ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200426

Joint 3D Face Reconstruction and Dense Face Alignment via Deep Face Feature Alignment

Jian Zhou[1] and **Zhangjin Huang***[2]

Abstract. Reconstructing a 3D face from a single face image is a challenging problem in a wide range of applications. Due to the lack of a large number of 3D face datasets with ground truth, previous methods usually adopt weakly supervised learning methods. However, most methods only utilize pixel level information, which causes the convolutional neural network models to easily fall into local minima. This paper proposes a novel method of 3D face reconstruction and dense face alignment based on a single face image under unknown pose, expression and illumination. We not only consider the difference between the input face image and the rendered image at the pixel level, but also consider their difference in the deep feature space. First, a 3D face model is constructed from a single face image by using a parameterized face model. Then, the 3D face model is rendered to a 2D plane through a differentiable renderer. Next, the correspondences between the input face image and the rendered image in the pixel space and the deep feature space are established, respectively. Finally, our model is trained by back propagation. Experiments on AFLW2000-3D and AFLW-LFPA show that the proposed method outperforms existing approaches in both 3D face reconstruction and dense face alignment.

1 Introduction

3D face reconstruction and dense face alignment based on a single face image is a challenging task in the field of computer vision and computer graphics. It has a wide range of applications in face recognition, face animation, facial expression migration, and face alignment. Traditional two-dimensional face alignment methods often encounter great difficulties for large poses and occluded face images. However, 3D face reconstruction can well solve these difficulties. Due to the strong topological nature of the 3D face grid model, the tens of thousands of vertices of the 3D face grid model can all be regarded as face features points to be utilized by dense 3D face alignment. Traditional 3D face reconstruction methods are mainly based on the optimization algorithms [26, 14] . However, such methods are usually time-consuming due to the high optimization complexity and suffer from local optimal solution and bad initialization. After the Convolutional Neural Networks (CNN) emerged, CNN-based methods [48, 43] have achieved significant success in 3D face reconstruction and dense face alignment. These methods usually use CNNs to predict the coefficients of a 3D Morphable Model (3DMM), which significantly improves the quality and efficiency of 3D face reconstruction. The methods based on CNN usually need a large number of datasets, which are expensive to collect and even not achievable in some cases. Some recent methods [34, 48, 29] use synthetic data for training. There are some public synthetic face datasets such as 300W-LP [48]. However, the face images generated by synthetic methods usually have a certain gap with the in-the-wild images. They generally lack diversity in face expression, environment illumination and image background, which often leads to poor generalization performance of the trained CNN models.

In order to solve the problem of missing 3D face reconstruction datasets, some recent work [43, 45, 42, 15] used weakly supervised learning methods, which only requires two-dimensional face images and their corresponding two-dimensional face feature points. Using this method, the trained 3D face reconstruction model can perform 3D face reconstruction and dense 3D face alignment well. At present, it is easy to obtain face image dataset with two-dimensional face feature points, so a large number of training sets can be established to meet the needs of CNN. At the same time, these 2D face feature points can also provide valuable face information. The key to the current weakly-supervised 3D face reconstruction algorithm is to render the reconstructed 3D face to the pixel level using a differentiable renderer and compare the difference between the rendered image and the input image. For example, Tewari et al. [43, 42] use the difference between the rendered image and the input image in the pixel color to create a loss function. Genova et al. [15] and Tran et al. [45] used the face recognition network to establish the loss of the rendered image and the input image.

Through experiments, we have observed that using only pixel-level information may cause CNN models to fall into local suboptimal solutions. To alleviate this phenomenon, we devise a CNN model with weakly-supervised learning to regress the 3DMM coefficients for accurate 3D face reconstruction and dense 3D face alignment from a single face image. At the same time, we design a new loss function which considers the differences between the input face image and the rendered image in both the pixel space and the CNN depth feature space. Our main contributions are summarized as follows:

- Using the DFF module [21], we propose an end-to-end weakly-supervised CNN network structure to effectively establish the corresponding relationship of face feature points in deep feature space.
- We present a new loss function which takes into consideration the difference between the input face image and the rendered image in the deep feature space. It effectively prevents the CNN model from being trapped in a local minimum when using only the pixel information. In the meantime, it enforces intensive training for the feature points that had large loss values in the previous training epoch.

[1] University of Science and Technology of China, China, email: zj199501@mail.ustc.edu.cn
[2] University of Science and Technology of China, China, email: zhuang@ustc.edu.cn, *corresponding author

- Compared with other existing methods on AFLW2000-3D and AFLWLFPA datasets, our model achieves better performance in both 3D face reconstruction and dense face alignment.

