

Face Search at Scale

Dayong Wang, *Member, IEEE*, Charles Otto, *Student Member, IEEE*, Anil K. Jain, *Fellow, IEEE*

Abstract—Given the prevalence of social media websites, one challenge facing computer vision researchers is to devise methods to search for persons of interest among the billions of shared photos on these websites. Despite significant progress in face recognition, searching a large collection of unconstrained face images remains a difficult problem. To address this challenge, we propose a face search system which combines a fast search procedure, coupled with a state-of-the-art commercial off the shelf (COTS) matcher, in a cascaded framework. Given a probe face, we first filter the large gallery of photos to find the top- k most similar faces using features learned by a convolutional neural network. The k retrieved candidates are re-ranked by combining similarities based on deep features and those output by the COTS matcher. We evaluate the proposed face search system on a gallery containing 80 million web-downloaded face images. Experimental results demonstrate that while the deep features perform worse than the COTS matcher on a mugshot dataset (93.7% vs. 98.6% TAR@FAR of 0.01%), fusing the deep features with the COTS matcher improves the overall performance (99.5% TAR@FAR of 0.01%). This shows that the learned deep features provide complementary information over representations used in state-of-the-art face matchers. On the unconstrained face image benchmarks, the performance of the learned deep features is competitive with reported accuracies. LFW database: 98.20% accuracy under the standard protocol and 88.03% TAR@FAR of 0.1% under the BLUFR protocol; IJB-A benchmark: 51.0% TAR@FAR of 0.1% (verification), rank 1 retrieval of 82.2% (closed-set search), 61.5% FNIR@FAR of 1% (open-set search). The proposed face search system offers an excellent trade-off between accuracy and scalability on galleries with millions of images. Additionally, in a face search experiment involving photos of the Tsarnaev brothers, convicted of the Boston Marathon bombing, the proposed cascade face search system could find the younger brother's (Dzhokhar Tsarnaev) photo at rank 1 in 1 second on a 5M gallery and at rank 8 in 7 seconds on an 80M gallery.

Index Terms—face search, unconstrained face recognition, deep learning, large face collections, cascaded system, scalability.

1 INTRODUCTION

Social media has become pervasive in our society. One popular aspect of social media is the sharing of personal photographs. Facebook, in a 2013 white paper, revealed that its users have uploaded more than 250 billion photos, and are uploading 350 million new photos each day¹. To enable automatic tagging of these images, accurate and robust face recognition capabilities are needed. Given an uploaded photo, Facebook and Google's tag suggestion systems automatically detect faces and then suggest possible name tags based on the similarity between facial templates generated from the input photo and previously tagged photographs in their datasets. In the law enforcement domain, the FBI plans to include over 50 million photographs in its Next Generation Identification (NGI) dataset², with the goal of providing investigative leads by searching the gallery for images similar to a suspect's photo. Both tag suggestion in social networks and searching for a suspect in criminal investigations are examples of face search at scale (Fig. 1). We address the large-scale face search problem in the context of social media and other web applications where face images are generally unconstrained in terms of pose, expression, and illumination [1], [2].

A major focus in face recognition has been to improve unconstrained face recognition accuracy, particularly on the Labeled Faces in the Wild (LFW) benchmark [3]. But, the problem of scale in face recognition has not been adequately addressed³. It is now generally agreed that the small size of the LFW dataset (13,233 images of 5,749 subjects) and the limitations in the LFW protocol do not address the two major challenges in large-

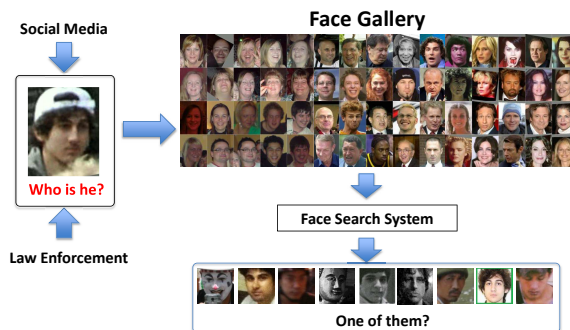


Fig. 1. An example of large-scale face search problem.

scale face search: (i) loss in search accuracy, and (ii) increase in computational complexity with increase gallery size.

The typical approach to scalability (used in e.g. content-based image retrieval [2]) is to represent objects with feature vectors and employ an indexing or approximate search scheme in the feature space. A vast majority of face recognition approaches, irrespective of the representation scheme, are ultimately based on fixed length feature vectors, so employing feature space methods is feasible. However, some techniques for improving face recognition accuracy, such as pairwise comparison models (e.g. Joint-Bayes [5]), are not compatible with feature space approaches. Additionally, most COTS face recognition SDKs define pairwise comparison scores but do not reveal the underlying feature vectors, so they are also incompatible with feature-space approaches. Therefore, using a feature space based approximation method alone may not be sufficient for large-scale search.

To address the tradeoff between search performance and search

1. <http://phys.org/news/2016-01-facebook.html>

2. goo.gl/UYIT8p

3. Our preliminary work on this topic appeared in the Proc. IEEE International Conference on Biometrics (ICB), Phuket, June 2015 [4]. A technical report describing this work appeared in [37]

time at scale (80M face images used here), we propose a cascaded face search framework (Fig. 2). In essence, we decompose the search problem into two steps: (i) a fast filtering step, which uses an approximation method to return a short candidate list, and (ii) a re-ranking step, which re-ranks the candidate list with a slower pairwise comparison operation, resulting in a more accurate search. The fast filtering step utilizes a deep convolutional network (ConvNet), which is an efficient implementation of the architecture in [6], with product quantization (PQ) [7] to speed up retrieval. For the re-ranking step, a COTS face matcher (one of the top performers in the 2014 NIST FRVT [8]) is used. The main contributions of this paper are as follows:

- An efficient deep convolutional network for face recognition, trained on a large public domain data (CASIA [6]), which improves upon the baseline results reported in [6].
- A large-scale face search system, leveraging the deep network representation combined with a state-of-the-art COTS face matcher in a cascaded scheme.
- Studies on three face datasets of increasing complexity: PCSO mugshot dataset, LFW dataset (only contains faces detectable by Viola-Jones face detector), and the IJB-A dataset (contains several faces which are not detectable by the Viola-Jones detector).
- The largest face search experiments conducted to date on the LFW [3] and IJB-A [9] benchmarks, with an 80M gallery.
- Using face images of the Tsarnaev brothers involved in the Boston Marathon bombing as queries, we show that Dzhokhar Tsarnaev's photo could be identified at rank 8 when searching against the 80M gallery.

2 RELATED WORK

Face search has been extensively studied in multimedia and computer vision literature [20]. Early studies primarily focused on faces captured under constrained conditions, e.g. the FERET dataset [14]. However, due to the growing need for strong face recognition capability in the social media context, ongoing research is focused on faces captured under more challenging conditions in terms of large variations in pose, expression, illumination and aging, similar to images in the LFW [3] and IJB-A [9] datasets.

The three main challenges in large-scale face search are: i) *face representation*, ii) *approximate k -NN search*, and iii) *gallery selection and evaluation protocol*. For face representation, features learned from deep convolutional networks (deep features) have been shown to saturate performance on the standard LFW evaluation protocol⁴.

The best accuracy to date on LFW is reported by Baidu [21] which is 99.77%; it leverages 70 deep learning models trained on 1.2M images of 18K individuals. A comparable result (99.63%) was achieved by Google [22] using a deep model and a training set with over 150M images of 8M subjects. It has even been reported that deep features exceed the human face recognition accuracy (99.20% [10]) on the LFW dataset. In order to recognize faces in web downloaded images, we also adopt a deep ConvNet based face representation by improving the architecture outlined in [6].

Given our goal of using deep features to filter a large gallery to a small set of candidate face images, we use an approximate

k -NN search method to improve scalability. There are three main approaches for approximate face search:

- *Inverted Indexing*. Following the traditional bag-of-words representation, Wu et al. [2] designed a component-based local face representation for inverted indexing. They first split aligned face images into a set of small blocks around the detected facial landmarks and then quantized each block into a visual word using an identity-based quantization scheme. The candidate images were retrieved from the inverted index of visual words. Chen et al. [1] improved the search performance in [2] by leveraging human attributes.
- *Hashing*. Yan et al. [15] proposed a spectral regression algorithm to project facial features into a discriminative space; a cascaded hashing scheme (similarity hashing) was used for efficient search. Wang et al. [23] proposed a weak label regularized sparse coding to enhance facial features and adopted the Locality-Sensitive Hash (LSH) [24] to index the gallery.
- *Product Quantization (PQ)*. Unlike inverted indexing and hashing which require index vectors to be stored in main memory, PQ [7] is a compact discrete encoding method that can be used either for exhaustive search or inverted indexing search. In this work, we adopt product quantization for fast filtering.

In the literature, face search systems have mainly been evaluated under the closed-set protocol (Table 1), which assumes that the subject in the probe image is present in the gallery. However, in many large scale applications (e.g., surveillance and watch list scenarios), open-set search protocol, where the probe subject may not be present in the gallery, is more appropriate. Recognizing this, several new protocols for unconstrained face recognition based on the LFW dataset have been proposed, including the open-set identification protocol [18] and the Benchmark of Large-scale Unconstrained Face Recognition (BLUFR) protocol [19]. However, even for these two protocols, the gallery sizes are fairly small (a few thousand images), due to the inherent small size of the LFW dataset. Table 1 shows that the largest face gallery reported in the literature to date is about 1M, which is not even close to being a representative of social media and forensic applications. To tackle these two limitations, we evaluate the proposed cascaded face search system with an 80M face gallery⁵ under both closed-set and open-set protocols.

3 FACE SEARCH FRAMEWORK

Given a probe image, a face search system aims to find the top- k most similar face images in the gallery. To handle large galleries containing tens of millions of images, we propose a cascaded face search structure similar to [13], [25], designed to speed up the search process while achieving acceptable accuracy.

Figure 2 outlines the proposed face search architecture consisting of three main steps: i) *template generation* module which extracts features for the N gallery faces (offline) as well as for the probe face (online); ii) *face filtering* module which compares the probe representation against the gallery representations using product quantization to retrieve the top- k most similar candidates

⁵ We got the links to these web images from a research collaborator, who was unwilling to release these images publicly. Our approach will also work on any other gallery size.

