Dense Face Alignment

Yaojie Liu[1], Amin Jourabloo[1], William Ren[2], and Xiaoming Liu[1]
[1]Department of Computer Science and Engineering, Michigan State University, MI
[2]Monta Vista High School, Cupertino, CA
[1]{liuyaoj1,jourablo,liuxm}@msu.edu, [2]williamyren@gmail.com

Abstract

Face alignment is a classic problem in the computer vision field. Previous works mostly focus on sparse alignment with a limited number of facial landmark points, i.e., facial landmark detection. In this paper, for the first time, we aim at providing a very dense 3D alignment for large-pose face images. To achieve this, we train a CNN to estimate the 3D face shape, which not only aligns limited facial landmarks but also fits face contours and SIFT feature points. Moreover, we also address the bottleneck of training CNN with multiple datasets, due to different landmark markups on different datasets, such as 5, 34, 68. Experimental results show our method not only provides high-quality, dense 3D face fitting but also outperforms the state-of-the-art facial landmark detection methods on challenging datasets. Our model can run at real time during testing and it's available at http:///cvlab.cse.msu.edu/project-pifa.html.

1. Introduction

Face alignment is a long-standing problem in the computer vision field, which is the process of aligning facial components, e.g., eye, nose, mouth, and contour. An accurate face alignment is an essential prerequisite for many face related tasks, such as face recognition [8], 3D face reconstruction [22, 21] and face animation [37]. There are fruitful previous works on face alignment, which can be categorized as either generative methods such as the early Active Shape Model [17] and Active Appearance Model (AAM) based approaches [13], or discriminative methods such as regression-based approaches[38, 28].

Most previous methods estimate a *sparse* set of landmarks, e.g., 68 landmarks. As this field is being developed, we believe that Dense Face Alignment (DeFA) is highly desirable. Here, DeFA denotes that it's doable to map any face-region pixel to the pixel in other face images, which has the *same* anatomical position in human faces. For example, given two face images from the same individual

Figure 1. A pair of images with their dense 3D shapes obtained by imposing landmark fitting constraint, contour fitting constraint and sift pair constraint.

but with different poses, lightings or expressions, a perfect DeFA can even predict a mole (i.e. darker pigment) on two faces as the same position. Moreover, DeFA should offer dense correspondence not only between two face images, but also between the face image and the canonical 3D face model. This level of detailed geometry interpretation of a face image is invaluable to many conventional facial analysis problems mentioned above.

Since this interpretation has gone beyond the sparse set of landmarks, fitting a dense 3D face model to the face image is a reasonable way to achieve DeFA. In this work, we choose to develop the idea of fitting a dense 3D face model to an image, where the model with thousands of vertexes makes it possible for face alignment to go very "dense". 3D face model fitting is well studied in the seminal work of 3D Morphorbal Model (3DMM) [4]. We see its recent surge in poplarith when it is applied to problems such as large-pose face alignment [10, 41], 3D reconstruction [5], and face recognition [1], especially using the convolutional neural network (CNN) architecture.

However, most prior works on 3D-model-fitting-based face alignment only utilize the sparse landmarks as supervision. There are two main challenges that need to be addressed in 3D face model fitting, in order to enable high-quality DeFA. First of all, to the best of our knowledge, no public face dataset has dense face shape labeling. All of the in-the-wild face alignment datasets have no more than 68 landmarks in the labeling. To provide a high-quality alignment for face-region pixels, we need a greater amount of information than just the landmark labeling. Hence, the first challenge is to seek valuable information for additional su-

pervision and integrate them in the learning framework.

In addition, similar to many other data-driven problems and solutions, it is preferred for multiple datasets to be involved for solving face alignment task since a single dataset has limited types of variations. However, many face alignment methods can not leverage multiple datasets, because each dataset either is labeled differently. For instance, AFLW dataset [23] contains a significant variation of poses, but has a few number of visible landmarks. In contrast, 300W dataset [23] contains a large number of faces with 68 visible landmarks, but all faces are in a near-frontal view. Therefore, the second challenge requires the proposed method to leverage multiple face datasets.

With the objective of addressing both challenges, we train a CNN to fit a 3D face model to the face image. While the proposed method works for any face image, we mainly pay attention to faces with large poses. Large-pose face alignment is a relatively new topic, and the performances in [10, 41] still have room to improve. To tackle the first challenge of limited landmark labeling, we propose to employ additional constraints. We include both contour constraint where the contour of the predicted shape should match the detected 2D face boundary, and SIFT constraint where the SIFT key points detected on two face images of the same individual should map to the same vertexes on the 3D face model. Both constraints are integrated into the CNN training as additional loss function terms, where the end-to-end training results in an enhanced CNN for 3D face model fitting. For the second challenge of leveraging multiple datasets, the 3D face model fitting approach has an inherent advantage in handling multiple training databases. Regardless of the landmark labeling number in a particular dataset, we can always define the corresponding 3D vertexes to guide the training.

Generally, our main contributions can be summarized as:

1. We identify and define a new problem of dense face alignment, seeking the alignment of face-region pixels beyond the sparse set of landmarks.

2. To achieve dense face alignment, we develop a novel 3D face model fitting algorithm that adopts multiple constraints and leverages multiple datasets.

3. Our dense face alignment algorithm outperforms the SOTA on challenging large-pose face alignment, and achieves competitive results on near-frontal face alignment. The model runs at real time.

