FACE ALBUM: TOWARDS AUTOMATIC PHOTO MANAGEMENT BASED ON PERSON IDENTITY ON MOBILE PHONES

Yuansheng Xu[⋆] *Fangyue Peng*[⋆] *Yu Yuan*[†] *Yizhou Wang*[⋆]

[⋆] Nat'l Engineering Laboratory for Video Technology Cooperative Medianet Innovation Center
Key Laboratory of Machine Perception (MoE) Sch'l of EECS, Peking University, Beijing, 100871, China
[†] Beijing University of Posts and Telecommunications
Email: xys-tc@hotmail.com, 752829006@qq.com, {1300012910, Yizhou.Wang}@pku.edu.cn

ABSTRACT

We implement a new photo management system '*Face Album*' on mobile phones, which organizes photos by person identity, as is shown in Fig. 1. We automatically group faces into clusters to release user workload. Our system is composed of two pools: a certain pool with reliable clusters consisting of faces from same identity, and an uncertain pool containing faces that are lacking in evidence to be recognized. Constantly as new faces increase, the certain pool and uncertain pool work together to either assign new faces to existing clusters or discover new identities in the album. In addition, user interaction is introduced for some deviation corrections. Experiments indicate that our results are close to offline hierarchical clustering method while a subjective survey shows our photo management system is favored by users.

Index Terms— Face Recognition, Mobile Album, Online Clustering

1. INTRODUCTION

Nowadays, more and more photos are taken by and stored on mobile phones, thus management for these photos becomes increasingly important. Since people are main subject of daily mobile photographing, organizing photos by person is expected.

We consider this problem as annotating faces in an album, which has been studied in past literatures [1, 2, 3, 4, 5] and some annotation systems were proposed including Apple's iPhoto[6]. Some methods, e.g. [2, 3], exploit clustering or partial clustering on a photo set. Then users tag on each cluster to annotate faces. However, these methods are proposed for offline photo tagging which is not suited to the fact users will continuously take photos and enlarge the album. Some works, e.g. [1] and Apple's iPhoto which are designed for practical album management, can handle continuous photo-input. They list some unlabeled faces to users for annotation.

This work is supported in part by 973-2015CB351800, NSFC-61527804, NSFC-61421062, NSFC-61210005.

(a) Main Interface (b) Editing Interface

Fig. 1. Screen capture of *Face Album*. (a)User can choose one face identity, and browse the photos in this sub-album by sliding. (b)User can identify an uncertain face by choosing one of the recommended face sub-album or creating a new one.

When a user annotates a face, they will recommend similar photos to the user, so that a few operations can annotate a number of photos. As these systems are semi-automatic, most critical tasks rely on human decisions (e.g. a new face cluster can only be created by human).

We expect a photo management system on mobile phones to be highly automatic and also be able to deal with continuous photo stream. Thus, two main difficulties emerge: 1) how to recognize new faces of existing identities and 2) how to discover new identities in album.

As for the first difficulty, face recognition, which has been improved a lot in recent years [7, 8, 9, 10], can be applied. The task face verification has achieved 99.47% accuracy by [8] on the *LFW* dataset[11]. However, identifying a face from a set of candidate faces[12] is still challenging in that N candidates are much more confusing than one single candidate[13]. To solve this problem, we constrain each recognized identity with a certain number of faces as its evidence. Then a new

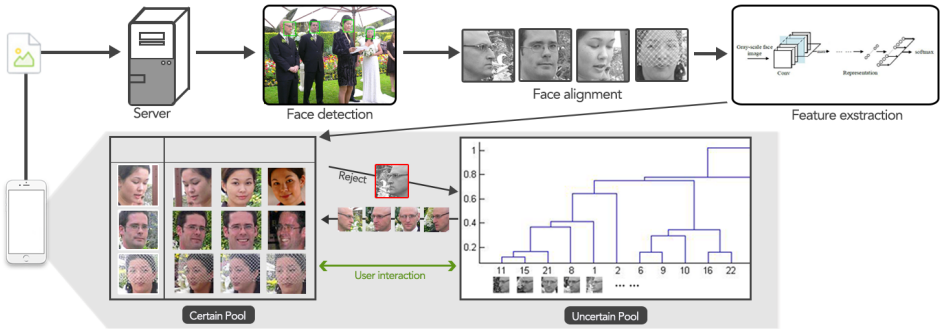


Fig. 2. The framework and work flow of *Face Album*.

face is compared with face clusters instead of candidate faces, which is more robust. The similarity of a face to a cluster is defined based on the overlap of the face's near neighbors with the cluster, which can reduce the influence caused by noises and false positives or hard cases. In our system, we propose a certain pool to contain these clusters of recognized faces.