2 Related Works

In this section, we will introduce the related work of the 3D face model, 3D face reconstruction and face alignment, respectively.

2.1 3D Face Model

The 3D face model is a widely used face grid model in 3D face reconstruction. Compared with the point cloud, the 3D face model can provide prior knowledge of face shape, expression and texture. It can convert a complex 3D model to a linear combination of a group of 3D basis models. Therefore, we can use a set of coefficient vectors to express the reconstructed 3D face model.

Cootes et al. [10] introduced an active appearance model (AAM) as a statistical deformation model of the two-dimensional shape and texture. AAM is a generic model that recovers the parameter description of an object through optimization during the fitting process. As an extension of the two-dimensional AAM, Blanz and Vetter [3] introduced a three-dimensional deformable model (3DMM), which used PCA to decompose geometric and texture information. This effectively reduces the size of the shape and texture space. Later, Gerig et al. [16] extended 3DMM by including expression in 3DMM. Booth et al. [4] give 9,663 large-scale face models (LSFM) with different facial features. The model contains statistics from a large number of different groups of people. Tran et al. [44] proposed a nonlinear 3D face deformable model. They directly trained a nonlinear 3D face deformable model from a large number of unconstrained face images without collecting 3D face scanning images. We use 3DMM as our 3D face model in this work.

2.2 3D Face Reconstruction

Monocular 3D face reconstruction algorithms are usually divided into two categories: optimization-based algorithm and regression-based algorithm. Optimization-based algorithm usually establishes energy function to describe some natural processes of image. Many methods use shape-from-shading [26, 37] or optical flow [14] to simulate image information. The main disadvantages of these algorithms are high computational complexity and slow reconstruction speed. At the same time, the optimization algorithm is very sensitive to initialization and needs accurate two-dimensional feature point detection [23, 46].

In recent years, regression-based methods [49, 20, 48, 43] have been fully developed. Especially with the emergence of the convolutional neural network (CNN), many methods are based on convolutional neural network [48, 43, 19, 34, 45]. Convolutional neural network usually needs to input a large amount of data for training. However, 3D face datasets are usually small, so current methods are mainly divided into using synthetic data [34, 48, 18, 29] and using weakly supervised [42, 15, 12, 44, 43]. However, there is usually a big difference between the synthetic face data and the real face image, which will lead to a significant decline in the generalization ability of the model, and the test results on the real face image are not good. Therefore, most of the current methods adopt the weakly supervised learning method, which does not need the label of face pictures.

Richardson et al. [34] used 3D deformable models to generate images of different shapes, expressions, and textures and rendered them into two-dimensional images. In this way, they obtained images of real 3D face model labels for network training. Tewari et al. [43] trained an auto-encoder to regress shapes, expressions, textures, postures, and illumination. The regression parameters are used to generate the 3D face model, and then the differentiable renderer is used to render the 3D face model to the 2D plane. The loss function is established by comparing the input face image with the rendered face image. This method does not need the label of the 3D face model corresponding to the 2D image, and the result is obviously better than the model trained with synthetic data. Tewari et al. [42] further extended 3DMM, which effectively improved the detailed information of human faces, especially the texture information of colored people. The model also significantly improves information such as facial hair. Richardson et al. [35] incorporated shape from shading into the learning process to learn more detailed information. Jackson et al. [19] trained a convolutional neural network to reconstruct the voxel representation of three-dimensional facial geometry directly from a two-dimensional image. This is a model-free approach that does not require a 3D face deformable model. Feng et al. [13] trained the convolutional neural network to regress UV position map from a single two-dimensional image and obtained the corresponding three-dimensional facial structure. The method also does not depend on any prior face model. Deng et al. [12] used multiple pictures of the same person for training. Multiple pictures can complement information from different angles and have the advantages of anti-occlusion. Shi et al. [40] trained a convolutional neural network to create a matching cartoon image from a face image in the game. Chang et al. [8] used three deep convolutional nerual networks (CNN) to estimate each of 3D face shape, viewpoint, and expression from a single, unconstrained photo separately. Jiang et al. [22] used a coarse-to-fine optimization strategy to reconstruct 3D faces from unconstrained 2D images.

Most of these methods only consider the information at the pixel level of the face image. We find that using only pixel information may cause the convolutional neural network model to fall into a local suboptimal solution. To solve this problem, we extract facial features from the depth feature space with DFF modules [21] and compare the difference between the input face image and the reconstructed face image in the feature space.

