Learning Pose-Aware Models for Pose-Invariant Face Recognition in the Wild

Iacopo Masi, Feng-Ju Chang, Jongmoo Choi, Shai Harel, Jungyeon Kim, Kanggeon Kim, Jatuporn Leksut, Stephen Rawls, Yue Wu, Tal Hassner, Wael AbdAlmageed, Gerard Medioni, Louis-Philippe Morency, Prem Natarajan, and Ram Nevatia

Abstract—We propose a method designed to push the frontiers of unconstrained face recognition in the wild with an emphasis on extreme out-of-plane pose variations. Existing methods either expect a single model to learn pose invariance by training on massive amounts of data or else normalize images by aligning faces to a single frontal pose. Contrary to these, our method is designed to explicitly tackle pose variations. Our proposed Pose-Aware Models (PAM) process a face image using several pose-specific, deep convolutional neural networks (CNN). 3D rendering is used to synthesize multiple face poses from input images to both train these models and to provide additional robustness to pose variations at test time. Our paper presents an extensive analysis of the IARPA Janus Benchmark A (IJB-A), evaluating the effects that landmark detection accuracy, CNN layer selection, and pose model selection all have on the performance of the recognition pipeline. It further provides comparative evaluations on IJB-A and the PIPA dataset. These tests show that our approach outperforms existing methods, even surprisingly matching the accuracy of methods that were specifically fine-tuned to the target dataset. Parts of this work previously appeared in [1] and [2].

Index Terms—Face recognition, CNN, pose-aware

1 Introduction and Motivation

There has been a flurry of advances in face recognition in recent years, with some techniques claiming to meet [3] or even surpass [4], [5] human face verification performance. This is, in particular, reflected in the saturated results reported under certain conditions on the standard Labeled Faces in the Wild (LFW) benchmark [6].

Recognizing that under real-world conditions current face verification systems still have shortcomings, a new benchmark was recently proposed in [8]: the IARPA Janus Benchmark A (IJB-A). IJB-A was designed to encourage work on novel aspects of unconstrained face recognition. One such aspect is an emphasis on a much wider range of facial poses compared to previous benchmarks (most notably, LFW). IJB-A consequently represents far more challenging viewing conditions and an elevated bar for face recognition systems to clear.

The broader range of IJB-A facial poses is illustrated in Fig. 1. It shows the distribution of facial yaw angles (out-of-plane yaw rotations of the head), comparing faces in LFW and the the more recent and much larger CASIA WebFace collection [7] with those in IJB-A. Evidently, IJB-A images encompass a wider variety of head poses than both previous sets. The two *bumps* in the extreme edges of the IJB-A distribution further suggest that in designing it, a particular emphasis was placed on injecting large numbers of profile and near-profile views.

One implication of these extreme poses is illustrated in Fig. 2, which provides example faces selected from the range of yaw angles available in LFW and IJB-A. Each example is accompanied by its frontalized (aligned, front-facing) view. These examples underscore the challenges of aligning such near-profile views to frontal positions, as proposed by, e.g., [9]. The figure shows that a single, frontal reference coordinate system is insufficient when processing images in near-profile views; rendering profile faces to a profile reference view introduces far less artifacts and better preserves facial appearances.

In addition to emphasizing poses, IJB-A also introduced the concept of set-to-set matching, where sets are composed of heterogeneous media rather than matching two single images, as in the LFW benchmark, or two sets of frames from two videos, as in the YouTube Faces (YTF) benchmark [10], IJB-A matches two *sets with mixed media type*. Each set contains images and videos from multiple sources. This specific case of set matching is also designed to reflect

- *I. Masi, F.-J. Chang, J. Choi, J. Kim, K. Kim, J. Leksut, G. Medioni, and R. Nevatia are with the Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA 90007. E-mail: {iacopoma, fengjuch, jongmooc, jungyeon, kanggeon.kim, leksut, medioni, nevatia}@usc.edu.*
- *T. Hassner is with the Information Sciences Institute, University of Southern California, Los Angeles, CA 90007, and also with the Open University of Israel, Ra'anana 4353701, Israel. E-mail: hassner@isi.edu.*
- *S. Rawls, Y. Wu, W. AbdAlmageed, and P. Natarajan are with the Information Sciences Institute, University of Southern California, Los Angeles, CA 90007. E-mail: {srawls, yue_wu, wamageed, pnataraj}@isi.edu.*
- *S. Harel is with the Open University of Israel, Ra'anana 4353701, Israel. E-mail: shaih82@gmail.com.*
- *L.P. Morency is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: morency@cs.cmu.edu.*

Manuscript received 16 Sept. 2016; revised 26 Sept. 2017; accepted 17 Dec. 2017. Date of publication 11 Jan. 2018; date of current version 16 Jan. 2019. (Corresponding author: Iacopo Masi.)
Recommended for acceptance by E. G. Learned-Miller.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2792452

0162-8828 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

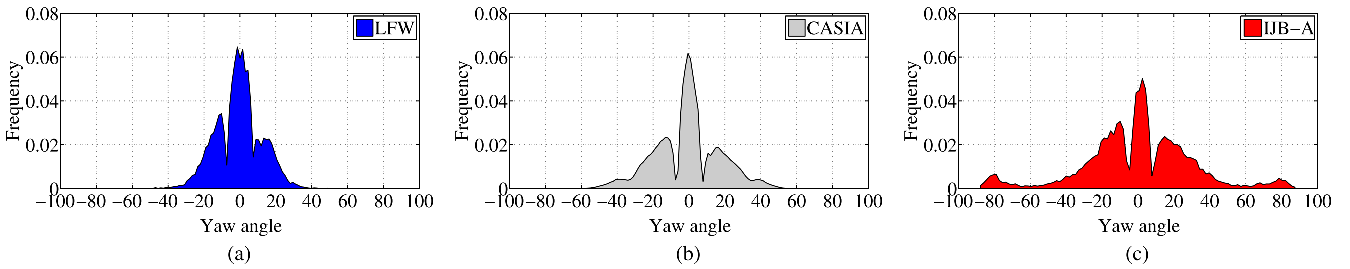


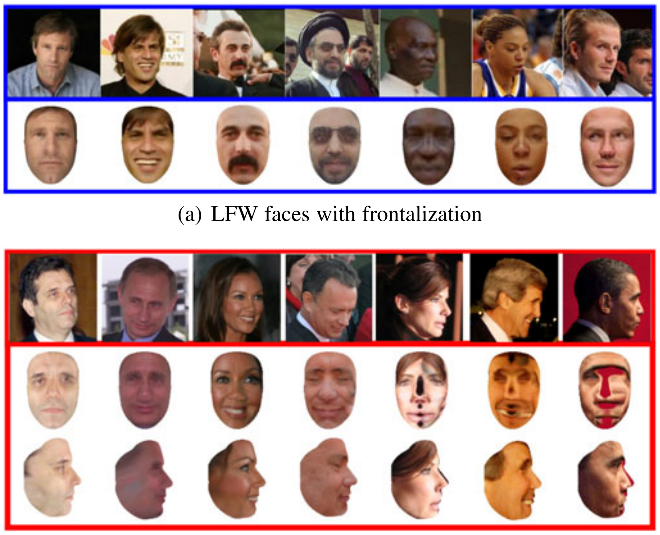
Fig. 1. Distributions of head yaw angles in LFW [6] and CASIA WebFaces [7] (the latter set used to train our system), compared to the new IJB-A benchmark on which our system is tested [8]. These demonstrate the far wider range of poses and the increased numbers of near-profile views in IJB-A.

real-world face verification settings where visual information from multiple sources can be collected and used to represent subject appearances. Since sets can contain images of a subject over multiple poses, it becomes important to consider how to handle set-to-set matching and how pose variations take part in the matching process.

Because much of the existing work on face recognition was developed and tested primarily on LFW, attention to extreme poses was never truly necessary. As far as we know, and as also noted by others [11], extreme poses were never directly addressed in previous work. Therefore, our work aims to address wide ranges of pose variations and large numbers of near-profile views.

Contributions. This paper offers the following contributions.

(i) *Pose-Aware Models for face recognition:* We propose a face recognition approach that explicitly considers and handles pose variability, including faces in extreme, near-profile views. Our method trains multiple pose-specific models and effectively exploits these models when matching images with faces appearing in different poses. Most previous approaches rely only on a single frontal-pose model [13], [14], possibly normalizing images via frontalization [3], [9]. We show why these approaches cannot be applied to wider pose variations. Contrary to these methods, we propose to handle pose variability by learning Pose-Aware Models (PAMs) for frontal, half-profile and full-profile poses. Similar to [15], [16], PAMs also allow us to overcome a main limitation of the pose-deficient training set, resulting in a better trained system.


(a) LFW faces with frontalization

(b) JANUS faces with frontalization and rendered to profile view

Fig. 2. Faces selected from the range of available yaw angles in the LFW and IJB-A benchmarks. Faces displayed in increasing yaw angles from left to right. Each face shows its frontalized view, and in the case of IJB-A, also a near-profile aligned view. *(a)* In LFW images, frontalization typically suffices as a means of compensating for pose variations. *(b)* Frontalization of IJB-A faces in extreme poses introduces serious artifacts, if performed following [9], [12]. These are addressed by aligning the images to near-profile views instead.

(ii) *Multiple ideal coordinates for out-of-plane face alignment:* We extend frontalization to multi-poses in order to mitigate the severe artifacts produced when frontalizing faces in extreme poses (demonstrated also in Fig. 2). We describe an entire face recognition pipeline developed in order to exploit these multi-pose models while performing pose-aware face recognition.

(iii) *Co-training:* We develop a method for effective training of deep Convolutional Neural Network (CNN) pose-aware models. Co-training is designed to address the problem of training CNN models for extreme poses where relatively few example faces are available for training (see, e.g., Fig. 1b). To clarify, we differentiate our contribution from the recent work of [17], which uses multi-task learning, and the multi-view perception (MVP) of [18]. Both of these methods train deep networks to interpolate between views, whereas our PAM uses rendering techniques to generate synthetic new views.

Our PAM model is rigorously tested, and we provide an analysis of its various components and their effect on recognition performance on the IJB-A benchmark. We further compare PAM to existing alternatives on the IJB-A and PIPA [19] benchmarks. Remarkably, PAM outperforms DeepFace [3] and [20] on PIPA despite having significantly less training data and without being tailored to that set. Finally, to promote reproduction of our work, we make our PAM CNN models publicly available.[1]

2 Related Work

Researchers have long acknowledged that face matching techniques struggle when presented with images of faces in extremely different poses [21], [22], [23], [24]. In order to address these problems, early work suggested training face classifiers for different poses, with one example being [25] which extends the basic Eigenface approach to multiple poses. Subsequent methods for addressing multiple poses mainly

[1] Available from http://goo.gl/forms/NK6adyd7DFJhlmHG2

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

considered face images obtained under controlled conditions, such as those available in the Multi-PIE data set [26].

