# Do We Really Need to Collect Millions of Faces for Effective Face Recognition?

Iacopo Masi[1(✉)], Anh Tuấn Trần[1], Tal Hassner[2,3(✉)], Jatuporn Toy Leksut[1], and Gérard Medioni[1]

[1] Institute for Robotics and Intelligent Systems, USC, Los Angeles, CA, USA
{iacopo.masi,anhttran,leksut,medioni}@usc.edu
[2] Information Sciences Institute, USC, Los Angeles, CA, USA
hassner@isi.edu
[3] The Open University of Israel, Ra'anana, Israel

**Abstract.** Face recognition capabilities have recently made extraordinary leaps. Though this progress is at least partially due to ballooning training set sizes – huge numbers of face images downloaded and labeled for identity – it is not clear if the formidable task of collecting so many images is truly necessary. We propose a far more accessible means of increasing training data sizes for face recognition systems: *Domain specific* data augmentation. We describe novel methods of enriching an existing dataset with important facial appearance variations by manipulating the faces it contains. This synthesis is also used when matching query images represented by standard convolutional neural networks. The effect of training and testing with synthesized images is tested on the LFW and IJB-A (verification and identification) benchmarks and Janus CS2. The performances obtained by our approach match state of the art results reported by systems trained on millions of downloaded images.

## 1 Introduction

The recent impact of deep Convolutional Neural Network (CNN) based methods on machine face recognition capabilities has been extraordinary. The conditions under which faces are now recognized and the numbers of faces which systems can now learn to identify improved to the point where some consider machines to be better than humans at this task. This progress is partially due to the introduction of new and improved network designs. However, alongside developments in network architectures, it is also the underlying ability of CNNs to learn from massive training sets that allows these techniques to be so effective.

Realizing that effective CNNs can be made even more effective by increasing their training data, many began focusing efforts on harvesting and labeling large image collections to better train their networks. In [39], a standard CNN was trained by Facebook using 4.4 million labeled faces and shown to achieve what was, at the time, state of the art performance on the Labeled Faces in the

---

I. Masi, A. Tuấn Trần and T. Hassner are equally contributed.

| Dataset | #ID | #Img | #Img/#ID |
|---------|-----|------|----------|
| CASIA [46] | 10,575 | 494,414 | 46 |
| Facebook DeepFace [39] | 4,030 | 4.4M | 1K |
| Google FaceNet [33] | 8M | 200M | 25 |
| VGG Face [28] | 2,622 | 2.6M | 1K |
| Facebook Fusion [40] | 500M | 10M | 50 |
| MegaFace [14] | 690,572 | 1.02M | 1.5 |
| Aug. pose+shape | 10,575 | 1,977,656 | 187 |
| Aug. pose+shape+expr | 10,575 | 2,472,070 | 234 |

**(a)** Face set statistics
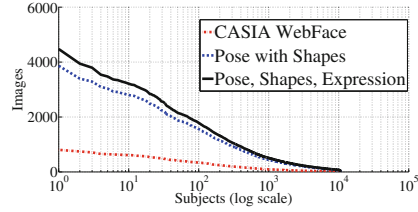


**(b)** Images for subjects

**Fig. 1.** (a) Comparison of our augmented dataset with other face datasets along with the average number of images per subject. (b) Our improvement by augmentation (Aug.) in the distribution of per-subject image numbers in order to avoid the long-tail effect of the CASIA set [46] (also shown in the last two rows of (a)).

Wild (LFW) benchmark [13]. Later, [28] proposed the VGG-Face representation, trained on 2.6 million faces, and Face++ proposed its Megvii System [47], trained on 5 million faces. All, however, pale in comparison to the Google FaceNet [33] which used 200 million labeled faces for its training.

Making CNNs better by collecting and labeling huge training sets is unfortunately not easy. The effort required to download, process and label millions of Internet images with reliable subject names is daunting. To emphasize this, the bigger sets, [39,40] and [33], required the efforts of affluent commercial organizations to assemble (Facebook and Google, resp.) and none of these sets was publicly released by its owners. By comparison, the largest publicly available face recognition training set is CASIA WebFaces [46] weighing in at a mere 495K images, several orders of magnitudes smaller than the two bigger commercial sets[1].

But downloading and labeling so many faces is more than just financially hard. Figure 1a provides some statistics for the larger face sets. Evidently, set sizes increase far faster than per-subject image numbers. This may imply that finding many images verified as belonging to the same subjects is hard even when resources are abundant. Regardless of the reason, this is a serious problem: recognition systems must learn to model not just inter-class appearance variations (differences between different people) but also intra-class variations (same person appearance variations) and this appears to remain a challenge for data collection efforts.

In light of these challenges, it is natural to ask: is there no alternative to this labor intensive, data harvesting and labeling approach to pushing recognition performances? Beyond mitigating the challenges of data collection, this question touches a more fundamental issue. Namely, can CNNs benefit from domain specific image preprocessing, and if so, how?

To answer this, we make the following contributions. **(1)** We propose *synthesizing data* in addition to collecting it. We inflate the size of an existing training

---

[1] MegaFace [14] is larger than CASIA, but was designed as a testing set and so provides few images per subject. It was consequently never used for training CNN systems.

set, the CASIA WebFace collection [46], to several times its size using domain (face) specific image synthesis methods (Fig. 1b). We generate images which introduce new intra-class facial appearance variations, including pose (Sect. 3.1), shape (Sect. 3.2) and expression (Sect. 3.3). **(2)** We describe a novel matching pipeline which uses similar synthesis methods at test time when processing query images. Finally, **(3)**, we test our approach on the LFW [13], IJB-A (verification and identification) and CS2 benchmarks [16]. Our results show that a CNN trained using these generated faces matches state of the art performances reported by systems trained on millions of manually downloaded and labeled faces[2].

