Image-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions

Takumi Ege and Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 JAPAN

ABSTRACT

Image-based food calorie estimation is crucial to diverse mobile applications for recording everyday meal. However, some of them need human help for calorie estimation, and even if it is automatic, food categories are often limited or images from multiple viewpoints are required. Then, it is not yet achieved to estimate food calorie with practical accuracy and estimating food calories from a food photo is an unsolved problem. Therefore, in this paper, we propose estimating food calorie from a food photo by simultaneous learning of food calories, categories, ingredients and cooking directions using deep learning. Since there exists a strong correlation between food calories and food categories, ingredients and cooking directions information in general, we expect that simultaneous training of them brings performance boosting compared to independent single training. To this end, we use a multi-task CNN [1]. In addition, in this research, we construct two kinds of datasets that is a dataset of calorie-annotated recipe collected from Japanese recipe sites on the Web and a dataset collected from an American recipe site. In this experiment, we trained multi-task and single-task CNNs. As a result, the multi-task CNN achieved the better performance on both food category estimation and food calorie estimation than single-task CNNs. For the Japanese recipe dataset, by introducing a multi-task CNN, 0.039 were improved on the correlation coefficient, while for the American recipe dataset, 0.090 were raised compared to the result by the single-task CNN.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; **Supervised learning by classification**; **Supervised learning by regression**; **Multi-task learning**;

KEYWORDS

calorie estimation, Multi-task CNN, food recognition

ThematicWorkshops'17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-5416-5/17/10...$15.00
https://doi.org/10.1145/3126686.3126742

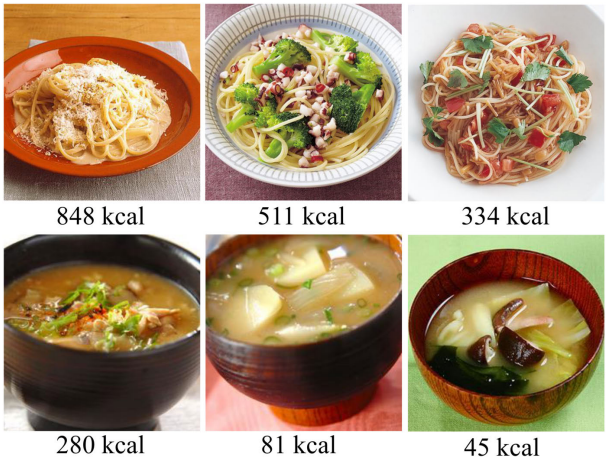


Figure 1: The differences of food calorie values within the same food categories. "Spaghetti" in the top row and "Miso soup" in the bottom row are shown, respectively.

ACM Reference Format:
Takumi Ege and Keiji Yanai. 2017. Image-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions. In *Proceedings of ThematicWorkshops'17, Mountain View, CA, USA, October 23–27, 2017,* 9 pages.
https://doi.org/10.1145/3126686.3126742

1 INTRODUCTION

In recent years, because of a rise in health thinking on eating, many mobile applications for recording everyday meals have been released so far. Some of them employ food image recognition which can estimate not only food names but also food calories. However, since these applications often require users to enter information such as food categories and size or volume, there are problems that it is troublesome and subjective evaluation. To solve these problems, automatic recognition of the food photo on the mobile devices is effective [2, 6, 9, 13, 14, 19]. However, in most of the cases, the estimated calories are just associated with the estimated food categories, or the relative size compared to the standard size of each food category which is usually indicated by a user manually. 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 lot, but also is promising as a new problem of image recognition studies.

Regarding food calorie estimation, a lot of approaches have been proposed so far. The main approach is to estimate calories based on the estimated food category and its size or volume, which is a quite standard approach [5, 6, 9, 13–15]. Since food calories strongly depend on food categories and volumes, this approach is effective and important.

The other approach is to estimate calories from food photos directly without based on food categories and volumes. The works adopting this approach are a few [12]. Food calories strongly depend on the food categories, volumes, ingredients and cooking directions, and these appear in the appearance of finished dishes as shown in Figure 1. Even if food categories are the same, the food calories are different depending on used ingredients and cooking directions. We think estimating calories from the appearance is important in the task of food calorie estimation, which cannot be resolved by only food category estimation. Estimating food calories directly from a food photo potentially make it possible to account for the intra-category differences as shown in Figure 1.