⁴ <http://vis-www.cs.umass.edu/lfw/results.html>

TABLE 1
A summary of face search systems reported in the literature

| Authors | Probe | | Gallery | | Dataset | Search Protocol |
|-------------------------|----------|------------|----------|------------|--|-------------------|
| | # Images | # Subjects | # Images | # Subjects | | |
| Wu et al. [2] | 220 | N/A | 1M+ | N/A | LFW [3] + web faces ^a | closed set |
| Chen et al. [1] | 120 | 12 | 13, 113 | 5, 749 | LFW [3] | closed set |
| | 4,300 | 43 | 54,497 | 200 | Pubfig [10] | closed set |
| Miller et al. [11] | 4,000 | 80 | 1M+ | N/A | FaceScrub [12] + Yahoo Images ^b | closed set |
| Yi et al. [13] | 1,195 | N/A | 201,196 | N/A | FERET [14] + web faces | closed set |
| Yan et al. [15] | 16,028 | N/A | 116,028 | N/A | FRGC [16] + web faces | closed set |
| Klare et al. [17] | 840 | 840 | 840 | 840 | LFW [3] | closed set |
| | 25,000 | 25,000 | 25,000 | 25,000 | PCSO [17] | closed set |
| Best-Rowden et al. [18] | 10,090 | 5,153 | 3,143 | 596 | LFW [3] | closed & open set |
| Liao et al. [19] | 8,707 | 4,249 | 1,000 | 1,000 | LFW [3] | closed & open set |
| Proposed System | 7,370 | 5,507 | 80M+ | N/A | LFW [3] + web faces | closed & open set |
| | 14,868 | 4,500 | 80M+ | N/A | IJB-A [9] + web faces | closed & open set |

a. Face images are downloaded from the web and used to augment the gallery; different face search systems use their own web downloaded face datasets.

b. <http://labs.yahoo.com/news/yfcc100m/>

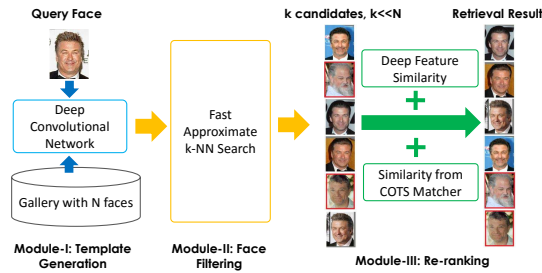


Fig. 2. Illustration of the proposed large-scale face search system.

($k \ll N$); and (iii) *re-ranking* module which fuses similarity scores of deep features with scores from a COTS face matcher to generate a new ordering of the k candidates. These three modules are discussed in detail in the remainder of this section.

3.1 Template Generation

Given a face image I , the template generator is a non-linear mapping function

$$\mathcal{F}(I) = \mathbf{x} \in \mathbb{R}^d \quad (1)$$

which projects I into a d -dimensional feature space. The discriminative ability of the template is critical for search system accuracy. Given state-of-the-art performance of deep learning techniques in various machine learning applications, particularly face recognition, we adopt deep learning for template generation.

The architecture of the proposed network (Fig. 3) is inspired by [6], [26]. There are four main differences between the proposed network and the one in [6]: i) input to the network is color images instead of gray scale images; ii) use of a robust face alignment procedure; iii) an additional data augmentation step that randomly crops a 100×100 region from the 110×110 input color image, followed by a horizontal reflection to generate additional images to train the network; and iv) deleting the contrastive cost layer for computational efficiency.

For the network's convolution layers, we adopt a very deep architecture [27] (10 convolution layers in total) and use filters with a small size (3×3). The small filter size reduces the total number of parameters to be learned, and the deep architecture

enhances the non-linearity of the network [27]. The network output is a 320-dimensional feature vector.

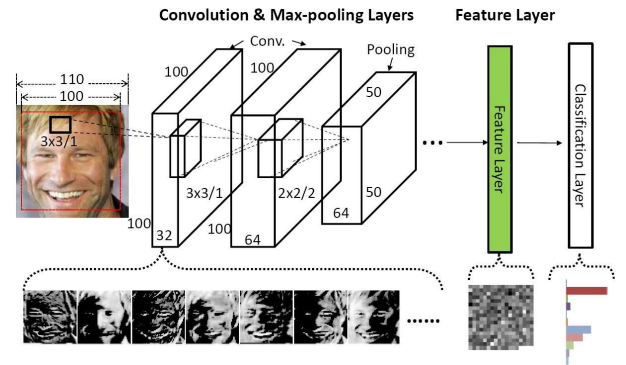


Fig. 3. Proposed deep convolutional neural network (ConvNet).

The input layer accepts the RGB pixel values of the aligned face images. Faces are aligned as follows: i. Use the DLIB⁶ implementation of Kazemi and Sullivan's ensemble of regression trees method [28] to detect 68 facial landmarks (see Fig. 4); ii. rotate the face in the image plane to make it upright based on the eye positions; iii. find a central point on the face (the blue point in Fig. 4) by taking the mid-point between the leftmost and rightmost landmarks; the center points of the eyes and mouth (red points in Fig. 4) are found by averaging all the landmarks in the eye and mouth regions, respectively; iv. center the faces along the x-axis, based on the central point (blue point); v. keep the aspect ratio and fix the position along the y-axis by placing the eye center point at 45% from the top of the image and the mouth center point at 25% from the bottom of the image, respectively; vi. resize the width and height of the image to 110×110 . Note that the computed midpoint is not consistent across pose. In faces exhibiting significant yaw, the computed midpoint will be different from the one computed in a frontal image, so facial landmarks are not aligned consistently across yaw.

Following the input layer, there are 10 convolutional layers, 4 max-pooling layers, and 1 average-pooling layer. Every pair of

6. <http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

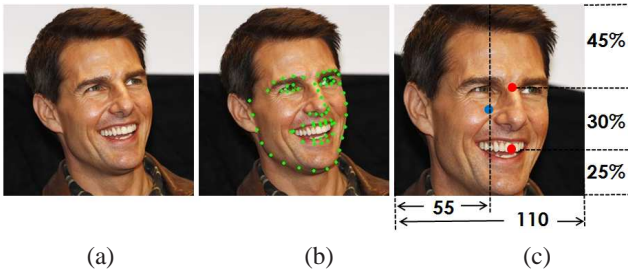


Fig. 4. A face image alignment example. The original image is shown in (a); (b) shows the 68 landmark points detected by the method in [28], and (c) is the final aligned face image, where the blue circle was used to center the face image along the x-axis, and the red circles denote the two points used for face cropping.

convolutional layers is grouped together and connected sequentially. The first four groups of convolutional layers are followed by a max-pooling layer with a 2×2 filter and a stride of 2, while the last group of convolutional layers is followed by an average-pooling layer with a 7×7 filter. The dimensionality of the feature representation layer is the same as the number of filters in the last convolutional layer. As discussed in [6], the ReLU [29] node produces a sparse vector, which is undesirable for a face representation layer. In our network, we use ReLU nodes [29] in all the convolutional layers, except the last one, which is combined with an average-pooling layer to generate a 320-dimensional face representation.

Although multiple fully-connected layers are used in [26], [29], in our network we directly feed the deep features generated by the feature layer to a P -way softmax, where P is the number of subjects in the training set. We regularize the feature representation layer using dropout [30], keeping 60% of the feature components as-is and randomly setting the remaining 40% to zero during training.

We use a softmax loss function for our network, and train it using the standard back-propagation method. The network is implemented using open source cuda-convnet⁷ library; weight decay is set to 5×10^{-4} . The learning rate for stochastic gradient descent (SGD) is initialized to 10^{-2} , and gradually reduced to 10^{-5} .

3.2 Face Filtering

Given a probe face I and a template generation function \mathcal{F} , finding the top- k most similar faces $C_k(I)$ in the gallery G is formulated as follows:

$$C_k(I) = \text{Rank}_k(\{\mathcal{S}(\mathcal{F}(I), \mathcal{F}(J_i)) | J_{i=1,2,\dots,N} \in G\}) \quad (2)$$

where N is the size of gallery G , \mathcal{S} is a function which measures the similarity of the probe face I and the gallery image J_i , and $\text{Rank}(\cdot)$ is a function that finds the top- k largest values in an array. The computational complexity of naïve face comparison functions is linear with respect to the gallery size N and the feature dimensionality d . To address large-scale search, approximate nearest neighbor (ANN) algorithms, which improve runtime without a significant loss in accuracy, have become popular.

Hashing based algorithms use compact binary representations to conduct an exhaustive nearest neighbor search in Hamming

space. Although multiple hash tables [24] can significantly improve performance and reduce distortion, their performance degrades quickly with increasing gallery size in face recognition applications. Product quantization (PQ) [7], where the feature template space is decomposed into a Cartesian product of low dimensional subspaces (each subspace is quantized separately) has been shown to achieve excellent search results [7]. Details of product quantization used in our implementation are described below.

Under the assumption that the feature dimensionality is a multiple of m , where m is an integer, any feature vector $\mathbf{x} \in \mathbb{R}^d$ can be written as a concatenation $(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m)$ of m sub-vectors, each of dimension d/m . In the i -th subspace $\mathbb{R}^{d/m}$, given a sub-codebook $\mathcal{C}^i = \{\mathbf{c}_{j=1,2,\dots,z}^i | \mathbf{c}_j^i \in \mathbb{R}^{d/m}\}$, where z is the size of codebook, the sub-vector \mathbf{x}^i can be mapped to a codeword \mathbf{c}_j^i in the codebook \mathcal{C}^i , with j as the index value. The index j can then be represented by a binary code with $\log_2(z)$ bits. In our system, each codebook is generated using the k -means clustering algorithm. Given all the m sub-codebooks $\{\mathcal{C}^1, \mathcal{C}^1, \dots, \mathcal{C}^m\}$, the product quantizer of feature template \mathbf{x} is

$$q(\mathbf{x}) = (q^1(\mathbf{x}^1), \dots, q^m(\mathbf{x}^m))$$

where $q^j(\mathbf{x}^j) \in \mathcal{C}^j$ is the nearest sub-centroid of sub-vector \mathbf{x}^j in \mathcal{C}^j , for $j = 1, 2, \dots, m$, and the quantizer $q(\mathbf{x})$ requires $m \log_2(z)$ bits. Given another feature template \mathbf{y} , the asymmetric squared Euclidean distance between \mathbf{x} and \mathbf{y} is approximated by

$$\mathcal{D}(\mathbf{y}, \mathbf{x}) = \|\mathbf{y} - q(\mathbf{x})\|^2 = \sum_{j=1}^m \|\mathbf{y}^j - q^j(\mathbf{x}^j)\|^2$$

where $q^j(\mathbf{x}^j) \in \mathcal{C}^j$, and the distances $\|\mathbf{y}^j - q^j(\mathbf{x}^j)\|$ are pre-computed for each sub-vector of \mathbf{y}^j , $j = 1, 2, \dots, m$ and each sub-centroid in \mathcal{C}^j , $j = 1, 2, \dots, m$. Since the distance computation requires $O(m)$ lookup and add operations [7], approximate nearest neighbor search with product quantizers is fast, and significantly reduces the memory requirements with binary coding. If no additional hashing scheme is used, this method is $O(N)$ with dataset size.