2.3 Face Alignment

Face alignment has attracted a lot of attention for a long time. The traditional two-dimensional face alignment method is mainly to locate a set of sparse face key points, such as AAM [10] and CLM [1]. In recent years, with the continuous development of deep learning, the convolutional neural network (CNN) [28, 33] based method has achieved the most advanced performance in two-dimensional face alignment. However, two-dimensional face alignment has certain limitations. It can only detect the visible feature points on the human face. Therefore, when the face posture is large or there is occlusion, some face feature points become invisible, which cannot be processed by these methods.

Recently, people have started to study three-dimensional face alignment. The main method is to use 3D deformable model (3DMM) to fit [48, 32, 17] or to match 2D face image with 3D face template [38, 11]. The 3D reconstruction method based on the 3D model can realize the 2D face alignment task by selecting the x and y coordinates in the 3D reconstruction model. [48, 47] all use specific

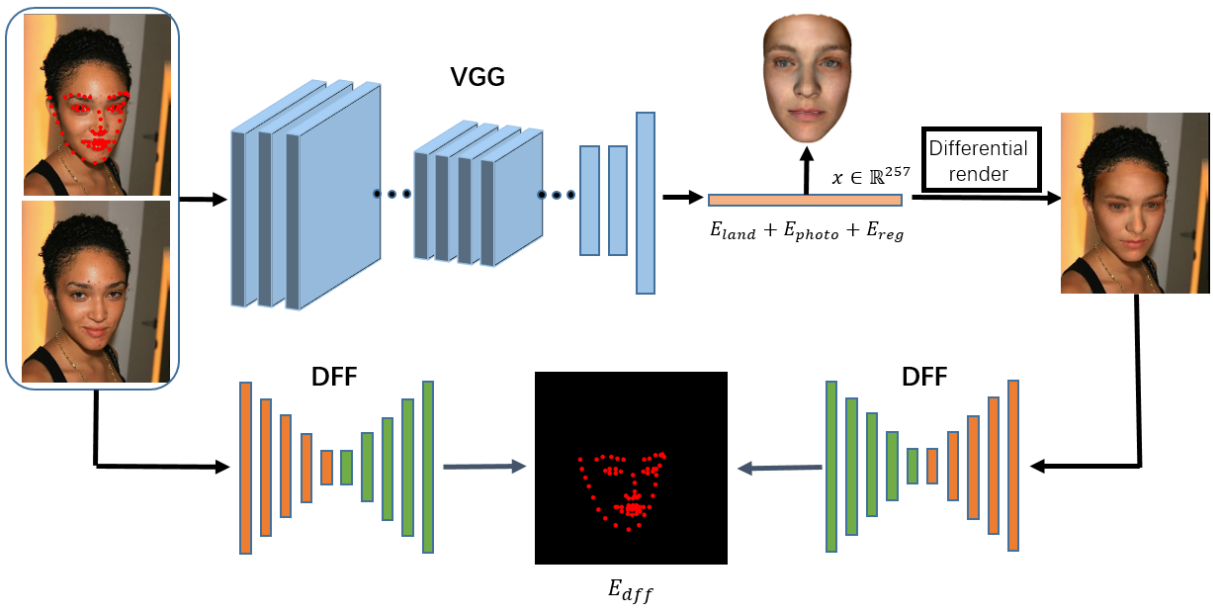

Figure 1. This is the algorithm flow chat we proposed. Our purpose is to train a convolutional neural network regression model. The regression model inputs a face image and its corresponding 2D face feature point information, and returns the 3D deformation model (3DMM) coefficients, camera parameters and spherical harmonic coefficients of the face through VGG-16 [41]. The corresponding 3D face model is reconstructed by 3D deformation model coefficients and spherical harmonic illumination. The reconstructed 3D face model is then rendered onto a 2D plane using a fully perspective projection through a differentiable renderer. Then the face image and the rendered face image are fed into the DFF network [21] (where red is the convolution layer and green is the deconvolution layer) to evaluate their difference in the deep feature space . Finally, the overall network is trained through backpropagation.

methods to fit the 3D deformable model (3DMM) to complete the task of face alignment. [5, 6] uses a deep neural network to directly predict the thermal map to obtain 3D face feature points and achieves the most advanced performance. Some methods [24, 7, 20] estimate the coefficients of the 3D deformable model (3DMM) and then project the estimated 3D face feature points into two-dimensional space, which can significantly improve the efficiency. Liu et al. [30] uses multiple constraints to train the CNN model and estimate the coefficient of 3DMM, which can achieve dense 3D estimation. [47] uses a deep convolutional neural network to learn the relationship between two-dimensional face images and three-dimensional templates, considering only the visible region.