In this work, basically we adopt the latter approach which is estimating food calories directly, and propose simultaneous learning of food calories, categories, ingredients and cooking directions. for the food calorie estimation from a food photo. Since there exists a strong correlation between food calories and food categories, ingredients and cooking directions information, we expect that simultaneous training of them brings performance boosting compared to independent single training. To this end, we use a multi-task CNN [1]. Chen and Ngo [4] proposed using a multi-task CNN to estimate food categories and food ingredients at the same time, and proved that simultaneous estimation boosted estimation performance on both tasks. Inspired by this work, we introduce a multi-task CNN for food calorie estimation. The food calorie estimation is treated as a regression problem that inputs a food photo and outputs food calorie. In that case, we assume that a given food image contains only one dish and outputs the value of the food calories for one person as shown in Figure 1. The food category estimation is treated as a normal classification problem. Regarding the food ingredient estimation, we convert food ingredients information into a real number vector by Word2Vec [11], and estimate the vectors as food ingredients information. Also, in the cooking directions estimation, the sentence of the cooking directions is converted into a real number vector, and estimate the vectors as the cooking directions information. In addition, in this paper we collected calorie-annotated recipe data from the online cooking recipe sites, and constructed two kinds of datasets. Food photo datasets such as Food-101 [3], UEC Food-100 [10] and VIREO Food-172 [4] have been published so far, but have no calorie annotation.

To summarize our contributions in this paper, we (1) propose to use multi-task CNN for the task of food calorie estimation with simultaneous learning of food calories, categories, ingredients and cooking directions, and (2) construct calorie-annotated food photo datasets by collecting recipe data from online cooking recipe sites and (3) showed the effectiveness of the multi-task CNN based approach by the comprehensive experiments. In addition note that this work is an extension of our previous work [7].

2 RELATED WORK

2.1 Image-based food calorie estimation

Various approaches has been proposed so far and the main approach is to estimate calories based on estimated food categories and its size or volume using the value of food calorie per unit area or volume.

Chen et al. [5] proposed an image-based food calorie estimation method that estimates food categories and volumes by depth cameras such as Kinect. Depth cameras such as Kinect are special devices, so it is thought that ordinary people are difficult to use usually.

Kong et al. [9] proposed a mobile application to estimate food calories from images multiple images,"DietCam". They carried out segmentation and food item recognition, and in addition reconstructed 3D volumes of food items and calculate food calories based estimated volumes. 3D reconstruction was performed with SIFT-based keypoint matching and homography estimation which were a standard method of 3D stereovision. Also, Dehais et al. [6] carried out detection of dishes, segmentation, food categorization. Furthermore, 3D reconstruction with images from multiple viewpoints is performed, finally estimate the amount of carbohydrates. In case of estimating food volumes with images from multiple viewpoints, it is often necessary to calibrate the camera or points where the photographing point, so it is burden on the user.

Im2Calories by Google [13], which estimates food categories, ingredients, volumes of each of the dishes included in a given food photo, and finally outputs food calories by calculation based on the estimated volumes and the calories density corresponding to the estimated food category. In the experiment, there is a problem that the calorie-annotated dataset is insufficient and evaluation is not sufficiently performed.

Pouladzadhe et al. [15] proposed a food calorie estimation system which needed two dish images taken from the top and the side and used a thumb of a user as a reference object. Their method to estimate volumes were calculated by multiplying the size of food items estimated from the top-view image by the height estimated from the side-view image, which was relatively a straight-forward way. Okamoto et al. [14] proposed an image-based calorie estimation system which estimate food calories automatically by simply taking a meal photo from the top with a pre-registered reference object.

As described above, estimating food categories and volumes is a standard approach for estimating food calories

from the food photo. In contrast to this, Miyazaki et al. [12] estimate calories from food photos directly without estimating food categories and volumes. The biggest difficulty on direct calorie estimation is creating datasets which contains calorie-annotated food images. They hired dietitians to annotate calories on 6512 food photos which up-loaded to the commercial food logging service, Food-Log[1]. Unfortunately, their dataset was not released, because they used the food photos picked up from the master image database of the commercial service. In their work, they adopted image-search based calorie estimation, in which they searched the calorie-annotated food photo database for the top k similar images based on conventional hand-crafted features such as SURF-based BoF and color histograms and estimated food calories by averaging the food calories of the top k food photos. Since their method ignored information on food categories, their method was applicable for any kinds of foods. However, the number of food images was not enough for the search-based method, and the employed image features was too simple. As results, they failed to estimate food calories with high accuracy. On the other hand, because in our work we use CNN which is successful in image recognition, we expect great improvement in terms of accuracy.

2.2 Multi-task CNNs

To learn multiple tasks simultaneously, multi-task CNN has been proposed so far [1]. In the original work, it was applied to the face attribute detection task.

Recently, it was applied to food category and ingredient estimation by Chen and Ngo [4]. They showed that simultaneous estimation boosted estimation performance on both tasks. Inspired by this work, we introduce a multi-task CNN for simultaneous learning of food calories, categories, ingredients and cooking directions.