2. Related Work

We review papers in three relevant areas: 3D face alignment from a single image, using multiple constraints in face alignment, and using multiple datasets for face alignment.

3D model fitting in face alignment Recently, there are increasingly attentions in conducting face alignment by fitting a 3D face model to a single 2D image [10, 41, 15, 16, 35, 11]. In [4], Blanz and Vetter proposed the 3DMM to represent the shape and texture of a range of individuals. The analysis-by-synthesis based methods are utilized to fit the 3DMM to the face image. In [41, 10] a set of cascade CNN regressors with the extracted 3D features is utilized to estimate the parameters of 3DMM and the projection matrix directly. Liu *et al*. [15] proposed to utilize two sets of regressors, one set for estimating update of 2D landmarks and the other set estimate update of dense 3D shape by using the 2D landmarks update. They apply these two sets of regressors alternatively. Compared to prior work, our method imposes additional constraints, which is the key to dense face alignment.

Multiple constraints in face alignment Other than landmarks, there are other features that are useful to describe the shape of a face, such as contours, pose and face attributes. Unlike landmarks, those features are often not labeled in the datasets. Hence, the most crucial step of leveraging those features is to find the correspondence between the features and the 3D shape. In [20], multiple features constraints in the cost function is utilized to estimate the 3D shape and texture of a 3D face. 2D edge is detected by Canny detector, and the corresponding 3D edges' vertices are matched by Iterative Closest Point (ICP) to use this information. Furthermore, [24] provides statistical analysis about the 2D face contours and the 3D face shape under different poses.

There are few works that use constraints as separate side tasks to facilitate face alignment. In [31], they set a pose classification task, predicting faces as left, right profile or frontal, in order to assist face alignment. Even with such a rough pose estimation, this information boosts the alignment accuracy. Zhang *et al*. [34] jointly estimates 2D landmarks update with the auxiliary attributes (e.g., gender, expression) in order to improve alignment accuracy. The "mirrorability" constraint is used in [32] to force the estimated 2D landmarks update be consistent between the image and its mirror image. In contrast, we integrate a set of constraints in an end-to-end trainable CNN to perform 3D face alignment.

Multiple datasets in face alignment Despite the huge advantages (e.g., avoiding dataset bias), there are only a few face alignment works utilizing multiple datasets, owing to the difficulty of leveraging different types of face landmark labeling. Zhu *et al*. [39] propose a transductive supervised descent method to transfer face annotation from a source dataset to a target dataset, and use both datasets for training. [25] ensembles a non-parametric appearance model, shape model and graph matching to estimate the superset of the landmarks. Even though achieving good results, it suffers from high computation cost. Zhang *et al*. [33] propose a deep regression network for predicting the superset of landmarks. For each training sample, the sparse shape regression is adopted to generate the different types of landmark

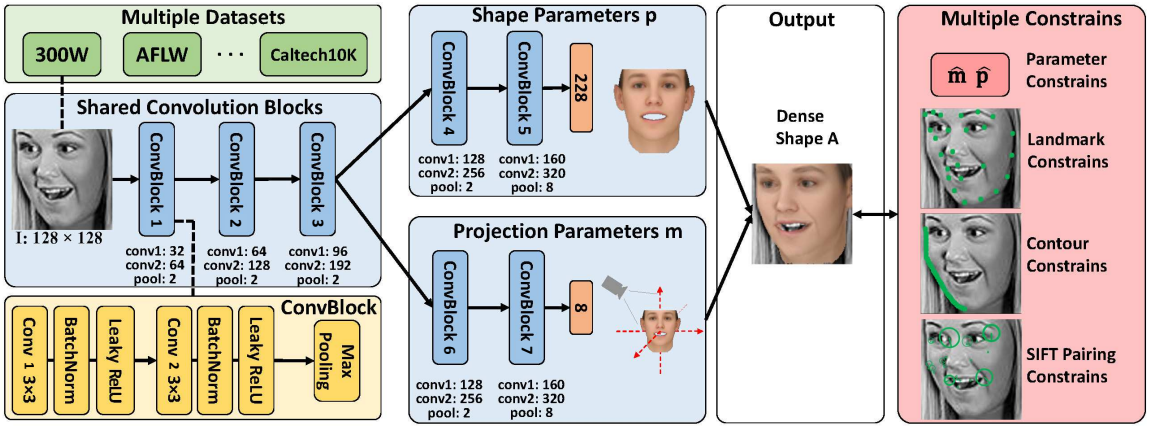


Figure 2. Architecture of CNN in the proposed DeFA method. The structure of each ConvBlock is shown in yellow area in the left bottom corner. Each convolution layer and fully connected layer is followed with one batch normalization layer (BN) and one leaky ReLU layer. The output dimension of each convolution layer is shown in the bottom of each unit, such as conv1: 32, which means the output has 32 channels. pool: 2 denotes the pooling layer adopts a stride of 2.

annotations. In general, most of the mentioned prior work learn to map landmarks between two datasets, while our method can readily handle an arbitrary number of datasets since the dense 3D face model can bridge the discrepancy of landmark definitions in various datasets.

3. Dense Face Alignment

In this section, we explain the details of the proposed dense face alignment method. We train a CNN for fitting the dense 3D face shape to a single input face image. We utilize the dense 3D shape representation to impose multiple constraints, e.g., landmark fitting constraint, contour fitting constraint and SIFT pairing constraint, to train such CNN.

