IEICE TRANS. INF. & SYST., VOL.E102–D, NO.11 NOVEMBER 2019

PAPER Special Section on Picture Coding and Image Media Processing

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

Qing YU[†a)], **Masashi ANZAWA**[†], ***Nonmembers***, **Sosuke AMANO**[†,††], ***Member***, ***and*** **Kiyoharu AIZAWA**[†], ***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.

2. Foodlog Dataset: A Real-World Food Dataset

We use the food record dataset named FoodLog Dataset (FLD) which meets our goal of developing a food image recognition system in the real-world setting. FoodLog Dataset was collected for about two years by a photo-based food tracking application for smartphones, called FoodLog. This dataset not only consists of 623,956 images including 1,508,171 foods uploaded by over 20,000 general users, but also contains each image's owner ID and time stamps. Some examples selected from FLD are shown in Fig. 1. Besides 1,870 kinds of general foods defined by the system, 97,457 kinds of other foods were defined by users.

This real-world food dataset is challenging for food im-

Manuscript received February 13, 2019.
Manuscript revised May 2, 2019.
Manuscript publicized June 21, 2019.

[†]The authors are with Dept. of Information and Communication Eng., The University of Tokyo, Tokyo, 113–0033 Japan.
[††]The author is with foo.log Inc., Tokyo, 113–0033 Japan.

*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.

a) E-mail: yu@hal.t.u-tokyo.ac.jp

DOI: 10.1587/transinf.2019PCP0005

**http://www.foodlog.jp/en

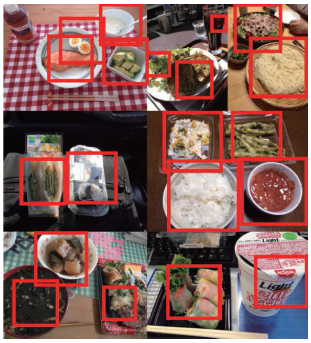


Fig. 1 Examples of FLD.

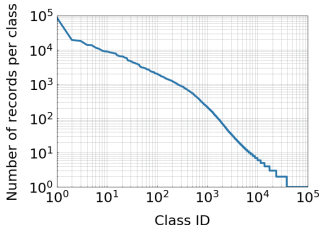


Fig. 2 The skewed class distribution of FLD.

(a) Examples of inter-class similarity. Pork curry (top), beef curry (middle), and chicken curry (bottom).

(b) Example of intra-class diversity. Each row shows different users' first five yogurt records.

Fig. 3 Inter-class similarity and intra-class diversity in FLD. Each image was cropped by the user's bounding box.

age recognition with following properties: (1) skewed class distribution (Fig. 2), (2) intra-class diversity and (3) inter-class similarity.

However, if we focus on one specific user, these problems become less complicated. Though three kinds of curries are visually indistinguishable (Fig. 3a), there is 42 percent of users, who recorded curry images five times or more, only eating one kind of curry. And Fig. 3b shows the images of yogurt recorded by different users. It is obviously that users have their photo preferences. As a result, designing a classifier for each user is essential.

For our experiments, we split FLD into two subsets. The first subset consisted of 469 major classes from the default label set and each class had 500 images. We detailed the method of determining the number of classes in Sect. 4. We called this subset FLD-469 and we used it to train a DCNN feature descriptor. The second subset consisted of 209,700 images, which contained the first 300 food records from 699 different users. We called this subset FLD-CLS and used this dataset in our personalized classification experiment. 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:

$$V_u = \{(\mathbf{x}_{ui}, w_{ui}, c_{ui}) \,|\, 1 \le i \le t\}, \tag{1}$$

where $\mathbf{x}_{ui}$, 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:

$$V_m = \{(\mathbf{x}_{mi}, w_{mi}, c_{mi}) | 1 \le i \le |C_m|\}, \tag{2}$$

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 $\mathbf{x}_{u(t+1)}$ and all vectors in database is calculated by:

$$s_i = w_i \frac{\mathbf{x}_{u(t+1)} \cdot \mathbf{x}_i}{\|\mathbf{x}_{u(t+1)}\|_2 \|\mathbf{x}_i\|_2}, (\mathbf{x}_i, w_i, c_i) \in V, \tag{3}$$