3 METHOD

In this work, for training our CNNs we use Chainer[2] [20] that the framework of Deep Learning. The food calorie estimation is treated as a regression problem that inputs a food photo and outputs the value of the food calorie. We assume that a given food image contains only one dish and outputs the value of the food calorie for one person. Regarding food ingredient estimation, for solving the problems of selecting ingredients to be recognized and notation deformation, we convert the food ingredients information to a real number vector by Word2Vec [11], and use the vectors for training of our CNNs. Also, in the cooking directions estimation, the sentence of the cooking directions is converted into a real number vector which is employed to training of our CNNs.

3.1 Overview of multi-task CNN

The architecture of our multi-task CNN is based on VGG-16 [17]. As shown in Figure 2, the fully-connected layer (fc6) is shared by all tasks, and the fc7 layer is branched to each task, so that each task has the fc7 layer and the output layer (fc8) independently. Chen and Ngo [4] showed that the best multi-task CNN architecture for food recognition which is based on VGG-16 is one having one shared fc layer and two individual fc layers. We follow this architecture in our work on multi-task food calorie estimation.

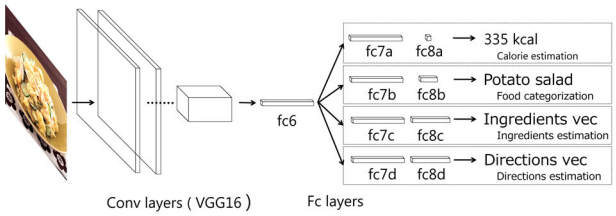


Figure 2: Overview of our multi-task CNN.

In this paper, we train the food calories, categories, ingredients and cooking directions simultaneously. Let L_{cal}, L_{cat}, L_{ing}, L_{dir} be the loss function of each task, and let N be the total number of learning data. The overall loss function L is as follows:

$$L = \frac{1}{N}\sum_{n=0}^{N}(L_{cal} + \lambda_{cat}L_{cat} + \lambda_{ing}L_{ing} + \lambda_{dir}L_{dir}) \quad (1)$$

We denote L_{ab} as an absolute error and L_{re} as a relative error, L_{cal} is defined as follows:

$$L_{cal} = \lambda_{re}L_{re} + \lambda_{ab}L_{ab}, \quad (2)$$

where $\lambda_{re}, \lambda_{ab}, \lambda_{cat}, \lambda_{ing}$ and λ_{dir} are the weight on the loss function of each task, and the value of each λ is usually determined so that all loss terms converge to the same value. Details are described in experiments.

3.2 Food calorie estimation

The food calorie estimation task has the fc7a layer with 4096 dimension and an output layer (fc8a) composed of one unit which outputs the food calorie. Because the food calories are the real value data, this task is treated as a regression problem. Generally, in the regression problem, a mean square error is used as the loss function, although in this paper we use the loss function of Equation (2). The absolute error is the absolute value of the difference between the estimated value and the ground-truth, and the relative error is the ratio of the absolute error to ground-truth. Since both errors are important indicators, we think that it is desirable to consider both. Combining absolute error and relative error as in Equation (2), both errors decrease in training. Let y be the estimated value of an image x and g be the ground-truth, L_{ab} and L_{re} are defined as following:

$$L_{ab} = |y - g| \quad (3)$$

$$L_{re} = \frac{|y - g|}{g} \quad (4)$$

3.3 Food category estimation

The food category estimation task has the fc7b layer with 4096 dimension and an output layer (fc8b) composed of units

[1] http://www.foodlog.jp/
[2] http://chainer.org/

corresponding to each category. Let y_i be the estimated value of unit i of an image x and g_i be the ground-truth, L_{cat} are defined as following:

$$L_{cat} = -\sum_{k=1}^{n} g_k \log y_k, \tag{5}$$

where g_k is a binary value. We set $g_k = 1$ if the unit i is a correct answer, while we set $g_k = 0$ if the unit i is not a correct answer. n represents the number of food categories. For example, in the case of 20 food categories, we set n as 20.

3.4 Food ingredient estimation

In this paper, we convert each word of food ingredient names into a real-value vector by Word2Vec [11]. Since each recipe contains multiple ingredients, we obtain a vector of ingredients information for each recipe by calculating a weighted linear combination of Word2Vec vectors of all the ingredient words for each recipe. We use this calcurated vector as a representation of ingredient information on food recipes. In case of using ingredient vectors as training data, it is difficult to recognize ingredients individually for a given food image. This is not a big problem, since our objective to introduce ingredient information is not the ingredients recognition. We expect to obtain the effect of simultaneous learning by multi-task CNNs and to improve estimation accuracy of food calorie values and categories. Therefore we adopt this method for training food ingredients information.

In this paper we use the model of Word2vec pre-trained with a large-scale recipe corpus. For the Japanese recipe dataset, we use sentences of cooking directions in the CookPad recipe dataset [3], while for the American recipe dataset, we use sentences of cooking directions in American recipe dataset described in 4.2. The sentences used for training in Word2Vec are pre-processed such as removal of low frequency words and subsampling of high frequency words. We use Skip-gram [11] as model and perform negative sampling [11] for training of Word2Vec.

