CH-006

コミュニティデータを活用した大規模食事画像認識

Food Image Recognition Using Community Data

安沢 昌志† 天野 宗佑† 相澤 清晴† 小川 誠‡
Masashi Anzawa Sosuke Amano Kiyoharu Aizawa Makoto Ogawa

1. はじめに

ImageNet [1]などを用いた固定データセットに対しては，畳込みニューラルネットワーク（CNN）を用いた画像認識がよく機能することが知られている．一方，人によってクラスの定義や数が異なる現実世界では個人適応を行い，新規クラスを逐次追加することで認識精度及び認識限界を向上されることが示されている [2]．しかし，主にユーザ自身のデータを利用しているため，初期段階においては認識限界が 0.42 と半分以上は新規クラスが占めているという問題がある．

本研究ではコミュニティ内の他のユーザデータを利用することで，新規クラスを減らし．それにより，認識精度及び認識限界の向上することが可能かの検討を行う．

2. 関連研究

2.1 画像認識器の逐次個人適応

堀口ら[2]は Sequential Personalized Classifier（SPC）を提案している．SPC ではまず，固定クラスのデータセットを用いて CNN によるソフトマックス分類器を学習し，CNN を特徴抽出器として用いて，固定クラスごとの平均ベクトルを計算する．この平均ベクトルを用いた Nearest Class Mean（NCM）分類器は CNN と同等の精度を達成する．各々のユーザは，先述の全員に共通する固定クラスによるデータベースV_mとユーザ個人のデータベースV_uに基づいて認識を行う．V_uはユーザの入力を逐次的に追加していくデータベースである．重み付き 1-Nearest Neighbor（1-NN）分類器により，V_mよりV_uに重きを置く個人適応を実現している．

しかし，この手法では図 1 で示したように，ユーザの各時点でV_m，V_uのどちらにも含まれない新規クラスの認識は必ず失敗するという問題があり，これが認識限界の上限ともなっている．

2.2 食事画像認識

食事画像認識に関する研究は[4]~[8]など多く存在するが，学習は逐次的ではなく，クラス数は固定されており現実世界におけるクラス数の増加は考慮されていない．

その他食事画像認識ではないが食事画像を用いた研究としては，食事の量やカロリーの推定[6][9][10]，食材認識 [11]，食事ベクトルの生成[12]，食事画像からレシピを推定する研究[13]など広く存在する．

また食事画像認識向けのデータセット[3]~[6]も多く公開されているが，ウェブで収集されておりユーザ情報は付与されていないため，個人性を考慮することはできない．

3. データセット

本稿では一般ユーザによる食事画像のデータセット（FoodLog Dataset，以下 FLD）を用いる．FLD はスマートフォン向け食事記録アプリケーション FoodLog App [14]によって収集されたデータセットである． 2015 年 4 月時点で，20,820 人のユーザにより 1,508,171 の食事があり，各ユーザによって正方形領域のアノテーションがなされている．食事の種類は膨大で，99,314 のクラスが存在する．クラスの内訳はデフォルトで用意された 1,857 クラスとユーザが新規に定義した 97,457 クラスである．各画像には撮影時刻が記録されており，登録順を再現することができる．本稿では，FLD-213，FLD-CLS という FLD のサブセットを用いる．FLD-213 は FLD に頻出の 213 クラスの画像で，CNN の学習に用いる．各々の画像はユーザの選択した領域で切り取り 256 画素×256 画素にリサイズする．FLD-CLS は 320 人のユーザそれぞれについて時系列順に 300 件，合計 960,000 件の食事記録からなる食事認識用のサブセットである．

4. コミュニティデータの活用

SPC [2]及び，SPC をベースラインとした，コミュニティ内の他のユーザデータを活用した 4 つの手法を用いる．

① SPC
② 全ユーザデータ追加 (SPC + $C1$)
③ 全ユーザデフォルトクラス追加 (SPC + $C2$)
④ 理想的なユーザデータ追加 (SPC + $C3$)