Our approach can be considered a novel face data augmentation method (Sect. 2): *Domain specific* data augmentation. Curiously, despite the success of existing generic augmentation methods, we are unaware of previous reports of applying this easily accessible approach to generate new training face images, or indeed training for any other image class.

## 2   Related Work

**Face recognition:**  Face recognition is one of the central problems in computer vision and, as such, the relevant work is extensive. Face recognition performances greatly improved with the recent introduction of deep learning techniques and in particular CNNs. Though CNNs have been used for face recognition as far back as [19], only when massive amounts of data became available did they achieve state of the art performance. This was originally shown by the Facebook DeepFace system [39], which used an architecture very similar to [19], but with over 4 million images used for training they obtained better results.

Since then, CNN based recognition systems continuously cross performance barriers with some notable examples including the Deep-ID 1-3 systems [36–38]. They and many others since, developed and trained their systems using far fewer training images, at the cost of somewhat more elaborate network architectures.

Though novel network designs can lead to better performance, further improvement can be achieved by collecting more training data. This was shown by Google FaceNet [33], which was trained on 200 million images. Besides improving results, they offered an analysis of the consequences of using more data: apparently, there is a significant diminishing returns effect when training with increasing image numbers. Thus, the performance gained by going from thousands of images to millions is substantial but increasing the numbers further provides smaller and smaller benefits. One way to explain this is that the data they used suffers from a *long tail* phenomenon [46], where most subjects in these sets have very few images for the network to learn intra-subject appearance variations.

These methods were evaluated on LFW, a *standard de facto* for measuring face recognition performances. Many recent LFW results, however, are reaching near-perfect performances, suggesting that LFW is no longer a challenging benchmark for today's systems. Another relevant and popular benchmark is the

---

[2] See www.openu.ac.il/home/hassner/projects/augmented_faces.

YouTube Faces (YTF) set [42]. It contains unconstrained face videos rather than images, but it too is quickly being saturated. Recently, a new benchmark was released, again aiming to push machine face recognition capabilities: the Janus set [16]. Its design offers several novelties compared to existing sets, including template based, rather than image based, recognition and a mix of both images and videos. It is also tougher than previous collections. Not surprisingly, dominating performance on Janus are CNN methods such as [5].

**Data augmentation:**   Data augmentation techniques are transformations applied to training or testing images, without altering their labels. Such methods are well known to improve CNN performances and avoid overfitting [3]. Popular augmentation methods include simple, geometric transformations such as *over-sampling* (multiple, translated versions of the input image obtained by cropping at different offsets) [18,20], *mirroring* (horizontal flipping) [3,44], rotating [43] the images as well as various photometric transformations [7,18,34].

Surprisingly, despite being widely recognized as highly beneficial to the training of CNN systems, we are unaware of previous attempts to go beyond these generic image processing transformations as described here. One notable exception is the recent work of [25] which proposes to augment training data for a person re-identification network by replacing image backgrounds. We propose a far more elaborate, yet easily accessible means of data augmentation.

**Face synthesis for face recognition:**   The idea that face images can be synthetically generated in order to aid face recognition is not new. To our knowledge, it was originally proposed in [10] and then effectively used by [8,11,23,39]. Contrary to us, they all produced frontal faces which are presumably better aligned and easier to compare. They did not use other transformations to generate new images (e.g., other poses, facial expressions). More importantly, their images were used to *reduce* appearance variability, whereas we propose the opposite: to dramatically *increase* it to improve training and testing.

## 3    Domain Specific Data Augmentation for Face Images

We next detail our approach to augmenting a generic face dataset. We use the CASIA WebFace collection [46], enriching it with far greater per-subject appearance variations, without changing existing subject labels or losing meaningful information. Specifically, we propose to generate (synthesize) new face images, by introducing the following face specific appearance variations:

1. **Pose:** Simulating face image appearances across unseen 3D viewpoints.
2. **Shape:** Producing facial appearances using different 3D generic face shapes.
3. **Expression:** Specifically, simulating closed mouth expressions.

As previously mentioned, (1) can be considered an extension of *frontalization* techniques [11] to multiple views. Conceptually, however, they rendered new views to reduce variability for better alignment whereas we do this to increase

variability and better capture intra-subject appearance variations. Also noteworthy is that (2) explicitly contradicts previous assumptions on the importance of 3D facial shape in recognizing faces (e.g., [39]): Contrary to their claim that shape carries important subject related information we ignore shape cues by rendering the same face using different underlying shapes. As we later show, this introduces subtle appearance variations at training without changing perceived identities.

### 3.1   Pose Variations

In order to generate unseen viewpoints given a face image $\mathbf{I}$, we use a technique similar to the frontalization proposed by [11]. We begin by applying the facial landmark detector from [2]. Given these detected landmarks we estimate the six degrees of freedom pose for the face in $\mathbf{I}$ using correspondences between the detected landmarks $\mathbf{p}_i \in \mathbb{R}^2$ and points $\mathbf{P}_i \doteq \mathbf{S}(\mathbf{i}) \in \mathbb{R}^3$, labeled on a 3D generic face model $\mathbf{S}$. Here, $i$ indexes specific facial landmarks in $\mathbf{I}$ and the 3D shape $\mathbf{S}$.

As mentioned earlier, we use CASIA faces for augmentation. These faces are roughly centered in their images, and so detecting face bounding boxes was unnecessary. Instead, we used a fixed bounding box determined once beforehand.

Given the corresponding landmarks $\mathbf{p}_i \leftrightarrow \mathbf{P}_i$ we use PnP [9] to estimate extrinsic camera parameters, assuming the principal point is in the image center and then refining the focal length by minimizing landmark re-projection errors. This process gives us a perspective camera model mapping the generic 3D shape $\mathbf{S}$ on the image such as $\mathbf{p}_i \sim \mathbf{M} \, \mathbf{P}_i$ where $\mathbf{M} = \mathbf{K} \, [\mathbf{R} \, \mathbf{t}]$ is the camera matrix.