Face Recognition with CNNs. An immensely popular recent approach is to address facial pose variations by training CNNs on massive data sets representing large variabilities of facial poses. These methods aim to obtain a single pose invariant representation. FaceNet [5] shows that it is possible to learn a compact embedding for faces with an end-to-end learning system trained on 260 million images. Though DeepID [27] uses ensembles of CNN representations, these are trained on different facial patches, rather than different poses, in an effort to learn pose invariance at the patch level. Along with their joint Bayesian metric, they showed remarkable performance on the LFW benchmark. Their work was later extended in [28] to show how the CNN learns sparse features that implicitly encode attribute information such as gender. Finally, an effort was made to produce more rich and varied data for training the CNN in [7], [29], [30].

Some previous work attempted to let the network disentangle the identity and the viewpoint by either performing multi-task learning [17] or using multi-view perceptrons [18]. The drawback of these latter methods is that they are only trained on constrained images in the Multi-PIE dataset in which the pose is manually controlled. These methods were consequently not tested on unconstrained benchmarks such as IJB-A.

Contrary to these, the recent work of Wang et al. [13] used CNNs to show accurate face identification on a gallery of 80 million images and on the IJB-A benchmark. Chen et al. [14], [31] demonstrated compelling results on IJB-A by using a single CNN trained from scratch on frontal views and fine-tuned to learn an effective metric on the target dataset.

Finally, recent efforts used 3D rendering techniques to inflate the training set by synthesizing novel poses, expressions, and using different 3D generic shapes [15]; others improved the method in [32] by jittering the VGGFace set [30] with new 3D views and illumination variations [16]. Contrary to our method, they used 3D rendering to massively augment the training set and to learn a single CNN on the augmented set.

Using 3D Models to Address Facial Poses. 3D computer graphics can be used to model the appearance of faces in different poses. In [33] a 3D average face model was used rather than relying on 3D cylindrical or ellipsoid models of earlier work. Prabhu et al. [34] proposed an efficient way to estimate a 3D model from a single frontal image using a Generic Elastic Model (GEM). The method was further improved by considering diverse average values of depth per ethnic group [35]. Their work is noteworthy because it was one of the first to match frontal versus profile images using rendering. 3D data was particularly used for matching rendered images of unconstrained 2D faces, accounting for small pose variations in [36]. Others have since introduced the idea of rotating a face to get different training poses [37].

In recent years, some researchers have proposed normalizing facial pose by synthesizing a novel view of the face in a canonical frontal view. The work of [38] was the first to report improved face recognition performance by rendering profile faces to frontal views. Others have since proposed different methods for the same underlying idea of *frontalization* (e.g., [3], [9], [39], [40], [41], [42]). The DeepFace of [3] applied a 3D face shape estimation method based on the one proposed in [12] to unconstrained face images. These 3D estimates were then used to frontalize a training set for their CNN system.

More recently, and particularly relevant to our work, is the frontalization of [9]. It proposed the use of a single generic 3D face shape, without modification, in order to simplify this frontalization process and improve results. We use a similar, single-generic approach here, though we extend it to multiple poses and apply it in a substantially different recognition pipeline.

Finally, rather than use an explicit 3D model, Zhu et al. [43] use a CNN to directly recover and normalize a near-frontal face to a frontal view, using that as an alternative alignment technique.

3 Overview of Our Approach

Given an image $\mathbf{I}$, we define the face pose distribution as $p(\mathbf{p}|\mathbf{I})$, where $\mathbf{p}$ is a vector of the three 3D rotation angles of the head. In this work we *do not* assume that this distribution is dominated by near-frontal faces. Instead, we propose to learn multiple pose-specific CNN models. An overview of our face matching pipeline is illustrated in Fig. 3. Given two subjects, each represented by a set which can contain multiple images and videos, we begin by detecting facial landmarks on all images and video frames. These landmarks are used to align each image to five different reference poses (see Section 4). Pose-aware CNN models, trained on the 500k images of the CASIA WebFace dataset [7] and described in Section 5, are used to extract features for the images in each pose. These features are matched across the two sets. We develop two different methods of matching these features and describe them in Section 6.2. We provide experimental results in Section 7.

4 Multiple Alignments

Detected facial landmarks provide an easy means of compensating—for facial roll when the face is near-frontal and for pitch when the face is near-profile—by using basic, in-plane alignment. We therefore focus our attention on compensating for yaw variations by assuming $p(\mathbf{p}|\mathbf{I}) \approx p(\psi|\mathbf{I})$, where ψ represents the face yaw angle. We further note that compensating for out-of-plane variations using frontalization [3], [9] could be a noisy process that becomes harder as the input face rotates closer to a profile view (see, e.g., Fig. 2). We therefore propose a method that extends the concept of frontalization to multiple modes of the pose distribution.

Finally, rather than choosing a single face alignment method, we fuse the outcome of multiple face alignment pipelines. This is done in order to prevent errors in landmark localization from producing misaligned or corrupt faces. This is a particular concern when using out-of-plane alignment methods such as frontalization which can project the face outside the output view when landmarks are incorrectly detected. In such cases, in-plane alignment (using, e.g., simple similarity transform) is a more robust transformation requiring fewer detected anchor points.

4.1 Extending the Training Set Pose Distribution

A key challenge in learning multiple pose-aware models is the limited data available for training effective CNNs for

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

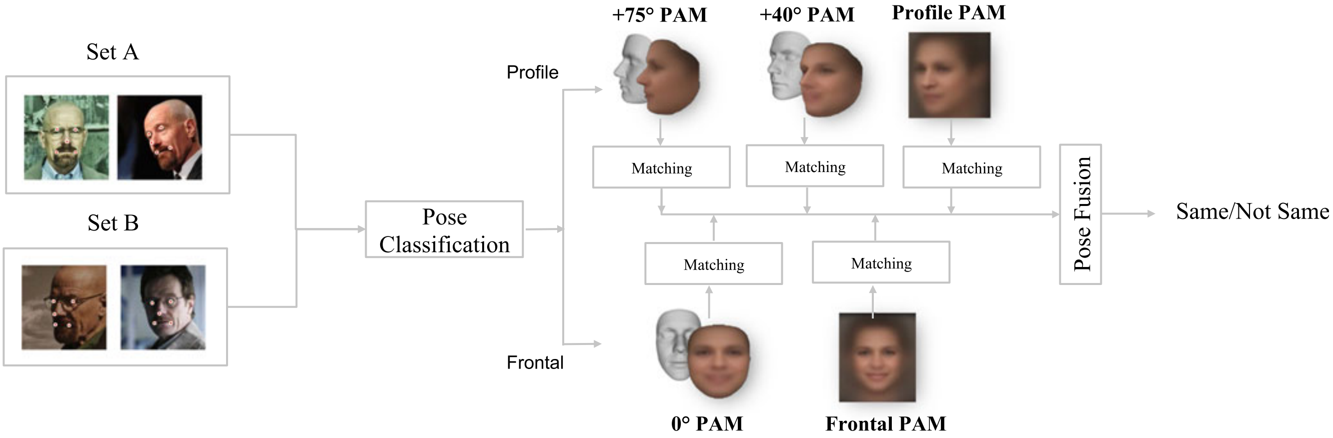


Fig. 3. Given two sets of face images and videos to match, pose classification is used to select a corresponding Pose-Aware CNN Model for processing. Given multiple images, each model extracts features and matches them at the set level, independently. Finally, the contribution of each model is pooled into a single, final score. Note how our approach uses 3D rendering to adjust the pose to frontal (0 degree), half-profile (40 degree) and full-profile views (75 degree).

each pose, particularly when developing a system intended to handle faces in extreme views—such as those in the IJB-A benchmark. In our work, we used the CASIA WebFace collection as a training set. Although it is far larger than LFW, it is still strongly biased towards frontal facing poses and very limited in the number of near-profile images it contains (see Fig. 1).

Contrary to methods that rely on multi-task learning [17] or [18] and model identity and viewpoint with a single network, we treat each type of alignment and data *independently*. That is, we learn a specific model for each type of alignment (in-plane and out-of-plane) and each mode of the pose distribution. Besides allowing for better modeling of appearances in different views, a key advantage of this approach is that it permits network *co-training*, thereby improving transferability of learned features (more on co-training in Section 5.2). We found this especially important for generalization to other datasets. This approach assumes, however, that sufficient examples are available for training each model, and again, this is not the case in CASIA.

To address this, we automatically *stretch* the distribution of facial poses in the CASIA training set in order to produce examples encompassing the range from frontal to full profile. To this end, we begin by partitioning the range of CASIA yaw angles into subsets. Faces in each subset will then be artificially mapped to extreme poses.

Specifically, given 2D landmarks, converted to homogeneous coordinates $\mathbf{l} \in \mathbb{R}^{3\times|\mathbf{J}|}$ and $\mathbf{J}$ a set of landmark indices, we compute the 3D pose in a manner similar to [12] by considering the positions of the same $\mathbf{J}$ facial landmarks on the surface of a generic 3D face shape $\mathbf{M}$, expressing them in homogeneous coordinates as: $\mathbf{L} \doteq \mathbf{M}(\mathbf{J}) \in \mathbb{R}^{4\times|\mathbf{J}|}$. The fixed 3D generic shape $\mathbf{M}$ was taken from a set of BlendShapes [44], selecting the one with neutral expression. We can then estimate a perspective camera model mapping the 3D face $\mathbf{M}$ onto the image such that

$$\mathbf{l} = \mathbf{p}\ \mathbf{L}, \tag{1}$$

where

$$\mathbf{p} = \mathbf{K}\ [\mathbf{R}\ \mathbf{t}]. \tag{2}$$

Although recent deep methods have shown improved accuracy in 3D face alignment [45], [46], we used the constrained local neural fields (CLNF) [47] to detect 68 facial landmarks. These were then used in the PnP method [48] to estimate the extrinsic camera parameters, assuming that the principal point is in the image center and refining the focal length by minimizing landmark re-projection errors.

From $\mathbf{p}$, we extract the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ representing the 3D rotation parameters of the model $\mathbf{M}$ with respect to the image. By decomposing $\mathbf{R}$ we obtain the yaw values ψ of the face across the entire dataset in the coordinate system of the generic 3D model. We accumulate all the $\{\psi_i\}_{i=1}^{N}$ values in order to estimate the training pose distribution $p(\psi|\mathbf{I})$. These distributions are visualized in Fig. 1.

