

**Personalized Food Image Classifier Considering Time-Dependent and Item-Dependent Food Distribution**

Qing YU\(^{(a)}\), Masashi ANZAWA\(^{(b)}\), Nonmembers, Sosuke AMANO\(^{(c),\dagger}\), Member, and Kiyoharu AIZAWA\(^{(c)}\), Fellow

---

**SUMMARY** Since the development of food diaries could enable people to develop healthy eating habits, food image recognition is in high demand to reduce the effort in food recording. Previous studies have worked on this challenging domain with datasets having fixed numbers of samples and classes. However, in the real-world setting, it is impossible to include all of the foods in the database because the number of classes of foods is large and increases continually. In addition to that, inter-class similarity and intra-class diversity also bring difficulties to the recognition. In this paper, we solve these problems by using deep convolutional neural network features to build a personalized classifier which incrementally learns the user’s data and adapts to the user’s eating habit. As a result, we achieved the state-of-the-art accuracy of food image recognition by the personalization of 300 food records per user.

**key words:** food image recognition, user-specific recognition, incremental learning, classifier adaptation

---

1. **Introduction**

Recently, more and more people have been using food tracking applications to manage their diet, control their portions, and stick to healthy eating habits. While in most of food tracking applications, users need to enter food names to get the nutrition information about the food, some photo-based food tracking applications like FoodLog App\(^{**}\) try to generate food dairies by recognizing the food in the photos uploaded by users. For example, FoodLog App will search photos related to food in users’ phone, detect the food area in each food photo and return the results of food image recognition. Users can select the right food name from the recognition results or enter the food name directly by themselves. To help users record their meals more easily by photo-based food tracking applications, it is necessary to achieve high accuracy in food image recognition.

Since image classification using deep convolutional neural networks (DCNNs) like ResNet\([2]\), has been widely developed for a wide range of tasks, a lot of previous studies have applied DCNNs to food image classification tasks\([3]–[8]\). Though the state-of-the-art accuracy has been achieved in these studies, food image recognition has been addressed as fixed-class recognition so far and fixed-class food image datasets like Food-101\([9]\) and UECFOOD-256\([10]\) are the benchmark datasets.

However, in the real-world setting, daily food data collected from consumers not only have a huge number of classes and imbalanced class distribution, but shows significant variation among consumers deriving from their nationality, locality, and preference\([11]\). Fixed-class food image recognition techniques are not capable of solving these problems.

In this paper, we extend the personalized classifier\([12]\) for large-scale daily food image recognition in the real-world setting to fit the user’s eating habit. We build a personalized classifier\([12]\) as our base framework which combines a Nearest Class Template (NCT) classifier and a Nearest Neighbor (NN) classifier for each user considering class imbalance problem. While new classes can be added to the classifier at nearly zero cost and the problem of food image variation among users can be avoided, the cold-start problem is also solved by the NCT classifier. We newly propose a time-dependent food distribution model and a weight optimization algorithm to make the personalized classifier learn the user’s data and adapt to the user’s eating habit.

The paper is organized as follows: Sect. 2 presents the dataset we used for our experiment. Our proposed method of personalized classification is detailed in Sect. 3. In Sect. 4, we explain how to extract deep features from food images. Experimental results are reported in Sect. 5 and we concluded our work in Sect. 6.

--

\(*\text{The authors are with Dept. of Information and Communication Eng., The University of Tokyo, Tokyo, 113–0033 Japan.}^{\dagger}\text{The author is with Foo.log Inc., Tokyo, 113–0033 Japan.}^{\ddagger}\text{A preliminary version of this paper was presented at ICIP 2018 as “Food Image Recognition by Personalized Classifier” [1] by the same author. The performance of our method is further improved by the optimization of common features in this version.}\)

\(\text{a) E-mail: yu@hal.t.u-tokyo.ac.jp}\)

DOI: 10.1587/transinf.2019PCP0005

\(\text{\textcopyright 2019 The Institute of Electronics, Information and Communication Engineers}\)
and used this dataset in our personalized classification ex-
periment. These two subsets did not overlap with each other. All of the images were cropped by the users’ annotations and resized to 256×256 pixels.

3. Personalized Classifier

In order to build a personalized classifier for each user, a nearest neighbor (NN) classifier is the most naive method. Though new classes defined by the user can be added into the classifier easily, the cold-start problem is significant because NN classifier needs enough data to achieve stable performance. Though other incremental learning methods have been proposed, most of them assume that the number of classes is limited [13], [14], and cannot learn from one sample [15], [16] or require high retraining costs with one sample [17]–[21].