To further reduce the search time, a non-exhaustive search scheme was proposed in [7] and [31] based on an inverted file system and a coarse quantizer; the query image is only compared against a portion of the image gallery, based on the coarse quantizer. However, we found that non-exhaustive search significantly reduces face search performance when used with the proposed feature vector.

Two important parameters in product quantization are the number of sub-vectors m and the size of the sub-codebook z , which together determine the length of the quantization code: $m \log_2 z$. Typically, z is set to 256. To find the optimal m , we empirically evaluate search accuracy and time per query for various values of m , based on a 1 million face gallery and over 3,000 queries. We noticed that the performance gap between product quantization (PQ) and brute force search becomes small when the length of the quantization code is longer than 512 bits ($m = 64$). Considering search time, the PQ-based approximate search is an order of magnitude faster than the brute force search. As a trade-off between efficiency and effectiveness, we set the number of sub-vectors m to 64; the quantization code is $64 \log_2(256) = 512$ bits long.

Although we use product quantization to compute face similarity scores, we also need to pick a distance or similarity metric. We

7. <https://code.google.com/p/cuda-convnet2/>

evaluated cosine similarity⁸, L1 distance, and L2 distance using a 5M gallery. The cosine similarity achieves the best performance among those three metrics, although the normalized L2 distance has identical performance.

3.3 Re-Ranking

Given the short candidate list, the *re-ranking module* aims to improve search accuracy by using additional face matchers to re-rank the candidate list. In particular, given a probe face I and its top- k most nearest faces returned from the filtering module (denoted with $C_k(I)$), the k candidate faces are re-ranked by fusing the similarity scores from L different matchers. The re-ranking module is formulated as:

$$\text{Re-Rank}(\{\text{Fusion}(\mathcal{S}_{j=1,\dots,L}(I, J_i)) | J_{i=1,\dots,k} \in C_k(I)\}) \quad (3)$$

where \mathcal{S}_j is the j -th matcher, and Re-Rank function sorts top- k samples in the descending order. To make our system simple yet effective; we set $L = 2$ and generate the final similarity score using sum-rule fusion [32] between cosine similarity scores computed from the learned deep features and the similarity scores generated by the COTS face matcher⁹. To reduce the effect of scale in sum-rule fusion, we adopt z-score normalization [33] over the top- k similarity values for each face matcher, respectively.

The main benefit of combining the similarities derived from deep features and scores output by a COTS matcher is to utilize the strength of two different face representations. We noticed that the set of impostor face images that are incorrectly assigned high similarity scores by deep features and the COTS matcher do not overlap, suggesting their representations are complementary, which is necessary for the success of fusion [32]. Since COTS matchers are widely deployed in many real world applications [8], the proposed cascade fusion scheme can be easily integrated in existing applications to improve both scalability and performance.

3.3.1 Impact of Candidate Set Size (k)

In the proposed cascaded face search system, the size of candidate list k is a key parameter. In general, we expect the optimal value of k to be related to the gallery size N (a larger gallery would require a larger candidate list to maintain good search performance). We evaluate the relationship between k and N by computing the mean average precision (mAP) as the gallery size (N) is increased from 100K to 5M and the size of candidate list (k) is increased from 50 to 500K.

Fig. 6 (a) shows that the search performance, as expected, decreases with increasing gallery size. Further, for a fixed N , search performance initially increases, then drops off when k gets too large. The optimal candidate set size k scales linearly with the gallery size N . Because the plots in Fig 6 (a) flatten out for large k , a near optimal value of k (e.g., $k = 0.01N$) can drastically reduce the candidate list with a very small loss in accuracy.

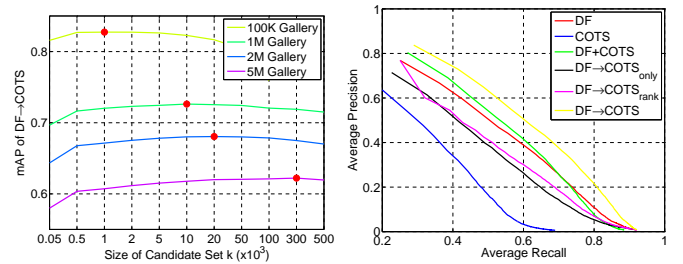
3.3.2 Fusion Method

Another important issue is the choice of fusion scheme to combine similarity scores from deep features (DF) and COTS. We empirically evaluated the following four fusion strategies:

- **DF+COTS:** Fusion of similarity scores based on deep features and the COTS matcher, without any filtering.

8. https://en.wikipedia.org/wiki/Cosine_similarity

9. We enforce the COTS matcher to compare two face images directly without using any metadata.



(a) Impact of candidate set size k (b) Various fusion strategies

Fig. 6. (a) Impact of candidate set size (k) as a function of the gallery size (N) on the search performance as measured in terms of Mean Average Precision (mAP). Red points mark the optimal value of k for different values of N . (b) Comparison of fusion strategies based on a 1M gallery and 3K probes.

- **DF→COTS:** Filter the gallery using deep features, then re-rank the candidate list based on fusion of similarity scores from deep features and the COTS matcher.
- **DF→COTS_{only}:** Only use the similarity scores of COTS matcher to rank the k candidate faces output by deep features.
- **DF→COTS_{rank}:** Filter the gallery using deep features, and rank k candidate faces using similarity scores of deep features and the COTS matcher, respectively, and then finally, combine two ranked lists using rank-level fusion. This is useful when the COTS matcher does not report similarity scores.

To keep the evaluation tractable, different fusion methods were evaluated using about 3K probes and a 1M face gallery. The *average precision vs. average recall* curves of these four fusion strategies are shown in Fig. 6 (b). As a baseline, we also show the retrieval performances of using DF and COTS alone. The fusion scheme (DF→COTS) consistently outperforms the other fusion methods, as well as simply using DF and COTS alone. Note that omitting the filtering step leads to poor retrieval results compared to the cascaded approach, which is consistent with results in the previous section: when k approaches N , the search accuracy decreases.

4 FACE DATASETS

We use one mugshot dataset and four different web face datasets in our experiments: PCSO, LFW [3], IJB-A [9], CASIA-WebFace [6] (abbreviated as “CASIA” in the following sections), and general web face images, referred to as “Web-Face”. We briefly introduce these datasets and show example face images from each dataset in Fig. 5.

- **PCSO:** This dataset is a subset of a large collection of mugshot images acquired from the Pinellas County Sheriff’s Office (PCSO) dataset, which contains 1,447,607 images of 403,619 subjects.
- **LFW [3]:** The LFW dataset is a collection of 13,233 face images of 5,749 individuals, downloaded from the web. Face images in this dataset contain significant variations in pose, illumination, and expression. All the images in this dataset contain faces that can be detected by the Viola-Jones face detector [3], [34].
- **IJB-A [9]** IARPA Janus Benchmark-A (IJB-A) contains 500 subjects with a total of 25,813 images (5,399 still

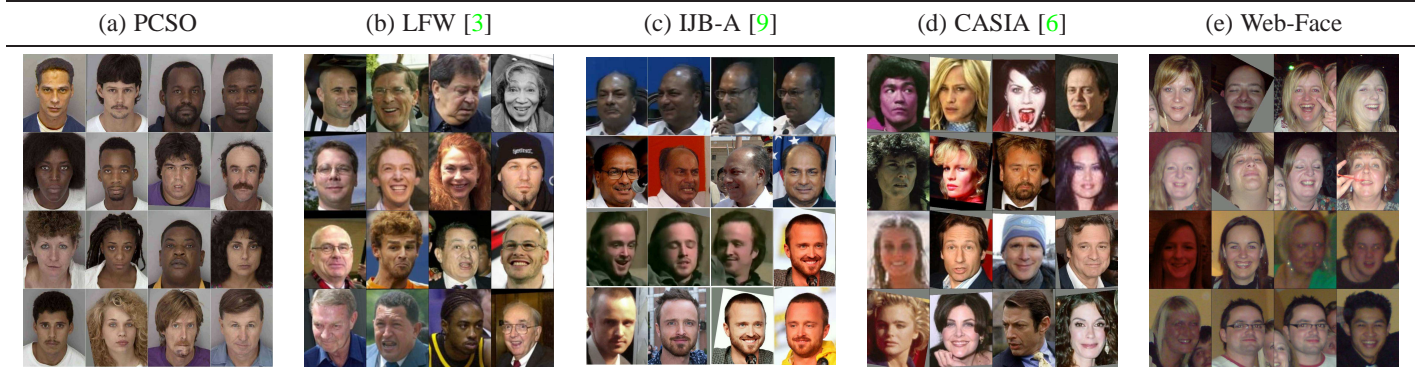


Fig. 5. Examples of face images in five face datasets used in our experiments.

images and 20,414 video frames). Compared to the LFW dataset, the IJB-A dataset is more challenging.

- **CASIA [6]** dataset provides a large collection of face images labeled with the subjects' identities, suitable for training deep networks. It contains 494,414 face images of 10,575 subjects. There are 22 overlapping subjects between the CASIA and IJB-A datasets. After removing the images of these 22 subjects and all images where face detection failed, we are left with 404,992 images of 10,553 subjects in the CASIA dataset.