Instead of treating all the images as belonging to the same single frontal model, irrespective of their underlying yaw distribution, we partition the faces into separate yaw distributions. In order to get the main T modes of the training yaw distribution, $p(\psi|\mathbf{I})$, we run K-means on $\{\psi_i\}_{i=1}^{N}$

$$\Psi = \{\mu_{\psi_t}\}_{t=1}^{T}, \tag{3}$$

along with the hard-assignment of each image to a certain mode. We can interpret these hard assignments of images to clusters as a function that maps each image to a specific mode according to

$$\delta(\mathbf{I}) = t \quad \text{where} \quad t \in [1 \ldots T]. \tag{4}$$

In practice, we consider $T = 5$ modes: one for roughly the frontal view and two additional views for positive and negative yaw angles. Each μ_{ψ_t} represents a mode in the yaw distribution and $\delta(\cdot)$ gives the assignments to each tth mode for each image. On the CASIA set, we found that the modes balanced each other and centered roughly on $\{-33.96^\circ, -15.87^\circ, 0.25^\circ, 16.50^\circ, 33.13^\circ\}$. As can be expected from Fig. 1, most of the images aggregate to the frontal and two near-frontal modes.

Generic versus Subject-Specific Shape. We motivate the use of a simple 3D generic shape $\mathbf{M}$ mainly following the findings in [9], that any generic face shape is as good as another, contingent on keeping it fixed across all the tests. In order to use subject-specific face shapes that are discriminative across

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

subjects, especially in profile views (e.g., nose contour), the system should be able to reconstruct the true 3D shape in order to correctly render profile views. Estimating correct and stable 3D shapes for IJB-A images, however, is exceedingly hard, especially at test-time. Attempting 3D reconstruction could therefore actually reduce recognition accuracy whenever 3D reconstructions fail. For this reason we favored the more robust solution of using a generic 3D model.

Discussion on More Fine-Grained PAMs. Obviously there is a trade-off in selecting T: If T equals one, we revert to training a single model irrespective of the pose; although we could increase T to presumably allow for better face recognition by having more dense PAMs. This has two main problems: first, by increasing the number of pose modes, each PAM would be trained on fewer images and so more likely to overfit; second, more PAMs require more storage and slower test times. We therefore chose $T = 5$ as an effective compromise.

4.2 In-Plane Alignment Models

Although Eq. (3) used five modes, we can use facial symmetry to simplify this model. Specifically, we flip one direction of the yaw distribution according to $\mu_{\psi_t} \rightarrow |\mu_{\psi_t}|$. This is done by flipping the images in these modes along their vertical axes and modifying the assignments in Eq. (4) accordingly. Thus, we can consider only one side of the distribution $p(\psi|\mathbf{I})$—for example, the left side—reducing the number of models we need to train and simplifying our system. The result is a set of three modes $\Psi' = \{\mu_{\text{frontal}}, \mu_{\text{near-frontal}}, \mu_{\text{profile}}\}$, corresponding to yaw values centered in $\{0.25°, 16.50°, 33.13°\}$.

We use these modes for 2D in-plane alignment as follows. We express $p(\psi|\mathbf{I})$ as a bi-modal distribution by partitioning the dataset into two classes: near-frontal faces with small pose variability and profile faces with high pose variability. In particular, we partition the images using the image assignments in Eq. (4), classifying an image as profile if it belongs to the third mode μ_{profile}, and frontal otherwise. In this way, the CASIA dataset is partitioned into two separate subsets which can be used to train two CNN models with in-plane aligned images. We denote these as $\text{PAM}_{\text{in-f}}$ and $\text{PAM}_{\text{in-p}}$. Moreover, since frontal images are separated from profiles, we can use different ideal target coordinates for each set. Frontal images are aligned using the nine most reliable landmarks for a frontal-facing face, while profile images, where half the face is less visible, are aligned using the tip of the nose and the center of the two eyes. For both alignments we use a non-reflective similarity transformation $\mathbf{S}(s, \theta, t_x, t_y)$. Alignment parameters for each set are recovered by standard techniques, solving a linear system of equations using detected and reference landmarks that are specific for each alignment.

4.3 Out-of-Plane Pose Models

The process described in Section 4.2 applies limited in-plane transformations to the images. It therefore cannot account for drastic pose variations and will not introduce new examples to the training set with higher yaw angles than those already present in the original CASIA set. For both reasons, we do not rely exclusively on in-plane aligned images and instead also learn models compensating for out-of-plane rotation; these models minimize pose variability and address the shortage of profile faces in the training data.

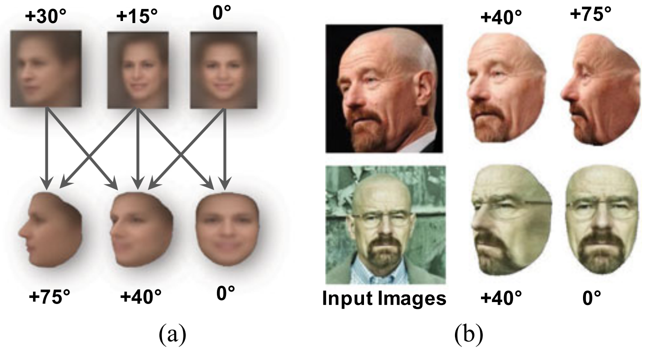


Fig. 4. (a) The directed graph used to map each mode of the CASIA yaw distribution to a desired target mode. (b) The process helps to properly render a face image: if the face is frontal we render it to both frontal and half-profile views; If it is far from frontal, we avoid frontalizing it and render only to profile views.

We again express $p(\psi|\mathbf{I})$ as a multi-modal distribution with three prominent modes and again exploit Eq. (3) and face symmetry, as we did in Section 4.2—this time adjusting the pose by aligning it to a target yaw value. This process is similar to the frontalization of [9], only here we use more than one target yaw angle.

Specifically, we would like our models to represent frontal, half-profile, and full-profile faces, but as we show in Section 4.1, all of our three modes concentrate on near-frontal views (the most extreme centered on 33.13 degree). We undo this training bias in the training set yaw distribution by transforming the training images to a new target distribution. For this, we use the 3D generic model $\mathbf{M}$ and the estimated pose $\mathbf{p}$ of Eq. (2) to render each training face to a new mode in the desired distribution. The specific rendering technique used is derived from [9].

In practice, we set the target distribution a priori to have frontal (0 degree), half-profile (40 degree), and full-profile modes (75 degree). Training images are mapped (rendered) to one *or more* of these modes in a manner that provides a trade-off between producing good rendered faces and generating sufficient side views. Specifically, an image is rendered to new views according to its estimated facial pose and the directed graph in Fig. 4; each edge in the graph represents the rendering process from a certain input pose to target poses. Thus, for example, an image in the middle mode (belongs to the cluster with mean yaw of 16.50 degree) is rendered to profile, half-profile, and frontal views, whereas an image belonging to the frontal mode is only rendered to frontal and half-profile views.

This process produces synthetic images which provide ample examples for each of the modes in the target distribution. The factor of increase in the training data is a function of the number of images assigned to each source mode and the number of edges entering the target node. Following this process, we can train three additional networks, one for each mode of the new desired pose distribution—namely, $\text{PAM}_{\text{out-0}}$, $\text{PAM}_{\text{out-40}}$, and $\text{PAM}_{\text{out-75}}$.

5 Learning Pose-Aware Models

An input image is aligned up to five times, depending on its estimated pose—either using 2D alignment to frontal or

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

profile views (Section 4.2) and up to three times more using out-of-plane alignment (Section 4.3). For each of these five, we train a separate CNN model, giving us an ensemble of five CNNs. Each CNN is *aware* of its viewpoint by learning its own pose-specific features, and hence the name for our approach.

5.1 Fine-Tuning PAMs

We train a separate, pose-aware CNN for each of the five modes described in Section 4. Since training a CNN from scratch for this purpose could require millions of annotated images, we train our Pose-Aware CNNs by fine-tuning state-of-the-art CNN models trained on ImageNet. In our implementation, we experimented with a CNN with eight layers (AlexNet) [49] and one with 19 (VGGNet) [50]. We tested different network types in order to demonstrate that our performance gain from fusing multiple pose-aware models is agnostic to the particular CNN architecture used.

All of these CNN models end with the fully connected layers *fc7* and *fc8*. The output of *fc8* is fed to a C-way SoftMax which gives a distribution over the subject labels $\mathcal{C}$. Denoting by $y_i(\mathbf{I})$ the ith output of the network on a given image $\mathbf{I}$, the probability assigned to the ith class is the output of the SoftMax function $p_i(\mathbf{I}) = \frac{e^{y_i(\mathbf{I})}}{\sum_{l=1}^{C} e^{y_l(\mathbf{I})}}$. Note that the *fc8* output in each PAM potentially includes different numbers of subjects, in accordance with the datasets produced in Section 4.

Training Parameters. As mentioned, we start from prelearned weights originally learned on the ImageNet data set. We then initialize the values of *fc8* to allow it to be trained from scratch. The initial values for *fc8* are drawn from a Gaussian distribution with zero mean and standard deviation 0.01. The networks are then fine-tuned on our face data sets through stochastic gradient descent (SGD) and standard back-propagation [51] with a mini-batch size of 60. We use an initial learning rate of 1e-3. This rate is applied to the entire network, except for the new *fc8*, which has a learning rate an order of magnitude greater. Biases are learned two times faster than weights. Finally, the learning rate is decreased by an order of magnitude when validation set error saturates and reaches a plateau. The final learning rate is 1e-4. The training is regularized by imposing ℓ_2 norm on the weights with a weight decay parameter 5e-4, and by applying dropout to the fully connected layer with probability 0.5; we used a momentum of 0.9. All of our input images are preprocessed to subtract the average image computed on the training set; no data-jittering is applied during training and testing.

For each mode, we apply the exact same learning regime: we retrain using only the data associated with a specific mode, including all training subjects with at least five images; we randomly select 80 percent of the images for training, and the remaining 20 percent are used for validation.

5.2 PAM Co-Training with Weight-Sharing Initializatio

Even after producing the training sets for each of our modes, our training set does not contain sufficient samples required to train our models from scratch. Therefore, following the analysis provided in [52], we fine-tune our networks from weights originally learned using the ImageNet training set.

To improve transferability of the learned models and to make the models less prone to overfitting, we propose to further *co-train* our models. This approach is designed to find a better optimization point in the loss minimization, and it acts as a sort of regularization for all the models. The motivation for co-training is in using convolutional filters and weights that were tuned on faces and not on generic images such those in ImageNet. Unlike the baseline model trained on plain CASIA, e.g., [7], we have multiple CNN models, each one designed for a specific viewpoint of the face and a specific alignment. We propose to exploit this plurality of models by optimizing different loss functions from different viewpoints using different sets.

We overview the co-training process in Algorithm 1. The networks intended for in-plane aligned modes were initially fine-tuned from ImageNet pre-learned weights ($\text{PAM}_{\text{in-f}}$). We empirically found that this was not enough, and that $\text{PAM}_{\text{in-p}}$ fine-tuned from ImageNet performed poorly due to the few profile images available for training. We thus co-train it by resuming the optimization from the weights obtained for $\text{PAM}_{\text{in-f}}$, which was trained on the substantially larger set of frontal faces. For the out-of-plane modes (Section 4.3), we again begin fine-tuning $\text{PAM}_{\text{out}-0}$ from ImageNet weights. Then we fine-tune $\text{PAM}_{\text{out}-40}$, initialized using the weights in $\text{PAM}_{\text{out}-0}$. We continue iterating by alternating the co-training until the validation accuracy saturates in both models.

