proxies 代理 / surrogates 替代
asymmetric 不对称

Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network

Anh Tuãn Trãn[1], Tal Hassner[2,3], Iacopo Masi[1], and Gérard Medioni[1]

[1] Institute for Robotics and Intelligent Systems, USC, CA, USA

[2] Information Sciences Institute, USC, CA, USA

[3] The Open University of Israel, Israel

Abstract

The 3D shapes of faces are well known to be discriminative. Yet despite this, they are rarely used for face recognition and always under controlled viewing conditions. We claim that this is a symptom of a serious but often overlooked problem with existing methods for single view 3D face reconstruction: when applied "in the wild", their 3D estimates are either unstable and change for different photos of the same subject or they are over-regularized and generic. In response, we describe a robust method for regressing discriminative 3D morphable face models (3DMM). We use a convolutional neural network (CNN) to regress 3DMM shape and texture parameters directly from an input photo. We overcome the shortage of training data required for this purpose by offering a method for generating huge numbers of labeled examples. The 3D estimates produced by our CNN surpass state of the art accuracy on the MICC data set. Coupled with a 3D-3D face matching pipeline, we show the first competitive face recognition results on the LFW, YTF and IJB-A benchmarks using 3D face shapes as representations, rather than the opaque deep feature vectors used by other modern systems.

1. Introduction

Single view 3D face shape estimation methods originally proposed using their 3D shapes for recognition [4, 7, 26]. This makes sense because 3D shapes are discriminative – different people have different face shapes – yet invariant to lighting, texture changes and more. Indeed, previous work showed that when available, high resolution 3D face scans are excellent face representations which can even be used to distinguish between the faces of identical twins [9].

Curiously, however, despite their widespread use, single view face reconstruction methods are rarely employed by modern face recognition systems. The highly successful 3D Morphable Models (3DMM), for example, were only ever used for recognition in limited, controlled viewing conditions [4, 7, 10, 16, 26]. To our knowledge, there are no reports of successfully using single view face shape estimation – 3DMM or any other method – to recognize faces in challenging unconstrained, *in the wild* settings.

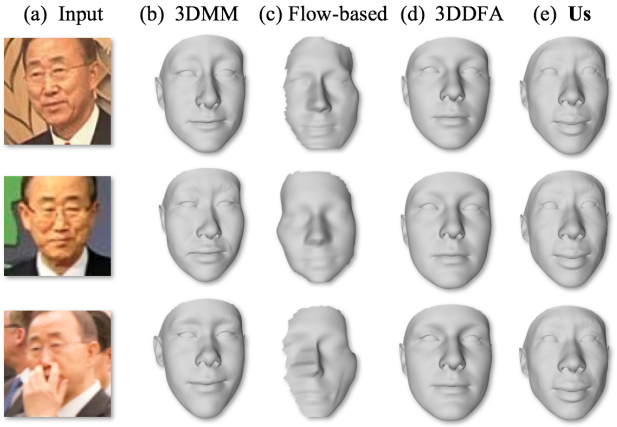

Figure 1: *Unconstrained, single view, 3D face shape reconstruction*. (a) Input images of the same subject with disruptive poses and occlusions. (b-e) 3D reconstructions using (b) single-view 3DMM [31], (c) flow based method [12] (d) 3DDFA [45], (e) Our proposed approach. (b-c) Present different 3D shapes for the same subject and (d) appears generic, whereas our method (e) is robust, producing similar discriminative 3D shapes for different views.

An important reason why this maybe so, is that these methods can be unstable in unconstrained viewing conditions. We later verify this quantitatively but it can also be seen in Fig. 1 which presents 3D shapes estimated from three unconstrained photos by three different methods (Fig. 1 (b-d)). Clearly, though the same subject appears in all photos, *shapes produced by the same method are either very different (b,c) or highly regularized and generic (d)*. It is therefore unsurprising that these shapes are poor representations for recognition. It also explains why some recently proposed using coarse, simple 3D shape approximations only as proxies when rendering faces to new views rather than as face representations [12, 14, 24, 37].

Contrary to previous work, we show that robust and dis-

criminative 3D face shapes can, in fact, be estimated from single, unconstrained images (Fig. 1 (e)). We propose estimating 3D facial shapes using a very deep convolutional neural network (CNN) to regress 3DMM shape and texture parameters directly from single face photos. We identify shortage of labeled training data as an obstacle to using data-hungry CNNs for this purpose. We address this problem with a novel means for generating a huge labeled training set of unconstrained faces and their 3DMM representations. Coupled with additional technical novelties, we obtain a method which is fast, robust and accurate.

The accuracy of our estimated shapes is verified on the MICC data set [1] and *quantitatively shown to surpass the accuracy of other 3D reconstruction methods*. We further show that our estimated shapes are robust and discriminative by presenting face recognition results on the Labeled Faces in the Wild (LFW) [17], YouTube Faces (YTF) [40] and IJB-A [22] benchmarks. To our knowledge, *this is the first time single image 3D face shapes are successfully used to represent faces from modern, unconstrained face recognition benchmarks*. Finally, to promote reproduction of our results, we publicly release our code and models.[1].

2. Related work

Over the years, many attempts were made to estimate the 3D surface of a face appearing in a single view. Before listing them, it is important to mention recent *multi image* methods which use image sets for reconstruction (e.g., [23, 28, 32, 33, 36]). Although these methods produce accurate 3D reconstructions, they require many images from multiple sources to produce a single 3D face shape whereas we reconstruct faces from single images.

Methods for *single view* 3D face reconstructions can broadly be categorized into the following types.

Statistical shape representations, such as the widely popular 3DMM [5, 6, 10, 26, 30, 38, 43], use many aligned 3D face shapes to learn a distribution of 3D faces, represented as a high dimensional subspace. Each point on this subspace is a parameter vector representing facial geometry and sometimes expression and texture. Reconstruction is performed by searching for a point on this subspace that represents a face similar to the one in the input image. These methods do not attempt to produce discriminative facial geometries and indeed, as mentioned earlier, were only used for face recognition under controlled settings.