3.1. 3D Face Representation

We represent the dense 3D shape of the face as, $\mathbf{S}$, which contains the 3D locations of Q vertices,

$$\mathbf{S} = \begin{pmatrix} x_1 & x_2 & \cdots & x_Q \\ y_1 & y_2 & \cdots & y_Q \\ z_1 & z_2 & \cdots & z_Q \end{pmatrix}. \tag{1}$$

To compute $\mathbf{S}$ for a face, we follow the 3DMM to represent it by a set of 3D shape bases,

$$\mathbf{S} = \bar{\mathbf{S}} + \sum_{i=1}^{N_{id}} p_{id}^i \mathbf{S}_{id}^i + \sum_{i=1}^{N_{exp}} p_{exp}^i \mathbf{S}_{exp}^i, \tag{2}$$

where the face shape $\mathbf{S}$ is the summation of the mean shape $\bar{\mathbf{S}}$ and the weighted PCA shape bases $\mathbf{S}_{id}$ and $\mathbf{S}_{exp}$ with corresponding weights of $\mathbf{p}_{id}, \mathbf{p}_{exp}$. In our work, we use 199 shape bases $\mathbf{S}_{id}^i, i = \{1, ..., 199\}$ for representing identification variances such as tall/short, light/heavy, and male/female, and 29 shape bases $\mathbf{S}_{exp}^i, i = \{1, ..., 29\}$ for representing expression variances such as mouth-opening, smile, kiss and etc. Each basis has $Q = 53,215$ vertices, which are corresponding to vertices over all the other bases. The mean shape $\bar{\mathbf{S}}$ and the identification bases $\mathbf{S}_{id}$ are from Basel Face Model [18], and the expression bases $\mathbf{S}_{exp}$ are from FaceWarehouse [7].

A subset of N vertices of the dense 3D face $\mathbf{U}$ corresponds to the location of 2D landmarks on the image,

$$\mathbf{U} = \begin{pmatrix} u_1 & u_2 & \cdots & u_N \\ v_1 & v_2 & \cdots & v_N \end{pmatrix}. \tag{3}$$

By considering weak perspective projection, we can estimate the dense shape of a 2D face based on the 3D face shape. The projection matrix has 6 degrees of freedom and can model changes w.r.t. scale, rotation angles (pitch α, yaw β, roll γ), and translations (t_x, t_y). The transformed dense face shape $\mathbf{A} \in \mathbb{R}^{3\times Q}$ can be represented as,

$$\mathbf{A} = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \end{bmatrix} \begin{bmatrix} \mathbf{S} \\ \mathbf{1}^\intercal \end{bmatrix} \tag{4}$$

$$\mathbf{U} = \mathbf{Pr} \cdot \mathbf{A}, \tag{5}$$

where $\mathbf{A}$ can be orthographically projected onto 2D plane to achieve $\mathbf{U}$. Hence, z-coordinate translation (m_{12}) is out of our interest and assigned to be 0. The orthographic projection can be denoted as matrix $\mathbf{Pr} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$.

Given the properties of projection matrix, the normalized third row of the projection matrix can be represented as the outer product of normalized first two rows,

$$[\bar{m}_9, \bar{m}_{10}, \bar{m}_{11}] = [\bar{m}_1, \bar{m}_2, \bar{m}_3] \times [\bar{m}_4, \bar{m}_5, \bar{m}_6]. \tag{6}$$

Therefore, the dense shape of an arbitrary 2D face can be determined by the first two rows of the projection parameters $\mathbf{m} = [m_1, \cdots, m_8] \in \mathbb{R}^8$ and the shape basis

coefficients $\mathbf{p} = [p_{id}^1, ..., p_{id}^{199}, p_{exp}^1, ...p_{exp}^{29}] \in \mathbb{R}^{228}$. The learning of the dense 3D shape is turned into the learning of $\mathbf{m}$ and $\mathbf{p}$, which is much more manageable in term of the dimensionality.

3.2. CNN Architecture

Due to the success of deep learning in computer vision, we employ a convolutional neural network (CNN) to learn the nonlinear mapping function $f(\Theta)$ from the input image $\mathbf{I}$ to the corresponding projection parameters $\mathbf{m}$ and shape parameters $\mathbf{p}$. The estimated parameters can then be utilized to construct the dense 3D face shape.

Our CNN network has two branches, one for predicting $\mathbf{m}$ and another for $\mathbf{p}$, shown in Fig. 2. Two branches share the first three convolutional blocks. After the third block, we use two separate convolutional blocks to extract task-specific features, and two fully connected layers to transfer the features to the final output. Each convolutional block is a stack of two convolutional layers and one max pooling layer, and each conv/fc layer is followed by one batch normalization layer and one leaky ReLU layer.

In order to improve the CNN learning, we employ a loss function including multiple constraints: Parameter Constraint (PC) J_{pr} minimizes the difference between the estimated parameters and the ground truth parameters; Landmark Fitting Constraint (LFC) J_{lm} reduces the alignment error of 2D landmarks; Contour Fitting Constraint (CFC) J_c enforces the match between the contour of the estimated 3D shape and the contour pixels of the input image; and SIFT Pairing Constraint (SPC) J_s encourages that the SIFT feature point pairs of two face images to correspond to the same 3D vertices.