Algorithm 1. Co-Training PAMs

```
Input: $\{\mathbf{W},\mathbf{b}\}_{\mathrm{I}}$, CNN weights trained on ImageNet
       $\mathcal{C}_a$ denotes the labels of (face) dataset A
       $\mathcal{C}_b$ denotes the labels of (face) dataset B
Output: $\{\mathbf{W},\mathbf{b}\}_a$, new optimized weights for $\mathcal{C}_A$
        $\{\mathbf{W},\mathbf{b}\}_b$, new optimized weights for $\mathcal{C}_B$
Initialize weights $\{\mathbf{W},\mathbf{b}\}_b$ using $\{\mathbf{W},\mathbf{b}\}_{\mathrm{I}}$
step = 0
while validation acc. on $\mathcal{C}_A$, $\mathcal{C}_B$ do not saturate do
    $\{\mathbf{W},\mathbf{b}\}_a$ = Fine-tune $\{\mathbf{W},\mathbf{b}\}_b$ on $\mathcal{C}_A$
    $\{\mathbf{W},\mathbf{b}\}_b$ = Fine-tune $\{\mathbf{W},\mathbf{b}\}_a$ on $\mathcal{C}_B$
    step = step + 1
end
```

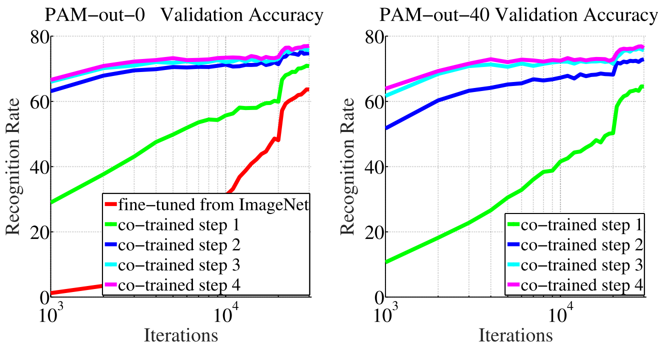
Co-training is a form of regularization, but it is different from weight sharing in the sense that in this case network weights from an already optimized network A are used as initialization for network B to avoid overfitting. Weight sharing, by comparison, assumes that the two networks A and B share the same weights in the same optimization process.

Fig. 5 shows the steep increase in the validation accuracy provided by co-training using the AlexNet architecture. After these models are trained, we proceed to co-train $\text{PAM}_{\text{out}-75}$ from $\text{PAM}_{\text{out}-40}$. Similar (though faster) gains were observed in the deeper architectures used to report performances in Section 7. We performed the same process for VGGNet, but this deeper model requires fewer steps of co-training to saturate the validation accuracy.

6 Pose-Aware Face Recognition

The pose-aware CNN models learned in Section 5 can be considered discriminative classifiers which are explicitly trained for a certain mode of the pose distribution. We next

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.



(a) Co-training $\text{PAM}_{\text{out-0}}$ (b) Co-training $\text{PAM}_{\text{out-40}}$

Fig. 5. Steep increase in validation accuracy as a function of the iterations in the fine-tuning process. Each curve represents a step in the co-training process. Iterations are shown in log scale.

describe how these individual models, $\{\text{PAM}_{\text{in}-\text{f}}, \text{PAM}_{\text{in}-\text{p}}, \text{PAM}_{\text{out}-0}, \text{PAM}_{\text{out}-40}, \text{PAM}_{\text{out}-75}\}$, are jointly used in our recognition pipeline.

6.1 Alignment at Test-Time

Given a test image, we choose an alignment based on the estimated pose of the face in the image (see Fig. 6). We further exploit facial symmetry in the same way that we did in the training set—aligning or rendering a face to one side and then flipping it back, if necessary, to align it with the corresponding PAM.

In-Plane Alignment. At test time, for an image $\mathbf{I}$ we first detect facial landmarks which are then used to estimate facial pose. The pose is then classified using the μ_{profile} corresponding to the mode found for the profile faces in Section 4.2. This process is the same as the one used for the in-plane aligned training images. Formally

$$\text{pose-class}(\mathbf{I}) = \begin{cases} \text{frontal}, & \text{if } |\psi| \le \mu_{\text{profile}} \\ \text{profile}, & \text{otherwise.} \end{cases} \tag{5}$$

We use the threshold value μ_{profile} corresponding to the mode found for the profile faces in Section 4.2. This decision is illustrated in Fig. 6 (left). In the general case, if a set contains multiple images (set-to-set matching), we proceed to align each face image either to $\text{PAM}_{\text{in}-\text{f}}$ or $\text{PAM}_{\text{in}-\text{p}}$, accordingly to the classified pose.

Out-of-Plane Alignment. The same image $\mathbf{I}$ also undergoes 3D alignment by rendering the face to the multiple modes $\{0^\circ, 40^\circ, 75^\circ\}$ defined in Section 4.3. We always render each image to a half-profile view (40 degree). Then, if the image is classified as near-frontal, we additionally frontalize it to 0 degree. If not, we render the image to the profile view (75 degree). These rendered images are processed using the PAM: $\text{PAM}_{\text{out}-0}$, $\text{PAM}_{\text{out}-40}$, $\text{PAM}_{\text{out}-75}$. This process is illustrated in Fig. 6 (right).

Mitigating Landmark Failures. The alignment process described in this section depends strongly on the landmark detector and its accuracy. The images in IJB-A or in other in-the-wild collections can cause the detector to fail, possibly often if the images are particularly challenging. This affects the quality of the alignment which, in turn, can degrade the performance of the recognition system. Whenever the detector fails to find landmarks on *all* the images of a set we then use unaligned face images instead. Since no pose information is available in these cases, these images are then simply processed using $\text{PAM}_{\text{in}-\text{f}}$.

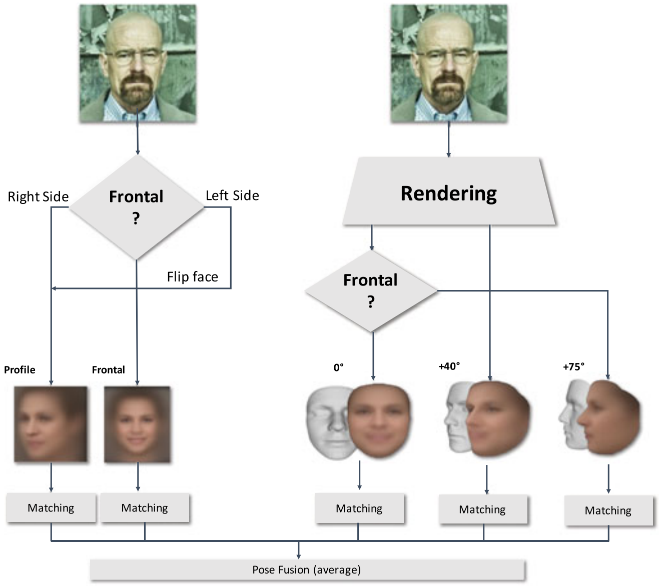


Fig. 6. Alignment is guided by the pose. (Left) Images are in-plane aligned separately for frontal or profile views. (Right) The approach renders faces only to near poses. We render an image to frontal and half-profile if the face is frontal; otherwise to profile and half-profile if the face is near-profile.

6.2 Matching Methodology

Face Representation and Matching. Each PAM processes the images aligned as described in Section 6.1 and produces a feature representation. For this representation we take the *fc7* layer response after non-linear ReLu activation. This feature representation is then transformed using principal components analysis (PCA) learned on the training images of the target data set. Following PCA, a non-linear transformation through element-wise signed square rooting (SSR) is applied. This transformation is commonly used with Fisher Vector encoding [53], and we found it to be beneficial here as well. The result is our final face representation, denoted by $\mathbf{x}$.

The matching score between two such representations $s(\mathbf{x}_1, \mathbf{x}_2)$ is computed by taking their correlation

$$s(\mathbf{x}_1, \mathbf{x}_2) = \frac{(\mathbf{x}_1 - \bar{\mathbf{x}}_1)(\mathbf{x}_2 - \bar{\mathbf{x}}_2)^T}{||\mathbf{x}_1 - \bar{\mathbf{x}}_1||\ ||\mathbf{x}_2 - \bar{\mathbf{x}}_2||}. \tag{6}$$

When matching entire sets (possibly containing both still images and video frames), we experimented with two very different matching approaches:

1) Pair-wise similarity comparison using Eq. (6) followed by score fusion using the SoftMax of Eq. (7).
2) Element-wise average of all the features in each set, pooled according to their source (still image and video frames pooled separately, see Eq. (8) below). This is followed by similarity of the pooled feature vectors.

Pair-Wise Similarity Comparison and SoftMax. Each PAM performs pair-wise comparison of feature pairs from the two sets using Eq. (6). These pair-wise scores are then fused by taking their weighted average, where each weight is a

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

function of the score using an exponential function. Formally

$$s_\beta(\mathcal{T}_1, \mathcal{T}_2) = \frac{\sum_{i\in\mathcal{T}_1, j\in\mathcal{T}_2} w_{ij}\; s(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{i\mathcal{T}_1, j\in\mathcal{T}_2} w_{ij}}, \quad w_{ij} \doteq e^{\beta\, s(\mathbf{x}_i, \mathbf{x}_j)}, \tag{7}$$

where $\mathcal{T}_1$ denotes a set as a collection of features $\mathcal{T}_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

A final set-to-set similarity score is defined by the average SoftMax responses over multiple values of $\beta = [0...20]$. It is interesting to note here that the SoftMax hyper-parameter β controls the trade-off between averaging the scores or taking its maximum. To clarify, we emphasize that this proposed SoftMax operator is *very different* from the one conventionally used when training CNNs for closed-set classification; here, it allows for open-set face recognition.

Frame and Image Pooling. A number of previous efforts proposed different ways of pooling feature descriptors [54], [55]. In this section we propose a much simpler yet highly effective alternative approach. Specifically, if a set contains multiple images, these are first pooled according to their type. Features extracted from still images, denoted by $\mathcal{T}_{\text{img}}$, and video frames, denoted by $\mathcal{T}_{\text{frm}}$, are pooled *separately* using element-wise averages as follows:

$$\text{pool}(\mathcal{T}) = \frac{1}{2|\mathcal{T}_{\text{img}}|} \sum_{i\in\mathcal{T}_{\text{img}}} \mathbf{x}_i + \frac{1}{2|\mathcal{T}_{\text{frm}}|} \sum_{j\in\mathcal{T}_{\text{frm}}} \mathbf{x}_j. \tag{8}$$

Once a set is *flattened* using the frame and image pooling of Eq. (8), then two sets are compared by taking the correlations between pooled features and obtaining a final set score: $s\big(\text{pool}(\mathcal{T}_1), \text{pool}(\mathcal{T}_2)\big)$.