As for the second difficulty of discovering new identities, creating a new identity from a face once it matches no cluster is too arbitrary. On one hand, the new face may be noise or hard case. On the other hand, even though it is a new identity, simply making it a new cluster is contrary to our design of certain pool. So we adopt a delayed decision strategy for these ambiguous faces to wait for further evidence. Thus we design an uncertain pool to reserve these faces. We periodically try assigning these faces again and finding new identities from them. Partial clustering is performed to find good clusters with small inner distance and certain size instead of a global clustering result, which is computationally efficient as well.

To conclude, we design our automatic photo management system - *Face Album* with a certain pool and an uncertain pool. These two pools work together to automatically group faces into different clusters even with continuous photo input. We obtain a good performance on real-life photos with the interaction of the two pools. Besides, our system also introduces user interaction to tag faces, correct misidentified faces and assign uncertain faces to existing identities.

2. FACE ALBUM SYSTEM

As illustrated in Fig.2, our system follows a classical server-client model. The server takes charge of the main computation work including detecting faces in photos and extracting recognition features from detected faces. The client maintains one certain pool and one uncertain pool and automatically 1) assigns a face to one of the clusters in certain pool, and 2) performs clustering in uncertain pool to find promising clusters in it. As shown in Fig.2, after receiving new face information, the client first tries to assign it to some cluster in the certain pool. If the assignment fails, this face will be put into the uncertain pool. As the client periodically performs clustering in the uncertain pool, a new identity will be found if some faces form a convincing cluster. Then these faces will be moved to the certain pool as a new identity. Beside this automatic album management, manual interaction is allowed to correct misidentified faces in the certain pool and to identify faces in the uncertain pool.

2.1. Face Recognition

On the server we follow the classical face recognition pipeline of face detection, face alignment and feature extraction[13]. We exploit facial point detector as in [14]. With facial key points detected, a face image patch is normalized to 128×128 as the input to CNN proposed in [7]. This CNN can derive a compact representation with features of only 256 dimensions, which is computationally efficient for mobile platform. After we implement the architecture B by *Caffe*[15], we follow the procedure in [7] to train the model. Training data are the face images from CASIA-WebFace[16] dataset. Each face is normalized to 144×144, and randomly cropped into 128×128 patches as the input of the first convolution layer. Finally, we obtain our deep feature. By simply using cosine distance, a 98.13% verification accuracy is achieved on *LFW* dataset, which indicates that the 256-dimensional feature is discriminative.

2.2. Assignment in Certain Pool

We denote x_i as the recognition feature of face i. The distance $d(i, j)$ between two faces i and j is defined as their cosine distance:

$$d(i,j) = 1 - \frac{x_i \cdot x_j}{\|x_i\|\|x_j\|} \tag{1}$$

The certain pool contains all the clusters: $\{C_i\}, i = 1 \ldots N$, where C_i is a cluster of faces.

We use the Epsilon Near Neighbors($\epsilon - NN$), neighbors within the distance ϵ of a face, to decide whether or not a new face x_i should be assigned to a specific cluster C. We calculate a similarity of x_i to C as:

$$S(x_i, C) = \frac{| N_\epsilon(x_i) \cap C |}{| C |}, \tag{2}$$

where $N_\epsilon(x_i)$ is the $\epsilon - NN$ of x_i:

$$N_\epsilon(x_i) = \{j \mid d(i,j) < \epsilon\}. \quad (3)$$

The similarity focus on shared part of x_i's neighbors and cluster C. Since x_i may fall on the edge of a cluster, it may not be close to every face in the cluster but part of it. Also, such similarity can greatly reduce the influence caused by false positive faces and noises in cluster C since they won't affect the shared part much.

We then compute the similarities of x to all the clusters in the certain pool and choose the largest one. We assign x to the cluster with largest similarity if the similarity exceeds a threshold δ.

2.3. Clustering in Uncertain Pool

We periodically find reliable clusters in the uncertain pool and move these clusters to the certain pool. A reliable cluster should have two properties: 1) pairwise distances in the cluster are small, and 2) the size of the cluster is no less than S_{min}. Thus, we adopt agglomerative hierarchical clustering[17] and, for property 1, stop clustering if no pair of clusters is closer than ϵ.

Algorithm 1 Hierarchical Clustering

Input: Distance Matrix of Faces
Output: Reliable Clusters
1: Each Face As an Individual Cluster.
2: **while** $MinimumDistanceOfTwoClusters < \epsilon$ **do**
3: Merge the closest two clusters.
4: Update the distance matrix to reflect the distance between the new cluster and the original clusters.
5: **end while**
6: **return** The clusters with size over S_{min}

The clustering is shown as Algorithm 1. At the beginning we consider each face as an individual cluster. Then we keep merging the closest pair of clusters, one pair a time, until no pair is closer than ϵ. The distance of two clusters is defined as average pairwise distance of all pairs from different clusters. Then, we will move all the clusters with size over S_{min} to the certain pool as new identities.