For each recipe data, we use only the words of food ingredient name that is top N_{max} of the value of the *tf-idf*. In the experiments, We set N_{max} to the average number of food ingredient words of each recipe. Finally, a food ingredient vector for each recipe data is calculated from the vectors obtained from Word2Vec and the values of *tf-idf*. Let w_i as the words of food ingredient name at recipe data r_j, food ingredient vector v_j of recipe data r_j are defined as following:

$$v_j = \sum_{k=1}^{N} tfidf_{k,j} * word2vec(w_k) \tag{6}$$

N is the number of words used in each recipe data. $word2vec(w_k)$ is a real number vector of w_k obtained from Word2Vec and $tfidf_{k,j}$ is the value of *tf-idf* of w_k at recipe data r_j. Then v_j is L2 normalized.

Training of food ingredients information is realized as a task of estimating food ingredient vector. This food ingredient estimation task has the fc7c layer of 4096 dimensions and an output layer (fc8c) composed of units of dimensions of food ingredient vector. Let y_i as the output of unit i and g_i as the ground-truth, L_{ing} are defined as following:

$$L_{ing} = \frac{1}{2}\sum_{k=1}^{n}(g_k - y_k)^2 \tag{7}$$

3.5 Cooking directions estimation

In addition to ingredient information, we use cooking directions as addition information for multi-task learning as well.

In the same way as ingredient information, we convert each word in the sentences of cooking directions into a real-value vector by Word2Vec [11], and calculate a weighted linear combination of them for each recipe.

To obtain cooking direction vectors, we only use nouns, verbs and adjectives in the sentences of cooking directions, and use words with high *tf-idf* values. For each recipe data, we use only the words of sentences of cooking directions that are top N_{max} of the value of the *tf-idf*. In the experiments, We set N_{max} to the average number of words included in cooking directions of each recipe. Finally, cooking direction vector for each recipe data is calculated from the vectors obtained from Word2Vec with the *tf-idf* weights. Let w_i be the words of sentences of cooking directions at recipe data r_j. Cooking direction vector v_j of recipe data r_j is calculated by Equation (6). Training of cooking direction information is realized as a task of estimating this cooking directions vector. This cooking direction estimation task has the fc7d layer of 4096 dimensions and an output layer (fc8d) composed of units of dimensions of cooking directions vector. L_{ing} is defined as Equation (7).

4 CONSTRUCTION OF CALORIE-ANNOTATED FOOD PHOTO DATASET

As far as I know, there is currently no publicly available dataset with both food image and food calorie. It costs too much to create calorie-annotated food image dataset by hand. Instead, we focus on collecting such data from the Web. In fact, some commercial cooking recipe sites provide recipes annotated with calorie values. In addition, they provide information on a food ingredient list and a description of cooking direction for each recipe as well. In this paper, we collect such information from some commercial Web sites, and create recipe datasets annotated with calorie values. In order to confirm an effect of multi-task learning sufficiently, we construct two kinds of datasets, Japanese and American, and use them respectively in experiments.

4.1 Japanese calorie-annotated food photo dataset

In the Japanese datasets, about 83,000 calorie-annotated recipe data were collected from six recipe sites (“Ajinomoto”[4],

[3] http://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html

[4] http://park.ajinomoto.co.jp/

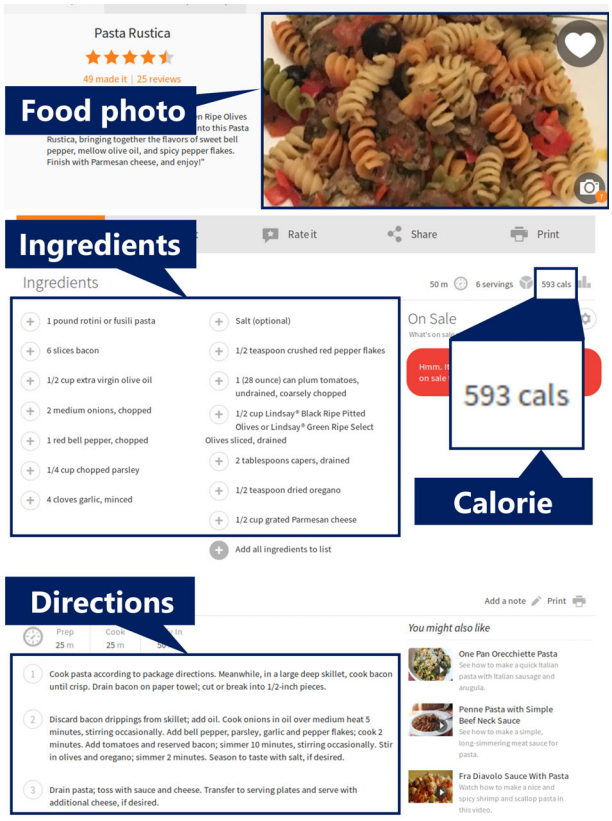
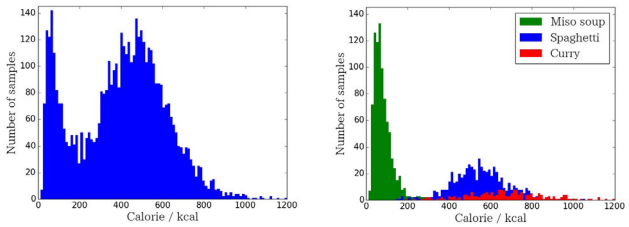


Figure 3: Example of a recipe page[10].