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

Final predicted class $c^*_{u(t+1)}$ of $\mathbf{x}_{u(t+1)}$ is calculated by:

$$c^*_{u(t+1)} = c_j, j = \arg\max_i \{s_i\}, \tag{4}$$

where $1 \le i \le |C_m| + t$. For top-N results, duplicate classes

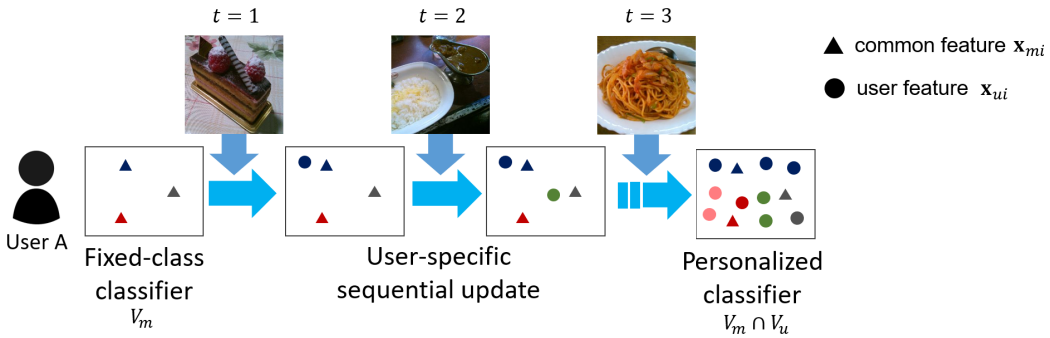


Fig. 4 Overview of the proposed personalized classifier. Each user has a common fixed-class classifier initially. The classifier is incrementally updated by learning new samples from existing or novel classes using a very limited number of samples for personalization.

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:

$$w_{mi} = \eta \ (1 \le i \le |C_m|) \tag{5}$$

$$w_{ui} = 1 \ (1 \le i \le t), \tag{6}$$

and $0 <= \eta <= 1$ 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 $\mathbf{s}$, 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, $\mathbf{s}$ 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:

$$s''_i = s'_i \times (\pi^{(t)}_{c_i})^{\lambda}, \tag{8}$$

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

$$\pi^{(t)}_{c_i} = \frac{n_t(c_i) + \alpha}{L + |C_m|\alpha}. \tag{9}$$

Finally, a set of time-dependent cosine similarities $\mathbf{s}'' = \{s_i | 1 \le i \le |C_m| + t\}$ is obtained and predicted class $c^*_{u(t+1)}$ of $\mathbf{x}_{u(t+1)}$ can be calculated based on $\mathbf{s}''$ by:

$$c^*_{u(t+1)} = c_j, j = \arg\max_i \{s''_i\}, \tag{10}$$

where $1 \le i \le |C_m| + t$.

Algorithm 1 Initial Weight Optimization for V_m

Input: Training data $\{\mathbf{x}_{ut}, c_{ut}\}$
Output: The parameters w_{mi}

```
w_mi ← η
for Epoch ← 1 to E do
    Shuffle U
    for u ← 1 to |U| do
        n ← 0
        L_total ← 0
        for t ← 1 to T do
            Compute similarity score s_mi by Eq. (3)
            if c_ut ∈ C_m ∧ c_ut ∉ C_u then
                Normalize s_mi by softmax function
                Compute loss L by Cross-Entropy Loss
                between label c_ut and confidence s_mi
                L_total ← L_total + L
                n ← n + 1
            end if
        end for
        Update the parameters w_mi by SGD on L_total/n
    end for
end for
```

3.3 Vector Weight Optimization

In the base model, we assign the same weight η 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 $\mathbf{x}_{ui}$ which can represent users' food images, and $\mathbf{x}_{mi}$ which can represent common classes, [12] proposed a fixed-class classifier which works as a feature descriptor to extract $\mathbf{x}_{ui}$ from each image and used the

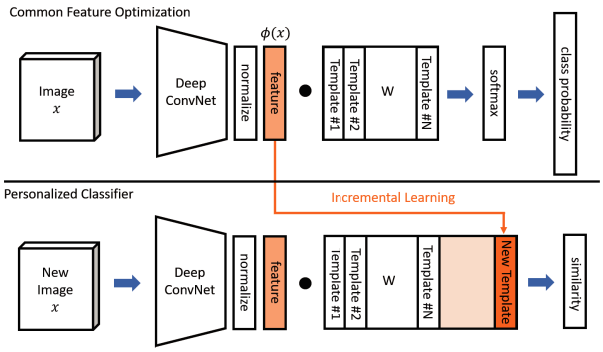


Fig. 5 The overall architecture of our feature descriptor and personalized classifier. The class templates in the weight matrix can be used as common features. Since the weight matrix can also be considered as the database of each user, user features can be added into the templates incrementally to build the personalized classifier.