We define the overall loss function as,

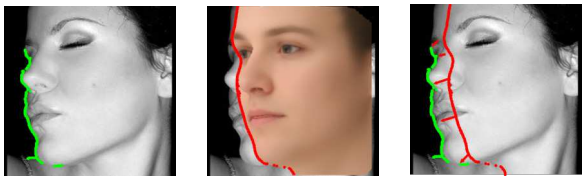
$$\arg\min_{\hat{\mathbf{m}},\hat{\mathbf{p}}} J = J_{pr} + \lambda_{lm} J_{lm} + \lambda_c J_c + \lambda_s J_s, \tag{7}$$

where the parameter constraint (PC) loss is defined as,

$$J_{pr} = \left\| \begin{bmatrix} \mathbf{m}^\intercal \\ \mathbf{p}^\intercal \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{m}}^\intercal \\ \hat{\mathbf{p}}^\intercal \end{bmatrix} \right\|^2. \tag{8}$$

Landmark Fitting Constraint (LFC) aims to minimize the difference between the estimated 2D landmarks and the ground truth 2D landmark labeling $\mathbf{U}_{lm} \in \mathbb{R}^{2\times N}$. Given 2D face images with a particular landmark labeling, we first manually mark the indexes of the 3D face vertices that are anatomically corresponding to these landmarks. The collection of these indexes is denoted as $\mathbf{i}_{lm}$. After the shape $\mathbf{A}$ is computed from Eqn. 4 with the estimated $\hat{\mathbf{m}}$ and $\hat{\mathbf{p}}$, the 3D landmarks can be extracted from $\mathbf{A}$ by $\mathbf{A}(:, \mathbf{i}_{lm})$. With projection of $\mathbf{A}(:, \mathbf{i}_{lm})$ to 2D plain, the LFC loss is defined as,

$$J_{lm} = \frac{1}{L} \cdot \|\mathbf{PrA}(:, \mathbf{i}_{lm}) - \mathbf{U}_{lm}\|_F^2, \tag{9}$$



(a) (b) (c)

Figure 3. The CFC fitting process. $\mathbf{A}_c$ is computed from estimated 3D face shape and $\mathbf{U}_c$ is computed from the off-the-shelf edge detector. Contour correspondence is obtained via Closest Pair Algorithm, and loss J_c is calculated based on Eqn. 10

where the subscript F represents the Frobenius Norm, and L is the number of pre-defined landmarks.

3.3. Contour Fitting Constraint (CFC)

Contour Fitting Constraint (CFC) aims to minimize the error between the projected outer contour (i.e., silhouette) of the dense 3D shape and the corresponding contour pixels in the input face image. The outer contour can be viewed as the boundary between the background and the 3D face while rendering 3D space onto a 2D plane. On databases such as AFLW where there is a lack of labeled landmarks on the silhouette due to self-occlusion, this constraint can be extremely helpful.

To utilize this contour fitting constraint, we need to follow these three steps: 1) Detect the true contour in the 2D face image; 2) Describe the contour vertices on the estimated 3D shape $\mathbf{A}$; and 3) Determine the correspondence between true contour and the estimated one, and back-propagate the fitting error.

First of all, we adopt an off-the-shelf edge detector, HED [29], to detect the contour on the face image, $\mathbf{U}_c \in \mathbb{R}^{2\times L}$. The HED has a high accuracy at detecting significant edges such as face contour in our case. Additionally, in certain datasets, such as 300W [23] and AFLW-LPFA [10], additional landmark labelings on the contours are available. Thus we can further refine the detected edges by only retaining edges that are within a narrow band determined by those contour landmarks, shown in Fig 3.a. This preprocessing step is done offline before the training starts.

In the second step, the contour on the estimated 3D shape $\mathbf{A}$ can be described as the set of boundary vertices $\mathbf{A}(:, \mathbf{i}_c) \in \mathbb{R}^{3\times L}$. $\mathbf{A}$ is computed from the estimated $\hat{\mathbf{m}}$ and $\hat{\mathbf{p}}$ parameters. By utilizing the Delaunay triangulation to represent shape $\mathbf{A}$, one edge of a triangle is defined as the boundary if the adjacent faces have a sign change in the z-values of the surface normals. This sign change indicates a change of visibility so that the edge can be considered as a boundary. The vertices associated with this edge are defined as boundary vertices, and their collection is denoted as $\mathbf{i}_c$. This process is shown in Fig 3.b.

In the third step, the point-to-point correspondences between $\mathbf{U}_c$ and $\mathbf{A}(:, \mathbf{i}_c)$ are needed in order to evaluate the

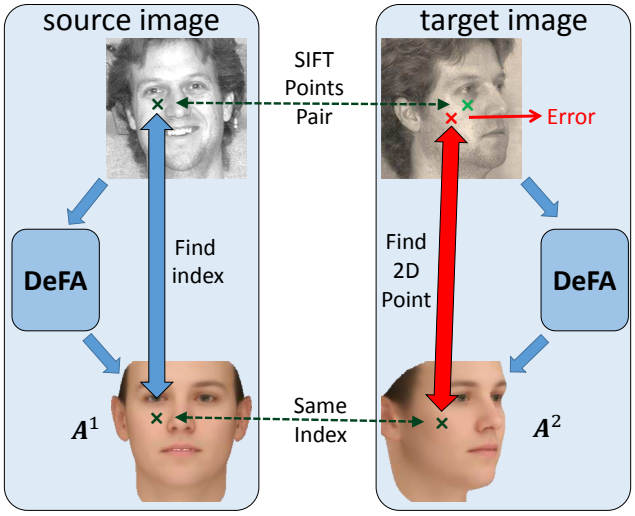


Figure 4. The illustration of the SIFT Matching process.