The very recent method of [29] also uses a CNN to regress 3DMM parameters for face photos. They too recognize absence of training data as a major concern. Contrary to us, they propose synthesizing training faces with known geometry by sampling from the 3DMM distribution. This approach produces synthetic looking photos which can easily cause overfitting problems when training large networks [24]. They were therefore able to train only a shallow residual network (seven layers compared to our 101) and their estimated shapes were not shown to be more robust or discriminative than other methods.

Scene assumption methods. In order to obtain *correct* reconstructions, some make strong assumptions on the scene and the viewing conditions in the input image. Shape from shading methods [20], for example, make assumptions on the light sources, facial reflectance and more. Others instead use facial symmetry [11]. The assumptions they and others make often do not hold in practice, limiting the application of these methods to controlled settings.

Example based methods, beginning from the work of [13] and more recently [12, 37], modify the 3D surface of example face shapes, fitting them to the face appearing in input photo. These methods favor robustness to challenging viewing conditions over detailed reconstructions. They were thus only used for face recognition to synthesize new views from unseen poses.

Landmark fitting methods. Finally, some reconstruction techniques fit a 3D surface to detected facial landmarks rather than to face intensities directly. These include methods designed for videos (e.g., [18, 34]) and the CNN based approaches of [19, 45]. These focus more on landmark detection than 3D shape estimation and so do not attempt to produce detailed and discriminative facial geometries.

3. Regressing 3DMM parameters with a CNN

We propose to regress 3DMM face shape parameters directly from an input photo using a very deep CNN. Ostensibly, CNNs are ideal for this task: After all, they are being successfully applied to many related computer vision tasks. But despite their success, apart from [29], we are unaware of published reports of using CNNs for 3DMM parameter regression.

We believe CNNs were not used here because this is a regression problem where both the input photo and the output 3DMM shape parameters are high dimensional. Solving such problems requires deep networks and these need massive amounts of training data. Unfortunately, existing unconstrained face sets with ground truth 3D shapes are far too small for this purpose and obtaining large quantities of 3D face scans is labor intensive and impractical.

We therefore instead leverage three key observations.

1. As discussed in Sec. 2, accurate 3D estimates can be obtained by using *multiple images* of the same face.

2. Unlike the limited availability of ground truth 3D face shapes, there is certainly no shortage of challenging face sets containing multiple photos per subject.

[1]Please see www.openu.ac.il/home/hassner/projects/CNN3DMM for updates.

3. Highly effective deep networks are available for the related task of extracting robust and discriminative face representations for face recognition.

From (1), we have a reasonable way of producing 3D face shape estimates for training, as surrogates for ground truth shapes: by using a robust method for multi-view 3DMM estimation. Getting multiple photos for enough subjects is very easy (2). This abundance of examples further allows balancing any reconstruction errors with potentially limitless subjects to train on. Finally, (3), a state of the art CNN for face recognition may be fine-tuned to this problem. It should already be tuned for unconstrained facial appearance variations and trained to produce similar, discriminative outputs for different images of the same face.

3.1. Generating training data

To generate training data, we use a simple yet effective *multi image* 3DMM estimation method, loosely based on the one recently proposed by [28]. We run it on the unconstrained faces in the CASIA WebFace dataset [44]. These multi image 3DMM estimates are then used as ground truth 3D face shapes when training our CNN 3DMM regressor.

Multi image 3DMM reconstruction is performed by first estimating 3DMM parameters from the 500k single images in CASIA. 3DMM estimates for images of the same subject are then aggregated into a single 3DMM per subject (∼10k subjects). This process is described next (see also, Fig. 2).

The 3DMM representation. Our system uses the popular Basel Face Model (BFM) [26]. It is a publicly available 3DMM representation and one of the state of the art methods for single view 3D face modeling.

A face is modeled by decoupling its shape and texture giving the following two independent generative models.

$$\mathbf{S}' = \widehat{\mathbf{s}} + \mathbf{W}_S\,\boldsymbol{\alpha} \quad , \quad \mathbf{T}' = \widehat{\mathbf{t}} + \mathbf{W}_T\,\boldsymbol{\beta}. \tag{1}$$

Here, the vectors $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{t}}$ are the mean face shape and texture, computed over the aligned facial 3D scans in the Basel Faces collection and represented by the concatenated 3D coordinates of the 3D point clouds and the concatenated RGB values of their textures. Matrices $\mathbf{W}_S$ and $\mathbf{W}_T$ are the principle components, computed from the same aligned facial scans. Finally, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are each 99D parameter vectors, representing shape and texture respectively.

Single image 3DMM fitting. Fitting a 3DMM to each training image is performed with a slightly modified version of the two standard methods of [8] and [31]. Given an image $\mathbf{I}$, we estimate parameter vectors $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta}^\star$ which represent a face similar to the one in $\mathbf{I}$ (Eq. (1)). Unlike previous work, we begin processing by applying the CLNF [21] state of the art facial landmark detector. It provides $K = 68$ facial landmarks $\mathbf{p}_k \in \mathbb{R}^2$, $k \in 1..K$, and a confidence score value w (which we use later on).

Landmarks are used to obtain an initial estimate for the pose of the input face, in the reference 3DMM coordinate system. Pose is represented by six degrees of freedom for rotation, $\mathbf{r} = [r_\alpha, r_\beta, r_\gamma]$, and translation, $\mathbf{t} = [t_X, t_Y, t_Z]$, and estimated similar to [12]. 3DMM fitting then proceeds by optimizing over the shape, texture, pose, illumination, and color model following [8]. We found that CLNF makes occasional localization errors. To introduce more stability, our optimization also uses the edge-based cost of [31]. For more details on this optimization, we refer to [8] and [31].