A fast personalization framework, which combines a nearest class mean (NCM) classifier [22] and a nearest neighbor (NN) classifier is proposed in [12]. We further improve this framework with a time-dependent food distribution model and a vector weight optimization strategy that help the classifier learn the user’s eating habit.

3.1 Base Model

Each user \( u \in U \) has his/her own database \( V_u \). The user’s records are registered into \( V_u \) at each time when the user makes record; thus \( V_u \) after the user’s \( t \)-th record is denoted by:

\[
V_u = \{(x_{ui}, w_{ui}, c_{ui}) | 1 \leq i \leq t\},
\]

(1)

where \( x_{ui}, w_{ui} \) and \( c_{ui} \) represent the user’s \( i \)-th record’s deep feature, the parameter of weight assigned to this vector and the class to which it belongs, respectively. \( C_u \) is defined by the set of classes observed in \( V_u \).

Personalized classification is conducted using \( V_u \) and the set of common vectors \( V_m \), which is common to all users initially. \( V_m \) is denoted by:

\[
V_m = \{(x_{mi}, w_{mi}, c_{mi}) | 1 \leq i \leq |C_m|\},
\]

(2)

where \( C_m \) is the set of classes observed in \( V_m \) that we used to train the feature descriptor.

When the user records the \((t+1)\)th dish, weighted cosine similarity \( s_i \) between \( x_{u(t+1)} \) and all vectors in database is calculated by:

\[
s_i = \frac{w_i \cdot x_{u(t+1)} \cdot x_i}{\|x_{u(t+1)}\|_2 \|x_i\|_2}, (x_i, w_i, c_i) \in V_i
\]

(3)

where \( V = V_u \cup V_m \) and we also get \( c_i \) that represent the class which \( x_i \) belongs to. A set of cosine similarities is defined as \( s = \{s_i | 1 \leq i \leq |C_m| + t\} \).

Final predicted class \( c_{u(t+1)}^* \) of \( x_{u(t+1)} \) is calculated by:

\[
c_{u(t+1)}^* = c_j, j = \arg \max_{i} s_i
\]

(4)

where \( 1 \leq i \leq |C_m| + t \). For top-N results, duplicate classes
are removed by keeping the highest \( s_i \) of each class.

Parameter \( w \) controls the degree of personalization, which is the balance between the common vectors \( V_m \) and the user’s vectors \( V_u \). For the base model, we simply set:

\[
\begin{align*}
    w_{mi} &= \eta \ (1 \leq i \leq |C_m|) \\
    w_{ui} &= 1 \ (1 \leq i \leq r),
\end{align*}
\]

and \( 0 \leq \eta \leq 1 \) that makes classifier to learn user’s past inputs \( V_u \) faster. The base model using fixed uniform weights is the same as our previous proposal [12].

Figure 4 shows the pipeline of the personalized classifier. Each user has a common fixed-class classifier and the classifier is gradually personalized by learning new samples incrementally from existing or novel classes using a very limited number of samples.

### 3.2 Time-Dependent Food Distribution Model

Instead of Eq. (3) which predicts result directly on weighted cosine similarities \( s \), we propose a time-dependent food distribution model to rerank the result with an aim to make the classifier adapt to the user’s eating habit.

First, \( s \) is normalized by:

\[
s_i" = \frac{\exp(s_i)}{\sum_j \exp(s_j)}. \tag{7}
\]

Then, we define a time-dependent food distribution by:

\[
s_j" = s_j" \times (\pi_{c_j}^{\text{in}})^\lambda, \tag{8}
\]

where \( \lambda \) is a parameter that controls the weight of \( \pi_{c_j}^{\text{in}} \). This time dependent factor is the same as [14] except the regularization parameter \( \lambda \). If we define \( n_i(c_j) \) as the number of the appearance of class \( c_j \) from \( \text{max}[1, t-50] \) to \( t \), \( L = \text{min}(r, 50) \) and set \( \alpha = 0.01 \) for smoothing, \( \pi_{c_j}^{\text{in}} \) [14] is denoted by:

\[
\pi_{c_j}^{\text{in}} = \frac{n_i(c_j) + \alpha}{L + |C_m|\alpha}. \tag{9}
\]

Finally, a set of time-dependent cosine similarities \( s" = \{s_i\} |1 \leq i \leq |C_m| + t \) is obtained and predicted class \( c_{\text{ut}(t+1)} \) of \( x_{\text{ut}(t+1)} \) can be calculated based on \( s" \) by:

\[
c_{\text{ut}(t+1)} = c_j, \ j = \text{arg max}_{i} [s_i"], \tag{10}
\]

where \( 1 \leq i \leq |C_m| + t \).