† 東京大学
‡ foo.log 株式会社

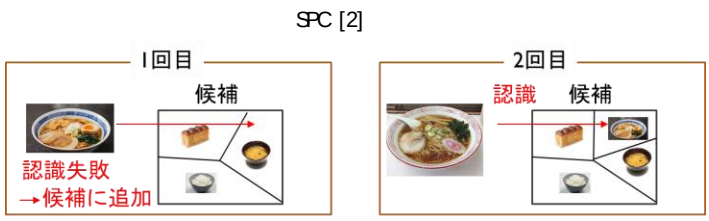

図 1　SPC の新規クラス認識
新規クラスの認識は必ず 1 回目は失敗する．2 回目以降は認識可能

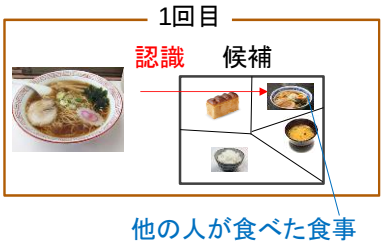

図 2 SPC + $C1$～$C3$の新規クラス認識
他のユーザデータに含まれていれば1回目でも新規クラスを認識可能

②, ③は認識限界を向上することを目的とし，④は認識精度を向上することを目的としている．図 3 で示すように，他のユーザデータを使用することによって，自分にとっては新規のクラスであっても認識できる可能性が向上すると考えられる．

4.1 SPC

全員に共通な固定クラスのデータベースV_mは固定クラスC_mの平均ベクトルからなる．

$$V_m = \{(X_{mi}, C_{mi}) | 1 \le i \le C_m\} \quad (1)$$

X_{mi}とC_{mi}はそれぞれ平均ベクトルとその所属クラスを表す．またユーザの入力を随時追加していくデータベースV_uは次式で表される．

$$V_u = \{(X_{ui}, C_{ui}) | 1 \le i \le t\} \quad (2)$$

X_{ui}とC_{ui}はユーザuのi番目のベクトルとその所属クラスを表す．ユーザuのt番目の入力X_{ut}の所属クラスc_{ut}^*は次式で予測される．

$$c_{ut}^* = \arg\max_{c \in C} [max\{s(c, X_{ut}, V_u), w_1 \cdot s(c, X_{ut}, V_m)\}] \quad (3)$$

CはV_mとV_uのいずれかに含まれるクラスの集合で，$w_1 > 0$はV_mとV_uの重視する度合いを表すパラメータである．ベクトルの類似度を測る$s(\cdot)$は内積で以下のように表す．

$$s(c, X_{ut}, V) = \begin{cases} \max_{(x,c) \in V_c} X^T X_{ut}, & V_c \neq \emptyset \\ -\infty & , otherwise \end{cases} \quad (4)$$

V_cはVのうちクラスcに所属するベクトルの部分集合である．ユーザは認識が正しいか否かを判断し，誤って入れば正しいクラスに修正する．その後V_uに(X_{ut}, C_{ut})を追加する．

4.2 全ユーザデータ追加（SPC + $C1$）

SPC で用いたV_m, V_uとは別に自分以外のユーザの全データを含むデータベース$V_{C1,u}$を用いる．

$$V_{C1,u} = \{(X_{u'}, C_{u'}) | u' \in U,\ u' \neq u\} \quad (5)$$

ユーザuのt番目の入力X_{ut}の所属クラスc_{ut}^*は次式で予測される．

$$c_{ut}^* = \arg\max_{c \in C} [max\{s'(V_u), w_1 \cdot s'(V_m), w_2 \cdot s'(V_{c1,u})\}] \quad (6)$$

$w_1, w_2 > 0$は$V_m, V_u, V_{c1,u}$の重視する度合いを表すパラメータである．

4.3 全ユーザデフォルトデータ追加(SPC + $C2$)

SPC で用いたV_m, V_uとは別に自分以外のユーザの全データのうち，デフォルトクラスとして用意された 1,857 クラス C_Dに含まれるデータを集めたデータベース$V_{C2,u}$を用いる．

$$V_{C2,u} = \{(X_{u'}, C_{u'}) | u' \in U,\ u' \neq u,\ C_{u'} \in C_D\} \quad (7)$$

所属クラスc_{ut}^*は次式で予測される．

$$c_{ut}^* = \arg\max_{c \in C} [max\{s'(V_u), w_1 \cdot s'(V_m), w_2 \cdot s'(V_{c1,u})\}] \quad (8)$$

4.4 理想的なユーザデータ追加（SPC + $C3$）

全時刻期間 T=300 を 50 毎の 6 つの期間T_1~T_6に分割する．SPC で用いたV_m, V_uとは別に，ユーザuにおける各期間T_iで最も Top 1 の精度が高くなる自分以外の 1 人のユーザu'のデータベースV_u'を用いる．ただし，どのユーザのデータベースを利用しても精度が下がる場合V_u'は用いない．V_u'の選択のためにすべてのユーザに対して全探索を行う．次の期間T_{i+1}においては，V_u'は用いずに再度全探索を行うことによって，新たなV_u'を決定する．ユーザuのt番目の入力X_{ut}の所属クラスc_{ut}^*は$V_{c1,u}$と同様に予測される．