Once the optimization converges, we take the shape and texture parameters, $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta}^\star$, from the last iteration as our single image 3DMM estimate for the input image $\mathbf{I}$. Importantly, though this process is known to be computationally expensive, it is applied in our pipeline only in preprocessing and once for every training image. We later show our CNN regressor to be much faster.

Multi image 3DMM fitting. Although a number of multi image 3D face shape estimation methods were proposed in the past, we found the following simple approach, inspired by the very recent work of [28], to be particularly effective.

Specifically, we pool the shape and texture 3DMM parameters $\boldsymbol{\gamma}_i = [\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i], i \in 1..N$ across all the N single view estimates belonging to the same subject. Pooling is performed by element wise weighted averaging of the N 3DMM vectors, resulting in a single 3DMM estimate for that subject, $\widehat{\boldsymbol{\gamma}}$. That is,

$$\widehat{\boldsymbol{\gamma}} = \sum_{i=1}^{N} w_i \cdot \boldsymbol{\gamma}_i \quad \text{and} \quad \sum_{i=1}^{N} w_i = 1, \tag{2}$$

where w_i are normalized per-image confidences provided by the CLNF facial landmark detector.

Note that unlike [28], we do not use a rank-list based on distances of normals as a quality measure to pool 3DMM parameters, instead taking the landmark detection confidence measure for these weights. Following this process, each CASIA subject is associated with a single, pooled 3DMM parameter vector $\widehat{\boldsymbol{\gamma}}$. For ease of notation, henceforth we will drop the *hat* when denoting pooled features, assuming all training set 3DMM parameters were pooled.

3.2. Learning to regress pooled 3DMM

Following the process described in Sec. 3.1, each subject in our data set is associated with a number of images and a single, pooled 3DMM. We now use this data to learn a function which, ideally, regresses the *same* pooled 3DMM feature vector for different photos of the same subject.

To this end, we use a state of the art CNN, trained for face recognition. We use the very deep ResNet architecture [15] with 101 layers, recently trained for face recognition by [24]. We modify its last fully-connected layer to output the 198D 3DMM feature vector $\boldsymbol{\gamma}$. The network is then

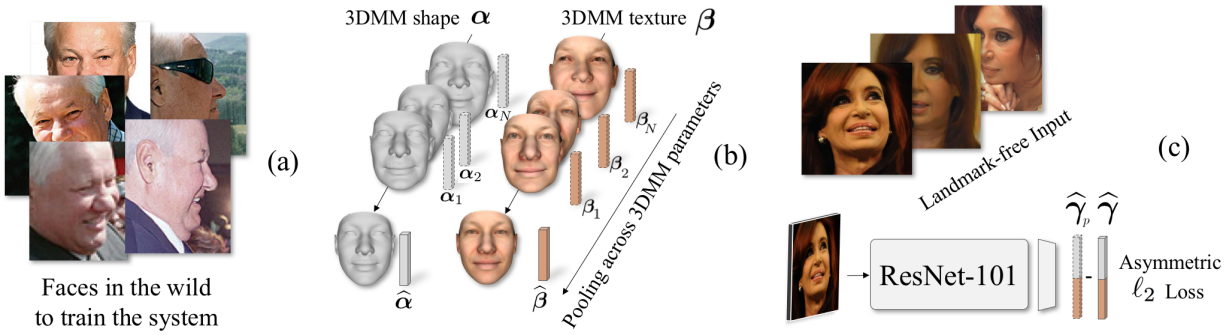

Figure 2: *Overview of our process.* (a) Large quantities of unconstrained photos are used to fit a single 3DMM for each subject. (b) This is done by first fitting single image 3DMM shape and texture parameters to each image separately. Then, all 3DMM estimates for the same subject are pooled together for a single estimate per subject. (c) These pooled estimates are used in place of expensive ground truth face scans to train a very deep CNN to regress 3DMM parameters directly.

fine-tuned on CASIA images using the pooled 3DMM estimates as target values; different images of the same subject presented to the CNN using the same target 3DMM shape. We note that we also tried using the VGG-Face CNN of [25] with 16 layers. Its results were similar to those obtained by the ResNet architecture, though somewhat lower.

The asymmetric Euclidean loss. Training our network requires some care when defining its loss function. 3DMM vectors, by construction, belong to a multivariate Gaussian distribution with its mean on the origin, representing the mean face (Sec. 3.1). Consequently, during training, using the standard Euclidean loss to minimize distances between estimated and target 3DMM vectors will favor estimates closer to the origin: these will have a higher probability of being closer to their target values than those further away. In practice, we found that a network trained with the Euclidean loss tends to output less detailed faces (Fig. 3).

To counter this bias towards a mean face shape, we introduce an *asymmetric Euclidean loss*. It is designed to encourage the network to favor estimates further away from the origin by decoupling under-estimation errors (errors on the side of the 3DMM target closer to the origin) from over-estimation errors (where the estimate is further out from the origin than the target). It is defined by:

$$\mathcal{L}(\boldsymbol{\gamma}_p, \boldsymbol{\gamma}) = \lambda_1 \cdot \underbrace{||\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}_{\max}||_2^2}_{\text{over-estimate}} + \lambda_2 \cdot \underbrace{||\boldsymbol{\gamma}_p^+ - \boldsymbol{\gamma}_{\max}||_2^2}_{\text{under-estimate}}, \quad (3)$$

using the element-wise operators:

$$\boldsymbol{\gamma}^+ \doteq \text{abs}(\boldsymbol{\gamma}) \doteq \text{sign}(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}; \quad \boldsymbol{\gamma}_p^+ \doteq \text{sign}(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}_p, \quad (4)$$

$$\boldsymbol{\gamma}_{\max} \doteq \max(\boldsymbol{\gamma}^+, \boldsymbol{\gamma}_p^+). \quad (5)$$