### 3.3 Vector Weight Optimization

In the base model, we assign the same weight \( \eta \) to \( V_m \) in Eq. (5). However, since the frequency of different classes varies in the real world, treating vectors in \( V_m \) equally is obviously not optimal. Therefore, we used back propagation by stochastic gradient descent to optimize parameters \( w_{mi} \). The algorithm is shown in Algorithm 1. Optimized \( w_{mi} \) obtained by this algorithm is defined as \( \overline{w}_{mi} \).

### 4. Deep Feature Selection

To extract the deep feature \( x_{\text{ut}} \) which can represent users’ food images, and \( x_{\text{ut}} \) which can represent common classes, [12] proposed a fixed-class classifier which works as a feature descriptor to extract \( x_{\text{ut}} \) from each image and used the
A probability distribution across all classes as the following logit scores followed by a softmax activation that produces max classifier $f$ of the correct class from reaching close to 1, Eq. (12) is modified by adding a trainable scalar $s$ shared across all classes to scale the inner product [23], denoted by:

$$f_i(\phi(x)) = \frac{\exp(w_i^T \phi(x))}{\sum_c \exp(w_c^T \phi(x))},$$

(13)

By using CT, the common features can be optimized by DCNN and we expect these CT features are more effective than NCM [22] features. As a result, we use $\phi(x)$ as users’ image feature $x_{mi}$ and $w_i$ as common feature $x_{mi}$. As shown in the lower part of Fig. 5, to combine this architecture with our personalized classifier described in Sect. 3.1, the weight matrix can be considered as the database of each user. The features of the images uploaded by the user can be stored as user features along with common features in the weight matrix incrementally. When a new image is uploaded, the prediction result can be calculated based on the similarity between the image feature and the templates in the weight matrix as discussed in Sect. 3.

### 4.2 The Number of Common Classes

In order to determine an appropriate number of classes for training the network, we created seven subsets of FLD and each subset had $[1196, 841, 469, 213, 83, 25, 6]$ classes with $[100, 200, 500, 1000, 2000, 5000, 10000]$ records by following the strategy in [12]. Each subset had $[119600, 168200, 234500, 213000, 166000, 125000, 60000]$ images. We used 80% of images for training and 20% for testing. While [12] used GoogLeNet [24] as Deep Convolutional Network, we used ResNet-50 [2] which is fine-tuned on these subsets from ImageNet [25] pretrained model. We expected ResNet-50 [2] could extract better features than GoogLeNet [24] due to its higher performance on ImageNet [25].

The fixed-class classification results are shown in Fig. 6. Though the top-1 accuracies of subsets which have fewer classes are higher, it does not mean that these feature descriptors have better generalization ability. So we roughly estimated the real-world accuracies by multiplying the top-1 accuracies of subsets and their coverage on the whole dataset (orange crosses in Fig. 6). The coverage was computed by the following equation:
use found that the model using CT is not sensitive to rate for each method.

result, CM is a ffing subset to decide parameters. We split FLD-CLS into training subset consisted of the first 300 food records from ment. We split FLD-CLS into training subset to decide parameters of this network during personalized classification.

5.2 Dataset and Procedure

First, we decided parameter \( \eta \), weight of common features, of the base model. As we will discuss in Sect. 5.4, we found that the model using CT is not sensitive to \( \eta \) and we use \( \eta = 1 \) in our experiments, while in [12] the model using CM is affected by \( \eta \) and \( \eta = 0.85 \) was used.

Then, we applied the time-dependent food distribution model on the base model (Time Model) and decided parameter \( \eta = 1 \) and \( \lambda = 0.01 \).

Finally, at the experiment the initial weight optimization for \( V_m \) (WOPT), we decided the learning rate at Algorithm 1 to be 0.001 and the number of epoch \( E \) to be 20 and \( w_m \) is obtained.

In addition, the average inference time of one image is 40.64 ms with one Titan Xp GPU.

