

Face Retriever: Pre-filtering the Gallery via Deep Neural Net

Dayong Wang and Anil K. Jain

Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824, U.S.A.

{dywang, jain}@msu.edu

Abstract

Face retrieval is an enabling technology for many applications, including automatic face annotation, de-duplication, and surveillance. In this paper, we propose a face retrieval system which combines a k -NN search procedure with a COTS matcher (PittPatt¹) in a cascaded manner. In particular, given a query face, we first pre-filter the gallery set and find the top- k most similar faces for the query image by using deep facial features that are learned with a deep convolutional neural network. The top- k most similar faces are then re-ranked based on score-level fusion of the similarities between deep features and the COTS matcher. To further boost the retrieval performance, we develop a manifold ranking algorithm. The proposed face retrieval system is evaluated on two large-scale face image databases: (i) a web face image database, which consists of over 3,880 query images of 1,507 subjects and a gallery of 5,000,000 faces, and (ii) a mugshot database, which consists of 1,000 query images of 1,000 subjects and a gallery of 1,000,000 faces. Experimental results demonstrate that the proposed face retrieval system can simultaneously improve the retrieval performance (CMC and precision-recall) and scalability for large-scale face retrieval problems.

1. Introduction

In the digital era, more and more face images are captured, stored, and shared over the Internet. Given a large collection of face images, an interesting challenge is automatic face retrieval, which aims to find one or several face images of interest from the collection [2, 15]. Face retrieval is useful for many applications, including automatic face annotation, de-duplication, surveillance, *etc.* In general, face retrieval can be solved based on two schemes: using face recognition models and using facial features for k -NN search. In the first scheme, followed by most

¹PittPatt, a face recognition company, was acquired by Google in 2011.

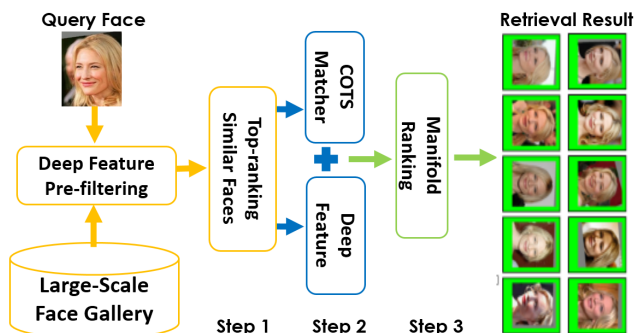


Figure 1. Illustration of the proposed cascaded face retrieval system.

face recognition techniques, a model of facial similarity is learnt and then used to rank faces in the dataset (gallery) according to their similarities with the query face image. Over the past few decades, face recognition in constrained environments has been extensively studied, with the result that commercial face SDKs have demonstrated impressive accuracies in these scenarios. However, one limitation of model-based schemes is the scalability to large-scale galleries, as the recognition models need to compare the query image with all face images in the gallery. While this problem can be addressed by parallelizing the matching process, we propose an alternative which filters the gallery.

The scalability problem can be addressed by directly using learned facial features for k -NN search, an approach that follows content-based image retrieval techniques [15]. Face images are represented as feature vectors and then efficiently indexed, searched, and re-ranked in the feature space. Based on traditional handcrafted features, the k -NN search schemes generally perform worse than state of the art model-based schemes. For example, in [1] the mean Average Precision (mAP²) of a nearest neighbor algorithm is 0.66, while the commercial software PittPatt achieves the best performance with mAP of 0.86. A recent breakthrough in feature representation is “deep learning”, which includes

²The mean of the average precision scores of a set of queries.

a family of machine learning algorithms that model high-level abstractions in data by employing architectures composed of multiple non-linear transformations. Many studies have reported state of the art performance by applying deep learning techniques to facial images [10, 11, 9, 12].

In this paper, we propose a face retrieval system which combines a k -NN search procedure with a COTS matcher (PittPatt³) in a cascaded manner, as shown in Fig. 1. In the first step, we first pre-filter the gallery set by using the deep learning based facial representations and find the top- k most similar faces to the query face image. Then, we re-rank the top- k most similar faces by fusing the similarities from deep facial features with the similarities output by PittPatt. In the third step, we use a manifold ranking algorithm to fully explore the intrinsic structural information among the top- k face images.