constraint. Given that we normally detect partial contour pixels on 2D images while the contour of 3D shape is typically complete, we match the contour pixel on the 2D images with closest point on 3D shape contour, and then calculate the minimun distance. The sum of all minimum distances is the error of CFC, as shown in the Eqn. 10. To make CFC loss differentiable, we rewrite Eqn. 10 to compute the vertex index of the closest contour projection point, i.e., $k^0 = \arg\min_{k \in \mathbf{i}_c} \|\mathbf{PrA}(:,k) - \mathbf{U}_c(:,j)\|^2$. Once k^0 is determined, the CFC loss will be differentiable, similar to Eqn. 9.

$$
\begin{aligned}
J_c &= \frac{1}{L}\sum_j \min_{k \in \mathbf{i}_c} \|\mathbf{PrA}(:,k) - \mathbf{U}_c(:,j)\|^2 \\
&= \frac{1}{L}\sum_j \|\mathbf{PrA}(:, \arg\min_{k \in \mathbf{i}_c} \|\mathbf{PrA}(:,k) - \mathbf{U}_c(:,j)\|^2) \\
&\qquad - \mathbf{U}_c(:,j)\|^2. \quad (10)
\end{aligned}
$$

Note that while $\mathbf{i}_c$ depends on the current estimation of $\{\mathbf{m}, \mathbf{p}\}$, for simplicity $\mathbf{i}_c$ is treated as constant when performing back-propagation w.r.t. $\{\mathbf{m}, \mathbf{p}\}$.

3.4. SIFT Pairing Constraint (SPC)

SIFT Pairing Constraint (SPC) regularizes the predictions of dense shape to be consistent on the significant facial points other than pre-defined landmarks, such as edges, wrinkles, and moles. The Scale-invariant feature transform (SIFT) descriptor is a classic local representation that is invariant to image scaling, noise, and illumination. It is widely used in many regression-based face alignment methods [30, 26] to extract the local information.

In our work, the SIFT descriptors are used to detect and represent the significant points within the face pair. The face pair can either come from the same people with different poses and expressions, or the same image with different augmentation, e.g., cropping, rotation and 3D augmentation, shown in Fig. 4. The more face pairs we have, the stronger this constraint is. Given a pair of faces i and j, we first detect and match SIFT points on two face images. The matched SIFT points are denoted as $\mathbf{U}_s^i$ and $\mathbf{U}_s^j \in \mathbb{R}^{2\times L_{ij}}$.

With a perfect dense face alignment, the matched SIFT points would overlay with exactly the same vertex in the estimated 3D face shapes, denoted as $\mathbf{A}^i$ and $\mathbf{A}^j$. In practices, to verify how likely this ideal world is true and leverage it as a constraint, we first find the 3D vertices $\mathbf{i}_s^i$ whose projections overlay with the 2D SIFT points, $\mathbf{U}_s^i$.

$$
\mathbf{i}_s^i = \arg\min_{i \in \{1,\dots,L_{ij}\}} \|\mathbf{A}^i\{\mathbf{i}_s^i\} - \mathbf{U}_s^i\|_F^2, \quad (11)
$$

Similarly, we find $\mathbf{j}_s^j$ based on $\mathbf{U}_s^j$. Now we define the SPC loss function as

$$
\begin{aligned}
&J_s(\hat{\mathbf{m}}^j, \hat{\mathbf{p}}^j, \hat{\mathbf{m}}^i, \hat{\mathbf{p}}^i) = \\
&\qquad \frac{1}{L_{ij}} \left(\|\mathbf{A}^i\{\mathbf{i}_s^j\} - \mathbf{U}_s^i\|_F^2 + \|\mathbf{A}^j\{\mathbf{i}_s^i\} - \mathbf{U}_s^j\|_F^2\right) \quad (12)
\end{aligned}
$$

where $\mathbf{A}^i$ is computed using $\{\mathbf{m}^i, \mathbf{p}^i\}$. As shown in Fig. 4, we map SIFT points from one face to the other and compute their distances w.r.t. the matched SIFT points on the other face. With the mapping from both images, we have two terms in the loss function of Eqn. 12.

4. Experimental Results

4.1. Datasets

We evaluate our proposed method on four benchmark datasets: AFLW-LFPA [9], AFLW2000-3D [41], 300W [23] and IJBA [12]. All datasets used in our training and testing phases are listed in Tab. 1.

AFLW-LFPA: AFLW contains around $25,000$ face images with yaw angles between $\pm 90°$, and each image is labeled with up to 21 visible landmarks. In [9], a subset of AFLW with a balanced distribution of the yaw angle is introduced as AFLW-LFPA. It consists of $3,901$ training images and $1,299$ testing images. Each image is labeled with 13 additional landmarks.