4.1 Gallery Augmentation: 80 Million Web Faces

To conduct face search at scale, we used a crawler to automatically download millions of web images (The links to these web images were provided by a different research collaborator). Following that, we filtered out all the non-face-detectable web images with the DLIB face detector¹⁰. A total of 80 million face images were collected in this manner. Since these images are unlabeled, we use them to augment the gallery size in our large-scale search experiments. We call this database "Web-Face."

We first evaluate the DLIB face detector on the unconstrained face detection benchmark FDDB [39], which contains 5,171 faces extracted from 2,845 Yahoo news images. Using default parameters provided by the DLIB face detector, the number of detected false positive faces is 140, and the corresponding true positive rate (TPR) is about 81.6% using discrete score evaluation metric [39]. Fixing the same number of false positive faces, the true positive rate (TPR) of the best commercial face detector¹¹ and academic face detector [40] are around 91% and 89%, respectively.

We further manually examined 10,000 randomly drawn samples from the entire 80 million dataset. A total of 50 non-face images and 164 non-human faces were detected, which indicates that approximately 2.1% of the 80 million web downloaded face images may be false positives. In Figure 7 we show some examples of non-face and non-human face images from the Web-Face dataset.

We made a rough estimate of the number of identities in this dataset by combining the full LFW dataset with the 80 million images, performing clustering using the method outlined in [41], and selecting the distance threshold which attained the best possible performance on the LFW data. We acquired

approximately 6.7 million non-singleton clusters by this method, and consider that an estimated lower bound on the potential true number of discrete identities.



Fig. 7. Examples of false positive samples detected by DLIB face detector. (a) non-face images, (b) non-human face images.

5 FACE RECOGNITION EVALUATION

In this section, we first evaluate the proposed deep ConvNet on a mugshot dataset (PCSO), followed by evaluations on two publicly available unconstrained face recognition benchmarks (LFW [3] and IJB-A [9]) to establish its performance relative to state-of-the-art results.

5.1 Mugshot Evaluation

We evaluate the proposed deep model on mugshot data obtained from PCSO. Some example mugshots are shown in Fig. 5 (a). Mugshot faces are captured in constrained environments (e.g., a police station) with near frontal views of the face. We compare the performance of our deep features with a state-of-the-art COTS face matcher. The COTS matcher used here was designed to work with mugshot-style images, and is one of the top performers in the 2014 NIST FRVT [8].

Since mugshot data is qualitatively different from the CASIA [9] dataset that we initially used to train our deep network, we first retrained the network with a subset of 471,130 mugshot images of 29,674 subjects taken from the full PCSO mugshot dataset. For evaluation, we compared deep features with the COTS matcher on a test subset of the PCSO dataset containing 100,00 images of 63,670 subjects, which does not contain any of the subjects and images included in the training set. We conduct the face verification experiment using 10-fold cross-validation; there are approximately 340K genuine comparisons and 4 billion impostor comparisons, on average in each fold.

Experimental results show that the COTS matcher outperforms the deep features, especially at low False Accept Rates (FAR)¹².

10. <http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

11. <http://idl.baidu.com/>

12. False Accept Rate (FAR) is defined as the fraction of impostor pairs incorrectly accepted at a particular threshold.

For example, at a FAR of 0.01%, the True Accept Rates (TAR)¹³ of deep features and the COTS matcher are $98.6\% \pm 0.1\%$ and $93.7\% \pm 0.1\%$, respectively. However, a score-level fusion between the deep features and COTS scores results in an improved overall performance: $99.5\% \pm 0.3\%$ ¹⁴. This result suggests the following: (i) deep features are not superior to other state-of-the-art face recognition approaches for all tasks, especially near frontal photos with neutral expression and controlled illumination, and (ii) Deep features and state-of-the-art COTS matchers provide complementary information.

5.2 LFW Evaluation

While mugshot data is of interest in several applications, many other applications require recognizing more difficult, unconstrained face images. We now evaluate the proposed deep models on the benchmark LFW [3] unconstrained face dataset, using two protocols: the standard LFW [3] protocol and the BLUFR protocol [19].

5.2.1 Standard Protocol

The standard LFW evaluation protocol defines a verification experiment under 10-fold cross-validation with 300 genuine comparisons and 300 impostor comparisons per fold, involving a total of 7,701 images of 4,281 subjects. In Table 2, following the standard protocol, we present the mean verification accuracy of the proposed deep models and the same COTS face matcher evaluated in section 5.1 for the mugshot dataset. More evaluation results are available on the LFW leaderboard.¹⁵

TABLE 2
Performance of various face recognition methods on the standard LFW verification protocol.

| Method | #Nets | Mean accuracy \pm s.d. |
|---------------------|-------|--------------------------|
| COTS | N/A | $90.4\% \pm 1.3\%$ |
| Proposed Deep Model | 1 | $96.2\% \pm 0.9\%$ |
| Proposed Deep Model | 9 | $98.2\% \pm 0.6\%$ |

We notice that the COTS matcher performs poorly relative to the deep learning based algorithms. This is to be expected since unlike deep models, most COTS matchers have been trained to handle face images captured in constrained environments, e.g. mugshot or driver license photos. Almost all the top-ranking algorithms on the LFW leaderboard are deep learning based algorithms. The superior performance of deep learning based algorithms can be attributed to (a) large number of training images of large number of subjects ($> 100K$) [22], (b) data augmentation methods, e.g., use of multiple deep models [21], and (c) supervised learning algorithms, such as Joint-Bayes [5], used to learn a verification model for a pair of faces in the training set.

To generate multiple deep models, we first trained three deep ConvNets independently based on training data that was preprocessed using the alignment method in Section 3.1. In addition, we cropped six different sub-regions from the aligned face images (by centering the positions of the left-eye, right-eye, nose, mouth, left-brow, and right-brow) and trained six additional

networks. By combining these nine deep models together and using Joint-Bayes [5], the mean verification accuracy of our deep model improves to 98.20% from 96.20% for a single network using the cosine similarity. Despite relying only on publicly available training data, the performance of our deep model is competitive with state-of-the-art on the standard LFW protocol as shown on the leaderboard.

5.2.2 BLUFR Protocol

It has been argued that the standard LFW evaluation protocol is not appropriate for many face recognition applications, which require high True Accept Rates (TAR) at low False Accept Rates (e.g. FAR = 0.1%). In this experiment, we further evaluate the proposed deep models using the BLUFR [19] protocol, which defines both 10-fold cross-validation face verification and open-set identification tests involving larger number of genuine and impostor comparisons.

For face verification, in each trial, the test set contains 9,708 face images of 4,249 subjects, on average. As a result, over 47 million face comparison scores need to be computed in each trial. For open-set identification, in each trial, the genuine probe set contains 4,350 face images of 1,000 subjects, the impostor probe set contains 4,357 images of 3,249 subjects, on average, and the gallery set contains 1,000 images. Following the protocol in [19], we report the True Accept Rate (TAR) at a False Accept Rate (FAR) of 0.1% for face verification¹⁶. For open-set identification, we report the detection and identification rate (DIR) at Rank-1 corresponding to a False Accept Rate (FAR) of 1%. See Table 3 for results.

TABLE 3
Performance of various face recognition methods on LFW using the BLUFR protocol reported as True Accept Rate (TAR) and Detection and Identification Rate (DIR).

| Method | #Nets | TAR @FAR=0.1% | DIR@FAR=1% Rank=1 |
|---------------------|-------|--------------------|----------------------|
| Li et al. [6] | 1 | 80.3% | 28.9% |
| COTS | N/A | $60.0\% \pm 1.5\%$ | $37.9\% \pm 1.5\%$ |
| Proposed Deep Model | 1 | $85.0\% \pm 1.9\%$ | $49.1\% \pm 2.8\%$ |
| Proposed Deep Model | 9 | $89.8\% \pm 1.8\%$ | $55.9\% \pm 3.3\%$ |

We notice that the TAR at a FAR of 0.1% under the BLUFR protocol is much lower than the accuracies reported on the standard LFW protocol. For example, the performance of the COTS matcher is only 58.56% under the BLUFR protocol compared to 90.35% in the standard LFW protocol. This indicates that the performance metrics for the BLUFR protocol are more stringent than those of the standard LFW protocol; however, as previously discussed practical applications require good performance at low FAR operating points. The deep models still outperform the COTS matcher. Using cosine similarity and a single deep model, our method achieves better performance (83.08%) than the one in [6], which indicates that our modifications to the network design (using RGB input, random cropping, and improved face alignment) help boost the recognition performance. Our performance is further improved to 88.03% when we fuse nine deep models. In this experiment, the Joint-Bayes approach [5] did not improve accuracy. In the open-set recognition results, a single deep model achieves a significantly

13. True Accept Rate (TAR) is defined as the fraction of genuine pairs correctly accepted at a particular threshold

14. The two-tailed P value equals 0.0001, which indicates the performance improvement of fusion scheme is statistically significant.

15. <http://vis-www.cs.umass.edu/lfw/results.html>

16. The original BLUFR protocol uses Verification Rate (VR). We changed it to True Accept Rate (TAR) for consistency in reporting our results.

better performance (55.90%) than the previous best reported result of 28.90% [6], as well as the COTS matcher (36.44%).

5.3 IJB-A Evaluation

The IJB-A dataset [9] was released in 2015 in an attempt to push the frontiers of unconstrained face recognition. Given that recognition performance on the LFW dataset has saturated under the standard protocol, the IJB-A dataset contains more challenging face images and specifies both verification and identification (open and close sets) protocols. The basic protocol consists of 10-fold cross-validation on pre-defined splits of the dataset, with a disjoint training set defined for each split.

One unique aspect of the IJB-A evaluation protocol is that it defines “templates,” consisting of one or more images (still images or video frames), and defines set-to-set comparisons, rather than face-to-face comparisons, as shown in Fig. 9. In particular, in the IJB-A evaluation protocol the number of images per template ranges from a single image to a maximum of 202 images. Both the search task (1:N comparisons) and verification task (1:1 comparison) are defined in terms of comparisons between templates (consisting of several face images), rather than single face images.

The verification protocol in IJB-A consists of 10 sets of pre-defined comparisons between templates (groups of images). Each set contains about 11,748 pairs of templates (1,756 genuine plus 9,992 impostor pairs), on average. For the search protocol, which evaluates both closed-set and open-set performance, 10 corresponding gallery and probe sets are defined, with both the gallery and probe sets consisting of templates. In each search fold, there are about 1,187 genuine probe templates, 576 impostor probe templates, and 112 gallery templates, on average.