The main contributions of this paper are:

- A cascaded scheme for large-scale face retrieval problem, which addresses the performance and scalability simultaneously.
- Improved retrieval results on both relatively unconstrained (web downloaded images) and constrained (mugshots) large-scale facial image databases, which contain 5 millions and 1 million gallery images respectively.

2. Related Work

Face verification has been extensively studied in multimedia, computer vision, and biometrics [7]. Many studies prior to the last decade focused on face recognition and verification for relatively constrained acquisitions. Recently, many significant publications have appeared dealing with face recognition in unconstrained conditions using the Labeled Faces in the Wild (LFW) dataset [5, 9, 10, 11, 12]. For example, Sun et al. [10] constructed a ConvNet for face identification and verification. The overall facial feature representation was constructed by concatenating 25 low-dimensional deep features generated from 25 independent deep ConvNets. Finally, a joint Bayesian verification model was learned. Sun et al. achieved the best verification accuracy to date on the LFW database (99.15%), which is close to the human performance of 99.20%. Instead of directly using 2D aligned faces, a deep feature representation framework based on 3D alignment techniques was proposed in [12]. Given a 2D face image, they first generated a 3D aligned frontal-view face, followed by learning a deep ConvNet; the outputs of the second to last fully-connected layer were used as the face representations, demonstrating comparably high accuracy to [10] on LFW.

Most of the existing feature representations learned from deep learning frameworks are fed to supervised recognition/verification models. But, only a few of these studies have evaluated the retrieval performance of the deep feature representations [3, 13]. Donahue et al. [3] evaluated whether features extracted from the activation of deep ConvNets, trained in a fully supervised fashion for object recognition tasks, could be re-purposed for novel generic recognition tasks. Wan et al. [13] mainly focused on evaluating the performance of deep features on general content-based image retrieval tasks. In this paper, we aim to train deep ConvNets on a large set of facial images and evaluate their retrieval performance on large-scale face image databases.

An important problem in using k -NN search for face retrieval is the implementation of an efficient indexing and search scheme, which depends on the feature representation [15, 2, 14]. For example, following the Bag-of-Words representation scheme, Wu et al. [15] proposed a face retrieval system using component-based local features with identity-based quantization to deal with scalability issues. Chen et al. [2] proposed to use a sparse coding algorithm and partial identity information to generate component-based semantic codewords. Using global facial feature vectors, Wang et al. [14] adopted the Locality-Sensitive Hash (LSH) [4] algorithm for indexing and retrieval, which can also be adopted for the deep face representations proposed in this paper.

3. Face Retrieval System

3.1. Deep Face Representation

The architecture of our deep ConvNet is inspired by [8, 13]. In [8], the deep ConvNets were trained on about 1 million images from 1,000 object categories. This system won the *ImageNet Large Scale Visual Recognition Challenge* in 2012 with a top-5 test error rate of 15.3%. The architecture of our deep ConvNet is shown in Fig. 2 (a), which mainly consists of three parts: i) convolution layers and max-pooling layers, ii) fully connected layers, and iii) output classification layer.

The input layer accepts raw intensity values of the face image pixels. All faces are first aligned and cropped by the PittPatt SDK and resized to 256×256 . To reduce over-fitting, a data augmentation procedure is performed: several transformations of the input image are generated with translations and horizontal reflections by extracting 224×224 random patches from the original 256×256 image.

Following the input layer, there are five convolutional layers. The first two convolutional layers are followed by a response normalization and a max-pooling layer. The third and fourth convolutional layers are interconnected