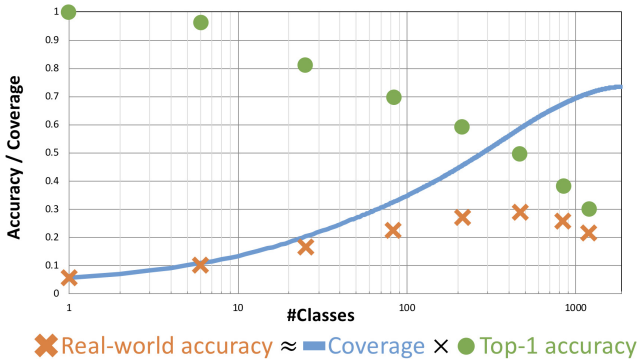


Fig. 6 Top-1 accuracies of various datasets.

class mean feature (CM) [22] of images belonging to each class as $\mathbf{x}_{mi}$. If n images are available for each class, $\mathbf{x}_{mi}$ is denoted by

$$\mathbf{x}_{mi} = \frac{1}{n}\sum_{j=1}^{n}\phi_i(x_j), \tag{11}$$

where $\phi_i(x_j)$ is the feature of j-th example of class i.

4.1 Nearest Class Template

Instead of CM [22], we adopt the feature learning proposed in [23] to generate optimal common features $\mathbf{x}_{mi}$.

The upper part of Fig. 5 shows the overall architecture of our feature descriptor. We use a DCNN to extract the feature of the input image $\phi(x)$. We also add an L2 normalization layer at the end of the feature extractor so that the output feature has unit length, *i.e.* $\|\phi(x)\|_2 = 1$. Then a softmax classifier $f(\phi(x))$ maps the feature into unnormalized logit scores followed by a softmax activation that produces a probability distribution across all classes as the following equation:

$$f_i(\phi(x)) = \frac{\exp(w_i^{\mathrm{T}}\phi(x))}{\sum_c \exp(w_c^{\mathrm{T}}\phi(x)))}, \tag{12}$$

where w_i is the i-th column of the weight matrix normalized to unit length. If we view each column of the weight matrix as a template feature which can represent the corresponding class, the last layer in our model computes the inner product between the feature of the input image $\phi(x)$ and all the template features w_i. If features and template features are normalized to unit lengths, the resulting prediction is equivalent to finding the most similar template features in the feature space, which is the same as the calculation of Eq. (3). So we call this method Nearest Class Template (NCT) and template features Class Template (CT).

Furthermore, to avoid the problem that the cosine similarity $w_i^{\mathrm{T}}\phi(x) \in [-1, 1]$ can prevent the softmax probability 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:

$$f_i(\phi(x)) = \frac{\exp(sw_i^{\mathrm{T}}\phi(x))}{\sum_c \exp(sw_c^{\mathrm{T}}\phi(x)))}. \tag{13}$$

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 $\mathbf{x}_{ui}$ and w_i as common feature $\mathbf{x}_{mi}$.

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:

Table 1 The results of personalized recognition. Each cell shows the average mean accuracy over 50 consecutive images in order of time. NVL shows whether of not the method could learn novel classes. The two upper limits show the rates at whether the class of the arriving sample was in V_m or $V_m \cup V_u$. Values in bold bace font are important results that show the effective of our methods.

Approach	NVL	$t_1 \sim t_{50}$		$t_{51} \sim t_{100}$		$t_{101} \sim t_{150}$		$t_{151} \sim t_{200}$		$t_{201} \sim t_{250}$		$t_{251} \sim t_{300}$	
		top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
CNN [2]		21.1	32.8	20.6	31.8	20.8	31.9	20.2	30.6	19.5	30.3	20.4	30.5
CEL [26]	✓	19.3	31.0	21.0	34.2	22.0	35.5	21.4	35.0	21.5	34.8	22.0	25.3
ABACOC [27]	✓	19.3	28.8	27.0	34.7	30.6	38.2	32.4	39.5	32.6	39.8	33.5	41.1
NCM [22]	✓	27.4	41.6	30.4	45.3	32.3	48.2	33.2	49.1	32.8	49.3	33.9	50.9
NCT [23]	✓	29.8	44.5	32.7	49.0	33.7	51.7	34.3	52.1	33.6	51.9	34.0	53.1
1-NN [28]	✓	22.9	28.5	33.1	42.1	37.0	47.7	38.1	49.3	38.5	50.3	39.2	52.6
Base Model (CM) [12]	✓	32.4	46.0	38.2	52.4	40.5	55.8	41.0	56.2	40.8	56.2	41.0	58.1
Base Model (ours)	✓	33.1	46.5	38.5	53.3	40.9	56.1	41.0	56.3	40.8	56.6	41.4	58.2
Time Model (ours)	✓	33.4	46.8	39.1	53.6	41.5	56.8	41.6	56.9	41.8	57.3	**42.3**	59.0
Base Model + WOPT (ours)	✓	**34.6**	47.0	39.2	53.4	41.7	56.4	41.7	56.8	41.6	57.1	42.3	58.7
Time Model + WOPT (ours)	✓	**34.8**	**47.5**	**39.9**	**53.9**	**42.2**	**57.1**	**42.1**	**57.3**	**42.2**	**57.7**	**42.9**	**59.2**
Upper limit		45.7		44.6		44.4		43.0		42.8		43.4	
Upper limit	✓	60.6		67.9		71.7		72.7		73.3		75.4	