These methods usually only establish their loss functions at the feature points, ignoring the information of the pixel-level and the depth feature space. Our method synthesizes the information of feature points, pixel-level and depth feature space. Experiments show that our method achieves better results.

3 Proposed Method

In this section, we introduce the proposed 3D face reconstruction and dense face alignment algorithm model. As shown in Fig. 1, our model takes as input one face image and learns its 3DMM, illumination and face pose parameters end-to-end. The 3D face model is reconstructed by 3DMM parameters and illumination coefficients, and the reconstructed 3D face model is projected onto a 2D plane by a differentiable renderer using a full perspective projection to establish the loss function of the pixel level and the deep feature level. Finally, the whole network is trained through backpropagation.

3.1 Background Knowledge

3.1.1 Face Model

We use a parametric 3D face geometry model $\mathbf{S} = \{\mathbf{s}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ and a parametric 3D face texture model $\mathbf{T} = \{\mathbf{t}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$:

$$\begin{aligned} \mathbf{S} &= \mathbf{S}(\boldsymbol{\alpha}, \boldsymbol{\delta}) = \bar{\mathbf{S}} + \mathbf{E}_{\text{shape}}\boldsymbol{\alpha} + \mathbf{E}_{\text{exp}}\boldsymbol{\delta} \\ \mathbf{T} &= \mathbf{T}(\boldsymbol{\beta}) = \bar{\mathbf{T}} + \mathbf{E}_{\text{tex}}\boldsymbol{\beta} \end{aligned} \tag{1}$$

where N = 35K, $\bar{\mathbf{S}}$ is the mean shape, $\bar{\mathbf{T}}$ is the mean texture. $\mathbf{E}_{\text{shape}} \in \mathbb{R}^{3N\times199}$, $\mathbf{E}_{\text{tex}} \in \mathbb{R}^{3N\times199}$and $\mathbf{E}_{\text{exp}} \in \mathbb{R}^{3N\times64}$are the basis of PCA for face shape, texture and expression, respectively. $\boldsymbol{\alpha} \in \mathbb{R}^{199}$, $\boldsymbol{\beta} \in \mathbb{R}^{199}$ and $\boldsymbol{\delta} \in \mathbb{R}^{64}$ are the coefficient of shape, texture and expression corresponding to the 3D face model. In this paper, $\bar{\mathbf{S}}$, $\bar{\mathbf{T}}$, $\mathbf{E}_{\text{shape}}$, $\mathbf{E}_{\text{tex}}$ are built from 2009 Basel Face Model [3], $\mathbf{E}_{\text{exp}}$ is built from FaceWarehourse [9].

3.1.2 Camera Model

The camera model is used to transform the 3D face model from the 3D space to the 2D plane. Same as [15], we use a full-perspective projection model. The 3D face pose is represented by the rotation $\mathbf{R} \in SO(3)$ and the translation $\mathbf{m} \in \mathbb{R}^3$:

$$\mathbf{q}_i = \Pi(\mathbf{R}\mathbf{p}_i + \mathbf{m}) \tag{2}$$

where $\mathbf{p}_i$ is the coordinate of the vertex in the world space, $\mathbf{q}_i$ is the coordinate of the vertex in the image plane, $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the full perspective projection model that changes the camera coordinates system to the screen coordinates system.

Figure 2. A schematic diagram of the face feature points. Among them, red is 52 fixed feature points and green is 16 contour feature points.

3.1.3 Illumination Model

We assume that the illumination is low frequency and approximate as a Lambert surface. Based on these two assumptions, we use the spherical harmonics [39] to represent the illumination. Vertex color $C(\mathbf{t}_i, \mathbf{n}_i, \boldsymbol{\gamma})$ is calculated by surface vertex texture $\mathbf{t}_i \in \mathbb{R}^3$, vertex normal vector $\mathbf{n}_i \in \mathbb{R}^3$ and the illumination coefficient $\boldsymbol{\gamma} \in \mathbb{R}^{27}$:

$$C(\mathbf{t}_i, \mathbf{n}_i, \boldsymbol{\gamma}) = \mathbf{t}_i \cdot \sum_{b=1}^{B^2} \mathbf{r}_b \mathbf{H}_b(\mathbf{n}_i) \tag{3}$$

where $\mathbf{H}_b : \mathbb{R}^3 \to \mathbb{R}$ is the spherical basis function, $\boldsymbol{\gamma} = \{\mathbf{r}_b \in \mathbb{R}^3 | 1 \le b \le B^2\}$ is the corresponding light coefficient. We use the first three orders ($B = 3$) in our model.