Given the estimated pose $\mathbf{M}$, we decompose it to obtain a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ containing rotation angles for the 3D head shape with respect to the image. We then create new rotation matrices $\mathbf{R}'_\theta \in \mathbb{R}^{3\times3}$ for unseen (novel) viewpoints by sampling different yaw angles $\theta$. In particular, since CASIA images are biased towards frontal faces, given an image $\mathbf{I}$ we render it at the fixed yaw values $\theta = \{0°, \pm40°, \pm75°\}$. Rendering itself is derived from [11], including soft-symmetry. Figure 2 shows viewpoint (pose) synthesis results for a training subject in CASIA, illustrating the 3D pose estimation process.

Note that in practice, faces are rendered with a uniform black background not shown here (original background from the image was not preserved in rendering).



**Fig. 2.** Adding pose variations by synthesizing novel viewpoints. Left: original image, detected landmarks, and 3D pose estimation. Right: rendered novel views.
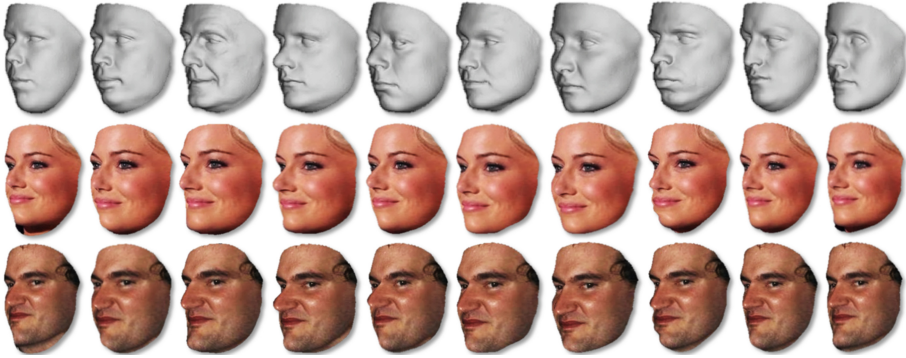
**Fig. 3.** Top: the ten generic 3D face shapes used for rendering. Bottom: faces rendered with the generic appearing right above them. Different shapes induce subtle appearance variations yet do not change the perceived identity of the face in the image.

### 3.2    3D Shape Variations

In the past, some argued that to truthfully capture the appearance of a subject's face under different viewpoints, its actual 3D shape must be used. They therefore attempted to estimate 3D face shapes directly from query images prior to frontalization [39]. Because this reconstruction process is unstable, particularly for challenging, unconstrained images, Hassner *et al.* [11] instead used a single generic 3D face to frontalize all faces. We propose the following simple compromise between these two approaches.

Rather than using a single generic 3D shape or estimating it from the image directly, we extend the procedure described in Sect. 3.1 to multiple generic 3D faces. In particular we add the set of generic 3D shapes $\mathcal{S} = \{\mathbf{S}_j\}_{j=1}^{10}$. We then simply repeat the pose synthesis procedure with these ten shapes rather than using only a single one.

We used generic 3D faces from the publicly available Basel set [29]. It provides ten high quality 3D face scans of different people representing different genders, ages, and weights. These scans are well aligned to one another. Hence, 3D landmarks need only be selected once, on one of these scans, and then directly transferred to the other nine. Figure 3 shows these ten generic models, along with images rendered to near profile view using each shape. Clearly, subjects in these images remain identifiable, despite the different underlying 3D shape, meeting the augmentation requirement of not changing subject labels. Yet each image is slightly but noticeably different from the rest, introducing appearance variations to this subject's image set.

### 3.3    Expression Variations

In addition to pose and shape, we also synthesize expression variations, specifically reducing deformations around the mouth. Given a face image **I** and its

**Fig. 4.** Expression synthesis examples. Top: example face images from the CASIA WebFace dataset. Bottom: synthesized images with closed mouths.

2D detected landmarks $\mathbf{p}_i$, and following pose estimation (Sect. 3.1) we estimate facial expression by fitting a 3D expression *Blendshape*, similarly to [21]. This is a linear combination of 3D generic face models with various basis expressions, including *mouth-closed*, *mouth-opened* and *smile*. Following alignment of the 3D face model and the 2D face image in both pose and expression, we perform image-based texture mapping to register the face texture onto the model. This is useful to quickly assign texture to our face model given that only one image is available. To synthesize expression, we manipulate the 3D textured face model to exhibit new expressions and render it back to the original image. This technique allows us to render a normalized expression where other image details, including hair and background, remain unchanged. In our experiments we do this to produce images with closed mouths. Some example synthesis results are provided in Fig. 4. Though slight artifacts are sometimes introduced by this process (some can be seen in Fig. 4) these typically do not alter the general facial appearances and are less pronounced than the noise often present in unconstrained images.

## 4    Face Recognition Pipeline

Data augmentation techniques are not restricted to training and are often also applied at test time. Our augmentations provide opportunities to modify the matching process by using different augmented versions of the input image. We next describe our recognition pipeline including these and other novel aspects.

### 4.1    CNN Training with Our Augmented Data

**Augmented training data:** Our pipeline employs a single CNN trained on both real and augmented data generated as described in Sect. 3. Specifically, training data is produced from original CASIA images. It consists of the following image types: (i) original CASIA images aligned by a simple similarity transform to two coordinate systems: roughly frontal facing faces (face yaw estimates in $[-30°... 30°]$) are aligned using nine landmarks on an ideal frontal template, while profile images (all other yaw angles) are aligned using the visible eye and the tip of the nose. (ii) Each image in CASIA is rendered from

three novel views in yaw angles $\{0°, \pm40°, \pm75°\}$, as described in Sect. 3.1. (iii) Synthesized views are produced by randomly selecting a 3D generic face model from $\mathcal{S}$ as the underlying face shape (see Sect. 3.2), thereby adding shape variations. (iv) Finally, a mouth neutralized version of each image is also added to the training (Sect. 3.3). This process raises the total number of training images from 494,414 in the original CASIA WebFace set to 2,472,070 images in our complete (pose+shape+expression) augmented dataset. Note that this process leaves the number of CASIA WebFace subjects unchanged, inflating only the number of images per subject (Fig. 1b).