AFLW2000-3D: Prepared by [41], this dataset contains $2,000$ images with yaw angles between $\pm 90°$ of the AFLW dataset. Each image is labeled with 68 landmarks. Both this dataset and AFLW-LFPA are widely used for evaluating large-pose face alignment.

IJBA: IARPA Janus Benchmark A (IJB-A) [12] is an in-the-wild dataset containing 500 subjects and $25,795$ images with three landmark, two landmarks at eye centers and one on the nose. While this dataset is mainly used for face

Table 1. The list of face datasets used for training and testing.

Database	Landmark	Pose	Images
Training			
300W [23]	68	Near-frontal	3,148
300W-LP [41]	68	$[-90°, 90°]$	96,268
Caltech10k [2]	4	Near-frontal	10,524
AFLW-LFPA [9]	21	$[-90°, 90°]$	3,901
COFW [6]	29	Near-frontal	1,007
Testing			
AFLW-LFPA [9]	34	$[-90°, 90°]$	1,299
AFLW2000-3D [41]	68	$[-90°, 90°]$	2,000
300W [23]	68	Near-frontal	689
IJB-A [12]	3	$[-90°, 90°]$	25,795
LFW [14]	0	Near-frontal	34,356

Table 2. The list of datasets used in each training stage, and the employed constraints for each dataset: Parameter Constraint (PC); Landmark Fitting Constraint (LFC); SIFT Pairing Constraint (SPC); Contour Fitting Constraint (CFC).

Dataset	Stage 1	Stage 2	Stage 3
300W-LP [41]	PC	PC LFC	-
Caltech10k [2]	-	LFC	-
COFW [6]	-	LFC	-
AFLW-LFPA [9]	-	-	LFC SPC CFC
300W [23]	-	-	LFC SPC CFC

recognition, the large dataset size and the challenging variations (e.g., $\pm 90°$ yaw and images resolution) make it suitable for evaluating face alignment as well.

300W: 300W [23] integrates multiple databases with standard 68 landmark labels, including AFW [43], LFPW [3], HELEN [36], and IBUG [23]. This is the widely used database for evaluating near-frontal face alignment.

COFW [6]: This dataset includes near-frontal face images with occlusion. We use this dataset in training to make the model more robust to occlusion.

Caltech10k [2]: It contains four labeled landmarks: two on eye centers, one on the top of the nose and one mouth center. We do not use the mouth center landmark since there is no corresponding vertex on the 3D shape existing for it.

LFW [14]: Despite having no landmark labels, LFW can be used to evaluate how dense face alignment method performs via the corresponding SIFT points between two images of the same individual.

4.2. Experimental setup

Training sets and procedures : While utilizing multiple datasets is beneficial for learning an effective model, it also poses challenges to the training procedure. To make the training more manageable, we train our DeFA model in three stages, with the intention to gradually increase the datasets and employed constraints. At stage 1, we use 300W-LP to train our DeFA network with parameter constraint (PL). At stage 2, we additionally include samples from the Caltech10K [2], and COFW [6] to continue the training of our network with the additional landmark fitting constraint (LFC). At stage 3, we fine-tune the model with SPC and CFC constraints. For large-pose face alignment, we fine-tune the model with AFLW-LFPA training set. For near-frontal face alignment, we fine-tune the model with 300W training set. All samples at the third stage are augmented 20 times with up to $\pm 20°$ random in-plain rotation and 15% random noise on the center, width, and length of the initial bounding box. Tab. 2 shows the datasets and constraints that are used at each stage.

Implementation details: Our DeFA model is implemented with MatConvNet [27]. To train the network, we use 20, 10, and 10 epochs for stage 1 to 3. We set the initial global learning rate as $1e-3$, and reduce the learning rate by a factor of 10 when the training error approaches a plateau. The minibatch size is 32, weight decay is 0.005, and the leak factor for Leaky ReLU is 0.01. In stage 2, the regularization weights λ_{pr} for PC is 1 and λ_{lm} for LFC is 5; In stage 3, the regularization weights λ_{lm}, λ_s, λ_c for LFC, SPC and CFC are set as 5, 1 and 1, respectively.

Evaluation metrics: For performance evaluation and comparison, we use two metrics for normalizing the MSE. We follow the normalization method in [10] for large-pose faces, which normalizes the MSE by using the bounding box size. We term this metric as "NME-lp". For the near-frontal view datasets such as 300W, we use the inter-ocular distance for normalizing the MSE, termed as "NME-nf".

4.3. Experiments on Large-pose Datasets

To evaluate the algorithm on large-pose datasets, we use the AFLW-LFPA, AFLW2000-3D, and IJB-A datasets. The results are presented in Tab. 3, where the performance of the baseline methods is either reported from the published papers or by running the publicly available source code. For AFLW-LFPA, our method outperforms the best methods with a large margin of 17.8% improvement. For AFLW2000-3D, our method also shows a large improvement. Specifically, for images with yaw angle in $[60°, 90°]$, our method improves the performance by 28% (from 7.93 to 5.68). For the IJB-A dataset, even though we are able to only compare the accuracy for the three labeled landmarks, our method still reaches a higher accuracy. Note that the best performing baselines, 3DDFA and PAWF, share the similar overall approach in estimating $\mathbf{m}$ and $\mathbf{p}$, and also aim for large-pose face alignment. The consistently superior performance of our DeFA indicates that we have advanced the state of the art in large-pose face alignment.