Here, $\boldsymbol{\gamma}$ is the target pooled 3DMM value, $\boldsymbol{\gamma}_p$ is the output, regressed 3DMM and $\lambda_{1,2}$ control the trade-off between the over and under estimation errors. When both equal 1, this reduces to the traditional Euclidean loss. In practice, we set $\lambda_1 = 1$, $\lambda_2 = 3$, thus changing the behavior of the training process, allowing it to escape under-fitting faster and

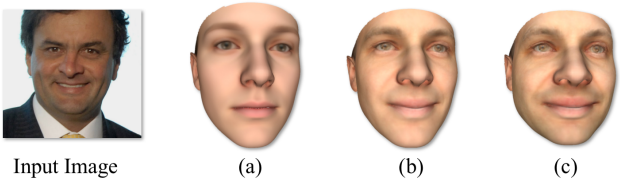

Figure 3: *Effect of our loss function:* (left) Input image, (a) generic model, (b) regressed shape and texture with a regular ℓ_2 loss and (c) our proposed asymmetric ℓ_2 loss.

encouraging the network to produce more detailed, realistic 3D face models (Fig. 3).

Network hyperparameters. Eq. (3) is solved using Stochastic Gradient Descent (SGD) with a mini-batch of size 144, momentum set to 0.9 and with regularization over the weights provided by ℓ_2 with a weight decay of 0.0005. When performing back-propagation, we learn the inner product layer (fc) after pool5 faster, setting the learning rate to 0.01, since it is trained from scratch for the regression problem. Other network weights are updated with a learning rate an order of magnitude lower. When the validation loss saturates, we decrease learning rates by an order of magnitude, until the validation loss stops decreasing.

Discussion: Render-free 3DMM estimator. It is important to note that by choosing to use a CNN to regress 3DMM parameters, we obtain a function that is *render-free*. That is, 3DMM parameters are regressed directly from the input image, without an optimization process which renders the face and compares it to the photo, as do existing methods for 3DMM estimation (including our method for generating training data in Sec. 3.1). By using a CNN, we therefore hope to gain not only improved accuracy, but also much faster 3DMM estimation speeds.

3.3. Parameter based 3D-3D recognition

The CNN we train in Sec. 3.2 represents a function $f : \mathbf{I} \mapsto \boldsymbol{\gamma}_p$, giving us 3DMM parameters $\boldsymbol{\gamma}_p$ for an input

image **I**. We later use our 3DMM estimates in face recognition benchmarks, to test how robust and discriminative they are. We next describe the method used for that purpose to evaluate the similarity of two face shapes and textures to determine if they represent the same subject.

3D-3D recognition with a single image. We perform face recognition using the 3DMM parameters regressed by our network: By using the 3DMM parameters γ_p as face descriptors. Because different benchmarks often exhibit specific appearance biases, we apply Principal Component Analysis (PCA), learned from the training splits of the test benchmark, to adapt our estimated parameter vectors to the benchmark. Signed, element wise square rooting of these vectors is then used to further improve representation power [27]. Finally, the similarity of two faces, $s(\gamma_{p1}, \gamma_{p2})$, is evaluated by computing their cosine score:

$$s(\gamma_1, \gamma_2) = \frac{\gamma_{p1} \cdot \gamma_{p2}^T}{||\gamma_{p1}|| \cdot ||\gamma_{p2}||}. \quad (6)$$

3D-3D recognition with multiple-images. In some scenarios, a subject is represented by a set of images, rather than just one. This is the case in the YTF benchmark [40] where videos are used, each containing multiple frames, and in the recent IJB-A [22], which uses *templates* containing heterogeneous visual data (images, videos and possibly more).

We use the same pipeline for single images also for image sets. Here, however, 3DMM parameters for different images or frames are first pooled using Eq. (2). Unlike the process applied in Sec. 3.1, all images here have equal weights, as we do not run landmark detection prior to 3DMM fitting with our CNN (see below). When using templates with both videos and images, following [24], we first pool the 3DMM estimates for frames in each video separately, obtaining one 3DMM per video. We then pool these 3DMMs with those of other images in the same template.

Face alignment. Facial landmark detection and face alignment are known to improve recognition accuracy (e.g., [41, 14]). In fact, the recent, related work of [16] manually assigned landmarks before using their 3DMM fitting method for recognition on controlled images. We, however, *did not align faces* beyond using the bounding boxes provided in their data sets. We found our method robust to misalignments and so spared the runtime this required.

4. Experimental results

We test our proposed method, comparing the accuracy of its estimated 3D shapes, its speed and its ability to represent faces for recognition with existing methods. Importantly, *we are unaware of any previous work on single view 3D face shape estimation which reported as many quantitative tests as we do*, in terms of the number of benchmarks used, the number of baseline methods compared with and the level of difficulty of the photos used in these tests.

Method	3DRMSE	RMSE	$\log_{10} \times 10^4$	Rel$\times 10^4$	Sec.
Generic	1.88±.52	3.48±.76	28±7	65±16	–
3DMM [31]	1.75±.42	3.64±.94	29±8	68±18	120
Flow-based [12]	1.83±.39	3.29±.70	27±6	62±14	13.3
Us	**1.57±.33**	**3.18±.77**	**26±6**	**59±14**	**.088**
Generic+*pool*	1.88±.52	3.48±.76	28±7	65±16	–
3DMM [31]+*pool**	1.60±.46	3.31±.98	27±9	62±20	120
3DDFA [45]+*pool*	1.83±.58	3.45±.85	28±7	65±17	.146
[18]	1.84±.32	3.73±.62	30±5	68±11	.372
[2]+*pool*	1.84±.58	3.45±.85	28±6	65±13	52.3
Us +*pool*	**1.53±.29**	**3.14±.70**	**25±6**	**58±13**	**.088**

Table 1: *3D estimation accuracy and per-image speed* on the MICC dataset. Top are single view methods, bottom are multi frame. See text for details on measures. 3DRMSE in real-world mm; $\log_{10}$ and Rel were both scaled to preserve space. * Denotes the method used to produce the training data in Sec. 3.1. Lower values are better.