Given an image or video frame from the IJB-A dataset, we first attempt to automatically detect 68 facial landmarks with DLIB. If the landmarks are successfully detected, we align the detected face using the alignment method proposed in Section 3.1. We call the images with automatically detected landmarks *well-aligned images*. If the landmarks cannot be automatically detected, as is the case for profile faces or when only the back of the head is showing (Fig. 8), we align the face based on the ground-truth landmarks provided with the IJB-A protocol. The ground truth landmarks consist of the left eye, right eye, and nose tip, but since these points are not visible in every image, landmarks which are not clearly visible are omitted. For example, in faces exhibiting a high degree of yaw, only one eye is typically visible, so the other eye will not be included in the ground truth landmarks. If all the three landmarks are available, we estimate the mouth position and align the face images using the alignment method in Section 3.1; otherwise, we directly crop a square face region using the provided ground-truth face region. We call images for which automatic landmark detection fails *poorly-aligned images*. Fig. 8 shows some face alignment examples in the IJB-A dataset.

The IJB-A protocol allows training for each fold. Since the IJB-A dataset is qualitatively different from the CASIA dataset that we used to train our network, we retrain our deep model using the IJB-A training set for each fold. The final face representations consists of a concatenation of the deep features from five different deep models trained just on the CASIA dataset, and one re-trained on the IJB-A training set for the current fold. We then use Principal Component Analysis (PCA) to reduce the dimensionality of the combined face representation to 100, which is the lowest value without performance reduction over the training set.

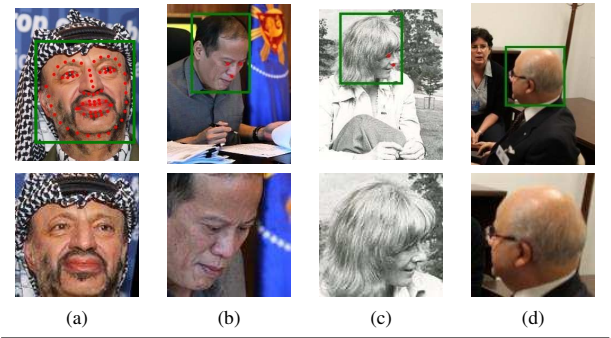


Fig. 8. Examples of web images in the IJB-A dataset with overlaid landmarks (top row), and the corresponding aligned face images (bottom row); (a) example of a well-aligned image obtained using automatically detected landmarks by DLIB [28]; (b), (c), and (d) examples of poorly-aligned images with 3, 2, and 0 ground-truth landmarks provided in IJB-A, respectively. DLIB fails to output landmarks for (b)-(d). The images in the top row have been cropped around the relevant face regions from the original images.

Since all the IJB-A comparisons are defined between sets of faces, we need to determine an appropriate set-to-set comparison method. Our set-to-set comparison strategy first determines if there are one or more *well-aligned images* in a template. If so, we only use the *well-aligned images* for the set comparison; we call the corresponding templates *well-aligned templates*. Otherwise, we use the *poorly-aligned images*, calling the corresponding templates *poorly-aligned templates*. The pairwise face-to-face similarity scores are computed using the cosine similarity, and the average score over the selected subset of images is the final set-to-set similarity score.

Key results of the proposed method, along with the baseline results reported in [9] and DCNN [36] are shown in Table 4. Our deep network based method performs significant better than the two baselines at all evaluated operating points, and slightly worse than DCNN [36]¹⁷. DCNN uses a similar network structure and the same training dataset as our deep model; however, it incorporates the recently proposed parametric rectified linear unit (PReLU) [38], instead of the rectified linear unit (ReLU) [29] used in our deep model. This indicates that the performance of our deep model could also be further improved using updated network architectures. Still, the main focus of this paper is to address the large-scale face retrieval problem.

Fig. 9 shows face search results for two probe templates, one where rank-1 retrieval is successful and the other where rank-1 retrieval is not successful. A template containing a single poorly-aligned image is much harder to recognize than the templates containing one or more well-aligned images. Fig. 10 shows the distribution of well-aligned images and poorly-aligned images in probe templates. Compared to the distribution of poorly aligned templates in the overall dataset, we fail to recognize a disproportionate number of templates containing only poorly-aligned face images.

6 LARGE-SCALE FACE SEARCH

In this section, we evaluate our face search system using an 80M gallery. The test datasets we use consist of LFW and IJB-A images. We use the images to construct the mated portion of a

¹⁷. DCNN was published after an earlier version of this paper [37] appeared on arXiv.

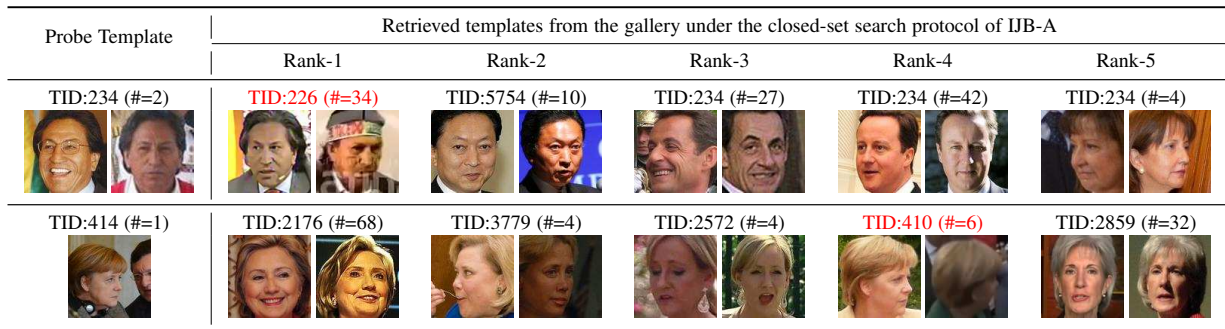


Fig. 9. Examples of face search in the first fold of the IJB-A closed-set search protocol, using “templates.” The first column contains the probe templates, and the following 5 columns contain the corresponding top-5 ranked gallery templates, where red text highlights the correct mated gallery template and the number of faces in the corresponding template is denoted by #. There are 112 gallery templates in total; only a subset (at most four) of the images for each template are shown.

TABLE 4

Recognition accuracies under the IJB-A protocol. Results for GOTS and OpenBR are taken from [9]. Results reported are the average \pm standard deviation over the 10 folds specified in the IJB-A protocol.

| Algorithm | TAR @ FAR (verification) | | | CMC* (closed-set search) | | FNIR @ FAR [†] (open-set search): | |
|--------------------------|--------------------------|------------------|------------------|--------------------------|------------------|--|------------------|
| | 10% | 1% | 0.1% | Rank-1 | Rank-5 | 10% | 1% |
| GOTS | 62.7% \pm 1.2% | 40.6% \pm 1.4% | 19.8% \pm 0.8% | 44.3% \pm 2.1% | 59.5% \pm 2.0% | 76.5% \pm 3.3% | 95.3% \pm 2.4% |
| OpenBR | 43.3% \pm 0.6% | 23.6% \pm 0.9% | 10.4% \pm 1.4% | 24.6% \pm 1.1% | 37.5% \pm 0.8% | 85.1% \pm 2.8% | 93.4% \pm 1.7% |
| DCNN _{all} [36] | 94.7% \pm 1.1% | 78.7% \pm 4.3% | N/A | 86.0% \pm 2.3% | 94.3% \pm 1.7% | N/A | N/A |
| Proposed Deep Model | 89.3% \pm 1.4% | 72.9% \pm 3.5% | 51.0% \pm 6.1% | 82.2% \pm 2.3% | 93.1% \pm 1.4% | 39.2% \pm 2.7% | 61.5% \pm 4.6% |

* Cumulative Match Characteristic (CMC) computes the fraction of genuine samples retrieved at or below a specific rank.

[†] For consistency, we use False Accept Rate (FAR) in place of the False Positive Identification Rate (FPIR) term used in [9]. This quantity is the fraction of impostor probe images accepted at a given threshold, and False Negative Identification Rate (FNIR) is the fraction of genuine probe images rejected at the same threshold.

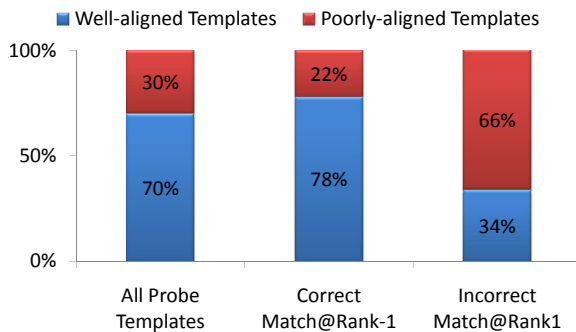


Fig. 10. Distribution of well-aligned templates and poorly-aligned templates in 1:N search protocol of IJB-A, averaged over 10 folds. Correct Match@Rank-1 means that the mated gallery template is correctly retrieved at rank 1. Landmarks in well-aligned images can be automatically detected by DLIB [28]. Poorly-aligned images mainly consist of profile views of faces. We align these images using the three ground-truth landmarks when available, or else by cropping the entire face region.

retrieval database with an extended gallery, rather than following the standard protocols for those datasets. We report search results under both open-set and closed-set protocols with increasing gallery size, up to 80M faces. We evaluate the following three face search schemes:

- **Deep Features (DF):** Use our deep features with product quantization (PQ) to directly retrieve the top- k most similar faces to the probe (no re-ranking step).
- **COTS:** Use a state-of-the-art COTS face matcher to compare the probe image with each gallery face, and output the top- k most similar faces to the probe (no

filtering step).

- **DF→COTS:** First filter the gallery using deep features. Next, re-rank the top- k candidate faces by fusing cosine similarities computed from deep features with the COTS matcher’s similarity scores for the k candidate faces.

For closed-set face search, we assume that the probe always has at least one corresponding face image in the gallery. For open-set face search, given a probe we first decide whether a corresponding image is present in the gallery. If it is determined that the probe’s identity is represented in the gallery, then we return the search results. For open-set performance evaluation, the probe set consists of two groups: i. genuine probe set that has mated images in the gallery, and ii. impostor probe set that has no mated images in the gallery.