³Version 5.1

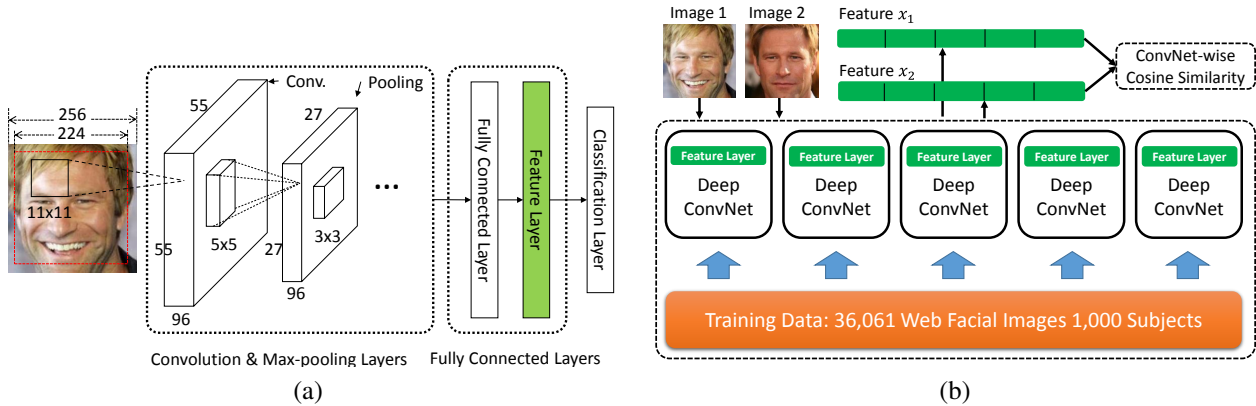


Figure 2. The proposed deep convolutional neural network (ConvNet) and deep feature similarity calculation.

without any intervening pooling or normalization. The fifth convolutional layer is followed by a max-pooling layer. Following the convolutional layers, there are two fully-connected layers with 1,024 neurons. Finally, the last layer is a softmax classification layer. The number of neurons in the classification layer is equal to the number of classes in the training set.

When using deep ConvNets to generate feature representations, generally, the last fully-connected layer produces the best retrieval performance [13]. In our deep ConvNet architecture shown in Fig. 2 (a), the last fully-connected layer is the Feature Layer; we denote the feature representation generated from the Feature Layer as “deep feature” (DF). Since there are millions of parameters in the deep ConvNet, it requires an enormous number of training images, and associated computational cost, to obtain a stable deep ConvNet. In order to reduce the training time and overcome the limitations of the number of training images, we initialize our deep ConvNets with the parameters of a pre-trained ImageNet-based ConvNet [13].

To improve the performance and robustness of the learned deep features, we train five deep ConvNets independently. For face retrieval, given a pair of facial images, we first generate their deep features with the learned deep ConvNets. Then, we compute the overall deep feature similarity by averaging the ConvNet-wise cosine similarities. The procedure of calculating deep feature similarity is shown in Fig. 2 (b). Since the similarity among two deep feature vectors only depends on the cosine similarity function, the retrieval problem can be easily accelerated by adopting any one of the many fast search and indexing techniques [4]. All the experiments on deep learning are conducted on a Linux server with Tesla K20 GPUs. To implement the system, we use the public-domain C++ implementation of ConvNet⁴.

⁴<https://code.google.com/p/cuda-convnet/>

3.2. Manifold Ranking

The similarity of deep features and the similarity output by PittPat SDK both aim to measure the possibility that two face images belong to the same subject. However, ranking based on the similarities of the query face image to the top- k face images alone does not make use of the intrinsic structure of the top- k most similar facial images. To fully exploit the intrinsic structure information, we adopt a manifold ranking algorithm that further utilizes the pairwise similarities between all of the top- k most similar face images [6].

Let \mathbf{x}_q and $X = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ denote the query face and its top- k most similar faces retrieved from the gallery, respectively, where $i = 1, 2, \dots, k$ is the ranking position. We obtain candidate face sets by sorting the fused similarity of deep features (DF) and PittPat (PP) in descending order. The fused similarity is computed by a sum-of-score fusion rule with equal weights applied to the similarities of PP and DF after z-score normalization. A sparse n -NN graph G is constructed over X , where two vertices are connected if and only if one is among the n nearest neighbors of the other. In our experiments, we used the top-5 nearest neighbors of \mathbf{x}_i in X to construct the graph G . We denote the edge affinity matrix of graph G as $W \in \mathbb{R}^{k \times k}$, where W_{ij} is the similarity score between \mathbf{x}_i and \mathbf{x}_j . Let $\hat{\mathbf{y}}$ and \mathbf{y} denote the initial and refined ranking scores, respectively. The manifold ranking algorithm aims to determine \mathbf{y} by minimizing the following objective function:

$$\min_{\mathbf{y}} \sum_{i,j=1}^k W_{ij} (\mathbf{y}_i - \mathbf{y}_j)^2 + \lambda \sum_{i=1}^k (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \quad (1)$$