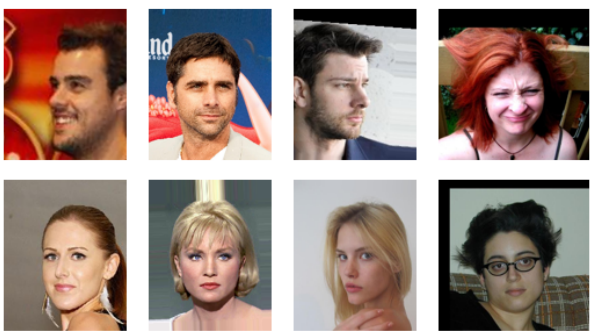

Figure 3. Some in-the-wild face images in the training set.

3.2 Network Architecture

The model of 3D face reconstruction and dense face alignment proposed in this paper mainly consists of two modules. One is a regression module based on convolutional neural network, which inputs a face image into the convolutional neural network, and returns the 3DMM coefficients, spherical harmonic parameters and camera model parameters of the corresponding 3D face. The other is a deconvolution module that extracts the features of face images on the deep convolutional layer.

We use VGG-16 [41] as the regression module. The input of regression module is a 224×224 RGB face image. The output 3D face parameter $\mathbf{x} \in \mathbb{R}^{257}$ includes the 3DMM shape parameter $\boldsymbol{\alpha} \in \mathbb{R}^{80}$, 3DMM texture parameter $\boldsymbol{\beta} \in \mathbb{R}^{80}$, 3DMM expression parameter $\boldsymbol{\delta} \in \mathbb{R}^{64}$, camera rotation $\mathbf{R} \in SO(3)$, camera translation $\mathbf{m} \in \mathbb{R}^3$ and spherical harmonics parameter $\boldsymbol{\gamma} \in \mathbb{R}^{27}$:

$$\mathbf{x} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \mathbf{R}, \mathbf{m}, \boldsymbol{\gamma}) \tag{4}$$

This module contains 13 convolutional layers, and each of them is followed by a ReLU activation layer. The 2nd, 4th, 7th, 10th, and 13th convolutional layers are followed by a maximum pooling layer. The convolutional layer is followed by three fully connected layers with sizes of 4096, 4096 and 257, respectively.

We adopt the DFF model [21] as the deconvolution module. The structure of the DFF model [21] is similar to U-Net [36]. The convolutional and deconvolution layers are applied symmetrically and connecting to a shallow feature map. DFF model inputs a 224×224 size RGB face image, and the output feature map size is 224×224 as well. Each pixel corresponds to a 32-dimensional feature vector.

3.3 Loss Function

The loss functions of previous approaches usually only consider face feature points or pixel-level information, which makes the CNN models easily fall into local optimal solutions. Our loss function takes into account information on face feature points, pixel spaces, and depth feature spaces together. The new loss function consists of five terms as follows:

$$\begin{aligned}\mathcal{L}_{\text{loss}}(\boldsymbol{x}) = \omega_{\text{land}}\mathcal{L}_{\text{land}}(\boldsymbol{x}) + \omega_{\text{land\_error}}\mathcal{L}_{\text{land\_error}}(\boldsymbol{x}) \\ + \omega_{\text{photo}}\mathcal{L}_{\text{photo}}(\boldsymbol{x}) + \omega_{\text{dff}}\mathcal{L}_{\text{dff}}(\boldsymbol{x}) + \omega_{\text{reg}}\mathcal{L}_{\text{reg}}(\boldsymbol{x})\end{aligned} \tag{5}$$

where $\mathcal{L}_{\text{land}}(\boldsymbol{x})$ and $\mathcal{L}_{\text{land\_error}}(\boldsymbol{x})$ are the landmark loss terms for feature point alignment and feature point enhanced training respectively, $\mathcal{L}_{\text{photo}}(\boldsymbol{x})$ is the photometric loss term for the difference between pixels, $\mathcal{L}_{\text{dff}}(\boldsymbol{x})$ is the depth feature alignment loss term established by the DFF model in the deep feature layer and $\mathcal{L}_{\text{reg}}(\boldsymbol{x})$ is a regularization term. To balance the contributions for each part, we set the weights to $\omega_{\text{land}} = 400$, $\omega_{\text{land\_error}} = 2000$, $\omega_{\text{photo}} = 100$, $\omega_{\text{dff}} = 10^{-6}$ and $\omega_{\text{reg}} = 1$.