$$Coverage(|C_m|) = \frac{\text{\#Images belonging to } C_m}{\text{\#Images in FLD}}, \tag{14}$$

where $|C_m|$ is the set of classes in the subset. This result indicates that the subset having 469 classes has the best performance in the real world. Consequently, we decided to use FLD-469 to train our feature descriptor and fixed the parameters of this network during personalized classification.

5. Experiment

5.1 Evaluation Protocol

To evaluate personalized image recognition performance, we calculated the mean accuracy for all users U:

$$MeanAccuracy(t) = \frac{1}{|U|} \sum_{u \in U} \mathbb{1}\left(c^*_{ut} = c_{ut}\right), \tag{15}$$

where $\mathbb{1}(\cdot)$ is the indicator function and c^*_{ut} is the predicted result, c_{ut} is the ground truth of the t-th record belonging to user u, respectively.

5.2 Dataset and Procedure

FLD-CLS is used in our personalized classification experiment. We split FLD-CLS into training/testing subsets. The training subset consisted of the first 300 food records from 400 different users and the testing subset consisted of the first 300 food records from the 299 users. We used the training subset to decide parameters η, λ, $\overline{w_{mi}}$ and the learning rate for each method.

First, we decided parameter η, 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 η and we use $\eta = 1$ in our experiments, while in [12] the model using CM is affected by η 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 $\overline{w_{mi}}$ 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 personalization 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 $\mathbf{x}_{mi}$ has higher performance during $t_1 \sim t_{50}$ than using CM [22] as $\mathbf{x}_{mi}$ 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 $t_{251} \sim t_{300}$ (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

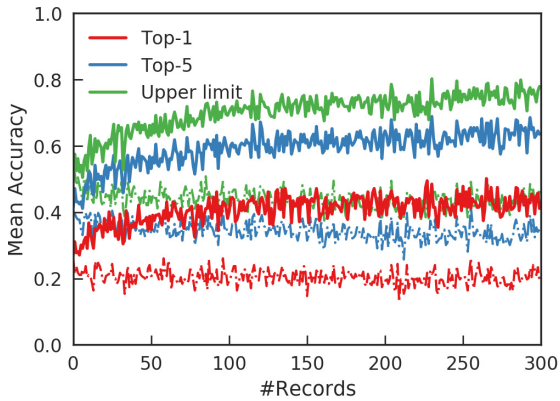
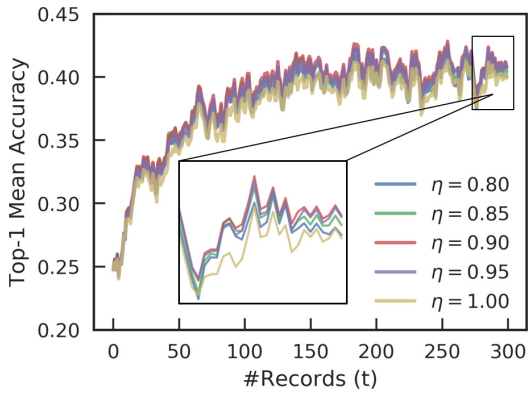
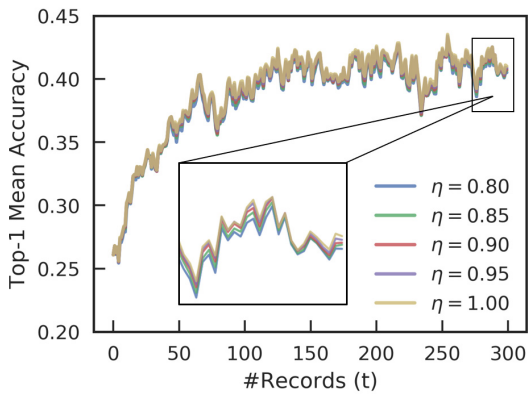


Fig. 7 Time series transition of mean accuracy. We show top-1 accuracy, top-5 accuracy, and upper limits of our method (Time Model + WOPT) in real lines and the CNN-based fixed-class method in dashed lines.