The first term in Eqn. 1 enforces that similar face images have close ranking scores, the second term incorporates the initial scores into the final result, and λ is a penalty parameter. A simple example of the manifold ranking algorithm is shown in Fig. 3. The optimization problem in Eqn. 1 has a closed-form solution: $\mathbf{y} = (L + \Lambda)^{-1} \Lambda \hat{\mathbf{y}}$,

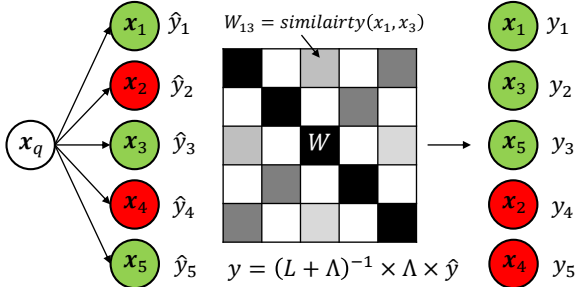


Figure 3. An example of manifold ranking. We pre-filtered the gallery set and retrieved the top- k ($k=5$ in this example) most similar face image. Firstly, we ranked these similar faces by fusing the similarities of deep features and PittPatt, and obtained the initial ranking score \hat{y} . Then, we constructed a n -NN similarity graph ($n=2$ in this example). Finally, we generated the refined ranking score y by adopting the proposed manifold ranking algorithm based on the initial ranking score \hat{y} and the edge affinity matrix W .

where $L = D - W$ is the graph Laplacian matrix of W , D is a diagonal matrix with the diagonal elements as $D_{ii} = \sum_{j=1}^k W_{ij}$, and Λ is a diagonal matrix with $\Lambda(i, i) = \lambda$

4. Experimental Protocol

4.1. Datasets

The proposed facial image retrieval system is evaluated on two databases. (i) *Web-based Face Database*, where web downloaded images are relatively unconstrained in a natural environment; (ii) *Mugshot Database*, where all the faces are captured in constrained environments and mostly have a frontal view.

Web-based Face Database is composed of three sources: WLFDB [14], LFW [5] and general web faces.

For WLFDB database, we randomly select 1,000 subjects for the *Training Set* to learn the deep ConvNets. Since WLFDB is constructed by querying the Google search engine and contains a large number of noise images (*i.e.* images that do not actually contain the person of interest), we only use his/her top 50 Google-ranked images for each subject.

For LFW database, we remove any overlapping subjects with the *Training Set*. We also remove all the subjects who only have 1 images. For each left subject, we randomly collect half of his/her images for the *Query Set* and use the left images for the *Gallery Set*. As a result, there are 3,880 query images of 1,507 subjects in the query set and 3,854 mated images in the gallery set.

To build a large-scale gallery set, we use a crawler to download millions of web images, which are filtered to only include images with faces detectable by the OpenCV implementation of the Viola-Jones face detector. As a result, there are 5 millions of web facial images in the



Figure 4. Examples of face images in the (a) web-based face database, and (b) mugshot database.

gallery set in total.

More details of the web-based face database are shown in Table 1. Several examples are shown in Fig. 4 (a).

	Source	# Subjects	# Images
Training Set	WLFDB [14]	1,000	36,061
Query Set	LFW [5]	1,507	3,880
Gallery Set	LFW [5]	1,507	3,854
	General Web Faces	-	4,996,146
Total		-	5,000,000

Mugshot Database. The Mugshot database was derived from the Pinellas County Sheriffs Office (PCSO) database, which is collected in the state of Florida, U.S.A. Firstly, we randomly select 2,000 images of 1,000 subjects. For each subject, we randomly add one image into the *Query Set* and the other into the *Gallery Set* as the mate image. Secondly, we randomly select 10,000 images of 1,000 subjects for the *Training Set*. Finally, we randomly select 900,000 images for background images in the *Gallery Set*. The details of the mugshot database are shown in Table 2. Several examples are shown in Fig. 4 (b).

	# Subjects	# Images
Training Set	1,000	10,000
Query Set	1,000	1,000
Gallery Set	1,000	1,000
	-	900,000
Total		1,000,000

Remark 1 There are no overlapping subjects between the training set and the query and mated images in the gallery.

4.2. Experimental Settings

The learning rate of deep ConvNets is initialized with 0.01, which is finally reduced to 0.0001. In our experiments, we train 5 deep ConvNets independently. To measure the similarity between deep features, we first compute ConvNet-wise cosine similarities, then compute

the overall DF similarity by averaging the five ConvNet-wise similarities.