Figure 4: 15 categories of Japanese recipe datasets.



(a) All the 15 foods. **(b) "Spaghetti", "Miso soup" and "Curry".**

Figure 5: The distribution of the calorie values of Japanese recipe datasets.

"e-Recipe"[5], "Kikkoman"[6], "Kyou no Ryour"[7], "Orange Page"[8], "Lettuce Club"[9]). Each recipe presented in these sites contains an ingredient list, descriptions on cooking directions, food images, and the value of a food calorie as shown in Figure 3. These websites does not provided the method for general users to post recipe information. All of the recipe sites except for "Ajinomoto" clearly indicate that professionals such as chefs provided recipe information which is expected to be reliable. Observing the collected data, it was found that most of the food photos contain one kind of dishes, and the value of food calorie are per serving. Therefore, in this study, we assume that giving a single-label food photo and estimating the calorie value for one person.

Since we use food categories on multi-task training, it is necessary to collect food category information as well. In this paper, we limited collecting recipes on the representative 15 categories included in the UEC Food-100 [10]. The UEC Food-100 contains food photos of 100 kinds of Japanese foods without food calories information.

We collect recipe data which has food calories information for one person. Then, we manually excluded the images with low resolution or multiple kinds of dishes. Finally, we excluded the food categories which is the number of samples less than 100. In the end, a total of 4877 images were collected on 15 categories as shown in Figure 4. Figure 5(a) shows the calorie distribution of all the collected recipes, and Figure 5(b) shows that of "Miso soup", "Spaghetti" and "Curry". Because of "Miso soup", the calories of which is less than 100 kcal are many, although we found that there are many foods with around 500 kcal. As shown in Figure 5(b), the values of food calories tend to gather depending on food categories. Although in some food categories, the range of food calories within the same food category are relatively broad.

4.2 American calorie-annotated food photo dataset

In the American datasets, about 24,000 calorie-annotated recipe data were collected from Allrecipes[10]. Allrecipes is a recipe site of a user contribution type, and the values of food

[5] http://erecipe.woman.excite.co.jp/
[6] https://www.kikkoman.co.jp/homecook/
[7] http://www.kyounoryouri.jp/
[8] http://www.orangepage.net/
[9] http://www.lettuceclub.net/recipe/
[10] http://allrecipes.com/

Figure 6: 21 categories of American recipe datasets.

calorie per serving is obtained from each recipe. We used the categories used in Allrecipes and excluded the images with low resolution or multiple kinds of dishes. In the end, a total of 2484 images were collected on 21 categories as shown in Figure 6. Compared with the Japanese recipe datasets, this datasets contains foods more similar in appearance.

5 EXPERIMENTS

In this paper, we extended VGG-16 [17] and implemented multi-task CNN as shown in the Figure 2. In the fc6 layer and fc7 layer, Batch Normalization [8] were used instead of Dropout [18]. In the layers other than Batch Normalization layers and the fc8 layers, the pre-train model of ImageNet 1000 classification tasks were used as the initial value. For optimization of the CNNs, we used SGD with the momentum value, 0.9 and the size of mini-batch was 8. The weights of the loss term of Equation (1) and Equation (2) were determined as follows. Firstly, the weights of the loss terms are set to 1 and train once. In the training, the values of the losses for each iteration are preserved. Finally, the inverse of the average value of the loss in all iterations is used as the weight for the loss term of each task. In this experiments, we fixed λ_{re} to 1.

For the test, 10 models obtained at the 100 iteration intervals from the last 1k iterations in training were used, and the average value of the estimated values obtained from each model was taken as the final estimated value.

5.1 The loss function of food calorie estimation

In this experiment, we tested the effectiveness of loss function combining L_{re} and L_{ab}. Equation (2) are compared to the loss function composed of each loss function. We used the Japanese recipe dataset 4.1. We used 70% of the dataset for training, and the rest for evaluation. We used 0.001 of the learning rate for 50k iterations, and then used 0.0001 for 20k iterations.

Table 1 shows the result of food calorie estimation. We show the average of the relative error representing the ratio between the estimated values and the ground-truth, and the absolute error representing the differences between both.