Time complexity of hierarchical clustering algorithm is $O(N^2 \log N)$[17], where N is the size of uncertain pool. Since we only merge clusters with a distance less than ϵ, this algorithm will finish at early stage, which is efficient in use.

2.4. User Interface

In this album, photos are classified by different face identities. User can name each sub-album and choose one face identity to browse the photos in this sub-album by sliding to left or right (as shown in Fig.1(a)). When browsing a photo, user can press 'Edit' to correct some deviations. In the 'Edit' interface, user can take two main actions: 1) Delete misidentified faces in a cluster; 2) Give one uncertain face an identity.

By pushing button 'Certain pool', the system shows some faces for each identity from its sub-album. If the classification is wrong, user can delete the face from this sub-album. As a result, this face will be moved to the uncertain pool. While pushing button 'Uncertain pool', 2 similar identities are recommended to each uncertain face. With a single click, users can either assign this uncertain face to one of the recommended identities or create a new one. Clicking on 'Finish' can go back to the previous interface.

3. EXPERIMENTS AND EVALUATION

We evaluate our *Face Album* with both objective experiments and subjective evaluation. In this section, we first introduce the datasets, then report our results on them and compare them with a baseline face image management system and an offline clustering algorithm. At last we present our subjective survey results. We give up using iPhoto as our baseline because it is hard to quantify iPhoto's performance due to its semi-automaticity. The baseline system we use is proposed as a simple probe-gallery identification protocol[12]. This baseline system maintains N clusters of faces in the gallery. For each new face, the system compares it N times with each representative. If the most similar one exceeds threshold, the new face will be included in the cluster, otherwise a new cluster will be created. We also compare our system to offline hierarchical clustering(HC), which performs clustering to the whole dataset.

3.1. Datasets

We test on two datasets: *LFW* and People In Photo Albums (*PIPA*) Dataset[18]. **The *LFW* dataset** includes 13,233 images of faces from 5,749 individuals. We use one half for training and the other half for testing. **The *PIPA* dataset** contains real-life photos collected from Flickr photo albums. The training set consists of 17,000 photos while the testing set contains 7,868 photos of 581 individuals. We perform our face detector on *PIPA* and register 52% of the labeled instances since the dataset is very challenging (even back view included). Finally the training set contains 11,920 instances and the testing set contains 6,073 instances with labels and 180 non-face images as false alarms.

3.2. Objective Experiments and Results

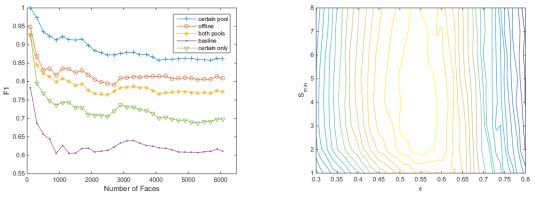
We evaluate the two pools over the quality of the face clusters we obtain. In the uncertain pool, each face is considered as an independent cluster. We compute pairwise Recall and Precision[19] over all the clusters and thus compute F1 to evaluate the clusters. Purity is also introduced for evaluation. We compare our results to two other methods, the *baseline*

Method	Recall	Precision	F1	Purity
baseline	0.569	0.676	0.618	0.747
offline HC	0.868	0.950	0.907	0.918
ours	0.838	0.923	0.879	0.905

Table 1. Experiment results on *LFW* dataset

system and the *offline HC* method as introduced before. We use the same distance measure in all these methods and select a different ϵ for each method by obtaining the best F1 score on training data. The parameter δ of our system is empirically set to 0.5, which implies at least half members in a cluster support the new face. The parameter S_{min} is set to 4 which will be explained later.

Results on *LFW* database are shown in Table 1. Our system achieves a better result than *baseline* system and a close performance to *offline HC* method. Considering the online background of our system, the close performance to the *offline HC* method is of great value.



(a) F1 value over time (b) Influence of parameters

Fig. 3. Experiment results on *PIPA* dataset.

We also evaluate our system on *PIPA* dataset. Since *PIPA* dataset can simulate realistic situation of daily album, we record the performance with every 200 photos as a batch processed, as shown in Figure 3(a). The performance of clusters in the certain pool is also concerned, represented as *certain pool*, since these clusters will be directly shown to users. Also, we test the performance of only using certain pool without uncertain pool, represented as *certain only*. With the same assignment method as described before, every time a new face is rejected, we will assign it as a new cluster in certain pool.