Table 3. The benchmark comparison (NME-lp) on three large-pose face alignment datasets.

Baseline	CFSS [38]	PIFA [9]	CCL [40]	3DDFA [41]	PAWF [10]	**Ours**
AFLW-LFPA	6.75	6.52	5.81	-	4.72	**3.86**
AFLW2000-3D	-	-	-	5.42	-	**4.50**
IJB-A	-	-	-	-	6.76	**6.03**

Table 4. The benchmark comparison (NME-nf) on 300W dataset. The top two performances are in bold.

Method	Common set	Challenging set	Full set
RCPR [6]	6.18	17.26	7.58
SDM [30]	5.57	15.40	7.50
LBF [19]	4.95	11.98	6.32
CFSS [38]	**4.73**	9.98	**5.76**
RAR [28]	**4.12**	**8.35**	**4.94**
3DDFA [41]	6.15	10.59	7.01
3DDFA+SDM	5.53	9.56	6.31
DeFA	5.37	**9.38**	6.10

4.4. Experiments on Near-frontal Datasets

Even though the proposed method can handle large-pose alignment, to show its performance on the near-frontal datasets, we evaluate our method on the 300W dataset. The result of the state-of-the-art method on the both common and challenging sets are shown in Tab. 4. To find the corresponding landmarks on the cheek, we apply the landmark marching [42] algorithm to move contour landmarks from self-occluded location to the silhouette. Our method is the second best method on the challenging set. In general, the performance of our method is comparable to other methods that are designed for near-frontal datasets, especially under the following consideration. That is, most prior face alignment methods do not employ shape constraints such as 3DMM, which could be an advantage for near-frontal face alignment, but might be a disadvantage for large-pose face alignment. The only exception in Tab. 4 in 3DDFA [41], which attempted to overcome the shape constraint by using the additional SDM-based finetuning. It is a strong testimony of our model in that DeFA, without further finetuning, outperforms both 3DDFA and its fine tuned version with SDM.

4.5. Ablation Study

To analyze the effectiveness of the DeFA method, we design two studies to compare the influence of each part in the DeFA and the improvement by adding each dataset.

Tab. 5 shows the consistent improvement achieved by utilizing more datasets in different stages and constraints according to Tab. 2 on both testing datasets. It shows the advantage and the ability of our method in leveraging more datasets. The accuracy of our method on the AFLW2000-3D consistently improves by adding more datasets. For the AFLW-PIFA dataset, our method achieves 9.5% and 20% relative improvement by utilizing the datasets in the stage 2 and stage 3 over the first stage, respectively. If including the datasets from both the second and third stages, we can have 26% relative improvement and achieve NME of 3.86%. Comparing the second and third rows in Tab. 5 shows that the effectiveness of CFC and SPC is more than LFC. This is due to the utilization of more facial matching in the CFC and SPC.

Table 5. The NME-lp when utilizing more datasets.

Training Stages	AFLW2000-3D	AFLW-LFPA
stage1	6.23	5.24
stage1 + stage2	5.68	4.74
stage1 + stage3	4.85	4.15
stage1 + stage2 + stage3	**4.50**	**3.86**

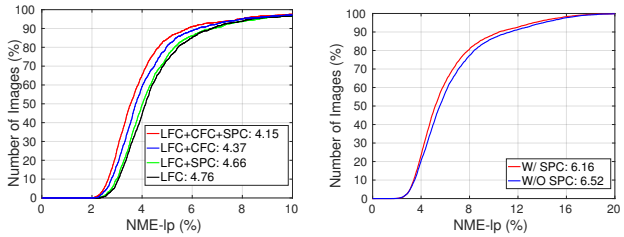


Figure 5. Left: The effect of constraints in enhancing the accuracy on the AFLW-LPFA testing set. The NME-lp of each setting is shown in legend. Right: The influence of the SIFT pairing constraint (SPC) in improving the performance for selected $5,000$ pairs from IJB-A.

The second study shows the performance improvement achieved by using the proposed constraints. We train models with different types of active constraints and test them on the AFLW-PIFA test set. Due to the time constraint, for this experiment, we did not apply 20 times augmentation of the third stage's dataset. We show the results in the left of Fig. 5. Comparing LFC+SPC and LFC+CFC performances shows that the CFC is more helpful than the SPC. The reason is that CFC is more helpful in correcting the pose of the face and leads to more landmark error reduction. Using all constraints achieves the best performance.

Finally, to evaluate the influence of using the SIFT pairing constraint (SPC), we use all of the three stages datasets to train our method. We select $5,000$ pairs of images from the IJB-A dataset and compute the NME-lp of all matched SIFT points according to Eqn. 12. The right plot in Fig. 5 illustrates the CED diagrams of NME-lp, for the trained models with and without the SIFT pairing constraint. This result shows that for the images with NME-lp between 5% and 15% the SPC is helpful.

Part of the reason DeFA works well is that it receives

Figure 6. The estimated dense 3D shape and their landmarks with visibility labels for different datasets. From top to bottom, the results on AFLW-LPFA, IJB-A and 300W datasets are shown in two rows each. The green landmark are visible and the red landmarks show the estimated locations for invisible landmarks. Our model can fit to diverse poses, resolutions, and expressions.