Fusion Across PAMs. Given a set (or image) pair to be matched, the process described above will yield up to five similarity scores, each produced by matching the features obtained from a specific PAM. In our implementation, these values are pooled together by simple, unweighted averaging. We found that this provides a good baseline for all our experiments.

7 Experimental Results

The key contribution of this paper is the notion of training and applying pose-aware models and matching. The system designed around this concept, however, includes many other design details and components. We begin our experiments with a diligent analysis of many of these components and the choices made in designing the face recognition system. These are performed on the IJB-A set. Additionally, comparative experimental results on two public benchmarks for face recognition in challenging unconstrained settings are provided in Sections 7.1 and 7.2.

We emphasize that our system does not perform any supervised training using the target data sets; supervised training is only performed by training our PAMs on the outside data of the CASIA WebFace. That is, contrary to [14], [31], we do not re-train our models or fine-tune them using the training splits provided on each benchmark, or learn any supervised embedding on the target training data. By avoiding this, we require a smaller training effort when applying our system to a new data set. Moreover, this underscores the transferability of our method beyond the restricted domain used to train it. Finally, in order to have a fair comparison, we removed all subjects in the CASIA set included also in IJB-A by string matching of subjects' names. In total, 26 overlapping subjects were thus removed from the training set.

7.1 IARPA Janus Benchmark A (IJB-A)

IJB-A is a new face recognition challenge proposed by IARPA and made available by NIST[2] in order to push the frontiers of face recognition in the wild. It followed the saturation of LFW results [6] (the existing, standard *de facto* benchmark). IJB-A provides 500 subjects under extreme viewing conditions, reflecting variations in pose, expression, illumination, and more. The IJB-A evaluation protocol consists of face verification (1:1) and face identification (1:N). As previously mentioned, each IJB-A subject is represented by a set containing images and/or video frames.

We study the effect on performance of a baseline trained on 2D in-plane aligned CASIA images and its key components. In particular, we examine the effects of selecting different facial landmark detection methods, face representations, and other components of our method. We also provide comparisons between this baseline, the multi-pose representation, and a network trained on pose-augmented data [15], [16]. Finally, we compare PAMs with co-training and CNNs trained with standard but more aggressive regularization methods. In these experiments the core matching system used is based on the pair-wise similarity and SoftMax set score fusion of Eq. (7). The dataset used is IJB-A.

Effect of Landmark Detection Method. Different landmark detection methods report different rates of accuracy on different benchmarks. Because landmark detection accuracy can directly impact the performance of a face recognition pipeline, it is therefore important to evaluate different landmark detection methods to assess the magnitude of impact and the robustness of the rest of the system to this choice.

Our baseline for face recognition uses the single model AlexNet for face representation trained on the plain CASIA set. We test three state-of-the-art facial landmark detection methods to see how they affect performance. These are: 1) DLIB [56] which provides 68 facial landmarks, 2) TDCNN [57] detects only five sparse landmarks, and 3) CLNF [47], [58] for 68 point detection.

We further experimented with a variant of CLNF that uses the seed landmarks provided by IJB-A [8] as part of its face recognition test protocol. These points were used to estimate an initial face shape before applying CLNF. Note that although, presumably, the same initialization technique may be applied to the other detectors, their off-the-shelf implementations do not offer an interface which allows this.

Detailed results are listed in Table 1. All of these landmark detectors share the same set of face bounding boxes. Evidently, the choice of a landmark detector does not greatly impact recognition performance. This implies that CNN features attain a certain level of spatial invariance. The use of seed landmarks to initialize the search for facial landmarks does seem to provide a substantial performance boost, suggesting that although there is little difference

[2] Available by request http://www.nist.gov/itl/iad/ig/ijba_request.cfm

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Landmark Influence on Face Recognition in IJB-A

Metric	DLIB	TDCNN	CLNF	CLNF$_{\text{seed}}$
TAR@FAR=0.01	.632 ± .018	.630 ± .019	.624 ± .022	.640 ± .027
TAR@FAR=0.10	.858 ± .014	.861 ± .012	.852 ± .011	.865 ± .011
FAR@TAR=0.85	.093 ± .016	.088 ± .016	.096 ± .017	.084 ± .012
FAR@TAR=0.95	.461 ± .064	.435 ± .065	.522 ± .091	.391 ± .048
Rank-1	.669 ± .023	.671 ± .021	.663 ± .020	.679 ± .020
Rank-5	.831 ± .012	.835 ± .013	.824 ± .013	.837 ± .014
Rank-10	.879 ± .009	.884 ± .007	.871 ± .010	.889 ± .010

TABLE 3
IJB-A Recognition Results with Varying Representations

Metric	AlexNet	VGGNet-16	VGGNet-19
TAR@FAR=0.01	.640 ± .027	.765 ± .020	.780 ± .017
TAR@FAR=0.10	.865 ± .011	.906 ± .005	.913 ± .007
FAR@TAR=0.85	.084 ± .012	.035 ± .006	.031 ± .005
FAR@TAR=0.95	.391 ± .048	.264 ± .041	.278 ± .047
Rank-1	.679 ± .020	.785 ± .012	.791 ± .011
Rank-5	.837 ± .014	.883 ± .011	.895 ± .008
Rank-10	.889 ± .010	.915 ± .009	.918 ± .009

between existing detection methods, there is still room for improving landmark detection accuracy.

Face Representation. As in many recognition problems, feature representation plays a key role in face recognition. Given a trained face recognition CNN, we may treat each layer's responses as potential facial features. It is instructive to investigate which combination of layer responses provides the best face recognition performance. In this experiment, we exhaustively try all possible layer combinations for the last five layers of each CNN architecture. We report face recognition performances on selected combinations with reasonable results in Table 2, where ✓ indicates which layers were used to extract features. Features are normalized using their mean and standard deviations in order to maintain similar dynamic ranges. Features from multiple layers are then concatenated together into a single representation. The experiment is performed again using the baseline based on AlexNet and VGG.

We notice that when we only use the *fc7* layer—the layer prior to classification—we obtain the best possible recognition rates compared to any other VGGNet and AlexNet feature combination. In this case, TAR@FAR=.01 exhibits a big improvement while Rank-10 is on a par with other feature combinations. Consequent to these findings, all our other results using both network architectures will use only *fc7* as a representation.

We further compare four face recognition pipelines based on different representations. Specifically, we test AlexNet, VGGNet with 16 layers, and VGGNet with 19 layers. These results are reported in Table 3. Unsurprisingly, the deeper architecture outperforms the shallow one. When stacking more convolutional layers together, face recognition performance improves as well.

Baseline versus Multi-Pose Representations Versus Pose-Augmented We explore the influence of using different component combinations of our proposed PAM representations—namely, our baseline, a single PAM, a single CNN trained on the pose-augmented CASIA set (Pose-Aug.) and the full PAMs. The network trained on the pose-augmented data is simply a network trained by using all the 2D in-plane aligned images in CASIA and augmenting this set by using 3D renderings of those images, as recently done in [15], [16]. Regarding each single PAM model, we report performance when using only one of the representations provided by the single network: PAM$_{\text{in-f}}$, PAM$_{\text{in-p}}$, PAM$_{\text{out-0}}$, PAM$_{\text{out-40}}$, and PAM$_{\text{out-75}}$.

As shown in Table 4, face recognition performance significantly improves as the number of pose representations increases, regardless of whether we use AlexNet or VGGNet. Interestingly, we can also see how much of this improvement comes from the fact that both the CNN trained on pose-augmented data and PAMs overcome the the pose-deficient, CASIA training set that offers only a limited amount of profile samples (see Fig. 1). Although much of the performance comes from handling this, we can see that applying pose-aware face recognition (Section 6) provides an additional boost to the performance of PAMs—especially at TAR at very low FAR=0.001 and at higher ranks—compared to augmenting for pose-variations at training time.

Component Analysis. Table 5 reports the improvements offered by various PAM components, tested here using two

TABLE 2
IJB-A Recognition Results with Varying Deep Feature Combinations

Layer Name	CNN Layer Response Combinations									
prob	✓	✓	✓	✓					✓	
fc8	✓				✓	✓	✓	✓	✓	
fc7		✓			✓		✓		✓	✓
fc6			✓					✓		
pool5				✓		✓	✓	✓	✓	
Metric	**Single Model (AlexNet)**									
TAR@FAR=.01	.573 ± .024	.585 ± .020	.590 ± .023	.523 ± .030	.611 ± .022	.597 ± .033	.608 ± .031	.604 ± .031	.604 ± .031	**.640 ± .027**
FAR@TAR=0.85	.114 ± .015	.103 ± .013	.102 ± .011	.146 ± .010	.099 ± .014	.108 ± .014	.103 ± .012	.103 ± .012	.103 ± .012	**.084 ± .012**
Rank-10	.893 ± .008	.887 ± .009	.891 ± .011	.869 ± .012	.893 ± .007	**.899 ± .008**	**.899 ± .009**	**.899 ± .008**	.897 ± .009	.889 ± .010
Metric	**Single Model (VGGNet)**									
TAR@FAR=.01	.729 ± .022	.735 ± .018	.730 ± .021	.669 ± .024	.755 ± .019	.720 ± .030	.731 ± .029	.726 ± .027	.728 ± .029	**.780 ± .017**
FAR@TAR=.85	.047 ± .007	.045 ± .005	.051 ± .007	.072 ± .009	.038 ± .006	.052 ± .008	.048 ± .008	.049 ± .007	.048 ± .007	**.031 ± .005**
Rank-10	.918 ± .008	.918 ± .008	.919 ± .007	.905 ± .009	.916 ± .007	**.921 ± .008**	**.921 ± .008**	.920 ± .008	**.921 ± .007**	.918 ± .009

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
IJB-A Recognition Results with Varying PAM Components