Specifically, we evaluate the accuracy of our estimated 3D shapes using videos and photos and their corresponding scanned, ground truth 3D shapes from the MICC Florence Faces dataset [1] (Sec. 4.1). To test how discriminative and robust our shapes are when estimated from unconstrained images, we perform single image and multi image face recognition using the LFW [17], YTF [40] and the new IARPA JANUS Benchmark-A (IJB-A) [22] (Sec. 4.3). Finally we also provide qualitative results in Sec. 4.4.

As baseline 3D reconstruction methods we used standard 3DMM fitting [31], which we implemented ourselves, the flow-based method of [12], the edge based method of [2], the multi resolution, multi-view approach of [18] and the recent 3DDFA [45], were all tested with their authors' implementations.

4.1. 3D shape reconstruction accuracy

The MICC dataset [1] contains challenging face videos of 53 subjects. The videos span the range of controlled to challenging unconstrained outdoor settings. For each of the subjects in these videos, the data set contains also a ground-truth 3D model acquired using a structured-light scanning system with high precision. This allows comparing our 3D face shape estimates with the ground truth shapes.

These videos were used for single image and multi frame 3D reconstructions, comparing our method to existing alternatives. In these tests, estimated and ground truth shape parameters were converted to 3D using Eq. (1), cropped at a radius of $95mm$ around the tip of the nose and globally aligned using the standard, rigid iterative closest point (ICP) method [3], obtaining $X, X^* \subseteq \mathbb{R}^3$, respectively. They were additionally projected to a frontal view, obtaining depth maps D_Q and D_Q^*. Estimation accuracy was then

Method	3D	Texture	Accuracy	100%-EER	AUC	TAR-10%	TAR-1%
Labeled Faces in the Wild							
EigenFaces [39]		–	60.02±0.79	–	–	25	6.2
Hybrid Descriptor [41]		–	78.47±0.51	–	–	66.60	42.4
DeepFace-ensemble [37]		–	97.35±0.25	–	–	99.6	93.7
AugNet [24]		–	98.06±0.60	98.00±0.73	–	99.5	94.2
3DMM [31]	✓	×	66.13±2.79	65.70±2.81	72.24±2.75	35.90±3.74	12.37±4.81
	×	✓	74.93±1.14	74.50±1.21	82.94±1.14	60.40±3.15	28.73±7.17
	✓	✓	75.25±2.12	74.73±2.56	83.21±1.93	59.4±4.64	29.67±4.73
3DDFA [45]	✓	×	66.98±2.56	67.13±1.90	73.30±2.49	36.76±6.27	10.00±3.22
Us	✓	×	90.53±1.34	90.63±1.61	96.6±0.79	91.13±2.62	58.20±12.14
	×	✓	90.6±1.07	90.70±1.17	96.75±0.59	91.23±2.42	52.60±8.14
	✓	✓	**92.35±1.29**	**92.33±1.33**	**97.71±0.64**	**94.2±2.00**	**65.57±6.93**
YouTube Faces							
MBGS LBP [40]		–	76.4±1.8	74.7	82.6	60.5	35.8
DeepFace-ensemble [37]		–	91.4±1.1	91.4	96.3	92	54
3DMM [31]+*pool**	✓	×	73.26±2.51	73.08±2.65	80.41±2.60	51.36±5.11	24.04±4.56
	×	✓	77.34±2.54	76.96±2.64	85.32±2.63	63.16±5.07	31.36±5.21
	✓	✓	79.56±2.08	79.20±2.07	87.35±1.92	69.08±5.00	34.56±6.89
3DDFA [45]+*pool*	✓	×	68.10±2.93	67.96±3.12	74.95±3.04	40.52±3.65	12.2±2.67
Us +*pool*	✓	×	88.28±1.84	88.32±2.16	**95.95±1.38**	86.60±3.95	**51.12±8.86**
	×	✓	87.56±2.56	87.68±2.25	94.44±1.38	84.80±4.89	40.92±8.26
	✓	✓	**88.80±2.21**	**88.84±2.40**	95.37±1.43	**87.92±4.18**	46.56±6.20

Table 2: *LFW and YTF face verification.* Comparing our 3DMM regression with others, including baseline face recognition methods. * Denotes the same method used to produce 3DMM target values for our CNN training (Sec. 3.1).

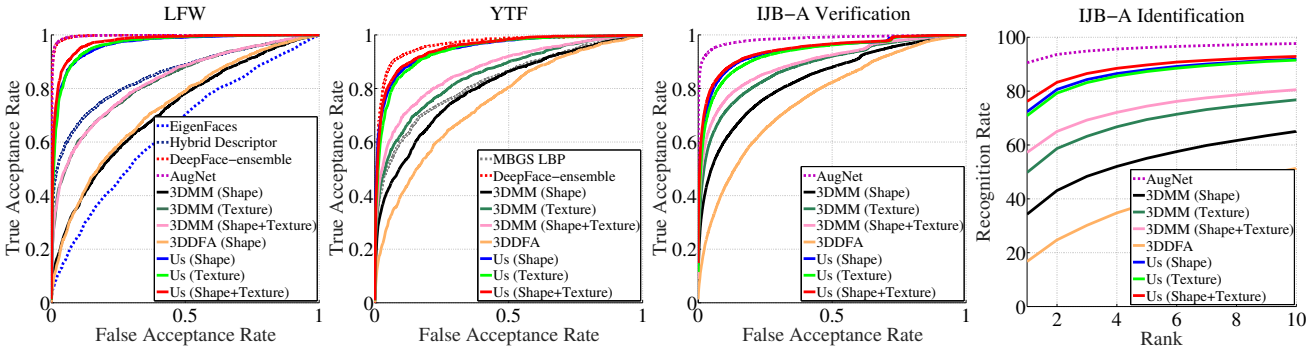

Figure 4: *Face verification and recognition results.* From left to right: Verification ROC curves for LFW, YTF, and IJB-A, and the recognition CMC for IJB-A.

computed with standard error measures [12, 35]:

- **3D Root Mean Square Error** (3DRMSE): $\sqrt{\sum_i (X - X^*)^2 / N_v}$
- **Root Mean Square Error** (RMSE): $\sqrt{\sum_i (D_{Q_i} - D^*_{Q_i})^2 / N_p}$
- **log10**: $|\log_{10}(D_Q) - \log_{10}(D^*_Q)|$
- **Relative error** (Rel): $|D_Q - D^*_Q| / |D^*_Q|$

Here, N_v is the number of 3D vertices and N_p the number of pixels in these representations.