As is shown in Fig.3(a), the *certain only(green line)* method reports a lower performance over time than the whole system(yellow line) but a higher performance than the *baseline(purple line)* system, indicating that our assignment strategy works well. Meanwhile, it's the uncertain pool that balances with the certain pool to achieve a overall better result. Surprisingly, we discover that the performance in the *certain pool(blue line)* is even better than that of *offline HC(orange line)* method. This reflects that the clusters in certain pool, which are the selected faces shown to users, are highly accurate.

Here we demonstrate the influence of parameter S_{min} and ϵ to our system. Fig.3(b) is the contour map of F1 score on *PIPA* dataset with respect to S_{min} and ϵ. There is an obvious plateau which indicates a robust parameter selection. When we set S_{min} to 1, our system degrade into *certain only*, leading to a sharp decline in performance. And when we choose S_{min} among 3 to 5, we get a wide range of ϵ with acceptable performance.

Finally, we add the noise images to *PIPA* dataset and test it over the methods. The result shows that our system is robust to noise since 173 of the 180 noise images remain in the uncertain pool while the 7 rest images form an independent cluster in the certain pool with no other images included. No noise image is falsely mixed up with any face cluster.

3.3. Subjective Evaluation

In the subjective evaluation, 14 subjects are asked to report their user experience of face annotation on our *Face Album* application. 78 photos about a wedding are provided and the subjects have 15 minutes to get familiar with the major characters in this event. Then they are asked to gradually add the photos in three steps. Meanwhile, they should correct mismatched faces and manually create albums for unidentified characters until the albums of the major characters are organized well. In the end, we conduct a UMUX[20] survey with the subjects. The UMUX is a four-item Likert scale used for the subjective assessment of an applications perceived usability. The items (as shown in Figure 4) have seven scale steps from 1 (strongly disagree) to 7 (strongly agree).

Question	Score
1. [Face Ablum's] capabilities meet my requirements.	
2. Using [Face Ablum] is a frustrating experience.	
3. [Face Ablum] is easy to use.	
4. I have to spend too much time correcting things with [Face Ablum].	

Fig. 4. Survey Questions

With scores collected, we recode them as hundred mark system as in [20] and get a mean UMUX score of 86.90. UMUX is designed to provide results similar to those obtained by SUS[21]. According to [21], the score indicates that our system is good.

4. CONCLUSIONS

We implement a novel photo management system *Face Album* to provide service for browsing photos of a particular person. To address the issue of continuous photo input, we propose two pools of faces: certain pool and uncertain pool. Confusing faces will be reserved in uncertain pool and can be identified later with the help of new faces and interaction of these two pools. This makes our identification highly reliable and results in a good performance in objective experiments. As a practical system, we also introduce user interaction for some deviation correction and subjective survey has demonstrated that our photo management system is favored by users.

5. REFERENCES

[1] Longbin Chen, Baogang Hu, Lei Zhang, Mingjing Li, and HongJiang Zhang, "Face annotation for family photo album management," *International Journal of Image and Graphics*, vol. 3, no. 01, pp. 81–94, 2003.

[2] Yuandong Tian, Wei Liu, Rong Xiao, Fang Wen, and Xiaoou Tang, "A face annotation framework with partial clustering and interactive labeling," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[3] Chunhui Zhu, Fang Wen, and Jian Sun, "A rank-order distance based clustering algorithm for face tagging," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 481–488.

[4] Hong-Wun Jheng, Bor-Chun Chen, Yan-Ying Chen, and Winston Hsu, "Automatic facial image annotation and retrieval by integrating voice label and visual appearance," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1001–1004.

[5] Jingyu Cui, Fang Wen, Rong Xiao, Yuandong Tian, and Xiaoou Tang, "Easyalbum: an interactive photo annotation system based on face clustering and re-ranking," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 367–376.

[6] Apple, *macOS - Photos*, `http://www.apple.com/mac/iphoto/`.

[7] Xiang Wu, Ran He, and Zhenan Sun, "A lightened cnn for deep face representation," *arXiv preprint arXiv:1511.02683*, 2015.

[8] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2892–2900.

[9] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.

[10] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[11] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller, "Learning to align from scratch," in *NIPS*, 2012.

[12] Lacey Best-Rowden, Hu Han, Christina Otto, Brendan F Klare, and Anubhav K Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 12, pp. 2144–2157, 2014.

[13] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[14] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[16] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[17] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, "Data mining cluster analysis: Basic concepts and algorithms," 2013.

[18] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev, "Beyond frontal faces: Improving person recognition using multiple cues," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 4804–4813.

[19] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al., *Introduction to information retrieval*, vol. 1, Cambridge university press Cambridge, 2008.

[20] Kraig Finstad, "The usability metric for user experience," *Interacting with Computers*, vol. 22, no. 5, pp. 323–327, 2010.

[21] Aaron Bangor, Philip Kortum, and James Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.