Metric	Baseline	PAM$_{\text{in-f}}$	PAM$_{\text{in-p}}$	PAM$_{\text{out-0}}$	PAM$_{\text{out-40}}$	PAM$_{\text{out-75}}$	Pose-Aug.	PAMs
				AlexNet				
TAR@FAR=0.001	.407 ± .043	.360 ± .042	.161 ± .023	.409 ± .035	.403 ± .063	.459 ± .055	.517 ± .040	**.548 ± .066**
TAR@FAR=0.01	.640 ± .027	.530 ± .031	.286 ± .018	.570 ± .026	.638 ± .042	.698 ± .040	.751 ± .031	**.756 ± .029**
TAR@FAR=0.10	.865 ± .011	.699 ± .014	.423 ± .011	.711 ± .014	.868 ± .015	.892 ± .010	**.911 ± .008**	.910 ± .008
FAR@TAR=0.85	.084 ± .012	.670 ± .024	.711 ± .008	.665 ± .026	.080 ± .016	.054 ± .013	.038 ± .009	**.034 ± .009**
FAR@TAR=0.95	.391 ± .048	.774 ± .013	.815 ± .008	.770 ± .014	.343 ± .040	.278 ± .044	**.256 ± .035**	.289 ± .063
Rank-1	.679 ± .020	.587 ± .021	.273 ± .017	.616 ± .026	.763 ± .016	.775 ± .012	**.776 ± .020**	.771 ± .011
Rank-5	.837 ± .014	.703 ± .020	.388 ± .015	.709 ± .022	.897 ± .008	.902 ± .009	.895 ± .010	**.897 ± .009**
Rank-10	.889 ± .010	.739 ± .017	.448 ± .014	.744 ± .020	.931 ± .008	.935 ± .009	.925 ± .007	**.928 ± .007**
				VGGNet-19				
TAR@FAR=0.001	.588 ± .036	.567 ± .029	.207 ± .023	.552 ± .033	.609 ± .043	.553 ± .046	.558 ± .068	**.652 ± .037**
TAR@FAR=0.01	.780 ± .017	.673 ± .016	.323 ± .022	.733 ± .019	.800 ± .018	.799 ± .015	.820 ± .014	**.826 ± .018**
TAR@FAR=0.10	.913 ± .007	.733 ± .014	.438 ± .011	.898 ± .008	.925 ± .006	.928 ± .004	**.942 ± .005**	.930 ± .007
FAR@TAR=0.85	.031 ± .005	.567 ± .019	.677 ± .012	.046 ± .008	.023 ± .005	.022 ± .004	**.016 ± .003**	**.016 ± .005**
FAR@TAR=0.95	.278 ± .047	.667 ± .014	.781 ± .012	.292 ± .037	.217 ± .040	.200 ± .032	**.135 ± .023**	.197 ± .042
Rank-1	.791 ± .011	.677 ± .019	.303 ± .020	.771 ± .016	.818 ± .013	.825 ± .010	.827 ± .011	**.840 ± .012**
Rank-5	.895 ± .008	.732 ± .020	.411 ± .015	.887 ± .009	.917 ± .008	**.925 ± .007**	.923 ± .007	**.925 ± .008**
Rank-10	.918 ± .009	.754 ± .018	.463 ± .018	.919 ± .009	.939 ± .009	.944 ± .006	.945 ± .007	**.946 ± .007**

CNN architectures: AlexNet and VGGNet. We show the performance on IJB-A splits, reporting the TAR at FAR=0.01 for the verification protocol and the recognition rate at Rank-10 for the identification protocol. Evident in the table is that significant improvement in performance is achieved by co-training. This improvement is much greater with AlexNet than it is with VGGNet, but even in the latter, co-training improves overall performance. Further improvement is obtained by applying PCA and signed square rooting (SSR). We also tested the regularization effect of the co-training algorithm compared to a much simpler method of regularizing the baseline model.

All the networks trained in this work (both baseline and PAMs) were regularized with weight decay and dropout (see Section 5.1). We show the effect of increasing the regularization of those baselines even more in order to compare with PAMs and co-training. We increased the weight decay from 5e-4 to 1e-3 in order to better bound weight changes with ℓ_2 norm; additionally we increase the dropout ratio from 0.5 to 0.75, dropping more connections in the fully connected layers. This experiment is provided in the ablation study of Table 5, *+Reg.A*. Moreover, we show an additional result in which the networks are even more regularized by increasing the weight decay to 5e-3 and dropout ratio to 0.75. Results are shown in Table 5, *+Reg.B*. It is interesting to see that, although AlexNet benefits from this more aggressive regularization procedure, VGG does not. More importantly, our proposed method results in better generalizing networks than standard regularization methods.

TABLE 5
Improvement for Each Component on the IJB-A Dataset

Networks	AlexNet		VGGNet-19	
Metrics	TAR	Rank-10	TAR	Rank-10
Basel. +*PCA*,+*SSR*	.640 ± .027	.889 ± .010	.780 ± .017	.918 ± .009
Basel. +*PCA*,+*SSR*,+*Reg.A*	.736 ± .022	.902 ± .008	.766 ± .016	.911 ± .011
Basel. +*PCA*,+*SSR*,+*Reg.B*	.727 ± .028	.900 ± .009	.743 ± .020	.893 ± .007
PAMs	.494 ± .059	.881 ± .010	.660 ± .028	.923 ± .011
PAMs +*co-train*	.612 ± .041	.903 ± .006	.701 ± .035	.936 ± .006
PAMs +*co-train*,+*PCA*	.666 ± .054	.912 ± .008	.768 ± .025	.938 ± .006
PAMs +*co-train*,+*PCA*,+*SSR*	**.756 ± .029**	**.928 ± .007**	**.826 ± .021**	**.946 ± .007**

TAR is reported at FAR=0.01 for verification. Recognition rate at Rank-10 is reported for identification.

Robustness to Pose Variations. In Fig. 7 we show the improvement offered by the proposed PAMs compared with learning a single CNN trained on the CASIA WebFace images following only standard in-plane alignment. In this experiment we use VGGNet for both of these methods.

We show that pose-aware face recognition provides a significant improvement over the baseline. Moreover, in order to better factor the effect of pose on performance, we designed an experiment with only image-to-image comparisons using IJB-A data (i.e., ignoring set structures), and classifying the pose of each image using our method. For each method, we then plot $\text{TAR@FAR} = 1\%$ of image-to-image matching across poses. Our results are reported in Fig. 8 which demonstrates how image-to-image TAR varies by pose. Note that since we artificially flip all faces to one side, we do not have any negative yaw values.

In particular, in Fig. 8c we show the relative improvement of PAM with respect to the baseline. It is clear that the major improvement comes from increased matching performance for fully profile images, even when matching them across different poses. Indeed, the largest TAR improvement is exhibited when matching different poses in the range 90 degree versus 40-50 degree. Evident from Fig. 8c is that, although the approach improves in general, this improvement is not uniform. Matching profile images up to 40 degree has the highest improvement (+50-30 percent); then moderate improvement (+20 percent) is shown for extreme cases (e.g., 80 versus 20 degree); small improvement is shown in the frontal region.

Effect of Input Resolution and Landmark Noise. Fig 9 shows the effects of two main variables which usually affect face recognition performance: face resolution and landmark noise. We tested our system by downsizing the original

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

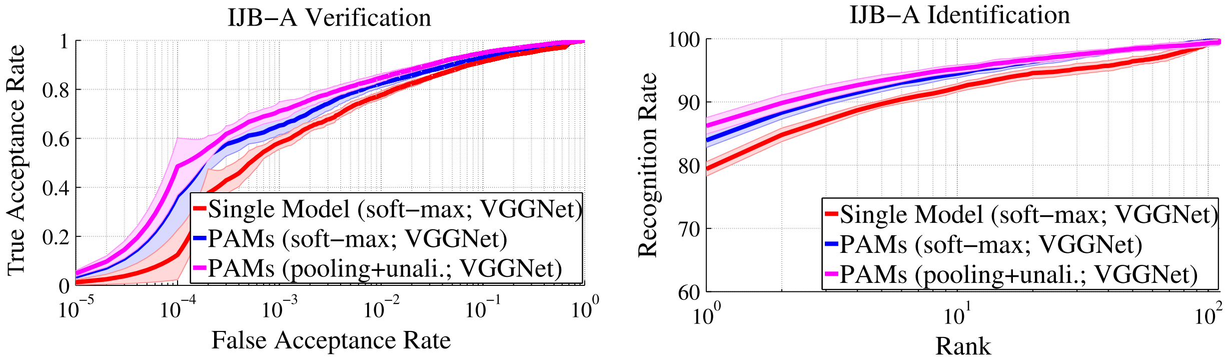


Fig. 7. ROC (a) and CMC (b) improvements comparing a single CNN and the proposed PAMs on the IJB-A challenge. Horizontal axes are shown in log scale. Each curve also shows the standard deviation.

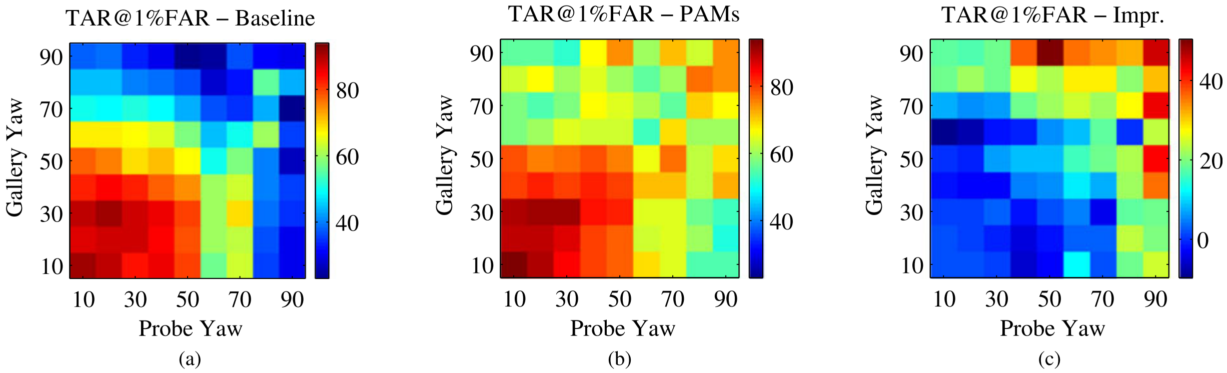


Fig. 8. Image-to-Image TAR across poses: PAMs (b) show better pose invariance than the baseline (a). The relative difference between PAMs and the baseline is visible in (c).

input images for different scaling values $\{100\%, 75\%, 50\%, 25\%\}$, leading to smaller and smaller bounding boxes. Fig. 9 reports two main metrics on IJB-A as functions of the average input box that was used to perform recognition. Moreover, we performed a similar test in which we stressed the system by injecting noise in landmark detection positions using Gaussian noise centered on each landmark, varying sigma σ_x, σ_y. Fig. 9 shows recognition metrics as function of this noise. Interestingly, it appears that performance drops linearly with increased landmark noise, while input resolution produces a much steeper drop.

Comparison with the State-of-the-Art. Table 6 reports a comparison with the state-of-the-art. For convenience, the table also provides details on the methods used and their components. Specifically, for each method we report whether or not it uses deep features (`Deep. Learn.`), if the CNN is fine-tuned to the target data set (fine-tuned ten times on IJB-A splits `Net. on train`), if supervised metric learning methods are used on the IJB-A training splits (`Met. Learn.`), and the number of CNNs used by the system (`#Net used`).