3.3.1 Landmark Loss

We train the neural network with the feature points of the 2D face image as a weakly supervised information. The current advanced face feature point detection algorithm [6] is used to detect 68 key points of the face image in the training set. The loss term $\mathcal{L}_{\text{land}}$ is as follows:

$$\mathcal{L}_{\text{land}}(\boldsymbol{x}) = \sum_{i=1}^{68} w_i \times \|\mathbf{v}_{k_i} - \hat{\mathbf{v}}_i\|_2^2 \tag{6}$$

where w_i is the weight corresponding to the feature points. The weight of the fixed 52 feature points is 1, and the weight of the 16 contour feature points is 0.5. The selection of the fixed feature points and contour feature points is shown in Fig. 2. $\hat{\mathbf{v}}_i \in \mathbb{R}^2$ is the real label of the 2D feature points of the face. $k_i \in \{1, \ldots, N\}$ is the corresponding model vertex index, and $\mathbf{v}_{k_i} \in \mathbb{R}^2$ is the coordinates of the reconstructed 3D face model projected on the pixel plane.

To strengthen the training of those feature points with relatively large errors, after the fifth epoch, we add the loss term $\mathcal{L}_{\text{land\_error}}$ as follows:

$$\mathcal{L}_{\text{land\_error}}(\boldsymbol{x}) = \sum_{i=1}^{52} e_i \times \|\mathbf{v}_{k_i} - \hat{\mathbf{v}}_i\|_2^2 \tag{7}$$

where e_i is the average error of the fixed 52 feature points in the previous epoch training.

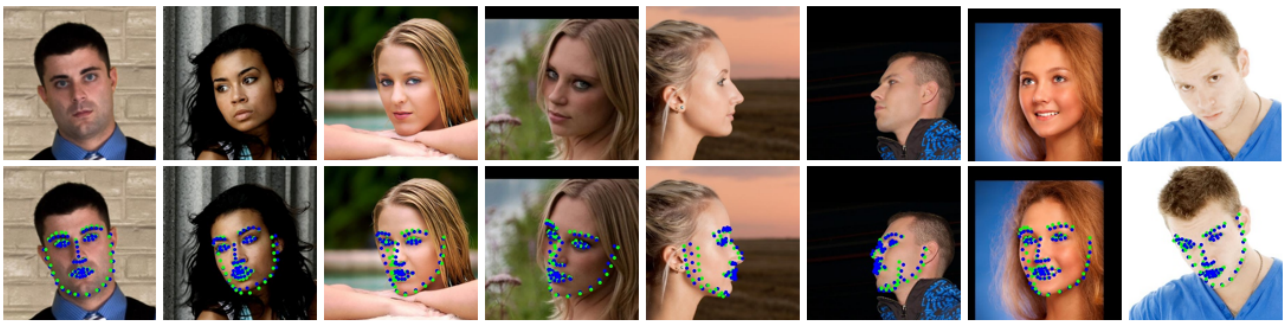

Figure 4. This is the face alignment effect of our method on ALFW2000-3D, in which the first line input picture, the second line face alignment effect, The green feature points are the predicted results of our method, and the blue feature points are the real labels on the test set.

3.3.2 *Photometric Loss*

The goal of the photometric loss term $\mathcal{L}_{\text{photo}}$ is to make the rendered image similar to the input image. The reconstructed 3D face model is rendered on the pixel space and aligned with the input monocular face image. To render a 3D face model into a 2D plane, we use a differentiable renderer [15] . We match the rendered image with the input monocular face image and compare their similarity in pixel space. The loss term $\mathcal{L}_{\text{photo}}$ is as follows:

$$\mathcal{L}_{\text{photo}}(x) = \frac{1}{N}\sum_{i\in\boldsymbol{\mathcal{V}}} \|\mathbf{I}_i - \mathbf{I}'_i\|_2 \tag{8}$$

where $\boldsymbol{\mathcal{V}}$ is the set of pixel points for all projected face regions on the pixel plane. N is the number of pixels in $\boldsymbol{\mathcal{V}}$. $\mathbf{I}$ is the input monocular face image, and $\mathbf{I}'$ is the image obtained by rendering the 3D face model into the pixel space.

3.3.3 *Depth Feature Alignment Loss*

To measure the difference between the input monocular face image and the rendered 3D face image in the CNN deep feature maps, we introduce a new depth feature alignment loss term $\mathcal{L}_{\text{dff}}$ deduced by the DFF (Deep Face Feature) model [21].