現実世界において，最も高くなるほかのユーザを事前に決定することはできないため，1 人のユーザを加えることによる最大の上がり幅を検証することが目的である．

5. 実験結果

5.1 評価手法

時刻t における性能は，次式のように，全ユーザの内t番目の入力を正しく認識できた人数で評価する．

$$MeanAccuracy(t) = \frac{1}{|U|} \sum_{u \in U} I(c_{ut}^* = c_{ut}) \quad (9)$$

ここで$I(\cdot)$は指示関数である．

5.2 結果

CNN の構造には Batch Normalization [15]を用いた GoogleNet [16]を使用した．ImageNet で事前学習された重みから FLD-213 でファインチューニングした pool5 層から抽出した特徴を L2 正規化して用いる．これらの平均ベクトルを計算することにより，V_m

表 1　4 手法の Mean Accuracy の時系列推移

各々の値は，その時間期間における Mean Accuracy の平均を示している．各期間は，SPC + $C3$のT_1~T_6に対応している．

手法	t_1~t_{50}		t_{51}~t_{100}		t_{101}~t_{150}		t_{151}~t_{200}		t_{201}~t_{250}		t_{251}~t_{300}	
	Top1	限界	Top1	限界	Top1	限界	Top1	限界	Top1	限界	Top1	限界
SPC	29.6	52.2	35.0	61.2	37.3	65.6	37.9	68.0	38.0	69.1	38.3	71.2
SPC + $C1$	28.0	**85.1**	34.1	**87.0**	36.6	**88.5**	37.1	**89.2**	37.7	**88.6**	38.5	**89.8**
SPC + $C2$	30.0	78.2	35.1	81.7	37.2	83.9	37.9	85.0	38.1	84.5	38.5	86.0
SPC + $C3$	**33.5**	56.2	**38.7**	64.7	**40.8**	68.8	**41.4**	70.9	**41.5**	71.8	**41.9**	74.0

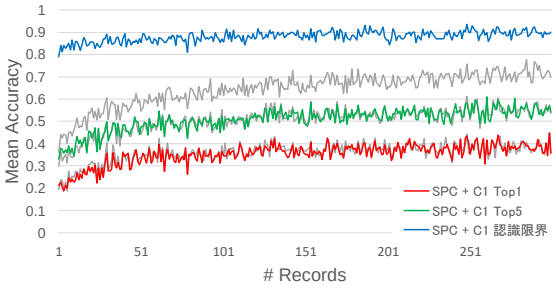

図 3　SPC + $C1$の Mean Accuracy 推移

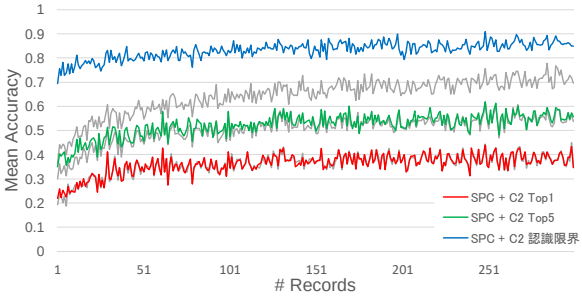

図 4　SPC + $C2$の Mean Accuracy 推移

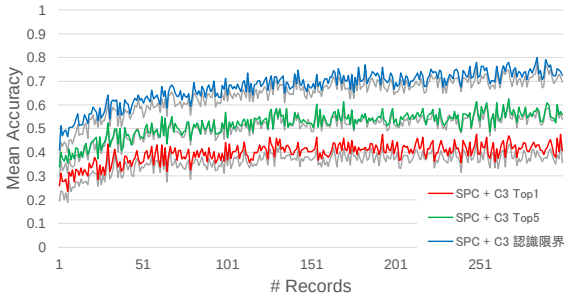

図 5　SPC + $C3$の Mean Accuracy 推移

を構築した．式(3)，(6)，(8)のw_1, w_2=0.85 と設定した．これは，ユーザ個人のデータベースV_u に重きを置いていることを意味する．評価には FLD-CLS を用いた．

4 手法による Mean Accuracy の推移を表 1 及び，結果を図 3 ～ 5 に示す．ただし，SPC の Top 1，Top 5，認識限界は各図に灰色の曲線で示している．