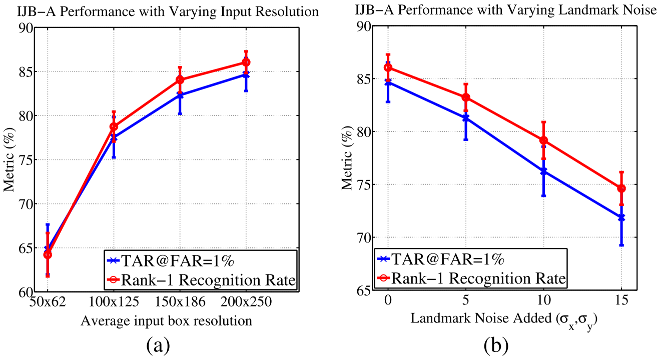


Fig. 9. IJB-A performance by varying (a) input resolution, (b) noise added to landmark points. Two main metrics are reported: $\mathrm{TAR@FAR} = 1\%$ and recognition rate at Rank-1.

Our PAM obtained better performance compared to GOTS (government off-the-shelf). Moreover, compared to deep learning based methods, PAMs show better performance than [13] which exploits seven networks and fuses the results with a commercial system. It is worth mentioning that PAMs improve in verification over [13] by about 11 percent TAR at FAR=0.01 and 20 percent TAR at FAR=0.001. PAMs also show a better rank-1 recognition rate.

Surprisingly, PAMs obtain comparable ROC with methods that explicitly fine-tuned their networks on the IJB-A training set (at additional computational costs) and/or applied metric-learning to these sets. PAMs, by comparison, did not perform supervised learning on the IJB-A training set. Specifically, our system shows better ROC curves compared to methods that tuned their CNNs ten times on IJBA training splits [14], [31], but it does not perform as well on IJB-A identification, where our method is less effective in ranking gallery sets.

We also performed a comparison with a single network trained on the pose-augmented CASIA set, following recent similar techniques [15], [16]. Although the network trained on the pose-augmented set is very competitive since most of the gap is filled, PAMs still improve TAR by performing pose-aware matching. That is, additional benefits of the method come from pose-aware recognition and 3D rendering at test-time.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

TABLE 6
Comparative Performance Analysis on IJB-A Benchmark for Verification (ROC) and Identification (CMC)

Methods ↓	Deep	Net	Met.	#Net	Pose-Aware	IJB-A Verification (TAR)			IJB-A Identification (Rec. Rate)		
Metrics →	learn.	on train	learn.	used	matching	@FAR=0.01	@FAR=0.001	FAR=0.0001	@Rank-1	@Rank-5	@Rank-10
GOTS	×	×	×	0	×	0.406 ± 0.014	0.198 ± 0.008	–	0.443 ± 0.021	0.595 ± 0.020	–
OpenBR [63]	×	×	×	0	×	0.236 ± 0.009	0.104 ± 0.014	–	0.246 ± 0.011	0.375 ± 0.008	–
Wang et al. [13]	✓	×	×	7	×	0.733 ± 0.034	0.514 ± 0.060	–	0.820 ± 0.024	0.929 ± 0.013	–
Chen et al. [14]	✓	✓	✓	1	×	0.787 ± 0.043	–	–	0.86 ± 0.023	0.943 ± 0.017	0.962 ± 0.012
Chen et al. [31]	✓	✓	✓	2	×	0.838 ± 0.042	–	–	**0.903 ± 0.012**	**0.965 ± 0.008**	**0.97.7 ± 0.007**
$\text{PAM}_{\text{out-0}}$ (frontalization only)	✓	×	×	1	×	0.733 ± 0.018	0.552 ± 0.032	0.318 ± 0.188	0.771 ± 0.016	0.887 ± 0.009	0.919 ± 0.009
Pose-augmented (soft-max)	✓	×	×	1	×	0.820 ± 0.014	0.558 ± 0.068	0.124 ± 0.135	0.827 ± 0.011	0.923 ± 0.007	0.945 ± 0.007
PAMs [1] (soft-max)	✓	×	×	5	✓	0.826 ± 0.018	0.652 ± 0.037	0.365 ± 0.203	0.840 ± 0.012	0.925 ± 0.008	0.946 ± 0.007
PAMs (pooling)	✓	×	×	5	✓	0.840 ± 0.014	0.699 ± 0.028	0.478 ± 0.103	0.862 ± 0.011	0.931 ± 0.008	0.948 ± 0.005
Pose-augmented (pooling+unal.)	✓	×	×	1	×	0.824 ± 0.018	0.662 ± 0.036	0.466 ± 0.111	0.856 ± 0.014	0.927 ± 0.009	0.942 ± 0.007
PAMs (pooling+unal.)	✓	×	×	5	✓	**0.847 ± 0.016**	**0.711 ± 0.037**	**0.486 ± 0.116**	0.862 ± 0.013	0.943 ± 0.009	0.953 ± 0.006

(Symbol "–" Indicates that the Metric is not Available for that Protocol. Standard Deviation is not Available for all the methods).

Finally, we note that other recent state-of-the-art IJB-A results reported after submission of this paper can be found in [15], [16], [32], [59], [60], [61], [62].

Comparison between Matching Methods. In Table 6 we additionally provide a comparison with the popular frontalization approach [3], [9], which corresponds to using the frontalized $\text{PAM}_{\text{out}-0}$ on all the images. Our PAM shows improvement over frontalization. We believe that the main reason for the advantage of PAM is the large number of images that are corrupted during the frontalization. This is particularly true for images of faces which are near-profile and far from frontal.

Additionally, in Table 6 we can see that of all the matching methods tested with PAM on IJB-A, substantial improvement over the matching proposed in [1] (soft-max) is obtained by separately pooling still images and frames. This is especially true at very low FAR (see the ROC curves in Fig. 7). This means that performing early fusion of deep features within a set is more effective than pair-wise feature comparison whenever the fusion considers the source of the input data (e.g., still image or video frame).

7.2 People in Photo Albums (PIPA)

We report additional results on the recently introduced People In Photo Albums (PIPA) dataset [19]. It consists of images originally appearing in public photo albums uploaded and available from Flickr. One of the characteristics of this dataset is that it contains extreme pose variations and so multiple cues, not just facial appearances, are assumed to be necessary in order to accurately recognize people appearing in these images. The benchmark test protocol also measures face recognition performance using a subset of the data where faces are clearly visible enough to provide meaningful information for face recognition.

We follow the same protocol described in [19], which uses two-fold cross-validation on the face recognition subset of PIPA.[3] Unlike them, we do not train a classifier for each subject, but rather classify each subject over the 581 identities described in Section 6. We first tested using only a single-pose frontal model, which matches the reported DeepFace [3] performance of 47.97 percent. More recent methods [20] increased accuracy to 67.89 percent by using a system similar to [27] which includes sixty CNNs. Note that as reported by [19], DeepFace results were influenced by landmark failures. Tests using our full PAM approach achieved a 57.65 percent ± 0.58 without dealing with landmark failures as well. By improving face-preprocessing using a newer and more robust version of the CLNF face detector [58], [64] and by using unaligned images, we were able to obtain state-of-the-art performance on PIPA with an accuracy of 70.13 percent ± 0.30. This increase in accuracy is notable not only because of the large gap in favor of our method, but also considering that DeepFace trained SVM classifiers on the PIPA training set, whereas the method in [20] uses a far larger ensemble of deep networks than our approach.

8 Conclusion and Discussion

In this paper we proposed a pose-aware method to perform face recognition with face images, demonstrating extreme pose variation. Our approach shows how we can rely, not only on a single frontal CNN model, but also on half-profile and full-profile models to perform face recognition in the wild—especially when the target dataset contains full-profile images. Unlike many existing, contemporary methods, which perform frontalization to neutralize pose variations, our approach proposes to render faces while taking into consideration their estimated poses in an adaptive manner. Our approach is agnostic to the underlying CNN architecture used. We demonstrate this and many other properties of our system in an extensive set of experiments.

Additional future work consists of training PAMs in a single optimization framework. The proposed method can be extended by learning a multi-branch network with different convolutional filters for faces in different poses (e.g., a sub-network for frontal, a sub-network for profile), and then a sub-network with weight sharing that jointly learns a common embedding, as similarly done for 3D generic objects in [65]. Finally, recent advances in estimating 3D face shapes under challenging, unconstrained conditions [66] offer potential new directions for improved rendering of faces in novel viewpoints.

[3] For data and splits see http://www.cs.berkeley.edu/~nzhang/piper.html.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

ACKNOWLEDGMENTS

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon. Moreover, we gratefully acknowledge USC HPC for hyper-computing and the support of NVIDIA Corporation for their donation of an NVIDIA Titan X. I.M. is the lead author; F.C., J.C., S.H., J.K., K.K., J.L., S.R., and Y.W. in alphabetical order; T.H. is the lead senior author; all remaining senior authors appear next in alphabetical order.

REFERENCES

[1] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, “Pose-aware face recognition in the wild,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4838–4846.

[2] W. AbdAlmageed, et al., “Face recognition using deep multi-pose representations,” in *Proc. Winter Conf. App. Comput. Vis.*, 2016, pp. 1–9.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[4] C. Lu and X. Tang, “Surpassing human-level face verification performance on LFW with GaussianFace,” in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3811–3819.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogniti.*, 2015, pp. 815–823.

[6] 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,” Univ. Massachusetts, Amherst, MA, Tech. Rep. 07–49, Oct. 2007.

[7] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” arXiv preprint, 2014. [Online]. Available: http://arxiv.org/abs/1411.7923

[8] B. F. Klare, et al., “Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus benchmark A,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1931–1939.

[9] T. Hassner, S. Harel, E. Paz, and R. Enbar, “Effective face frontalization in unconstrained images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4295–4304.

[10] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 529–534.

[11] A. R. Chowdhury, T. Y. Lin, S. Maji, and E. Learned-Miller, “One-to-many face recognition with bilinear CNNs,” *2016 IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Placid, NY, 2016, pp. 1–9, doi: 10.1109/WACV.2016.7477593.

[12] T. Hassner, “Viewing real-world faces in 3D,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3607–3614.

[13] D. Wang, C. Otto, and A. K. Jain, “Face search at scale: 80 million gallery,” arXiv preprint, 2015. [Online]. Available: http://arxiv.org/abs/1507.07242

[14] J. Chen, V. M. Patel, and R. Chellappa, “Unconstrained face verification using deep CNN features,” arXiv preprint, 2015. [Online]. Available: http://arxiv.org/abs/1508.01722v1

[15] I. Masi, A. Tran, T. Hassner, J. T. Leksut, and G. Medioni, “Do we really need to collect millions of faces for effective face recognition?” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 579–596.

[16] D. E. Crispell, O. Biris, N. Crosswhite, J. Byrne, and J. L. Mundy, “Dataset augmentation for pose and lighting invariant face recognition,” in *Proc. IEEE Appl. Imagery Pattern Recognit. Workshop*, 2016.

[17] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, “Rotating your face using multi-task deep neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 676–684.