The DFF model is an end-to-end method based on deep convolutional neural networks that extracts a feature vector considering global information for each face image pixel. DFF [21] uses a deep convolutional neural network to map each pixel of a face image to a high-dimensional vector and then normalizes the high-dimensional vector to a unit length. In order to effectively represent and distinguish facial features, the normalized DFF descriptor preserves the metric structure of the 3D face surface. Especially for two pixels of the same split region, even if they come from different images, different poses, different scales, different lighting conditions, their normalized DFF descriptors should also be close to each other. On the other hand, for corresponding pixels of different face parts, even if the surrounding image area have similar appearances, there should be sufficient distance between the normalized DFF descriptors.

After getting the predicted 3D face, we render the 3D face to the pixel space. The resulting image is denoted as $\mathbf{I}'$ and the input monocular face image is denoted as $\mathbf{I}$. We input $\mathbf{I}$ and $\mathbf{I}'$ into the DFF model [21] (as shown in the Fig. 1), and get the same size map as the original image $\mathbf{D}$ and $\mathbf{D}'$. The input image size is the same as $224 \times 224 \times 3$, and the output feature size is the same as $224 \times 224 \times 32$. The loss term $\mathcal{L}_{\text{dff}}$ is as follows:

$$\mathcal{L}_{\text{dff}} = \sum_{i=1}^{68} f_i \times \|\mathbf{d}_i - \hat{\mathbf{d}}_i\|_2^2 \tag{9}$$

where $\mathbf{d}_i \in \mathbb{R}^{32}$ and $\hat{\mathbf{d}}_i \in \mathbb{R}^{32}$ are the corresponding feature vectors for the real feature points in $\mathbf{D}$ and $\mathbf{D}'$ respectively. $f_i \in \{0, 1\}$ is the visible weight for facial feature points. If the feature point is visible then $f_i = 1$, otherwise $f_i = 0$. The visibility of feature points is determined by the normal vector of the corresponding point on the 3D face.

3.3.4 *Regularization*

In the process of training, to prevent the 3D face model deformation, like [43], we add the loss of regularization term to the regression 3DMM coefficients, which can make the predicted 3DMM coefficients satisfy a prior distribution. The loss term $\mathcal{L}_{\text{reg}}$ is as follows:

$$\mathcal{L}_{\text{reg}} = \omega_\alpha\|\boldsymbol{\alpha}\|^2 + \omega_\beta\|\boldsymbol{\beta}\|^2 + \omega_\delta\|\boldsymbol{\delta}\|^2 \tag{10}$$

To balance the contributions for each part, we choose $\omega_\alpha = 2\times10^{-5}$, $\omega_\beta = 2 \times 10^{-2}$ and $\omega_\delta = 4 \times 10^{-4}$.

4 Results

In this section, we evaluate the performance of our proposed method on the tasks of 3D face alignment and 3D face reconstruction. We first introduce the training details and the test datasets used in our experiments. Then we compare our results with other methods in both quantitative and qualitative way.

4.1 Training details

In order to train our convolutional neural network, we select about 250K face images from CelebA [31] and 300W-LP [48] (as shown in Fig. 3). Data enhancements are made to these images, including data flipping, random rotation of the image $-30^\circ \sim 30^\circ$, and the image color channel RGB randomly multiplied by $0.7 \sim 1.3$. The input image size is $224 \times 224 \times 3$. We use Adam optimizer [27] for the CNN regressor with a learning rate beginning at 1×10^{-5} and decays to 1×10^{-6} after five epoch. The batchsize is 16. We use the pre-trained weights of the DFF model [21], which are fixed during the training phase.