"dense" supervision. To show that, we take all matched SIFT points in the 300W-LP dataset, find their corresponding vertices, and plot the log of the number of SIFT points on each of the 3D face vertex. As shown in Fig. 7, SPC utilizes SIFT points to cover the whole 3D shape and the points in the highly textured areas are substantially used. We can expect that these SIFT constraints will act like anchors to guild the learning of the model fitting process.

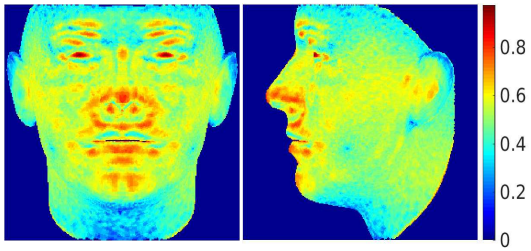


Figure 7. The log plot of the number of matched SIFT points in the 300W-LP training set. It shows that the SIFT constraints cover the whole face, especially the highly textured area.

5. Conclusion

We propose a large-pose face alignment method which estimates accurate 3D face shapes by utilizing a deep neural network. In addition to facial landmark fitting, we propose to align contours and the SIFT feature point pairs to extend the fitting beyond facial landmarks. Our method is able to leverage from utilizing multiple datasets with different landmark markups and numbers of landmarks. We achieve the state-of-the-art performance on three challenging large pose datasets and competitive performance on hard medium pose datasets.

References

[1] A. T. an Trãn, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. *arXiv preprint arXiv:1612.04904*, 2016. 1

[2] A. Angelova, Y. Abu-Mostafam, and P. Perona. Pruning training sets for learning of object categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 494–501. IEEE, 2005. 6

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940, 2013. 6

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 1, 2

[5] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, and S. Zafeiriou. 3d face morphable models" in-the-wild". 2017. 1

[6] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1513–1520, 2013. 6, 7

[7] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014. 3

[8] C. Ding, J. Choi, D. Tao, and L. S. Davis. Multi-directional multi-level dual-cross patterns for robust face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):518–531, 2016. 1

[9] A. Jourabloo and X. Liu. Pose-invariant 3D face alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3694–3702, 2015. 5, 6, 7

[10] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3D model fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4188–4196, 2016. 1, 2, 4, 6, 7

[11] A. Jourabloo and X. Liu. Pose-invariant face alignment via cnn-based dense 3D model fitting. April 2017. 2

[12] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, 2015. 5, 6

[13] J. Kossaifi, Y. Tzimiropoulos, and M. Pantic. Fast and exact newton and bidirectional fitting of active appearance models. *IEEE Transactions on Image Processing*, 2016. 1

[14] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016. 6

[15] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3D face reconstruction. In *European Conference on Computer Vision*, pages 545–560. Springer, 2016. 2

[16] J. McDonagh and G. Tzimiropoulos. Joint face detection and alignment with a deformable hough transform model. In *European Conference on Computer Vision*, pages 569–580. Springer, 2016. 2

[17] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *European conference on computer vision*, pages 504–513. Springer, 2008. 1

[18] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. IEEE, 2009. 3

[19] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014. 7

[20] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 986–993. IEEE, 2005. 2

[21] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *Proc. IEEE Computer Vision and Pattern Recognition*, Bostan, MA, June 2015. 1

[22] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016. 1

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013. 2, 4, 5, 6

[24] D. Sánchez-Escobedo, M. Castelán, and W. A. Smith. Statistical 3D face shape estimation from occluding contours. *Computer Vision and Image Understanding*, 142:111–124, 2016. 2

[25] B. M. Smith and L. Zhang. Collaborative facial landmark localization for transferring annotations across datasets. In *European Conference on Computer Vision*, pages 78–93. Springer, 2014. 2

[26] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2014. 5

[27] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015. 6

[28] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *European Conference on Computer Vision*, pages 57–72. Springer, 2016. 1, 7

[29] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015. 4

[30] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013. 5, 7

[31] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson. Face alignment assisted by head pose estimation. In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 130.1–130.13, 2015. 2

[32] H. Yang and I. Patras. Mirror, mirror on the wall, tell me, is the error small? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4685–4693, 2015. 2

[33] J. Zhang, M. Kan, S. Shan, and X. Chen. Leveraging datasets with varying annotations for face alignment via deep regression network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3801–3809, 2015. 2

[34] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):918–930, 2016. 2

[35] R. Zhao, Y. Wang, C. F. Benitez-Quiroz, Y. Liu, and A. M. Martinez. Fast and precise face alignment and 3d shape reconstruction from a single 2D image. In *European Conference on Computer Vision*, pages 590–603. Springer, 2016. 2

[36] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 386–391, 2013. 6

[37] K. Zhou, Y. Weng, and C. Cao. Method for real-time face animation based on single video camera, June 7 2016. US Patent 9,361,723. 1

[38] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4998–5006, 2015. 1, 7

[39] S. Zhu, C. Li, C. C. Loy, and X. Tang. Transferring landmark annotations for cross-dataset face alignment. *arXiv preprint arXiv:1409.0602*, 2014. 2

[40] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3409–3417, 2016. 7

[41] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Li. Face alignment across large poses: A 3D solution. In *Proc. IEEE Computer Vision and Pattern Recognition*, Las Vegas, NV, June 2016. 1, 2, 5, 6, 7

[42] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796, 2015. 7

[43] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012. 6