**CNN fine-tuning:** We use the very deep VGGNet [34] CNN with 19 layers, trained on the large scale image recognition benchmark (ILSVRC) [30]. We fine tune this network using our augmented data. To this end, we keep all layers $\{\mathbf{W}_k, \mathbf{b}_k\}_{k=1}^{19}$ of VGGNet except for the last linear layer (FC8) which we train from scratch. This layer produces a mapping from the embedded feature $\mathbf{x} \in \mathbb{R}^D$ (FC7) to the subject labels $N = 10,575$ of the augmented dataset. It computes $\mathbf{y} = \mathbf{W}_{19}\mathbf{x} + \mathbf{b}_{19}$, where $\mathbf{y} \in \mathbb{R}^N$ is the linear response of FC8. Fine-tuning is performed by minimizing the soft-max loss:

$$\mathcal{L}(\{\mathbf{W}_k, \mathbf{b}_k\}) = -\sum_t \log\left(\frac{e^{\mathbf{y}_l}}{\sum_{g=1}^{N} e^{\mathbf{y}_g}}\right) \tag{1}$$

where $l$ is the ground-truth index over $N$ subjects and $t$ indexes all training images. Equation (1) is optimized using Stochastic Gradient Descent (SGC) with standard L2 norm over the learned weights. When performing back-propagation, we learn FC8 faster since it is trained from scratch while other network weights are updated with a learning rate an order of magnitude lower than FC8.

Specifically, we initialize FC8 with parameters drawn from a Gaussian distribution with zero mean and standard deviation 0.01. Bias is initialized with zero. The overall learning rate $\mu$ for the entire CNN is set to 0.001, except FC8 which uses learning rate of $\mu \times 10$. We decrease learning rate by an order of magnitude once validation accuracy for the fine tuned network saturates. Meanwhile, biases are learned twice as fast as the other weights. For all the other parameter settings we use the same values as originally described in [18].

### 4.2   Face Recognition with Synthesized Faces

**General matching process:** After training the CNN, we use the embedded feature vector $\mathbf{x} = f(\mathbf{I}; \{\mathbf{W}_k, \mathbf{b}_k\})$ from each image $\mathbf{I}$ as a face representation. Given two input images $\mathbf{I}_p$ and $\mathbf{I}_q$, their similarity, $s(\mathbf{x}_p, \mathbf{x}_q)$ is simply the normalized cross correlation (NCC) of their feature vectors.

The value $s(\mathbf{x}_p, \mathbf{x}_q)$ is the recognition score at the image level. In some cases a subject is represented by multiple images (e.g., a *template*, as in the Janus benchmark [16]). This plurality of images can be exploited to improve recognition at test time. In such cases, image sets are defined by $\mathcal{P} = \{\mathbf{x}_1, ..., \mathbf{x}_P\}$ and $\mathcal{Q} = \{\mathbf{x}_1, ..., \mathbf{x}_Q\}$ and a similarity score is defined between them: $s(\mathcal{P}, \mathcal{Q})$.

Specifically, we compute the pair-wise image similarity scores, $s(\mathbf{x}_p, \mathbf{x}_q)$, for all $\mathbf{x}_p \in \mathcal{P}$ and $\mathbf{x}_q \in \mathcal{Q}$, and pool these scores using a SoftMax operator, $s_\beta(\mathcal{P}, \mathcal{Q})$ (Eq. (2), below). Though our use of SoftMax is inspired by the SoftMax loss often used by CNNs, our aim is to get a robust score regression instead of a distribution over the subjects. SoftMax for set fusion can be seen as a weighted average in which the weight depends on the score when performing recognition. It is interesting to note that the SoftMax hyper-parameter $\beta$ controls the trade-off between averaging the scores or taking the max. (or min.). That is:

$$s_\beta(\cdot, \cdot) = \begin{cases} \max(\cdot) & \text{if } \beta \to \infty \\ \text{avg}(\cdot) & \text{if } \beta = 0 \\ \min(\cdot) & \text{if } \beta \to -\infty \end{cases} \quad \text{and } s_\beta(\mathcal{P}, \mathcal{Q}) = \frac{\sum_{p \in \mathcal{P}, q \in \mathcal{Q}} s(\mathbf{x}_p, \mathbf{x}_q) e^{\beta \; s(\mathbf{x}_p, \mathbf{x}_q)}}{\sum_{p \in \mathcal{P}, q \in \mathcal{Q}} e^{\beta \; s(\mathbf{x}_p, \mathbf{x}_q)}}.$$

$$(2)$$

Pair-wise scores are pooled using Eq. (2) and we finally average the SoftMax responses over multiple values of $\beta = [0...20]$ to get the final similarity score:

$$s(\mathcal{P}, \mathcal{Q}) = \frac{1}{21} \sum_{\beta=0}^{20} s_\beta(\mathcal{P}, \mathcal{Q}). \tag{3}$$

The use of $\beta$ positive values is due to our use of a score for recognition, so the higher the value, the better. In our experiments we found that the SoftMax operator reaches a remarkable trade-off between averaging the scores and taking the maximum. The improvement given by the proposed SoftMax fusion is shown in Table 1: we can see how the proposed method largely outperforms standard fusion techniques on IJB-A, in which subjects are described by templates.