For the manifold ranking algorithm, we construct the graph of similar faces with the top-5 nearest samples. The penalty parameter λ is set to 1. The initial ranking scores \hat{y} are set as: $\hat{y}_i = \frac{n-i}{n}$ where i is the ranking position. We find the above ranking scores are empirically better than using similarity scores. For the mugshot database, we evaluate the retrieval performance by using a Cumulative Match Characteristic (CMC) curve, since only one mate image exists for each query image.

5. Experimental Results

5.1. Comparison of Distance Measures

In the proposed face retrieval system, we use cosine similarity to compute the distance between two deep facial representations, as Fig. 2 (b). In this experiment, we evaluate the performance of two other common distance measures: L1 and L2, as shown in Table. 3. We use a 100K gallery set for this experiment.

Table 3. Comparison of Distance Measures.

	Cosine	L2	L1
mAP of DF	0.448	0.309	0.283

We can observe that cosine similarity achieves the best performance.

5.2. Comparison of Fusion Schemes

In the proposed face retrieval framework, we first retrieve the top- k most similar images by deep feature similarities, then re-rank the similar faces by fusing the similarities of deep features and PittPatt. We denote such a unifying scheme as DF→PP. In this experiment, we also evaluate two other kinds of unifying schemes, as follows:

- PP→DF: We first use the similarities output by PittPatt to find the top- k most similar faces over the whole retrieval set, then we fuse the similarities of deep features and PittPatt over the top- k most similar faces for re-ranking.
- PP+DF: We fuse the similarities of deep features and PittPatt over the whole retrieval set, and rank all the images in retrieval set.

In PP→DF and DF→PP schemes, we retrieve the top- $k=1,000$ most similar faces then re-rank them. Experimental results for web-based face database are shown in Fig. 5. We use a 100K gallery set for this experiment. We can draw several observations from these results.

Firstly, deep features alone (*i.e.* ConvNet-wise cosine similarities) remarkably outperform commercial PittPatt

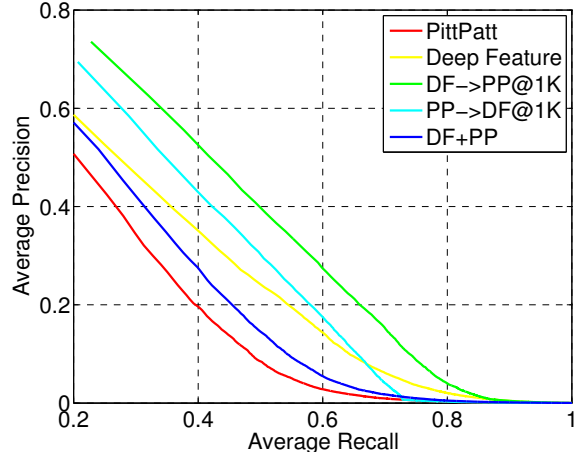


Figure 5. Comparison of fusion schemes for web-based face database. Top- $k=1,000$ most similar faces were first retrieved in PP→DF and DF→PP schemes with PP and DF, respectively.

SDK, which indicates that the deep ConvNets can efficiently learn and abstract high-level facial representations by training on a large set of web facial images.

Secondly, by fusing the similarities of deep features and PittPatt, we get better performance. However, the performance of the three fusion schemes (DF→PP, PP→DF, and DF+PP) varies. We notice that DF→PP produces the best performance, which is the one used in the proposed face retrieval system. The fusion scheme DF+PP performs worse than DF→PP and also requires more computation cost.

5.3. Impact of Number of Retrieved Faces, k

For the proposed face retrieval system, one important parameter is k , the number of retrieved similar faces from the gallery set, as it affects both the retrieval performance and computational cost/scalability. In this experiment, we examine the impact of k by increasing it from 100 to 1,500. Based on the re-ranking results of the DF→PP scheme, we sequentially adopt the manifold ranking algorithm, which is denoted as DF→PP+MR. The experimental results are shown in Fig. 6. We use a 100K gallery set for this experiment. We can draw several observations from the results.