5.3 Results and Discussion

Table 1 shows the results evaluated on testing subset. First, the general fixed-class CNN shows constant low performance because it cannot learn new classes defined by users. We also reimplement CEL [26] which is a method of personali- zation of fixed-class classifier using cross-entropy considering label frequency in personal data. We modify the frequency-based part of this method to classify novel classes, but it did not perform well. It is also difficult to train ABACOC [27], another incremental learning method, with few samples per class. NCM [22], NCT [23] and 1-NN [28] classifier can learn the user’s data incrementally and achieve better performance that CNN, but the speed of personalization of NCM and NCT is slow and 1-NN has a cold-start problem.

The results of our methods show that (1) the base model outperforms other methods (2) using CT as common features \( x_m \) has higher performance during \( t_1 \sim t_50 \) than using CM [22] as \( x_m \) and the further comparisons are detailed in Sect. 5.4 (3) by considering the time-dependent food distribution, top-1 accuracy is improved and achieves 0.9% higher than the base model during \( t_{51} \sim t_{500} \) (4) optimizing the initial weights of common vectors has 1.5% higher accuracy than the base model during \( t_1 \sim t_50 \) which shows weights of common vectors are helpful (5) the personalized classifier combined the time-dependent food distribution model and the initial vector weight optimization achieves the highest performance and has about 2% higher
Overall, these results show that our architecture achieves the state-of-the-art accuracy of personalized food image recognition. Our personalized classifier with the time-dependent food distribution model and the initial weight optimization achieves the best performance.

5.4 Ablation Study

In Sect. 4, we introduced two different methods to compute common features \( x_{cm} \). To demonstrate the characteristic of each method, we show how the accuracies varied when the parameter value \( \eta \), which is the weight on common features \( x_{cm} \), was changed in Fig. 8. Figure 8a shows that when CM features are used as common features, the performance is relatively sensitive to the parameter \( \eta \). On the other hand, when CT features, which obtained from the weight matrix optimized by DCNN, are used as common features, Fig. 8b shows that the performance is relatively robust to variations in the parameter value. Consequently, CT features have better performance on representing each class than CM features and we use \( \eta = 1.0 \) in our experiment using CT for common features.

6. Conclusion

In this paper, we have presented a personalized classifier for large-scale daily food images recognition in the real-world setting. Our architecture combines a NCT classifier and a NN classifier for each user and we also introduced a time-dependent food distribution model and a weight optimization algorithm to achieve higher performance. Our technique can learn the user’s data and adapt to the user’s eating habit at nearly zero cost. We evaluated personalization performance on FoodLog Dataset which is a real-world food dataset. Our proposed method significantly outperforms the existing methods.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 18H03254 and JST CREST Grant Number JP-MJCR1686, Japan.

References


Qing Yu received B.S. in Information and Communication Engineering from the University of Tokyo in 2018. He is currently a M.S. student of Interdisciplinary Studies in Information Science at the University of Tokyo.

Masashi Anzawa received B.S. in Information and Communication Engineering and M.S. in Interdisciplinary Studies in Information Science from the University of Tokyo in 2016 and 2018, respectively. He is currently an employee of NTT DOCOMO Inc.

Sosuke Amano received B.S. in Information and Communication Eng. and M.S. in Interdisciplinary Studies in Information Science from the University of Tokyo in 2012 and 2015 respectively. He is a Ph.D. student of Dept. of Information and Communication Eng. He also joined foo.log Inc. in 2015.

Kiyoharu Aizawa received the B.E., the M.E., and the Dr.Eng. degrees in Electrical Engineering all from the University of Tokyo, in 1983, 1985, 1988, respectively. He is currently a Professor at Department of Information and Communication Engineering of the University of Tokyo. He was a Visiting Assistant Professor at University of Illinois from 1990 to 1992. His research interest is in image processing and multimedia applications. He received the 1987 Young Engineer Award and the 1990, 1998 Best Paper Awards, the 1991 Achievement Award, 1999 Electronics Society Award from IEICE Japan, and the 1998 Fujio Frontier Award, the 2002 and 2009 Best Paper Award, and 2013 Achievement award from ITE Japan. He received the IBM Japan Science Prize in 2002. He is currently a Senior Associate Editor of IEEE Tras. Image Processing, and on Editorial Board of ACM TOMM, APSIPA Transactions on Signal and Information Processing, and International Journal of Multimedia Information Retrieval. He served as the Editor in Chief of Journal of ITE Japan, an Associate Editor of IEEE Trans. Image Processing, IEEE Trans. CSVT and IEEE Trans. Multimedia. He has served a number of international and domestic conferences; he was a General Co-Chair of ACM Multimedia 2012. He is a council member of Science Council of Japan.