(a) Class Mean (CM) features as common features.



(b) Class Template (CT) features as commmon features.

Fig. 8 Time series transition of Top-1 mean accuracy with different η.

top-1 accuracy that the base model at any t.

Figure 7 shows the mean accuracy transition of our method (real lines) and CNN [2] (dashed lines) and it is clear that our method improves classification accuracy. Figure 7 also demonstrates that our method can learn the user's data and adapt to the user's eating habit from a small number of incremental samples.

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 $\mathbf{x}_{mi}$. To demonstrate the characteristic of each method, we show how the accuracies varied when the parameter value η, which is the weight on common features $\mathbf{x}_{mi}$, was changed in Fig. 8. Figure 8a shows that when CM features are used as as common features, the performance is relatively sensitive to the parameter η. 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 JPMJCR1686, Japan.

References

[1] Q. Yu, M. Anzawa, S. Amano, M. Ogawa, and K. Aizawa, "Food image recognition by personalized classifier," ICIP, pp.171–175, 2018.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, pp.770–778, 2016.
[3] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," ACMMM, pp.1085–1088, 2014.
[4] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," CVPR, pp.3140–3145, 2016.
[5] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment," iCOST, pp.37–48, 2016.
[6] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very

deep convolutional networks," MADiMA, pp.41–49, 2016.
[7] L. Herranz, S. Jiang, and R. Xu, "Modeling restaurant context for food recognition," IEEE Trans. Multimedia, vol.19, no.2, pp.430–440, Feb. 2017.
[8] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," ACMMM Workshop on CEA, pp.589–593, 2014.
[9] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," ECCV, pp.446–461, 2014.
[10] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," ECCV Workshop on TASK-CV, 2014.
[11] K. Kitamura, C. De Silva, T. Yamasaki, and K. Aizawa, "Image processing based approach to food balance analysis for personal food logging," ICME, pp.625–630, 2010.
[12] S. Horiguchi, S. Amano, K. Aizawa, and M. Ogawa, "Personalized classifier for food image recognition," IEEE Trans. Multimedia, vol.20, no.10, pp.2836–2848, Oct. 2018. DOI:10.1109/TMM.2018.2814339.
[13] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," CVPR, pp.577–584, 2011.
[14] A. Royer and C.H. Lampert, "Classifier adaptation at prediction time," CVPR, pp.1401–1409, 2015.
[15] L. Cao, J. Hsiao, P. de Juan, Y. Li, and B. Thomee, "Incremental learning for fine-grained image recognition," ICMR, pp.363–366, 2016.
[16] S.A. Rebuffi, A. Kolesnikov, G. Sperl, and C.H. Lampert, "iCaRL: Incremental classifier and representation learning," CVPR, pp.2001–2010, 2017.
[17] F.M. Castro, M.J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," ECCV, pp.233–248, 2018.
[18] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, "Incremental learning of random forests for large-scale image classification," IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.3, pp.490–503, March 2016.
[19] A. Pronobis, L. Jie, and B. Caputo, "The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition," Image and Vision Computing, vol.28, no.7, pp.1080–1097, July 2010.
[20] T. Yeh and T. Darrell, "Dynamic visual category learning," CVPR, pp.1–8, 2008.
[21] T. Poggio and G. Cauwenberghs, "Incremental and decremental support vector machine learning," NIPS, pp.409–415, 2001.
[22] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," ECCV, pp.488–501, 2012.
[23] H. Qi, M. Brown, and D.G. Lowe, "Learning with imprinted weights," CVPR, pp.5822–5830, 2017.
[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," CVPR, pp.1–9, 2015.
[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," IJCV, vol.115, no.3, pp.211–252, Dec. 2015.
[26] X. Li, E. Gavves, C.G. Snoek, M. Worring, and A.W. Smeulders, "Personalizing automated image annotation using cross-entropy," ACMMM, pp.233–242, 2011.
[27] R. De Rosa, F. Orabona, and N. Cesa-Bianchi, "The ABACOC algorithm: a novel approach for nonparametric classification of data streams," ICDM, pp.733–738, 2015.
[28] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol.13, no.1, pp.21–27, Jan. 1967.

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 Commnication 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.