6.1 Search Dataset

We construct a large-scale search dataset using the four face datasets introduced in Section 4. The dataset consists of five parts, as shown in Table 5: 1) *training* set, which is used to train our deep network; 2) *genuine probe* set, the probe set which has corresponding gallery images; 3) *mate* set, the part of the gallery containing the same subjects as the *genuine probe* set; 4) *impostor probe* set, which has no overlapping subjects with the *genuine probe* set; 5) *background* set, which has no identity labels and is simply used as background images to enlarge the gallery size.

We use the LFW and IJB-A datasets to construct the *genuine probe* set and the corresponding *mate* set. For the LFW dataset, we first remove all the subjects who have only a single image, leaving 1, 507 subjects with 2 or more images. For each of these subjects, we take half of the images for the *genuine probe* set and use

the remaining images for the *mate* set in the gallery. We repeat this process 10 times to generate 10 groups of probe and mate sets. To construct the *impostor probe* set, we take 4,000 subjects from LFW, each having only one image. For the IJB-A dataset, a similar process is adopted to generate 10 groups of probe and mate sets. To build a large-scale *background* set, a crawler was used to download millions of web images from the Internet, then filter them to only include those with detectable faces by DLIB¹⁸. By combining the *mate* set and *background* set, we compose an 80 million image gallery. More details are shown in Table 5.

TABLE 5
Large-scale web face search dataset overview.

| | Source | # Subjects | # Images |
|---------------------------------|-----------|------------|------------|
| Training Set | CASIA [6] | 10,553 | 404,992 |
| LFW based probe and mate sets | | | |
| Genuine Probe Set | LFW [3] | 1,507 | 3,370 |
| Mate Set | LFW [3] | 1,507 | 3,845 |
| IJB-A based probe and mate sets | | | |
| Genuine Probe Set | IJB-A [9] | 500 | 10,868 |
| Mate Set | IJB-A [9] | 500 | 10,626 |
| Impostor Probe Set | LFW [3] | 4,000 | 4,000 |
| Background Set | Web-Faces | N/A | 80,000,000 |

6.2 Dataset Segmentation

In the retrieval experiments, we use LFW or IJB-A for the *probe* and *mate* sets, and 80M web faces for the *background* set. Although all the three datasets consist of unconstrained face images from the web, they are collected from different sources. In particular, LFW and IJB-A are from news images (with IJB-A containing more challenging images) and Web-Face is from photos on social media websites. As such, the different characteristics of the datasets may lead to a segmentation effect, where images from one dataset may easily be distinguished from others based on differing image acquisition properties, rather than the identities of the faces being compared. In other words, the *background* set should have a similar distribution to the *probe* set and the *mate* set, otherwise, the use of the background set will not effectively demonstrates the search performance that would be seen with a large gallery of images with more uniform properties.

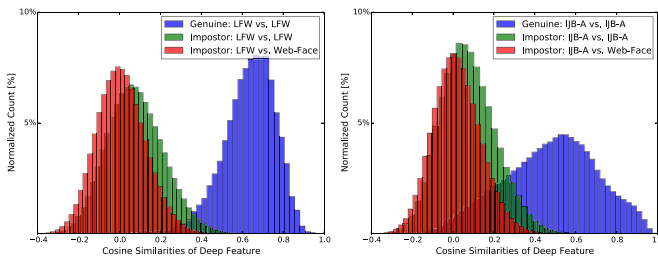


Fig. 11. Distributions of cosine similarities of the genuine pairs, within-dataset impostor pairs and between-dataset impostor pairs for the combinations of LFW+Web-Face (left) and IJB-A+Web-Face (right).

To examine the differences between the Web-Face and labeled datasets (LFW and IJB-A), we first randomly sample 10K images

¹⁸. The links to these web images were provided by a different research collaborator. We downloaded the raw web images, and filtered out all non-face-detectable web images using the DLIB face detector.

from the Web-Face dataset, and combine this subset with the LFW and IJB-A datasets, respectively. We then extract features using the proposed deep model, and compute all pairwise cosine similarities. We examine the distributions of these scores in Fig. 11, which plots the genuine, within-dataset impostor, and between-dataset impostor score distributions for both the LFW+Web-Face and IJB-A+Web-Face datasets.

We observe that for both LFW and IJB-A datasets the distributions of cosine similarities of the within-dataset and between-dataset impostor pairs have a significant overlap, but the distribution of cosine similarities of the between-dataset impostor pairs is left-shifted. This indicates that the “effective” background gallery size is smaller than 80 million, since typical impostor images from the background dataset score relatively lower than impostors from the labeled datasets. We analyze this effect in terms of the verification problem, by estimating what size sample from the within-dataset impostor would result in the same total number of false accepts seen from the cross-dataset impostor score distribution (using the empirical score distributions directly). For the 1% False Reject Rate operating point, a sample of approximately 23 million images following the observed within-LFW impostor score distribution would generate as many false accept errors as were generated from the full 80 million Web-Face dataset. For a lower FRR of 0.01%, matching the number of false accepts generated from the cross-dataset impostor distribution would require approximately 72 million images following the within-LFW impostor score distribution.

6.3 Performance Measures

Face search aims to find all the mated faces in the gallery, which is broader than the traditional biometric problems, e.g. authentication (1:1 search) or identification (1:N search). Hence, we evaluate our face search system with the widely used retrieval evaluation metrics: *precision*, the fraction of the search set consisting of mated face images, and *recall*, the fraction of all mated face images for a given probe face that were returned in the search results.

Various trade-offs between *precision* and *recall* are possible (for example, high recall can be achieved by returning a large result set, but a large result set will also lead to lower precision), so we summarize the overall closed-set face search performance using *mean Average Precision* (mAP), which is also widely used for search system evaluation [7]. mAP is defined as follows: given a set of n probe face images $Q = \{\mathbf{x}_q^1, \mathbf{x}_q^2, \dots, \mathbf{x}_q^n\}$ and a gallery set with N images, the *average precision* of \mathbf{x}_q^i is:

$$\text{avgP}(\mathbf{x}_q^i) = \sum_{j=1}^N P(\mathbf{x}_q^i, j) \times [R(\mathbf{x}_q^i, j) - R(\mathbf{x}_q^i, j-1)] \quad (4)$$

where $P(\mathbf{x}_q^i, j)$ is *precision* at the j -th position for \mathbf{x}_q^i and $R(\mathbf{x}_q^i, j)$ is *recall* at the j -th position for \mathbf{x}_q^i ($R(0) = 0$). Note that this measure includes the ranks of all gallery images matching a given query, so having more gallery images is not a strictly easier problem. The mean Average Precision (mAP) of the entire probe set is:

$$\text{mAP}(Q) = \text{mean}(\text{avgP}(\mathbf{x}_q^i)), i = 1, 2, \dots, n$$

When the gallery size N is too large, for efficiency, we compute the *average precision* using the top-100K retrieval results. Since mAP uses the unweighted average, each query image has the same

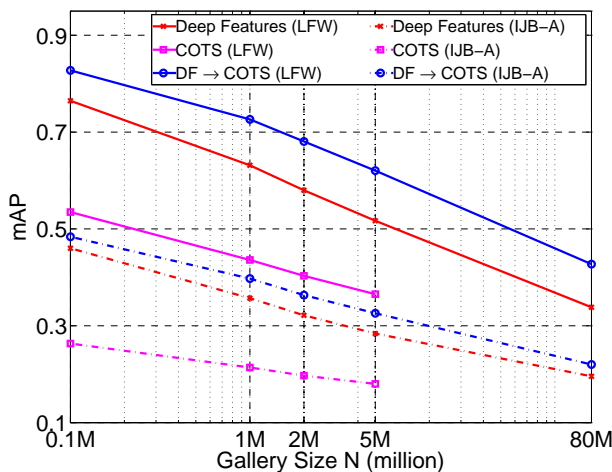
impact on the aggregate, regardless of the number of matching gallery images.

In the open-set scenario, we evaluate search performance as a trade-off between mean average precision (mAP) and false accept rate (FAR) (the fraction of impostor probe images which are not rejected at a given threshold). Given a genuine probe, its *average precision* is set to 0 if it is rejected at a given threshold, otherwise, its *average precision* is computed using Eq. 4.

For a large background gallery dataset like the 80 million Web-Face used here, it is difficult to ensure that there are no subjects which overlap between the query and background sets. As a result, in our evaluation, if one of the unlabeled background images is actually the same person as the query image, we consider it an “incorrect” retrieval result. In other words, while we cannot guarantee that no images in the background set have the same identity as the query image, any such images, if present, will bias our results in the direction of lower accuracy.

6.4 Closed-set Face Search

We examine closed-set face search performance with the gallery size N ranging from 100K to 80M. Enrolling the complete 80M gallery in the COTS matcher would take a prohibitive amount of time (over 80 days), due to limitations of the SDK we have, so the maximum gallery set used for the COTS matcher is 5M. For the proposed face search scheme DF→COTS, we chose the size of candidate set k using the heuristic $k = 1/100N$ when the gallery size is smaller than 5M and $k = 1,000$ when the gallery set size is 80M. We use a fixed k for the full 80M gallery since using a larger k would take a prohibitive amount of time, due to the need to enroll the filtered images in the COTS matcher. Experimental results for the LFW and IJB-A datasets under closed-set search are shown in Fig. 12.



Closed-set Search Evaluation on LFW and IJB-A datasets

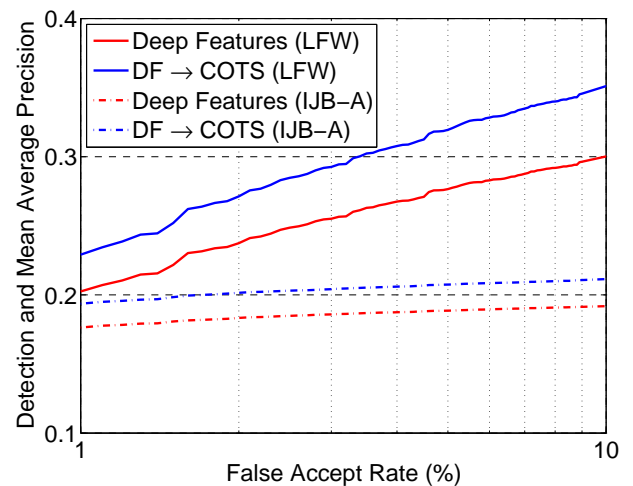
Fig. 12. Closed-set face search performance (mAP) vs. gallery size N (log-scale), on LFW and IJB-A datasets. The performance of COTS matcher on 80M gallery is not shown, since enrolling the complete 80M gallery with the COTS matcher would have taken a prohibitive amount of time (over 80 days).