Single view estimation was performed on the most frontal frame. Multi frame reconstructions were given the entire videos. Our multi frame results were produced by pooling 3DMM estimates from different frames, using Eq. (2), with equal weights used for all frames. For all 3DMM fitting baselines [2, 18, 31, 45], we found that estimating shape, texture *and expression* parameters but using only shape and texture for comparisons, gave the best results. This approach was therefore used in all our tests.

Method	3D	Text.	TAR-10%	TAR-1%	Rank-1	Rank-5	Rank-10
AugNet	–	–	88.6±1.6	90.6±1.2	96.2±0.6	97.7±0.4	
3DMM* +*p.*	✓	×	60.7±2.0	30.6±3.2	34.3±2.2	55.1±2.1	65.1±2.0
	×	✓	71.1±1.8	39.5±4.8	49.8±2.5	69.5±1.4	76.8±1.0
	✓	✓	75.4±1.6	46.6±5.1	57.2±1.9	74.4±1.3	80.5±1.1
3DDFA+*p.*	✓	×	43.3±2.5	12.5±1.9	16.7±1.9	38.3±2.7	51.3±3.0
Us +*pool*	✓	×	86.0±1.7	55.9±5.5	72.3±1.4	88.0±1.4	91.8±1.1
	×	✓	83.5±2.2	50.3±5.8	70.9±1.5	87.3±1.1	91.5±1.0
	✓	✓	**87.0±1.5**	**60.0±5.6**	**76.2±1.8**	**89.7±1.0**	**92.9±1.0**

Table 3: *IJB-A face verification and recognition.* Comparing our 3DMM regression with others, including baseline face recognition methods.* Denotes the same method used to produce 3DMM target values for our CNN training.

Results are reported in Tab. 1. Error rates are averaged across all videos and provided ± standard deviation. Our method is clearly the most accurate. Remarkably, both its single view and multiple frame versions outperform the method used to produce the training set target 3DMM labels (3DMM+*pool*). This may be due to our use of such a large dataset to train the CNN and their known robustness to training label errors and noise [42].

Our estimates are more accurate than the very recent state-of-the-art. This includes 3DDFA [45] which fits 3DMM parameters by using a CNN to deal with large pose variations as well as [18] and [2]. To better appreciate these numbers, note that our improvement over standard 3DMM fitting is comparable to their improvement over using a unmodified, generic Basel face shape [26].

4.2. 3DMM regression speed

Tab. 1 (rightmost column) also reports the average, per image runtime in seconds, required by the various methods to predict 3D face shapes. We compared our approach with iterative methods such as classic 3DMM implementations [2, 18, 31], the flow-based method of [12] and also with a recent CNN based method [45].

As mentioned earlier, our method is render-free, without optimization loops which render the estimated parameters and compare them to the input photo. Unsurprisingly, at 0.088s (∼11Hz), our CNN is *several orders of magnitude faster* predicting 3DMM parameters than most of the methods we tested. The second fastest method, by a wide gap, is the 3DDFA of [45], requiring 0.146s (∼7Hz) for prediction.

Runtime was measured on two different systems. All our baselines required MS-Windows to run and were tested on an Intel Core i7-4820K CPU @ 3.7GHz with 16GB RAM and a NVIDIA GeForce GTX 770. Our method requires Linux and so was tested on an Intel Xeon CPU @ 3.60GHz, with 12 GB of RAM and GeForce GTX 590. Importantly, the system used to measure our runtime is the slower of the two. Our runtimes may therefore be exaggerated.

4.3. Face recognition in the wild

We next consider the robustness of our 3DMM estimates and how discriminative they are. We aim to see if our 3DMM estimates for different unconstrained photos of the same person are more similar to each other than to those of other subjects. An effective way of doing this is by testing our 3DMM estimates on face recognition benchmarks. We emphasize that our goal is *not* to set new face recognition records. Doing so would require competing with state of the art systems designed exclusively for that problem. We provide performances of relevant (though not necessarily state of the art) recognition systems only as a reference. Nevertheless, our results below are the highest we know of that were obtained with meaningful features (here, shape and texture parameters) rather than opaque representations.

Our tests use the pipeline described in Sec. 3.3 and report multiple recognition metrics for verification (in LFW and YTF) and identification metrics (in IJB-A). These metrics are verification accuracy, 100%-EER (Equal Error Rate), Area Under the Curve (AUC), and recall (True Acceptance Rate) at two cut-off points of the False Alarm Rate (TAR-{10%,1%}). For identification we report the recognition rates at various ranks from the CMC (Cumulative Matching Characteristic). For each tested method we also indicate its use of estimated 3D shape and/or texture. Finally, bold values indicate best scoring 3D reconstruction methods.

Labeled Faces in the Wild (LFW) [17] results are provided in Tab. 2 (top) and Fig. 4 (left). Evidently, the shapes estimated by 3DDFA [45] are only slightly more robust and discriminative than the classical eigenfaces [39]. Fitting 3DMMs using [31] does better, but falls behind the Hybrid method of [41], one of the first results on LFW, now nearly a decade old. Both results suggest that the shapes estimated by these methods are unstable in unconstrained settings and/or are too generic. By comparison, recognition performances with our estimated 3DMM parameters is not far behind those recently reported by Facebook, using their multi-CNN approach trained on four million images [37].