Firstly, we observe that increasing the number of retrieved faces k , generally leads to better overall retrieval performance, while the improvement becomes marginal when k is larger than 1,000 (see Fig. 6 (a)). However, increasing the number of retrieved faces k will increase the computational cost, since we have to adopt PittPatt SDK for $O(k)$ times in the DF→PP and DF→PP+MR schemes (the similarity matrix W can be pre-computed offline). In our experiments, we set $k=1,000$ as a trade-off between performance and efficiency. Secondly, the manifold rank-

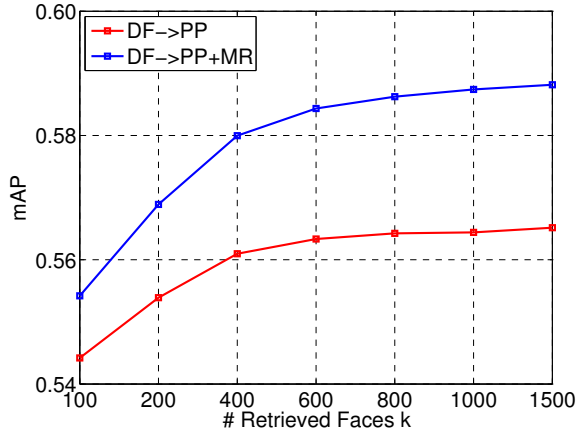


Figure 6. Impact of retrieved faces k for web-based face database.

ing algorithm improves the retrieval performance, which indicates that the propagation of similarities among the retrieved similar faces is helpful to increase the ranking values of the correct faces.

5.4. Retrieval Performance

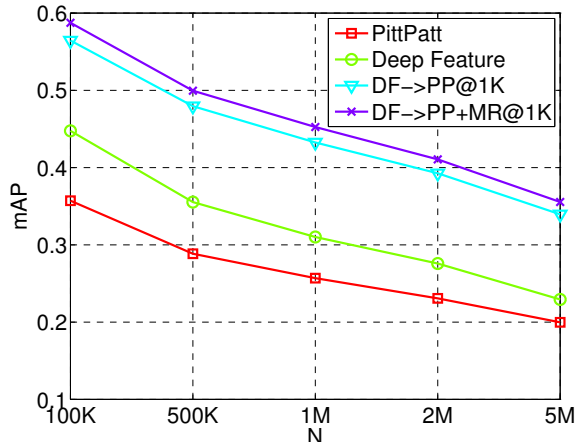


Figure 7. Retrieval Performance vs. Database Size, N .

In this experiment, we examine the relationship between the retrieval performance and the size of the gallery, which is increased from 100,000 to 5,000,000. The experimental results are shown in Fig. 7. The retrieval performance decreases with increased size of the gallery set. The proposed scheme consistently achieve the best performance.

5.5. Retrieval Time

In general, the retrieval time of all schemes increases linearly with the size of the gallery set. Table 4 shows evaluation of the running time of the various face retrieval schemes for the largest gallery set (5M). For one face query, the average retrieval time of PP is around 440 seconds compared with 14.2 seconds for DF->PP+MR. This indicates

the proposed face retrieval system is able to address the performance and scalability issues, simultaneously.

Table 4. The average retrieval time of one query face image for web-based face database with 5M face images in gallery set.

	PP	DF	DF->PP	DF->PP+MR
Time (s)	444.3	6.65	11.55	14.2
mAP	0.200	0.230	0.340	0.355

5.6. Experiments on Mugshot Retrieval

On the mugshot database, we first evaluate the performances of three kinds of deep features generated with the following three schemes on a 100K gallery set:

- DF1: Using the web-based face ConvNets that are trained and used in our previous experiments.
- DF2: Training new ConvNets with the mugshot database training set. The deep ConvNets are initialized with an ImageNet-based ConvNet.
- DF3: Training new ConvNets with the mugshot database training set. The deep ConvNets are initialized with a facial ConvNet that was used in our previous experiments.

Table 5. Comparison of various deep features for mugshot retrieval.

	DF1	DF2	DF3
Accuracy @ Rank-1	0.468	0.617	0.704

The experimental results are shown in Table 5. We can observe that the performance of DF1 scheme is very poor, which indicates the web face based ConvNets have bias when used for the mugshot database. To overcome the bias, five new deep ConvNets are trained with the mugshot training set in DF2 and DF3 schemes. Comparing with initializing the deep ConvNets with the ImageNet-based ConvNet (DF2), substantially better performance can be achieved by using the facial ConvNet for initialization (DF3). It indicates that the facial ConvNet, which was trained on a large set of web face images, provides a better start-point for deep ConvNet training on mugshot image databases.