**Table 1.** SoftMax template fusion for score pooling vs. other standard fusion techniques on the IJB-A benchmark for verification (ROC) and identification (CMC) resp.

| Fusion↓ | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | |
|---------|---------|----------|--------|--------|---------|
| Metrics → | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 |
| Min | 26.3 | 11.2 | 33.1 | 56.1 | 66.8 |
| Max | 77.6 | 46.4 | 84.8 | 93.3 | 95.6 |
| Mean | 79.9 | 53.0 | 84.6 | 94.7 | 96.6 |
| SoftMax | **86.6** | **63.6** | **87.2** | **94.9** | **96.9** |

**Exploiting pose augmentation at test time:** The Achilles heel of many face recognition systems is cross pose face matching; particularly when one of the two images is viewed at an extreme, near profile angle [6,22,45]. Directly matching two images viewed from extremely different viewpoints often leads to poor accuracy as the difference in viewpoints affects the similarity more than subject identities. To mitigate this problem, we suggest rendering both images from the same view: one that is close enough to the viewpoint of both images.

To this end, we leverage our pose synthesis method of Sect. 3.1 to produce images in poses better suited for recognition and matching.

Cross pose rendering can, however, come at a price: Synthesizing novel views for faces runs the risk of producing meaningless images whenever facial landmarks are not accurately localized and the pose estimate is wrong. Even if pose was correctly estimated, warping images across poses involves interpolating intensities, which leads to smoothing artifacts and information loss. Though this may affect training, it is far more serious at test time where we have few images to compare and ruining one or both can directly affect recognition accuracy.

Rather than commit to pose synthesis or its standard alternative, simple yet robust in-plane alignment, we use both: We found that pose synthesis and in-plane alignment are complimentary and by combining the two, recognition performance improves. For an image pair $(\mathbf{I}_p, \mathbf{I}_q)$ we compute two similarity scores: one using in-plane aligned images and the other using images rendered to a mutually convenient view. This view is determined as follows: If the two images are near frontal then we frontalize them [11], if they are both near profile we render to $75°$, otherwise we render to $40°$.

When matching templates, $(\mathcal{P}, \mathcal{Q})$, scores computed for in-plane aligned image pairs and pose synthesized pairs are pooled separately using Eq. (2). This is equivalent to comparing the sets $\mathcal{P}$ and $\mathcal{Q}$ twice, once with each alignment. The two similarities are then averaged for a final template level score.

## 5    Experiments

We tested our approach extensively on the IARPA Janus benchmarks [16] and LFW [13]. We perform a minimum of database specific training, using the training images prescribed by each benchmark protocol. Specifically, we perform Principal Component Analysis (PCA) on the training images of the target dataset with the features $\mathbf{x}$ extracted from the CNN trained on augmented data. This did not include dimensionality reduction; we did not cut any component after PCA projection. Following this, we apply root normalization to the new projected feature, i.e., $\mathbf{x} \rightarrow \text{sign}(\mathbf{x}) \, |\mathbf{x}|^c$, as previously proposed for the Fisher Vector encoding in [31]. We found that a value of $c = 0.65$ provides a good baseline across all the experiments. For each dataset we report the contribution of our augmentations compared with state-of-the-art methods which use millions of training images.

### 5.1    Results on the IJB-A Benchmarks

IJB-A is a new publicly available benchmark released by NIST to raise the challenges of unconstrained face identification and verification methods. Both IJB-A and the Janus CS2 benchmark share the same subject identities, represented by images viewed in extreme conditions, including pose, expression and illumination variations, with IJB-A splits generally considered more difficult than those in CS2. The IJB-A benchmarks consist of face verification (1:1) and face

identification (1:N) tests. Contrary to LFW, Janus subjects are described using templates containing mixtures of still-images and video frames.

It is important to note that the Janus set has some overlap with the images in the CASIA WebFace collection. In order to provide fair comparisons, our CNNs were fine tuned on CASIA subjects that are *not* included in Janus (Sect. 4.1).

**Face detection:** Our pipeline uses the face landmark detector of [2] for head pose estimation and alignment. Although we found this detector quite robust, it failed to detect landmarks on some of the more challenging Janus faces. Whenever the detector failed on all the images in the same template, we use the images cropped to their facial bounding boxes as provided in the Janus data.

**Video pooling:** Whenever face templates include multiple frames from a single video we pool together CNN features extracted from the same video: this, by simple element wise average over all the features extracted from that video's frames (i.e., features are not pooled across videos but only within each video). Similar pooling techniques were very recently demonstrated to provide substantial performance enhancements (e.g., [35]). We refer to this technique as *video pooling* and report its influence on our system, and, whenever possible, for our baselines. This process does not change the general matching process explained in Sect. 4.2: When performing video-pooling an entire video is represented as a single pooled feature vector and treated as a single image.

In all our IJB-A and Janus CS2 results this method provided noticeable performance boosts: we compare video pooling to pair-wise single image comparisons (referred as *without video pooling* in our results).

**Ablation Study:** We analyze the contribution of each augmentation technique on the IJB-A dataset. Clearly, the biggest gain is given by pose augmentation (red curve) over the baseline (blue curve) in Fig. 5a. The improvement is especially noticeable in the rank-1 recognition rate for identification. The effect of video pooling along with each data augmentation method is provided in Table 2.

We next evaluate the effect of pose synthesis at test time combined with the standard in-plane alignment (Sect. 4.2), in Table 3 and in Fig. 5b. Evidently, these methods combined contribute to achieving state-of-the-art performance on the IJB-A benchmark. We conjecture that this is mainly due to three contributions:
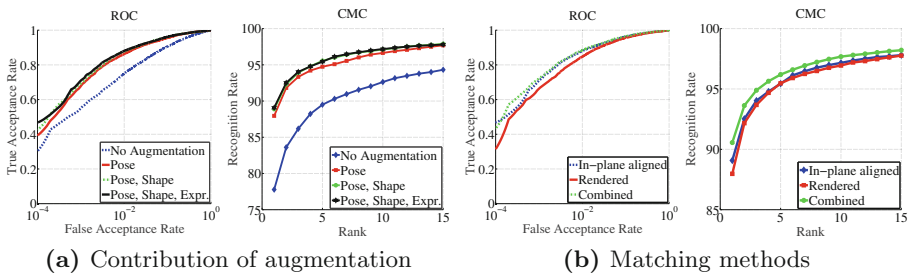