YouTube Faces (YTF) [40] Accuracy on YTF videos is reported in Tab. 2 (bottom) and Fig. 4 (mid-left). Though video frames in this set are often low in quality and resolution, our method performs well. It is outperformed by the Facebook CNN ensemble system [37], explicitly designed for face recognition, by an AUC gap of only ∼1%. The 3DMM shapes and textures estimated by other methods perform far worst, with [31] doing only slightly better than the MBGS face recognition system [40], which is the oldest result on that benchmark and [45] falling far behind.

IARPA Janus Benchmark A. (IJB-A) [22] Released recently, IJB-A was designed to offer elevated challenges compared to other face recognition benchmarks. In particular, it presents faces in near profile poses, almost nonex-

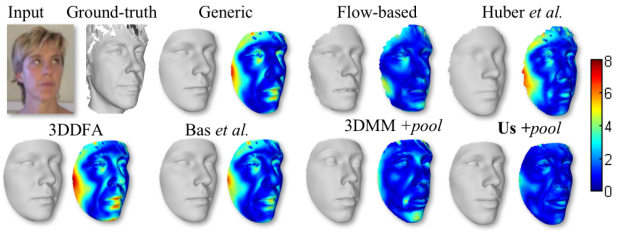

Figure 5: *Qualitative comparison of surface errors,* visualized as heat maps with real world mm errors on faces from MICC videos and their ground truth 3D shapes. Left to right, top to bottom: frame from input and 3D ground-truth shape; the generic face; estimates for flow-based method [12], Huber et al. [18], 3DDFA [45], Bas et al. [2], 3DMM +*pool* [31], our method +*pool*.

istent in previous face sets. It further contains faces in extremely low resolution and often strongly affected by noise.

We evaluated both the face verification (1:1) and recognition (1:N) protocols and report results in Tab. 3 and Fig. 4 (mid-right, right). Here too, performances adopt the same pattern as in the previous two benchmarks, with 3D shapes estimated by 3DDFA [45] performing far worst than other methods. Our own method performs quite well, though it is outperformed by a wide margin by the very recent face recognition system of [24], which was designed for this set.

4.4. Qualitative Results

Fig. 5 provides a qualitative comparison of the surface errors in mm for different methods for a subject in the MICC dataset. Our method produces visually smaller errors compared to ground-truth. The areas around the nose and mouth in particular have very low errors, while other methods are more sensitive in these regions (e.g 3DDFA [45]). We provide also qualitative 3D reconstructions of faces in the wild, using images from LFW and single frames from YTF videos. Fig. 6 presents these results showing both rendered 3D shapes and (when available) also its estimated texture. These results show that our method generates more visually plausible 3D and texture estimates compared with those produced by other methods. Fig. 5 also shows a few failure cases, here due to facial hair which was missing from the original 3DMM representation and extreme out-of-plane rotation which produced a thin, unrealistic 3D shape.

5. Conclusions

We show that existing methods for estimating 3D face shapes may either be sensitive to changing viewing conditions, particularly in unconstrained settings, or too generic. Their estimated shapes therefore do not capture identity very well, despite the fact that true 3D face shapes are known to be highly discriminative.

We propose instead to use a very deep CNN architecture

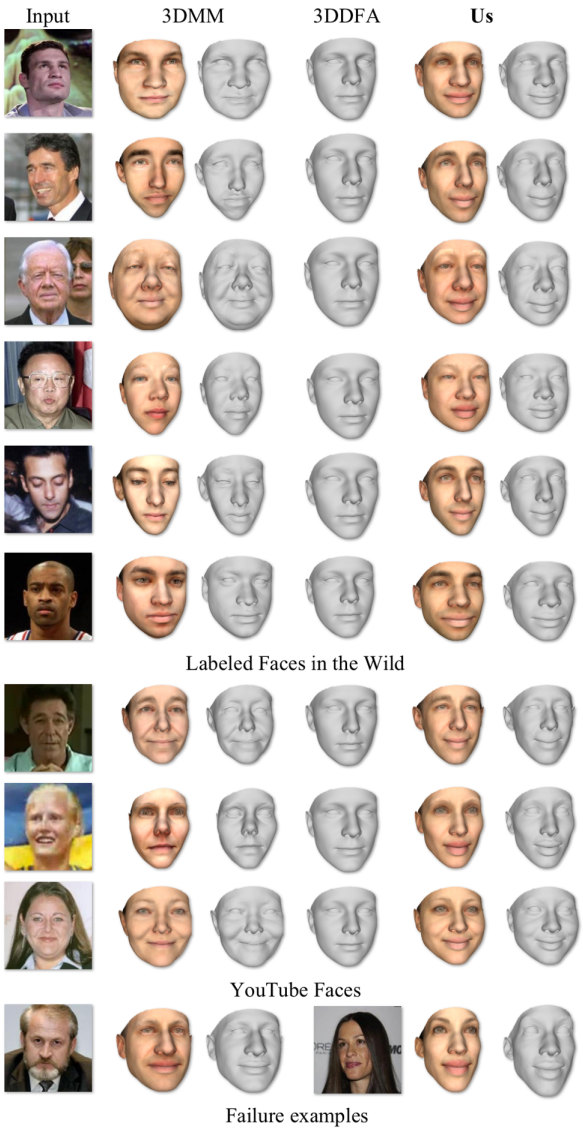

Figure 6: *Qualitative results*, produced by 3DMM [31], 3DDFA [45] and our method on still-images from LFW and single frames from YTF. Bottom: Two failure examples.

to regress 3DMM parameters directly from input images. We provide a solution to the problem of obtaining sufficient labeled data to train this network. We show our regressed 3D shapes to be more accurate than those of alternative methods. We further run extensive face recognition tests showing these shapes to be robust to unconstrained viewing conditions and discriminative. Our results are furthermore the highest recognition results we know of, obtained with interpretable representations rather than opaque features. We leave it to future work to regress more 3DMM parameters (e.g., expressions) and design state of the art recognition systems using these shapes instead of the abstract features used by others.