どの手法においても，t が増加するにつれて個人適応が進み，認識精度と認識限界は上昇している．

SPC + $C1$ではどの期間においても，認識限界が最も高くなっている．一方で，SPC に比べて Top 1，Top 5，の精度が低くなっている．これは加えた多くの他のユーザデータがノイズとして働いてしまったのではないかと考えられる．

SPC + $C2$は SPC + $C1$には劣るものの SPC に比べて各期間で認識限界が 30~50%向上した．さらにt_1~t_{50}において，Top 5 は SPC に比べて約 3%向上した．Top 1 は同様の精度を示した．これは，デフォルトクラスのみを選択したことにより，ノイズとして働きづらくなったと考えられる．

SPC + $C3$は Top 1 は約 3%向上したが，認識限界の向上は SPC + $C2$, SPC + $C3$に比べて微小であった．Top 1 の精度が上がるように加える他のユーザデータを選択しているため，Top 1 の精度が向上するのは当然ではあるが，その上がり幅は小さく 1 人のデータを加えることでは効果が薄いと考えられる．

6. まとめ

画像認識において，現実世界では逐次個人適応の認識限界を向上させるため，コミュニティ内の他のユーザデータを利用することで新規クラスを減らし，認識精度及び認識限界の向上することが可能かを検討した．SPC + $C1$では認識限界が大幅に向上したが，認識精度は低下した．SPC + $C2$ は認識限界が向上し，認識精度もわずかに向上し，ある程度有効であると考えられる．SPC + $C3$は認識精度の向上はしたものの，認識限界の向上は微小であった．今後の展望として，ユーザの特徴毎に用いるデータベースを変更したり，他人のユーザがノイズとならないように集約したりすることなどを検討している．

謝辞

本研究の一部は，JST CREST JPMJCR1686 の支援を受けた．

参考文献

[1] O. Russakovsky, J. Deng, H. Su. J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Brenstein, A. C. Berg and L. Fei-Fei, “ImageNet large scale visual recognition challenge”, IJCV, Vol.115, No.3, pp. 211-252, (2015).

[2] 堀口 翔太, 天野 宗佑, 相澤 清晴, 小川 誠, “画像認識器の逐次個人適応”, 信学技報, PRMU, Vol.116, No.461, pp. 149-154 (2016).

[3] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar and J. Yang: “PFID: Pittsburgh fast-food image dataset”, ICIP, pp. 289–292 (2009).

[4] Y. Matsuda, H. Hoashi and K. Yanai, “Recognition of multiple-food images by detecting candidate regions”, ICME, pp. 25–30 (2012).

[5] L. Bossard, M. Guillaumin and L. Van Gool, “Food-101–mining discriminative components with random forests”, ECCV, pp. 446-461 (2014).

[6] O. Beijbom, N. Joshi, D. Morris, S. Saponas and S. Khullar, “Menu-Match: Restaurant-specific food logging from images”, WACV, pp. 844–851 (2015).

[7] H. Kagaya, K. Aizawa and M. Ogawa: “Food detection and recognition using convolutional neural network”, ACM MM, pp. 1085–1088 (2014).

[8] M. Bolanos and P. Radeva, “Simultaneous Food Localization and Recognition on Egocentric Images”, ICPR, pp. 3140-3145 (2016).

[9] T. Miyazaki, G. C. de Silva and K. Aizawa, “Image-based calorie content estimation for dietary assessment”, ISM, pp.363–368 (2011).

[10] A. Myers, N. Johnston, V. Rathod, A. Korattikara and A. Gordan, “Im2Calories: towards an automated mobile vision food diary”, ICCV, pp. 1233–1241 (2015).

[11] J. Chen and C.-w. Ngo, “Deep-based ingredient recognition for cooking recipe retrieval”, ACM MM, pp. 32–41 (2016).

[12] M. Wilber, I. S. Kwak, D. Kriegman and S. Belongie, “Learning concept embeddings with combined human machine expertise”, ICCV, pp. 981–989 (2015).

[13] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I.Weber, A. Torralba, “Learning Cross-modal Embeddings for Cooking Recipes and Food Images”, CVPR (2017).

[14] K. Aizawa and M. Ogawa, “FoodLog: Multimedia tool for healthcare applications”, IEEE MultiMedia, Vol.22, No.2, pp. 4-8 (2015).

[15] S. loffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift”, ICML, pp. 448-455 (2015).

[16] 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).