**(a)** Contribution of augmentation          **(b)** Matching methods

**Fig. 5.** Ablation study of our data synthesis and test time matching methods on IJB-A.

**Table 2.** Effect of each augmentation on IJB-A performance on verification (ROC) and identification (CMC), resp. Only in-plane aligned images used in these tests.

| | Without Video pooling | | | | | With Video pooling | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | |
| Augmentation ↓ | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 |
| Metrics → | | | | | | | | | | |
| No Augmentation | 74.5±1.88 | 54.3±2.81 | 77.1±1.21 | 89.0±1.02 | 92.3±0.89 | 75.0±2.11 | 55.0±2.55 | 77.8±1.32 | 89.5±0.99 | 92.6±0.69 |
| Pose | 84.9±1.56 | 62.3±6.53 | 86.3±0.97 | 94.5±0.61 | 96.5±0.55 | 86.3±1.81 | 67.9±5.37 | 88.0±1.18 | 94.7±0.80 | 96.6±0.59 |
| Pose, Shapes | 86.3±1.66 | 62.0±6.21 | 87.0±0.81 | 94.8±0.69 | **96.9±0.48** | 87.8±1.46 | 69.2±4.80 | 88.9±0.96 | **95.6±0.69** | 97.1±0.52 |
| Pose, Shapes, Expr. | **86.6±1.27** | **63.6±6.00** | **87.2±0.96** | 94.9±0.77 | 96.9±0.58 | **88.1±1.42** | **71.0±5.24** | **89.1±1.04** | 95.4±0.88 | **97.2±0.59** |

**Table 3.** Effect of in-plane alignment and pose synthesis at test-time (matching) on IJB-A dataset respectively for verification (ROC) and identification (CMC).

| | Without Video pooling | | | | | With Video pooling | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | |
| Image type ↓ | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 |
| Metrics → | | | | | | | | | | |
| In-plane aligned | 86.6±1.27 | 63.6±6.00 | 87.2±0.96 | 94.9±0.77 | 96.9±0.58 | 88.1±1.42 | 71.0±5.24 | 89.1±1.04 | 95.4±0.88 | 97.2±0.59 |
| Rendered | 84.7±1.50 | 64.6±4.40 | 87.3±1.21 | 95.0±0.59 | 96.8±0.69 | 84.8±1.38 | 66.4±4.88 | 88.0±1.15 | 95.5±0.69 | 96.9±0.58 |
| Combined | **87.8±1.22** | **67.4±4.32** | **89.5±1.16** | **95.8±0.59** | **97.4±0.39** | **88.6±1.65** | **72.5±4.41** | **90.6±1.27** | **96.2±0.65** | **97.7±0.42** |

domain-specific augmentation when training the CNN, combination of SoftMax operator, video pooling and finally pose synthesis at test time.

**Comparison with the state-of-the-art:** Our proposed method achieves state of the art results in the IJB-A benchmark and Janus CS2 dataset (see Tab. 4). In particular, it largely improves over the off the shelf commercial systems COTS and GOTS [16] and Fisher Vector encoding using frontalization [4]. This gap can be explained by the use of deep learning alone. Even compared with deep learning based methods, however, our approach achieves superior performance and with very wide margins. This is true even comparing our results to [41], who use seven networks and fuse their output with the COTS system. Moreover, our method improves in IJB-A verification over [41] in 15 % TAR at FAR = 0.01 and ~20 % TAR at FAR = 0.001, also showing a better rank-1 recognition rate.

It is interesting to compare our results to those reported by [5,32]. Both fine tuned their deep networks on the ten training splits of each benchmark, at substantial computational costs. Some idea of the impact this fine tuning can have on performance is available by considering the huge performance gap between results reported before and after fine tuning in [5]. Our own results, obtained by training our CNN once on augmented data, far outperform those of [32] also largely outperforming those reported by [5]. We conjecture that by training the CNN with augmented data we avoid further specializing all the parameters of the network on the target dataset. Tuning deep models on in-domain data is computationally expensive and thus, avoiding overfitting the network at training time is preferable.

**Table 4.** Comparative performance analysis on JANUS CS2 and IJB-A respectively for verification (ROC) and identification (CMC). f.t. denotes fine tuning a deep network multiple times for each training split. A network trained once with our augmented data achieves mostly superior results, without this effort.

| Methods ↓ | CS2 Ver. (TAR) | | CS2 Id. (Rec. Rate) | | | IJB-A Ver. (TAR) | | IJB-A Id. (Rec. Rate) | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metrics → | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 | FAR0.01 | FAR0.001 | Rank-1 | Rank-5 | Rank-10 |
| COTS [16] | 58.1±5.4 | 37 | 55.1±3.0 | 69.4±1.7 | 74.1±1.7 | – | – | – | – | – |
| GOTS [16] | 46.7±6.6 | 25 | 41.3±2.2 | 57.1±1.7 | 62.4±1.8 | 40.6±1.4 | 19.8±0.8 | 44.3±2.1 | 59.5±2.0 | – |
| OpenBR [17] | – | – | – | – | – | 23.6±0.9 | 10.4±1.4 | 24.6±1.1 | 37.5±0.8 | – |
| Fisher Vector [4] | 41.1±8.1 | 25.0 | 38.1±1.8 | 55.9±2.1 | 63.7±2.5 | – | – | – | – | – |
| Wang et al. [41] | – | – | – | – | – | 73.3±3.4 | 51.4±6.0 | 82.0±2.4 | 92.9±1.3 | – |
| Chen et al. [5] | 64.9±1.5 | 45 | 69.4±1.2 | 80.9±1.1 | 85.0±0.9 | 57.3±2.0 | – | 72.6±3.4 | 84.0±2.3 | 88.4±2.5 |
| Pooling Faces [12] | 87.8 | 74.5 | 82.6 | 91.8 | 94.0 | 81.9 | 63.1 | 84.6 | 93.3 | 95.1 |
| Deep Multi-Pose [1] | 89.7 | – | 86.5 | 93.4 | 94.9 | 78.7 | – | 84.6 | 92.7 | 94.7 |
| PAMs [24] | 89.5±0.6 | 78.0±1.4 | 86.2±0.9 | 93.1±0.5 | 94.9±0.6 | 82.6±1.8 | 65.2±3.7 | 84.0±1.2 | 92.5±0.8 | 94.6±0.7 |
| Chen et al. [5] (f.t.) | 92.1±1.3 | 78 | 89.1±1.0 | **95.7±0.7** | **97.2±0.5** | 83.8±4.2 | – | 90.3±1.2 | **96.5±0.8** | **97.7±0.7** |
| Swami S. et al. [32] (f.t.) | – | – | – | | | 79±3.0 | 59±5.0 | 88±1.0 | 95±0.7 | – |
| Ours | **92.6±0.61** | **82.4±1.52** | **89.8±0.97** | 95.6±0.58 | 96.9±0.60 | **88.6±1.65** | **72.5±4.41** | **90.6±1.27** | 96.2±0.65 | **97.7±0.42** |