[18] Z. Zhu, P. Luo, X. Wang, and X. Tang, “Multi-view perceptron: A deep model for learning face identity and view representations,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 217–225.

[19] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev, “Beyond frontal faces: Improving person recognition using multiple cues,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4804–4813.

[20] H. Li, J. Brandt, Z. Lin, X. Shen, and G. Hua, “A multi-level contextual model for person recognition in photo albums,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1297–1305.

[21] T. Kanade and A. Yamada, “Multi-subregion based probabilistic approach toward pose-invariant face recognition,” in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2003, pp. 954–959.

[22] A. B. Ashraf, S. Lucey, and T. Chen, “Learning patch correspondences for improved viewpoint invariant face recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[23] X. Zhang and Y. Gao, “Face recognition across pose: A review,” *Pattern Recognit.*, vol. 42, no. 11, pp. 2876–2896, 2009.

[24] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith, “Fully automatic pose-invariant face recognition via 3D pose normalization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 937–944.

[25] A. Pentland, B. Moghaddam, and T. Starner, “View-based and modular eigenspaces for face recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 84–91.

[26] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-PIE,” *Image Vis. Comput.*, vol. 28, pp. 807–813, 2010.

[27] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1891–1898.

[28] Y. Sun, X. Wang, and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2892–2900.

[29] E. Zhou, Z. Cao, and Q. Yin, “Naive-deep face recognition: Touching the limit of LFW benchmark or not?” arXiv preprint, 2015. [Online]. Available: http://arxiv.org/abs/1501.04690

[30] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 41.1–41.12.

[31] J. C. Chen, V. M. Patel, and R. Chellappa, “Unconstrained face verification using deep CNN features,” in *Proc. Winter Conf. App. Comput. Vis.*, 2016, pp. 1–9.

[32] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, “Template adaptation for face verification and identification,” in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 1–8.

[33] A. Li, S. Shan, X. Chen, and W. Gao, “Maximizing intra-individual correlations for face recognition across pose differences,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 605–611.

[34] U. Prabhu, J. Heo, and M. Savvides, “Unconstrained pose-invariant face recognition using 3D generic elastic models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1952–1961, Oct. 2011.

[35] J. Heo and M. Savvides, “Gender and ethnicity specific generic elastic models from a single 2D image for novel 2D pose face synthesis and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2341–2350, Dec. 2011.

[36] I. Masi, G. Lisanti, A. Bagdanov, P. Pala, and A. Del Bimbo, “Using 3D models to recognize 2D faces in the wild,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2013, pp. 775–780.

[37] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, “Dictionary-based face recognition from video,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 766–779.

[38] A. Asthana, T. Marks, M. Jones, K. Tieu, and M. Rohith, “Fully automatic pose-invariant face recognition via 3D pose normalization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 937–944.

[39] D. Yi, Z. Lei, and S. Li, “Towards pose robust face recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 3539–3545.

[40] I. Masi, C. Ferrari, A. DelBimbo, and G. Medioni, “Pose independent face recognition by localizing local binary patterns via deformation components,” in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 4477–4482.

[41] T. Hassner, et al., “Pooling faces: Template based face recognition with pooled face images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 59–67.

[42] C. Ferrari, G. Lisanti, S. Berretti, and A. DelBimbo, “Dictionary learning based 3D morphable model construction for face recognition with varying expression and pose,” in *Proc. Int. Conf. 3D Vis.*, 2015, pp. 509–517.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

[43] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," arXiv preprint, 2014. [Online]. Available: http://arxiv.org/abs/1404.3543

[44] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng, "Practice and Theory of Blendshape Facial Models," *Eurographics 2014 - State Art Rep.*, L. Sylvain and S. Michela, The Eurographics Association, 2014, doi: 10.2312/egst.20141042.

[45] F. ju Chang, A. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "FacePoseNet: Making a case for landmark-free face alignment," in *Proc. 7th IEEE Int. Workshop Anal. Model. Faces Gestures ICCV Workshops*, 2017, pp. 11599–1608.

[46] Y. Wu, T. Hassner, K. Kim, G. Medioni, and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, p. 1, 2015, doi: 10.1109/TPAMI.2017.2787130.

[47] T. Baltrušsaitis, P. Robinson, and L.-P. Morency, "3D constrained local model for rigid and non-rigid facial tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2610–2617.

[48] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[51] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[52] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[53] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.

[54] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 897–902.

[55] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1978–1990, Oct. 2011.

[56] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1867–1874.

[57] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis*, 2014, pp. 94–108.

[58] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jun. 2013, pp. 354–361.

[59] I. Masi, T. Hassner, A. T. Tran, and G. Medioni, "Rapid synthesis of massive face sets for improved face recognition," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 604–611.

[60] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 17–24.

[61] C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo, "Investigating nuisance factors in face recognition with DCNN representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 583–591.

[62] J. Yang, et al., "Neural aggregation network for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5216–5225.

[63] J. Klontz, B. Klare, S. Klum, E. Taborsky, M. Burge, and A. K. Jain, "Open source biometric recognition," in *Proc. Int. Conf. Biometrics: Theory Appl. Syst.*, 2013, pp. 1–8.

[64] K. Kim, T. Baltrusaitis, A. B. Zadeh, L.-P. Morency, and G. G. Medioni, "Holistically constrained local model: Going beyond frontal poses for facial landmark detection," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 95.1–95.12.

[65] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.

[66] A. Tuan Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1493–1502.

Iacopo Masi received the BS, MS and PhD degrees in computer engineering from the Università di Firenze, Italy, in 2006, 2009 and 2014 respectively. He has been a postdoctoral researcher in the Media Integration and Communication Center (MICC), Firenze, Italy, and is currently a postdoctoral scholar with the University of Southern California. His research interests include pattern recognition and computer vision—specifically the areas of tracking, person re-identification, 2D/3D face recognition and modeling.

Feng-Ju Chang received the BS degree in electrical engineering from National Cheng Kung University and the MS degree in communication engineering from National Taiwan University, in 2009 and 2011 respectively. She was a research assistant in Academia Sinica from 2011 to 2013, and is currently working toward the PhD degree in the Department of Electrical Engineering with the University of Southern California. Her research interests include computer vision and machine learning.

Jongmoo Choi received the BS degree in physics, the MS degree in cognitive science and the PhD degree in computer engineering from Sungkyunkwan University, South Korea. He was a research assistant professor with the ISRC, Sungkyunkwan University, and is currently a senior research associate with IRIS Labs, University of Southern California. His research interests include 3D face modeling and recognition.

Shai Harel received the MSc degree from the Open University of Israel in the field of computer vision, under the supervision of Professor Tal Hassner. He has more than 15 years of experience in computer programming—10 years in the field of statistical machine learning. He is an expert in field of 2D and 3D alignment and deep learning.

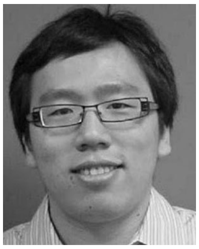
Jungyeon Kim received the BS degree from the Electronics and Computer Engineering Department, Pusan National University, and the MS degree in electrical engineering from the Pohang University of Science and Technology, in 2001 and 2003 respectively. She has been working with Samsung Electronics since 2003, and is currently working toward the PhD degree in the Department of Electrical Engineering with the University of Southern California. Her research interests include computer vision and machine learning.

KangGeon Kim received the BS and MS degrees in mechanical engineering from Seoul National University, Korea, in 2004 and 2008, respectively. From 2008 to 2013, he was a research scientist in the Korea Institute of Science and Technology (KIST). He is currently working toward the PhD degree in the Department of Computer Science, University of Southern California. His research interests include computer vision, image processing and machine learning.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.

Jatuporn Leksut received the BS degree in computer science from George Washington University, in 2012, and the PhD degree from the Department of Computer Science, University of Southern California. Her interests include computer vision and computer graphics.

Stephen Rawls is a researcher with ISI USC, working on both computer vision and document processing. He is currently working toward the PhD degree with the University of Southern California under Dr. Prem Natarajan. Prior to working at ISI, he was with BBN Technologies.



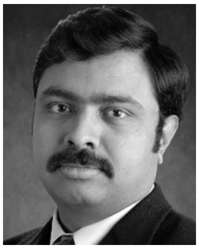
Yue Wu received the BE degree in telecommunication engineering from the Huazhong University of Sciences and Technologies, the MSc degree in applied mathematics from the University of Toledo, and the PhD degree in electrical engineering from Tufts University, in 2001, 2008, and 2012. He is a computer scientist in the Information Sciences Institute (USC). Previously, he was a scientist with Raytheon BBN Technologies in Cambridge, Massachusetts. His research focuses on information security, image processing, and pattern recognition.

Tal Hassner received the MSc and PhD degrees in applied mathematics and computer science from the Weizmann Institute of Science, in 2002 and 2006, respectively. In 2008 he joined the Department of Math. and Computer Science, The Open University of Israel where he is currently an associate professor. Since 2015, he has also been a senior computer scientist at the Information Sciences Institute, USC, California.

Wael AbdAlmageed received the PhD degree with distinction from the University of New Mexico, in 2003, where he was also selected as the Outstanding Graduate Student. He is a senior scientist in USC's Information Sciences Institute. His research focus is on applying learning techniques to computer vision and bioinformatics. Prior to ISI, he was a research scientist with the University of Maryland.

Gérard Medioni received the diplôme d'Ingenieur degree from ENST, Paris, in 1977, and the MS and PhD degrees from the University of Southern California, in 1980 and 1983 respectively. He has been with USC since then, and is currently a professor of computer science and electrical engineering, and co-director of the Institute for Robotics and Intelligent Systems (IRIS). He is also currently the director of research at Amazon.

Louis-Philippe Morency received the PhD degree from the Massachusetts Institute of Technology. He is an ssistant professor with Carnegie Mellon University where he leads the Multimodal Communication and Machine Learning Laboratory. He was formerly part of the research faculty with the University of Southern California. Research focuses on building the computational foundations to analyze and recognize subtle human communicative behaviors during social interactions.



Prem Natarajan is the Michael Keston executive director of Information Sciences Institute, a vice dean of the USC Viterbi School of Engineering and a professor of computer science. At ISI, he leads managerial and technical directions Institute-wide, including research, development and the MOSIS electronic chip brokerage. He also heads teams in his areas of expertise: novel approaches to face, character, handwriting and speech recognition, along with other deep learning and natural language processing directions.

Ram Nevatia received the PhD degree in electrical engineering from Stanford University, California. He has been with the University of Southern California since 1975, where he is currently a professor of computer science and electrical engineering and the director of the Institute for Robotics and Intelligent Systems (IRIS). He has made contributions to the areas of object recognition, stereo analysis, modeling from aerial images, action recognition and tracking.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on June 23,2020 at 13:27:43 UTC from IEEE Xplore. Restrictions apply.