Table 1: Comparison of loss functions of estimating food calorie.

	rel. err.(%)	abs. err.(kcal)	correlation	20% err.(%) ⩒	40% err.(%) ⩒
only L_{re}	**29.4**	105.9	0.776	42.0	77.0
only L_{ab}	59.8	134.9	0.589	36.7	63.5
$L_{re} + L_{ab}$	**29.4**	**100.7**	**0.778**	**45.9**	**77.7**

In addition we show the correlation coefficient between estimated value and ground-truth and the ratio of the estimated value within the relative error of 20% and 40%. In table 1, the absolute error and the ratio of the estimated value within the relative error of 20% indicate the accuracy is improved by using both errors. Therefore, in this paper, we use Equation (2) as the loss function of food calorie estimation.

5.2 Cooking direction vector

In this experiment, we tested the effective cooking directions vector for food calorie estimation. We used the Japanese recipe dataset 4.1. We used 70% of the dataset for training, and the rest for performance evaluation. We used 0.001 of the learning rate for 50k iterations, and then used 0.0001 for 20k iterations. we trained Word2Vec with about 8,710,000 sentences of cooking directions in CookPad recipe dataset. The dimension of the word vector is $n = 500$. In the sentences of cooking directions for each recipe, because of $N_{max} = 44$ which is the average value of the number of words of sentences of cooking directions extracted from one recipe data, we only used the words of the top 44 of the *tf-idf* value. Then, in order to take time information into account simply, the sentence is divided into m in time order, and for each divided sentence, created a cooking directions vector by Equation (6). Finally, the divided vectors are concatenated.

Table 2 shows the result of food calorie estimation. We show the average of the relative error and the absolute error, the correlation coefficient and the ratio of the estimated value within the relative error of 20% and 40%. Since the sentences of cooking directions are basically short, it is considered that the effect of dividing was not obtained. Therefore, in this paper, we use $m = 1$ as the cooking directions vector.

5.3 Food calorie estimation with Japanese recipe dataset

We used the Japanese recipe dataset 4.1. We used 70% of the dataset for training, and the rest 30% for performance evaluation. We used 0.001 of the learning rate for 50k iterations, and then used 0.0001 for 20k iterations. For food ingredient vector and cooking directions vector, we trained Word2Vec with about 8,710,000 sentences of cooking directions in CookPad recipe dataset. The dimension of the word vector is $n = 500$. Regarding to food ingredient, because of $N_{max} = 12$, we only used the words of the top 12 ingredient

Table 2: Comparison of cooking directions vectors.

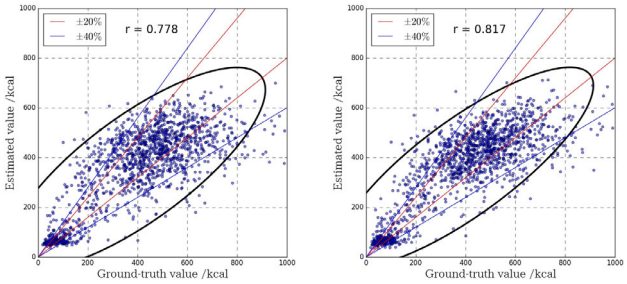
	rel. err.(%)	abs. err.(kcal)	correlation	20% err.(%) ⩛	40% err.(%) ⩛
calorie(single)	29.4	100.7	0.778	45.9	77.7
+directions(BoW+PCA)	29.0	96.6	0.793	46.6	78.8
+directions(m=1, 500-d)	**28.2**	**95.5**	**0.808**	48.1	**80.7**
+directions(m=2, 1000-d)	28.4	97.7	0.800	48.1	79.3
+directions(m=3, 1500-d)	29.1	98.8	0.786	**48.2**	77.4
+directions(m=4, 2000-d)	28.9	99.5	0.785	46.6	78.1

Table 3: The estimation results in Japanese recipe dataset.

	rel. err.(%)	abs. err.(kcal)	correlation	20% err.(%) ⩛	Top-1(%)
calorie(single)	29.4	100.7	0.778	45.9	—
+categories	27.9	95.2	0.802	48.8	82.8
++ingredients	27.6	94.4	0.811	49.5	**85.2**
+++directions	27.4	**91.2**	**0.817**	**50.1**	84.1
+ingredients	29.2	96.8	0.795	46.8	—
++directions	28.0	97.9	0.806	47.2	—
+directions	28.2	95.5	0.808	48.1	—
++categories	**27.3**	96.0	0.808	48.8	84.8
categories(single)	—	—	—	—	81.2

names of the *tf-idf* value for each recipe data, and created a food ingredient vector by Equation (6). In the sentences of cooking directions for each recipe, because of $N_{max} = 44$, we only used the words of the top 44 of the *tf-idf* value, and created a cooking directions vector.