## 5.2    Results on Labeled Faces in the Wild

For many years LFW [13] was the standard benchmark for unconstrained face verification. Recent methods dominating LFW scores use millions of images collected and labeled by hand in order to obtain their remarkable performances. To test our approach, we follow the standard protocol for unrestricted, labeled outside data and report the mean classification accuracy as well as the 100 % - EER (Equal Error Rate). We prefer to use 100 % - EER in general because it is not dependent on the selected classification threshold but we still report verification accuracy to be comparable with the other methods.

**Improvement for each augmentation:**  Figure 6a provides ROC curves for each augmentation technique used in our approach. The green curve represents our baseline, that is the CNN trained on in-plane aligned images with respect to a frontal template. The ROC improves by a good margin when we inject unseen rendered images across poses into each subject. Indeed the 100 % - EER improves by +1.67 %. Moreover, by adding both shapes and expressions, performance improves even more, reaching 100 % - EER rate of 98.00 % (red curve). See Fig. 6b for a comparison with methods trained on millions of downloaded images.

## 5.3    Summary of Results and Discussion

The results in this section clearly show that synthesizing training images using domain tools leads to dramatic increase in recognition accuracy. This may be attributed to the potential of domain specific augmentation to infuse training data with important intra-subject appearance variations; the very variations that seem hardest to obtain by simply downloading more images. We believe that the extent to which this domain specific data augmentation affects performance depends on the data set used as seed for the augmentation (in this paper, CASIA): Different sets have their own appearance biases and so would benefit differently from the introduction of different synthesized appearance variations.
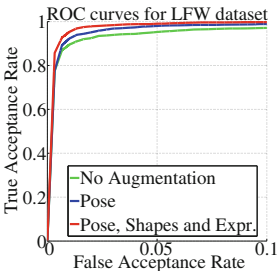


| Method | Real | Synth | Net | Acc. (%) | 100% - EER |
|---|---|---|---|---|---|
| Fisher Vector Faces [27] | – | – | – | 93.0 | 93.1 |
| DeepFace [39] | 4M | – | 3 | 97.35 | – |
| Fusion [40] | 500M | – | 5 | 98.37 | – |
| FaceNet [33] | 200M | – | 1 | 98.87 | – |
| FaceNet + Alignment [33] | 200M | – | 1 | 99.63 | – |
| VGG Face [28] | 2.6M | – | 1 | – | 97.27 |
| VGG Face (triplet loss) [28] | 2.6M | – | 1 | 98.95 | 99.13 |
| Us, no aug. | 495K | – | 1 | 95.31 | 95.26 |
| Us, aug. pose | 495K | 2M | 1 | 97.01 | 96.93 |
| Us, aug. pose, shape, expr. | 495K | 2.4M | 1 | 98.06 | 98.00 |

**(a)** Ablation Study          **(b)** Results for methods trained on millions of images

**Fig. 6.** LFW verification results. (a) Break-down of the influence of different training data augmentation methods. (b) Performance comparison with state of the art methods, showing the numbers of real (original) and synthesized training images, number of CNNs used by each system, accuracy and 100 %-EER

We caution that despite the ability to simulate novel face samples, our system still relies on a deep-learned model and alone, a generative model which synthesizes gallery images to match the probe is still insufficient. The learned model currently handles appearance variations such as occlusions, image quality and other confounding factors, all of which may affect the final face representation and are not handled by our augmentation process. Discriminative deep-learned models are therefore required to accurately compare these representations.

Finally, a comparison of our results on LFW to those reported by methods trained on millions of images [28,33,39,40], shows that with the initial set of less than 500K publicly available images, our method surpasses [39] and [28] (without their metric learning, not applied here), falling only slightly behind the rest.

## 6    Conclusions

This paper makes several important contributions. First, we show how domain specific data augmentation can be used to generate (synthesize) valuable additional data to train effective face recognition systems, as an alternative to expensive data collection and labeling. Second, we describe a face recognition pipeline with several novel details. In particular, its use of our data augmentation for matching across poses in a natural manner. Finally, *in answer to the question in the title*, our extensive analysis shows that though there is certainly a benefit to downloading increasingly larger training sets, much of this effort can be substituted by simply synthesizing more face images.

There are several compelling directions of extending our work. Primarily, additional face specific data augmentation methods can be used to provide additional intra subject appearance variations. Appealing potential augmentation techniques, not used here, are facial age synthesis [15] or facial hair manipulations [26]. Finally, beyond faces there may be other domains where such approach is relevant and where the introduction of synthetically generated training data can help mitigate the many problems of data collection for CNN training.