For both LFW and IJB-A face images, as expected, the recognition performance of all three face search schemes evaluated here decreases with increasing gallery set size. In particular, for all the search schemes, mAP linearly decreases with the gallery size

N on log scale; the performance gap between a 100K gallery and a 5M gallery is about the same as the performance gap between a 5M gallery and an 80M gallery. While deep features outperform the COTS matcher alone, the proposed cascaded face search system (which leverages both deep features and the COTS matcher) gives better search accuracy than either method individually. Results on the IJB-A dataset are similar to the LFW results, except for a lower overall accuracy. The lower accuracy on IJB-A data is to be expected given that IJB-A contains more challenging face images.

6.5 Open-set Face Search

Open-set search is important for several practical applications (e.g., de-duplication), one cannot assume that the gallery will contain images of all potential probe subjects. We evaluate open-set search performance on the 80M gallery, and plot the search performance (mAP) at varying FAR values in Figs. 13.



Open-set Search Evaluation on LFW and IJB-A datasets

Fig. 13. Open-set face search performance (mAP) vs. false accept rate (FAR) on LFW and IJB-A datasets, using the 80M face gallery. The performance of COTS matcher is not shown due to computational issues. FAR is shown only up to 10% since operational systems are not likely to operate beyond this value.

For both the LFW and IJB-A datasets, the open-set face search problem is much harder than closed-set face search. At a FAR of 1%, the search performance (mAP) of both algorithms is much lower than the closed-set face search at 80M show in Fig. 12, indicating that a large number of genuine probe images are rejected at the threshold needed to attain 1% FAR.

6.6 Scalability

In addition to the mAP performance measure, we also report the search times in Table 6. We run all the experiments on a PC with an Intel(R) Xeon(R) CPU (E5-2687W) clocked at 3.10GHz. For a fair comparison, all the compared algorithms use only one CPU core. The deep features are extracted using a Tesla K40 graphics card.

In our experiments, template generation for the entire gallery is done off-line and the gallery is indexed using product quantization before processing the probe images. Gallery images (up to 5M) are enrolled into the COTS matcher. The run time of the proposed face search system after the gallery is enrolled and indexed consists of

TABLE 6
The average search time (seconds) per probe face and the corresponding search performance (mAP).

| | 5M Face Gallery | | | 80M Face Gallery | | |
|------------|-----------------|------|-----------------|------------------|------|----------------|
| | COTS | DF | DF→COTS @50K | COTS | DF | DF→COTS @1K |
| Enrollment | 0.09 | 0.05 | 0.14 | 0.09 | 0.05 | 0.14 |
| Search | 30 | 0.84 | 1.15 | 480.0* | 6.63 | 6.64 |
| Total Time | 30.09 | 0.89 | 1.29 | 480.1* | 6.68 | 6.88 |
| mAP | 0.36 | 0.52 | 0.62 | N/A | 0.34 | 0.40 |

* Estimated by assuming that search time increases linearly with gallery size.

two parts: i) *enrollment time* including face detection, alignment and feature extraction, and ii) *search time* consisting of the time taken to find the top- k search results given the probe template. Since we did not enroll all 80M gallery images using the COTS matcher, we estimate the query time for the 80M gallery by assuming that search time increases linearly with the gallery size.

Using product quantization for fast matching based on deep features, we can retrieve the top- k candidate faces in about 0.9 seconds for a 5M image gallery and in about 6.7 seconds for an 80M gallery. On the other hand, the COTS matcher takes about 30 and 480 seconds to carry out brute-force comparison over the complete galleries of 5 and 80 million images, respectively. In the proposed cascaded face search system, we mitigate the impact of the slow exhaustive search required by the COTS matcher by only using it on a short candidate list. The proposed cascaded scheme takes about 1 second for the 5M gallery and about 6.9 seconds for the 80M gallery, which is only a minor increase over the time taken using deep features alone (6.68 seconds). While the search time could be further reduced by using a non-exhaustive search method, it would most likely result in a significant loss in search accuracy.

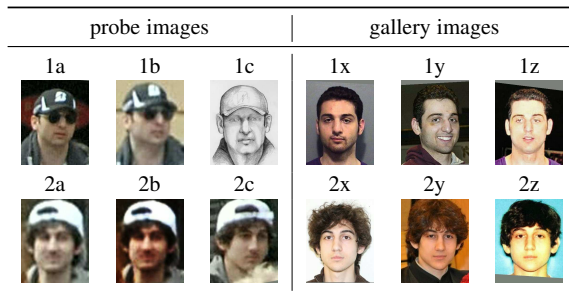


Fig. 14. Probe and gallery images of Dzhokhar Tsarnaev and Tamerlan Tsarnaev, responsible for the April 15, 2013 Boston marathon bombing. Face images 1a and 1b are the two probe images used for Suspect 1 (Tamerlan Tsarnaev), and 1c is his sketch image drawn by a forensic sketch artist based on 1a and 1b. Face images 2a, 2b and 2c are the three probe images used for Suspect 2 (Dzhokhar Tsarnaev). The gallery images of the two suspects became available on media websites following the identification of the two suspects based on investigative leads. Face images 1x, 1y and 1z are the three gallery images for Suspect 1 and images 2x, 2y and 2z are the three gallery images for Suspect 2.

7 BOSTON MARATHON BOMBING CASE STUDY

In addition to the large-scale face search experiments reported above, we report on a case-study: finding the identity of Boston marathon bombing suspects¹⁹ in an 80M face gallery.

19. https://en.wikipedia.org/wiki/Boston_Marathon_bombing

Klontz and Jain [35] made an attempt to identify the face images of the Boston marathon bombing suspects in a 1M gallery of mugshot images. Video frames of the two suspects were matched against a background set of mugshots using two state-of-the-art COTS face matchers. Five low resolution images (1a, 1b, 2a, 2b, 2c) of the two suspects, released by the FBI (shown in the left side of Fig. 14) were used as probe images, and six images (1x, 1y, 1z, 2x, 2y, 2z) of the suspects released by the media (shown in the right side of Fig. 14) were used as the mates in the gallery. These mated images were augmented with 1 million mugshot images. One of the COTS matchers was successful in finding the true mate (2y) of one of the probe image (2c) of Dzhokhar Tsarnaev at rank 1. Neither of the two probe images for the older brother could retrieve the true mates at a reasonable rank.

To evaluate the performance of our cascaded face search system, we construct a similar search problem under more challenging conditions by adding the gallery images to a background set of up to 80 million web faces. In addition, we also use one sketch image (1c in Fig. 14) of the older brother as the probe image. We argue that the unconstrained web faces are more consistent with the quality of the images of the suspects used in the gallery than mugshot images and therefore comprise a more meaningful gallery set. We evaluate the search results using gallery sizes of 5M and 80M leveraging the same background set used in our prior search experiments.

The search results are shown in Table 7. Considering the images of subject 1, although the performance of deep features is better than the COTS matcher, both the deep features and the COTS matcher return the matching gallery images at excessively high ranks for all three probe images. We noticed that the retrieval performance of the sketch image (1c) is much better than the retrieval results of the two probe faces extracted from video frames (1a and 1b). Still, even for the sketch image, the best retrieval result is a true match at rank 66,427 on the 5M gallery.

For the second subject, results are relatively better. For the 5M gallery, the COTS matcher found a mate (2y) for probe 2c at rank 625, while the deep features returned gallery image 2x for probe 2c at rank 9. The proposed cascaded search system returned gallery image 2y at rank 1, by combining the COTS matcher and deep features to re-rank the top 1K or top 10K candidate faces, demonstrating the strength of the proposed cascade framework. The retrieval results for probe 2c are slightly worse on the 80M image gallery, which is to be expected. Using deep features alone, we now find gallery image 2x at rank 109 and gallery image 2y at rank 2,952. However, using the cascaded search system, we retrieve gallery image 2x at rank 46 by re-ranking the top-1K faces, and retrieve gallery image 2y at rank 8 by re-ranking the top-10K faces. So, even with an 80M image gallery, we can successfully find a match for one of the probe images (2c) within the top-10 retrieved faces. The face search results for the 80M galleries are shown in Fig. 15.

8 CONCLUSIONS

We have proposed a cascaded search system suitable for large-scale face search problems. We have developed a deep learning based face representation trained on the publicly available CASIA dataset [6]. The deep features are used in a product quantization based approximate k -NN search to first obtain a short list of

TABLE 7

Rank retrieval results of the two Boston bomber suspects based on 5M and 80M face gallery. The six probe images are designated as 1a, 1b, 1c, 2a, 2b, and 2c. The six mated images in the gallery are designated as 1x, 1y, 1z, 2x, 2y, and 2z. The corresponding images are shown in Fig. 14

| | COTS (5M Gallery) | | | Deep Features (5M Gallery) | | | Deep Features (80M Gallery) | | |
|--------------------------------------|-------------------|-----------|-----------|----------------------------|-----------|-----------|-----------------------------|------------|------------|
| | 1x | 1y | 1z | 1x | 1y | 1z | 1x | 1y | 1z |
| 1a | 2,041,004 | 595,265 | 1,750,309 | 132,613 | 232,275 | 1,401,474 | 2,566,917 | 5,398,454 | 31,960,091 |
| 1b | 3,816,874 | 3,688,368 | 2,756,641 | 1,511,123 | 1,153,036 | 1,699,951 | 33,783,360 | 27,439,526 | 44,282,173 |
| 1c | 126,217 | 608,899 | 535,815 | 66,427 | 199,083 | 1,529,169 | 753,653 | 2,408,392 | 29,383,945 |
| | 2x | 2y | 2z | 2x | 2y | 2z | 2x | 2y | 2z |
| | | | | | | | | | |
| 2a | 67,766 | 86,747 | 301,868 | 174,440 | 39,417 | 105,879 | 2,461,664 | 875,168 | 1,547,895 |
| 2b | 352,062 | 48,335 | 865,043 | 71,795 | 26,525 | 84,013 | 1,417,768 | 972,411 | 1,367,694 |
| 2c | 158,341 | 625 | 515,851 | 9 | 341 | 9,975 | 109 | 2,952 | 136,651 |
| Proposed Cascaded Face Search System | | | | | | | | | |
| 2c | DF→COTS@1K | | | 7 | 1 | 9,975 | 46 | 2,952 | 136,651 |
| 2c | DF→COTS@10K | | | 10 | 1 | 1,580 | 160 | 8 | 136,651 |























| Method | Probe | Top 10 most similar retrieved images from an 80M face gallery | | | | | | | | | |
|---------------|-------|---|---|---|---|---|---|--|---|---|---|
| Deep Features | 1a |  |  |  |  |  |  |  |  |  |  |
| Deep Features | 1b |  |  |  |  |  |  |  |  |  |  |
| Deep Features | 1c |  |  |  |  |  |  |  |  |  |  |
| Deep Features | 2a |  |  |  |  |  |  |  |  |  |  |
| Deep Features | 2b |  |  |  |  |  |  |  |  |  |  |
| Deep Features | 2c |  |  |  |  |  |  |  |  |  |  |
| DF→COTS @ 10K | |  |  |  |  |  |  |  |  |  |  |