Table 3 shows the results of food calorie estimation. Regarding the food calorie estimation, we show the average of the relative error and the absolute error, the correlation coefficient and the ratio of the estimated value within the relative error of 20%. In addition regarding food category estimation, we show the top-1 classification accuracy. Table 3 indicates the performance improve by multi-task CNNs in any evaluation. In the case of multi-task of all task, 2.0% and 9.5 kcal were reduced on the relative error and the absolute error, and 0.039 and 4.2% were increased on the correlation coefficient and the ratio of the estimated calories within 20% error. In addition, 2.9% were increased on the top-1 accuracy. Figure 7(a) shows the relation between the ground truth values and the estimated calorie values by the single-task CNNs, while Figure 7(b) shows the relation by the multi-task CNN. Comparing Figure 7(a) with Figure 7(b), we can confirm that the accuracy is improved by multi-task CNN from 95% confidence ellipses. Figure 8 and Figure 9 shows four examples of successfully estimated results and wrongly estimated results, respectively.



(a) Single-task CNNs. The correlation coefficient is 0.778.

(b) Multi-task CNNs. The correlation coefficient is 0.817.

Figure 7: The relation between the ground-truth values and the estimated calorie values.

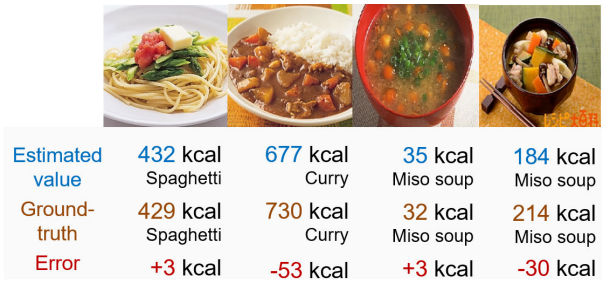


Figure 8: Examples on successful estimation of food calories.

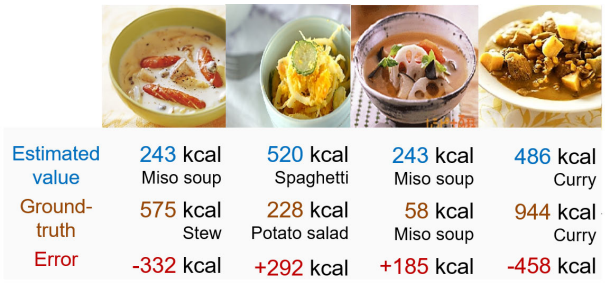


Figure 9: Examples on failure estimation of food calories.

5.4 Food calorie estimation with American recipe dataset

We used the American recipe dataset 4.2. For performing 5-fold cross-validation, we used 80% of the dataset for training, and the rest 20% , since the size of test data is small. We used 0.001 of the learning rate for 30k iterations, and then used 0.0001 for 40k iterations. For food ingredient vector and cooking directions vector, we trained Word2Vec with

Table 4: The estimation results in American recipe dataset.

	rel. err.(%)	abs. err.(kcal)	correlation	20% err.(%) ∨|	Top-1(%)
calorie(single)	43.3	128.5	0.293	32.2	—
+categories	42.9	120.5	0.361	34.3	58.7
++ingredients	**41.7**	**119.6**	**0.383**	**35.1**	61.1
+++directions	42.9	120.6	0.369	33.7	**61.3**
+ingredients	43.0	124.2	0.335	32.5	—
++directions	42.1	122.5	0.351	32.5	—
+directions	42.0	123.0	0.349	33.6	—
++categories	42.4	120.8	0.365	33.5	59.3
categories(single)	—	—	—	—	54.4

about 82,000 sentences of cooking directions in American recipe dataset. The dimension of the word vector is $n = 500$. Regarding to food ingredient, because of $N_{max} = 26$, we only used the words of the top 26 ingredient names of the *tf-idf* value in each recipe data. In the sentences of cooking directions for each recipe, because of $N_{max} = 66$, we only used the words of the top 66 of the *tf-idf* value.

We performed 5-fold cross-validation and indicate the results of food calorie estimation in the Table 4. In the case of multi-task of food categories and ingredients, 1.6% and 8.9 kcal were reduced on the relative error and the absolute error, and 0.090 and 2.9% were increased on the correlation coefficient and the ratio of the estimated calories within 20% error. In addition, 6.7% were increased on the top-1 accuracy. Similar to Japanese recipe dataset, multi-task CNNs advantage was confirmed.

5.5 Comparison with a baseline

In this experiment, we compared with Miyazaki et al. [12]. However, since the used dataset is different, it is a reference comparison. Miyazaki et al. [12] estimated calories from food photos directly without estimating food categories and volumes. In their work, they adopted image-search based calorie estimation, in which they searched the calorie-annotated food photo database for the top 5 similar images based on conventional hand-crafted features such as SURF-based BoF and color histograms and estimated food calories by averaging the food calories of the top 5 food photos. Note that this comparison is only for reference and not formal, since the detailed conditions and the datasets are different. The dataset used in [12] contains multiple-dish food photos and has no limitation on food categories, both of which are different from our dataset.