## References

1. AbdAlmageed, W., Wu, Y., Rawls, S., Harel, S., Hassner, T., Masi, I., Choi, J., Leksut, J., Kim, J., Natarajan, P., Nevatia, R., Medioni, G.: Face recognition using deep multi-pose representations. In: Winter Conference on Applications of Computer Vision (2016)

2. Baltrusaitis, T., Robinson, P., Morency, L.P.: Constrained local neural fields for robust facial landmark detection in the wild. In: Proceedings of the International Conference on Computer Vision Workshops (2013)

3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: Proceedings of the British Machine Vision Conference (2014)

4. Chen, J.C., Sankaranarayanan, S., Patel, V.M., Chellappa, R.: Unconstrained face verification using fisher vectors computed from frontalized faces. In: International Conference on Biometrics: Theory, Applications and Systems (2015)

5. Chen, J.C., Patel, V.M., Chellappa, R.: Unconstrained face verification using deep cnn features. In: Winter Conference on Applications of Computer Vision (2016)

6. Ding, C., Xu, C., Tao, D.: Multi-task pose-invariant face recognition. Trans. Image Process. **24**(3), 980–993 (2015)

7. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the International Conference on Computer Vision, pp. 2650–2658 (2015)

8. Ferrari, C., Lisanti, G., Berretti, S., Del Bimbo, A.: Dictionary learning based 3D morphable model construction for face recognition with varying expression and pose. In: 3DV (2015)

9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)

10. Hassner, T.: Viewing real-world faces in 3d. In: Proceedings of the International Conference on Computer Vision, pp. 3607–3614 (2013)

11. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: Proceedings of the International Conference on Computer Vision Pattern Recognition (2015)

12. Hassner, T., Masi, I., Kim, J., Choi, J., Harel, S., Natarajan, P., Medioni, G.: Pooling faces: template based face recognition with pooled face images. In: Proceedings of the International Conference on Computer Vision Pattern Recognition Workshops, June 2016

13. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07-49, UMass, Amherst, October 2007

14. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The MegaFace benchmark: 1 million faces for recognition at scale. In: Proceedings of the International Conference on Computer Vision Pattern Recognition (2016)

15. Kemelmacher-Shlizerman, I., Suwajanakorn, S., Seitz, S.M.: Illumination-aware age progression. In: Proceedings of the International Conference on Computer Vision Pattern Recognition, pp. 3334–3341. IEEE (2014)

16. Klare, B.F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Burge, M., Jain, A.K.: Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus benchmark A. In: Proceedings of the International Conference on Computer Vision Pattern Recognition, pp. 1931–1939 (2015)

17. Klontz, J., Klare, B., Klum, S., Taborsky, E., Burge, M., Jain, A.K.: Open source biometric recognition. In: International Conference on Biometrics: Theory, Applications and Systems (2013)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems, pp. 1097–1105 (2012)

19. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. Trans. Neural Netw. **8**(1), 98–113 (1997)
20. Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. In: Proceedings of the International Conference on Computer Vision Pattern Recognition Workshops, June 2015. http://www.openu.ac.il/home/hassner/projects/cnn_agegender
21. Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F., Deng, Z.: Practice and theory of blendshape facial models. In: Eurographics 2014 (2014)
22. Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J.: Probabilistic elastic matching for pose variant face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3499–3506 (2013)
23. Masi, I., Ferrari, C., Del Bimbo, A., Medioni, G.: Pose independent face recognition by localizing local binary patterns via deformation components. In: International Conference on Pattern Recognition (2014)
24. Masi, I., Rawls, S., Medioni, G., Natarajan, P.: Pose-aware face recognition in the wild. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2016)
25. McLaughlin, N., Martinez Del Rincon, J., Miller, P.: Data-augmentation for reducing dataset bias in person re-identification. In: International Conference on Advanced Video and Signal Based Surveillance. IEEE (2015)
26. Nguyen, M.H., Lalonde, J.F., Efros, A.A., De la Torre, F.: Image-based shaving. Comput. Graph. Forum **27**(2), 627–635 (2008)
27. Parkhi, O.M., Simonyan, K., Vedaldi, A., Zisserman, A.: A compact and discriminative face track descriptor. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2014)
28. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Machine Vision Conference (2015)
29. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3d face model for pose and illumination invariant face recognition. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, pp. 296–301, September 2009
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
31. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
32. Sankaranarayanan, S., Alavi, A., Chellappa, R.: Triplet similarity embedding for face verification (2016). arXiv preprint: arXiv:1602.03418
33. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
35. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the International Conference on Computer Vision, pp. 945–953 (2015)
36. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Neural Information Processing System, pp. 1988–1996 (2014)

37. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: face recognition with very deep neural networks (2015). arXiv preprint: arXiv:1502.00873
38. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the International Conference on Computer Vision Pattern Recognition. IEEE (2014)
39. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 1701–1708. IEEE (2014)
40. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Web-scale training for face identification. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (2015)
41. Wang, D., Otto, C., Jain, A.K.: Face search at scale: 80 million gallery (2015). arXiv preprint: arXiv:1507.07242
42. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Proceedings of the International Conference on Computer Vision Pattern Recognition, pp. 529–534. IEEE (2011)
43. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the International Conference on Computer Vision (2015)
44. Yang, H., Patras, I.: Mirror, mirror on the wall, tell me, is the error small? In: Proceedings of the International Conference on Computer Vision Pattern Recognition (2015)
45. Yi, D., Lei, Z., Li, S.: Towards pose robust face recognition. In: Proceedings of the International Conference on Computer Vision Pattern Recognition, pp. 3539–3545 (2013)
46. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch (2014). arXiv preprint: arXiv:1411.7923, http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html
47. Zhou, E., Cao, Z., Yin, Q.: Naive-deep face recognition: touching the limit of LFW benchmark or not? (2015). arXiv preprint: arXiv:1501.04690