Fig. 15. Top 10 search results for the two Boston marathon bombers on the 80M face gallery. The first three probe faces (1c is a sketch) are of the older brother (Tamerlan Tsarnaev) and the last three probe faces are of the younger brother (Dzhokhar Tsarnaev). For each probe face, the retrieved gallery image with green border is the correctly retrieved image. Images with the red border are “near-duplicate” images present in the gallery. Note that we were not aware of the existence of these near-duplicate images in the 80M gallery before the search.

candidate faces. This short list of candidate faces is then re-ranked using the similarity scores provided by a state-of-the-art COTS face matcher. We demonstrate the performance of our deep features on three face recognition datasets, of increasing difficulty: the PCSO mugshot dataset, the LFW unconstrained face dataset, and the IJB-A dataset. On the mugshot data, deep feature performance (TAR of 93.5% at FAR of 0.01%) is worse than a COTS matcher (98.5%), but fusing our deep features with the COTS matcher does improve the overall performance (99.2%). Our performance on the standard LFW protocol (98.20% accuracy) is comparable to state-of-the-art accuracies reported in the literature. On the BLUFR protocol for the LFW database we attain the best reported performance to date (TAR of 88.03% at FAR of 0.1%). Our deep features outperform the benchmarks reported in [9] on the IJB-A dataset, as follows: TAR of 51.0% at FAR of 0.1% (verification); Rank 1 retrieval of 82.2% (closed-set search); FNIR of 61.5% at FAR of 1% (open-set search). In addition to the evaluations on the LFW and the IJB-A benchmarks, we evaluate the proposed search scheme on an 80

million face gallery, and show that the proposed scheme offers an attractive tradeoff between recognition accuracy and runtime. We also demonstrate search performance on an operational case study involving the video frames of the Tsarnaev brothers implicated in the 2013 Boston marathon bombing. In this case study, the proposed system can find one of the suspects’ images at rank 1 in 1 second on a 5M gallery and at rank 8 in 7 seconds on an 80M gallery.

We consider non-exhaustive face search an avenue for further research. Although we made an attempt to employ indexing methods, they resulted in a drastic decrease in search performance. If only a few searches need to be made, the current system’s search speed is adequate, but if the number of searches required is on the order of the gallery size, the current runtime is inadequate. We are also interested in improving the underlying face representation, via improved network architectures (e.g. joint-training of multiple patches, and using different kinds of layers), or by using larger training sets.

ACKNOWLEDGEMENTS

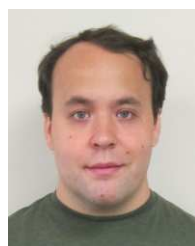
This research was supported by the National Institute of Justice (NIJ) grant 2011-IJ-CX-K057.

REFERENCES

- [1] B. Chen, Y. Chen, Y. Kuo, and W. Hsu, "Scalable face image retrieval using attribute-enhanced sparse codewords," *IEEE Trans. on Multimedia*, vol.15 no.5, pp.1163-1173, 2012. 1, 2, 3
- [2] Z. Wu, Q. Ke, J. Sun, and H. Y. Shum, "Scalable face image retrieval with identity-based quantization and multi-reference re-ranking," *CVPR*, 2010. 1, 2, 3
- [3] 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. 1, 2, 3, 5, 6, 7, 10
- [4] D. Wang and A. K. Jain, "Face retriever: Pre-filtering the gallery via deep neural net," *ICB*, 2015. 1
- [5] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," *ECCV*, 2012. 1, 7
- [6] D. Yi, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv:1411.7923v1*, 2014. 2, 3, 4, 5, 6, 7, 8, 10, 12
- [7] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.33, no.1, pp.117-128, 2011. 2, 4, 10
- [8] P. Grother and M. Ngan, "Face recognition vendor test (frvt): Performance of face identification algorithms," *NIST Interagency Report 8009*, 2014. 2, 5, 6
- [9] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mash, M. Burge, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark," *CVPR*, 2015. 2, 3, 5, 6, 8, 9, 10, 13
- [10] N. Kumar, A. C. Berg, A. C. Berg, and S. K. Nayar, "Attribute and simile classifiers for face verification," *ICCV*, 2009. 2, 3
- [11] D. Miller, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Megaface: A million faces for recognition at scale," *arXiv:1505.02108*, 2015. 3
- [12] H. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," *ICIP*, pp. 343-347, 2014. 3
- [13] D. Yi, Z. Lei, Y. Hu, and S. Z. Li, "Fast matching by 2 lines of code for large scale face recognition systems," *arXiv:1302.7180*, 2013. 2, 3
- [14] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no.10, pp. 1090-1104, 2000. 2, 3
- [15] J. Yan, Z. Lei, D. Yi, and S. Li, "Towards incremental and large scale face recognition," *ICB*, 2011. 2, 3
- [16] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," *CVPR*, 2005. 3
- [17] B. Klare, A. Blanton, and B. Klein, "Efficient face retrieval using synecdoches," *ICB*, 2014. 3
- [18] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Transactions on Information Forensics and Security*, vol.9, no.12, pp.2144-2157, 2014. 2, 3
- [19] S. Liao, Z. Lei, D. Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," *IJCB*, 2014. 2, 3, 7
- [20] A. K. Jain and S. Z. L. (eds.), "Handbook of Face Recognition, 2nd edition," *Springer-Verlag*, 2011. 2
- [21] J. Liu Y. Deng T. Bai Z. Wei C. Huang "Targeting Ultimate Accuracy: Face Recognition via Deep Embedding," *arXiv:1506.07310*, 2015. 2, 7
- [22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CVPR*, 2015. 2, 7
- [23] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol.36, no.3, pp.550-563, 2014. 2
- [24] W. Dong, Z. Wang, W. Josephson, M. Charikar, and K. Li, "Modeling LSH for performance tuning," *CKM*, 2008. 2, 4
- [25] J. Zhou and T. Pavlidis, "Discrimination of characters by a multi-stage recognition process," *Pattern Recognition*, vol.27, pp.1539-1549, 1994. 2
- [26] J. Wan, D. Wang, S. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," *ACM Multimedia*, 2014. 3, 4
- [27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014. 3
- [28] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *CVPR*, 2014. 3, 4, 8, 9
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012. 4, 8
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol.15, pp.1929-1958, 2014. 4
- [31] Y. Kalantidis and Y. Avrithis, "Locally optimized product quantization for approximate nearest neighbor search," *CVPR*, 2014. 4
- [32] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, no.3, pp.226-239, 1998. 5
- [33] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol.38, no.12, pp.2270-2285, 2005. 5
- [34] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol.57, pp.137-154, 2004. 5
- [35] J. Klontz and A. Jain, "A case study of automated face recognition: The Boston marathon bombings suspects," *Computer*, vol.46, no.11, pp.91-94, 2013. 12
- [36] J. Chen, V. Patel, and R. Chellappa, "Unconstrained Face Verification using Deep CNN Features," *arXiv:1508.01722*, 2015. 8, 9
- [37] D. Wang, C. Otto, and A. Jain, "Face Search at Scale: 80 Million Gallery," *arXiv:1507.07242*, 2015. 1, 8
- [38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification," *arXiv:1502.01852*, 2015. 8
- [39] V. Jain and Erik Learned-Miller, "FDDB: A Benchmark for Face Detection in Unconstrained Settings", *University of Massachusetts, Amherst*, 2010. 6
- [40] R. Ranjan, V. Patel, R. Chellappa. "A Deep Pyramid Deformable Part Model for Face Detection," *BTAS*, 2015. 6
- [41] C. Otto, D. Wang and A. K. Jain "Clustering Millions of Faces by Identity" *arXiv:1604.00989*, 2016 6



Dayong Wang received his bachelor degree from Tsinghua University in 2008 and his Ph.D. degree from Nanyang Technological University, Singapore in 2014. His research interests include machine learning, pattern recognition, and multimedia information retrieval. He has published over 10 papers in top venues, including TPAMI, TKDE, ACM MM, SIGIR. He received the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2014.



Charles Otto received his B.S. degree in the Department of Computer Science and Engineering at Michigan State University in 2008. He was a research engineer at IBM during 2006-2011. Since 2012, he has been working towards the Ph.D. degree in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition, image processing, and computer vision, with applications to face recognition.



Anil K. Jain is a university distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. He served as the editor-in-chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (1991-1994). He is the coauthor of a number of books, including Handbook of Fingerprint Recognition (2009), Handbook of Biometrics (2007), Handbook of Multibiometrics (2006), Handbook of Face Recognition (2011), BIOMETRICS: Personal Identification in Networked Society (1999), and Algorithms for Clustering Data (1988). He served as a member of the Defense Science Board and The National Academies committees on Whither Biometrics and Improvised Explosive Devices. He received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award and the Pattern Recognition Society best paper awards in 1987, 1991, and 2005. He has received Fulbright, Guggenheim, Alexander von Humboldt, IEEE Computer Society Technical Achievement, IEEE Wallace McDowell, ICDM Research Contributions, and IAPR King-Sun Fu awards. He was elected a member of the National Academy of Engineering in 2016.