Based on DF3 feature representation scheme, we evaluate the face retrieval performance of the proposed scheme on the whole 1M gallery set, as shown in Fig. 8. Since there is a single mate image for all query images, in this experiment, we do not adopt the manifold ranking algorithm. Several observations can be drawn from the results. First, the performance of PittPatt on the mugshot database is substantially better than on the web-based face images, which indicates that it has been well developed on face images captured in constrained environments. Second,

the recognition rate of DF3 is worse than PittPatt for small rank positions, and better at high rank positions. Third, the proposed face retrieval system (DF3→PP) obtains the best performance over all ranking positions.

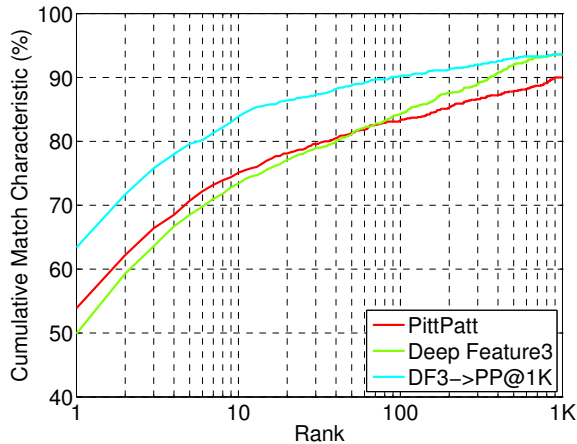


Figure 8. Retrieval performance on the mugshot database. Top- $k=1,000$ most similar face images are first retrieved from the gallery and then re-ranked. The size of the gallery database is 1,000,000.

6. Conclusions

In this paper, we propose a face retrieval system which combines a k -NN search procedure and a COTS matcher in a cascaded manner. We propose to pre-filter the gallery set by using deep face representations and rank the top- k most similar faces by fusing the similarities of deep features and a COTS matcher (PittPatt). A manifold ranking algorithm is also adopted to refine the retrieval performance. Experimental results demonstrate the proposed face retrieval system can simultaneously address the performance and scalability issues and achieve favorable retrieval results on two large-scale face databases collected in different scenarios.

Acknowledgement

This research was supported by the National Institute of Justice (NIJ) grant 2011-IJ-CX-K057.

References

- [1] B. Becker and E. Ortiz. Evaluating open-universe face identification on the web. In *Computer Vision and Pattern Recognition Workshops*, 2013. 1
- [2] B. Chen, Y. Kuo, Y. Chen, K. Chu, W. Hsu. Scalable face image retrieval using attribute-enhanced sparse codewords. *IEEE Trans. on Multimedia*, 2012. 1, 2
- [3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional

activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013. 2

- [4] W. Dong, Z. Wang, W. Josephson, M. Charikar, and K. Li. Modeling LSH for performance tuning. In *CIKM'08*. 2, 3
- [5] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report, University of Massachusetts, Amherst, 2007. 2, 4
- [6] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas. Image retrieval via probabilistic hypergraph ranking. In *CVPR'10*, 2010. 3
- [7] S. Z. Li and A. K. Jain (eds.). *Handbook of Face Recognition, 2nd Edition*. Springer-Verlag, 2011. 2
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS'12*, 2012. 2
- [9] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *ICCV'13*, 2013. 2
- [10] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *CoRR*, abs/1406.4773, 2014. 2
- [11] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR'14*. 2
- [12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR'14*, 2014. 2
- [13] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *ACM MM'14*, 2014. 2, 3
- [14] D. Wang, S. Hoi, Y. He, J. Zhu, T. Mei, and J. Luo. Retrieval-based face annotation by weak label regularized local coordinate coding. *IEEE Trans. PAMI*, 2014. 2, 4
- [15] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum. Scalable face image retrieval with identity-based quantization and multi-reference re-ranking. In *CVPR'10*, 2010. 1, 2
- [16] X. T. W. Zhang. Image reranking by example: A semi-supervised learning formulation. Technical report, The Chinese University of Hong Kong, 2014.