Table 5 shows the results. In Japanese recipe data, our result by multi-task CNNs outperformed the result of [12]. 0.5 and 13% were improved on the correlation coefficient and the ratio of the estimated calories within 20% error. In American recipe data, 0.06% were improved on the correlation coefficient.

Table 5: Comparison with the result of [12]

	correlation	≤ 20% err.(%)	≤ 40% err.(%)
Baseline	0.32	35	79
Multi-task (Japanese)	0.82	50	80
Multi-task (American)	0.38	35	65

6 DISCUSSION

In this paper, we collected calorie-annotated food photos from the online cooking recipe sites, and used the value of food calories as ground-truth for training and testing. However, we cannot guarantee the accuracy of this value of food calorie, and it is considered to contain many erroneous values. Therefore, it is considered that it is difficult to estimate the food calorie with high precision based on our datasets. In despite of this, it is thought that it was beneficial regardless of the datasets that the results of the performance improvement of both food calorie estimation and food category classification task by multi-task CNNs. In order to realize highly accurate food calorie estimation, it is considered to be urgent to create high-quality datasets. It is the biggest issue how to build a large-scale calorie-annotated food photo dataset.

7 CONCLUSIONS

In this paper, we proposed estimating food calorie from a food photo by simultaneous learning of food calories, categories, ingredients and cooking directions using multi-task CNNs. Also, we constructed two kinds of datasets that is a dataset of calorie-annotated recipe collected from Japanese recipe sites on the Web and a dataset collected from an American recipe site. In this experiments, in both datasets, the performances of multi-task CNNs outperformed the result of independent single-task CNNs.

As future work, we plan to estimation of volume for food calorie estimation, and construction of large and high-quality datasets. For highly accurate food calorie estimation, food detection and segmentation are important, and using a pre-registered reference object as [14, 16] may be necessary. It is also conceivable to estimate the volume of food from images from multiple viewpoints [6, 9]. We expect that much more accurate calorie estimation is possible by introducing multi-task CNN-based calorie estimation into volume/size-based calorie estimation approaches.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 15H05915, 17H01745, 17H05972, 17H06026 and 17H06100.

REFERENCES

[1] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. 2015. Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia* 17, 11 (2015), 1949–1959.

[2] V. Bettadapura, E. Thomaz, A. Parnami, D. G. Abowd, and A. Essa. 2015. Leveraging context to support automated food recognition in restaurant. In *Proc. of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[3] 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*.

[4] J. Chen and C. W. Ngo. 2016. Deep-based Ingredient Recognition for Cooking Recipe Retrival. In *Proc. of ACM International Conference Multimedia*.

[5] M. Chen, Y. Yang, C. Ho, S. Wang, E. Liu, E. Chang, C. Yeh, and M. Ouhyoung. 2012. Automatic Chinese Food Identification and Quantity Estimation. In *Proc. of SIGGRAPH Asia Technical Briefs*. 29.

[6] J. Dehais, M. Anthimopoulos, and S. Mougiakakou. 2016. GoCARB: A Smartphone Application for Automatic Assessment of Carbohydrate Intake. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*.

[7] T. Ege and K Yanai. 2017. Simultaneous Estimation of Food Categories and Calories with Multi-task CNN. In *Proc. of IAPR International Conference on Machine Vision Applications(MVA)*.

[8] S. Ioffe and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.. In *Proc. of International Conference on Machine Learning*.

[9] F. Kong and J. Tan. 2012. DietCam: Automatic dietary assessment with mobile camera phones. In *Proc. of Pervasive and Mobile Computin*. 147–163.

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

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*.

[12] T. Miyazaki, G. Chaminda, D. Silva, and K. Aizawa. 2011. Image - based Calorie Content Estimation for Dietary Assessment. In *Proc. of IEEE ISM Workshop on Multimedia for Cooking and Eating Activities*.

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

[14] 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*.

[15] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi. 2014. Measuring calorie and nutrition from food image. In *IEEE Transactions on Instrumentation and Measurement*. 1947–1956.

[16] W. Shimoda and K. Yanai. 2015. CNN-Based Food Image Segmentation Without Pixel-Wise Annotation. In *Proc. of IAPR International Conference on Image Analysis and Processing*.

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

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.

[19] R. Tanno, K. Okamoto, and K. Yanai. 2016. DeepFoodCam: A DCNN-based Real-time Mobile Food Recognition System. In *Proc. of ACM MM Workshop on Multimedia Assisted Dietary Management*.

[20] S. Tokui, K. Oono, S. Hido, and J. Clayton. 2015. Chainer: a Next-Generation Open Source Framework for Deep Learning. In *Advances in Neural Information Processing Systems*.