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 2014-14071600011. 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.

References

[1] A. Bagdanov, A. D. Bimbo, and I. Masi. The florence 2D/3D hybrid face dataset. In *ACM Multimedia Conf. Workshops*, 2011. Available: www.micc.unifi.it/masi/research/ffd. 2, 5

[2] A. Bas, W. A. P. Smith, T. Bolkart, and S. Wuhrer. Fitting a 3D morphable model to edges: A comparison between hard and soft correspondences. *arxiv preprint*, abs/1602.01125, 2016. 5, 6, 7, 8

[3] P. J. Besl and N. McKay. A method for registration of 3-D shapes. *Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. 5

[4] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 192–197, 2002. 1

[5] V. Blanz, K. Scherbaum, T. Vetter, and H. Seidel. Exchanging faces in images. *Comput. Graphics Forum*, 23(3), 2004. 2

[6] V. Blanz and T. Vetter. Morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH Conf. Comput. Graphics*, 1999. 2

[7] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003. 1

[8] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, Sept 2003. 3

[9] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *Int. J. Comput. Vision*, 64(1):5–30, 2005. 1

[10] B. Chu, S. Romdhani, and L. Chen. 3D-aided face recognition robust to expression and pose variations. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2014. 1, 2

[11] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3D structure recovery of faces. *European Conf. Comput. Vision*, pages 99–113, 2004. 2

[12] T. Hassner. Viewing real-world faces in 3D. In *Proc. Int. Conf. Comput. Vision*, pages 3607–3614. IEEE, 2013. Available: www.openu.ac.il/home/hassner/projects/poses. 1, 2, 3, 5, 6, 7, 8

[13] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *Proc. Conf. Comput. Vision Pattern Recognition Workshops*. IEEE, 2006. 2

[14] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2015. 1, 5

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2016. 3

[16] G. Hu, F. Yan, C.-H. Chan, W. Deng, W. Christmas, J. Kittler, and N. M. Robertson. Face recognition using a unified 3D morphable model. In *European Conf. Comput. Vision*, pages 73–89. Springer, 2016. 1, 5

[17] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, UMass, Amherst, October 2007. 2, 5, 7

[18] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. Koppen, W. Christmas, M. Rtsch, and J. Kittler. A multiresolution 3D morphable face model and fitting framework. In *Int. Conf. on Computer Vision Theory and Applications*, 2016. 2, 5, 6, 7, 8

[19] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3D model fitting. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016. 2

[20] I. Kemelmacher-Shlizerman and R. Basri. 3D face reconstruction from a single image using a single reference face shape. *Trans. Pattern Anal. Mach. Intell.*, 33(2):394–405, 2011. 2

[21] K. Kim, T. Baltruaitis, A. Zadeh, L.-P. Morency, and G. Medioni. Holistically constrained local model: Going beyond frontal poses for facial landmark detection. In *Proc. British Mach. Vision Conf.*, 2016. 3

[22] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark-A. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2015. 2, 5, 7

[23] S. Liang, L. G. Shapiro, and I. Kemelmacher-Shlizerman. Head reconstruction from internet photos. In *European Conf. Comput. Vision*, pages 360–374. Springer, 2016. 2

[24] 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 *European Conf. Comput. Vision*, 2016. Available www.openu.ac.il/home/hassner/projects/augmented_faces. 1, 2, 3, 5, 6, 8

[25] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. British Mach. Vision Conf.*, 2015. 4

[26] P. Paysan, R. Knothe, B. Amberg, S. Romhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *Int. Conf. on Advanced Video and Signal based Surveillance*, 2009. 1, 2, 3, 7

[27] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conf. Comput. Vision*, pages 143–156. Springer, 2010. 5

[28] M. Piotraschke and V. Blanz. Automated 3D face reconstruction from multiple images using quality measures. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2016. 2, 3

[29] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In *Int. Conf. on 3D Vision*, 2016. 2

[30] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proc. Int. Conf. Comput. Vision*, 2003. 2

[31] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. Conf. Comput. Vision Pattern Recognition*, volume 2, pages 986–993, 2005. 1, 3, 5, 6, 7, 8

[32] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2015. 2

[33] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2016. 2

[34] S. Saito, T. Li, and H. Li. Real-time facial segmentation and performance capture from RGB input. In *European Conf. Comput. Vision*, 2016. 2

[35] A. Saxena, M. Sun, and A. Ng. Make3D: Learning 3d scene structure from a single still image. In *Trans. Pattern Anal. Mach. Intell.*, volume 31, page 824840, 2008. 6

[36] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz. Total moving face reconstruction. In *European Conf. Comput. Vision*, pages 796–812. Springer, 2014. 2

[37] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2014. 1, 2, 6, 7

[38] H. Tang, Y. Hu, Y. Fu, M. Hasegawa-Johnson, and T. S. Huang. Real-time conversion from a single 2d face image to a 3D text-driven emotive audio-visual avatar. In *Int. Conf. on Multimedia and Expo*, pages 1205–1208. IEEE, 2008. 2

[39] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. Conf. Comput. Vision Pattern Recognition*, 1991. 6, 7

[40] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2011. 2, 5, 6, 7

[41] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, October 2008. 5, 6, 7

[42] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian. DisturbLabel: Regularizing CNN on the loss layer. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016. 7

[43] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3D-aware face component transfer. *ACM Trans. on Graphics*, 30(4):60, 2011. 2

[44] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. Available: http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html. 3

[45] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016. 1, 2, 5, 6, 7, 8