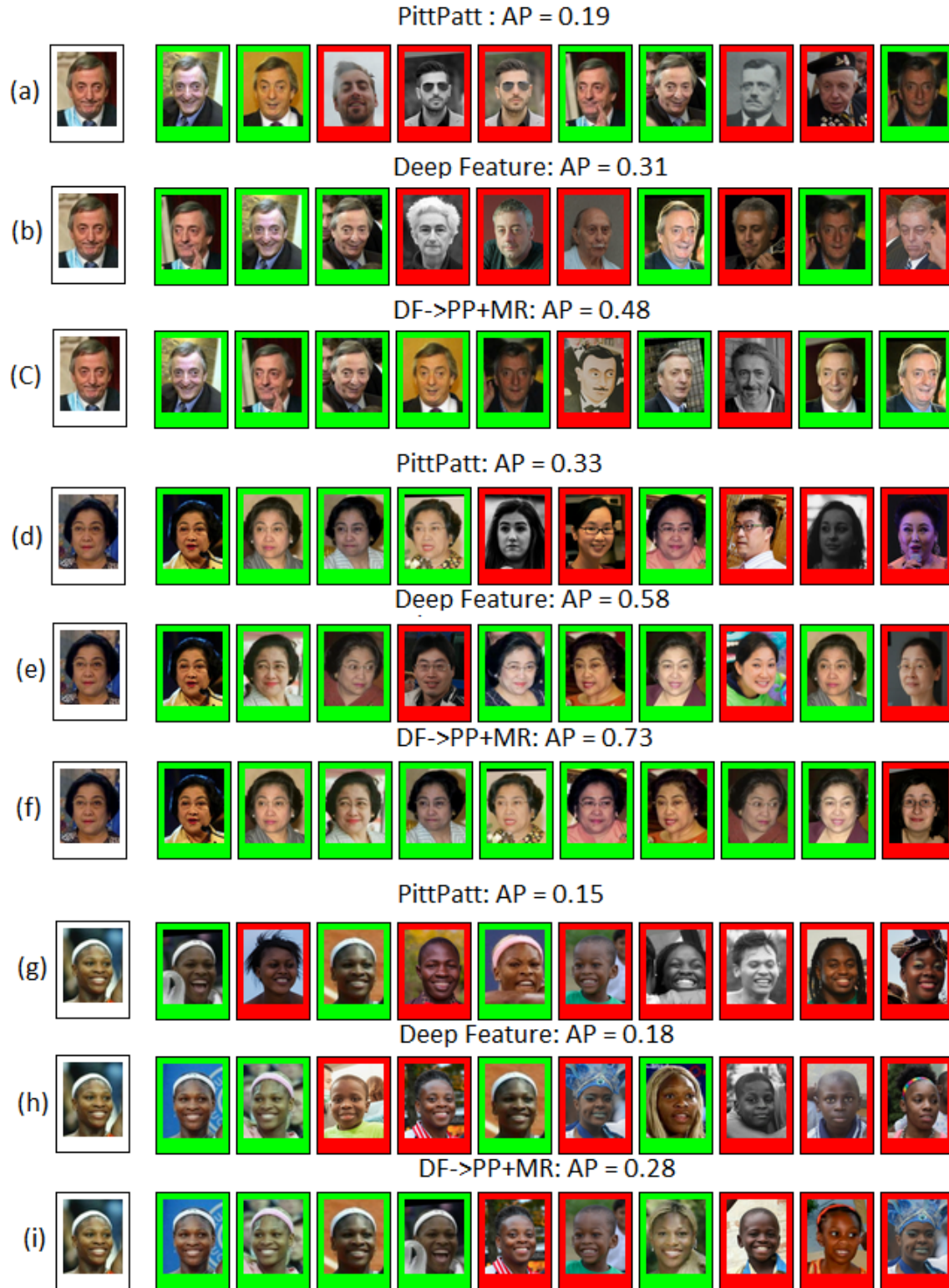


Figure 9. Face retrieval results for three face queries on web-based face database. (a), (d) and (g) top-10 retrieved faces by using PittPatt SDK (PP); (b), (e) and (h) top-10 retrieved faces by using deep features (DF); (c), (f) and (i) top-10 retrieved faces by using the proposed cascaded face retrieval system (DF→PP+MR): top- $k=1,000$ most similar faces were first filtered by deep features, and then were ranked by fusing the similarities of deep features and PittPatt SKD, finally, were re-ranked by adopting a manifold ranking (MR) algorithm. The size of the gallery set is 5,000,000.