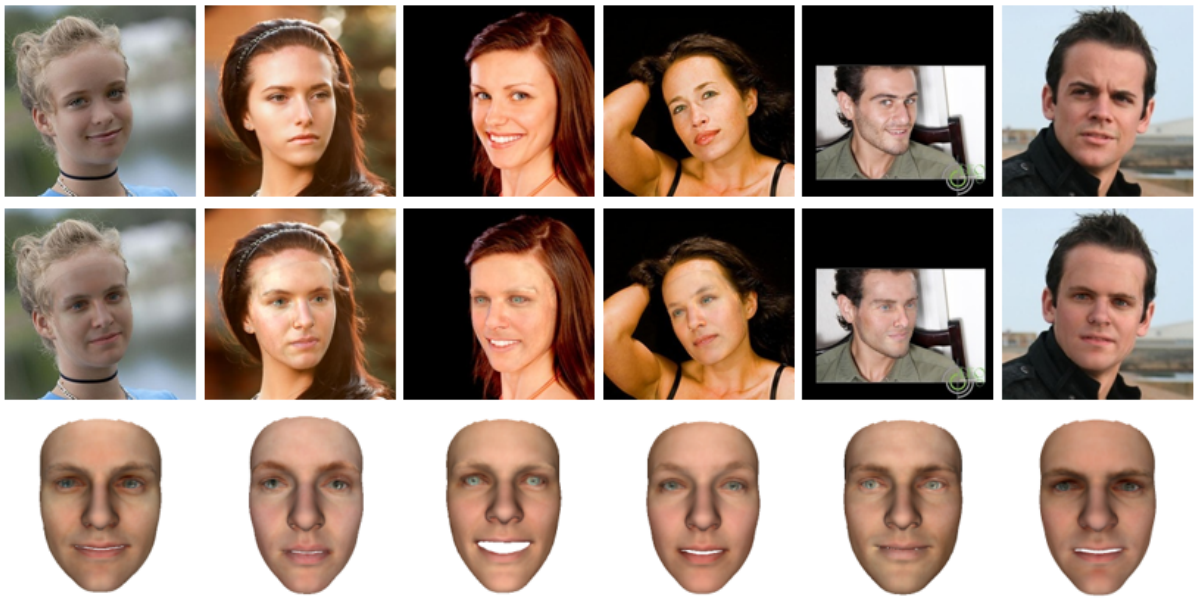

Figure 5. This picture shows the effect of our method on 3D face on ALFW2000-3D. The first column is the input picture, the second column is the effect of rendering the reconstructed 3D face back to the original image, and the third column is the 3D face model.

4.2 Test datsets

AFLW2000-3D [48] is built by selecting the first 2000 images in AFLW. Each face image has a corresponding 3DMM coefficient and positions of 68 3D face feature points. We use this dataset to evaluate our method on both 3D face reconstruction and dense face alignment.

AFLW-LFPA [25] is another extension of AFLW dataset. The authors construct this dataset by picking images from AFLW according to the poses. It contains 1299 test images with a balanced distribution of the yaw angle. This dataset is evaluated on the task of dense face alignment by using 34 visible landmarks as the ground truth.

4.3 Dense face alignment

We first compare the qualitative results of our method and corresponding ground truths in Fig. 4. In order to compare our method, we use the normalized mean error (NME) [48] as an indicator to evaluate the performance of our method. The normalized mean error is normalized according to the size of the face bounding box. The formula is as follows:

$$\mathbf{NME} = \frac{1}{N}\sum_{k=1}^{N}\frac{\|\mathbf{x}_k - \mathbf{y}_k\|_2}{d} \tag{11}$$

where N is the number of points, and d is the square root of the product of the width and height of the ground truth bounding box of the face, calculated as $d = \sqrt{w_{\text{bbox}} * h_{\text{bbox}}}$. $\mathbf{x_k} \in \mathbb{R}^2$ and $\mathbf{y_k} \in \mathbb{R}^2$ are the predicted point coordinates and the labels on the test set, respectively.

We use 68 sparse feature points to measure, and the 68 sparse feature points can be seen as a sample of dense face feature points. We compare the normalized mean errors in AFLW2000-3D [48] and AFLW-LFPA [25] with other methods. Since the distribution of the yaw angle of the face in the AFLW2000-3D is not uniform, in order to make the absolute value of the yaw angle evenly distributed on $0^\circ \sim 30^\circ$, $30^\circ \sim 60^\circ$ and $60^\circ \sim 90^\circ$, we randomly select 696 face images from AFLW2000-3D as well as [48]. So that the proportion of face images from all angles is $1:1:1$.

Tab. 1 shows the comparison of our method and other methods. The data information of related methods is directly obtained from related papers. As can be seen from this table, our method exhibits a lower normalized mean error than the other methods on both AFLW2000-3D [48] and AFLW-LFPA [25] datasets. At the same time, our method shows good robustness in different face angles, and it can well perform dense face alignment on large-scale face images.

4.4 3D face reconstruction

We validate the effect of 3D face reconstruction on AFLW2000-3D [48] as shown in Fig. 5. We compare our method to 3DDFA [48] and DeFA [30] on the AFLW2000-3D. We first use the Iterative Closest Point (ICP) to align the predicted 3D face with the ground truth 3D point cloud. We then calculate the mean square error (MSE) between the point clouds by the face bounding box size. Tab. 2 shows the comparison between our method and other methods. It can be seen that our approach shows better results.

5 Conclusion

In this paper, we propose an end-to-end CNN model with weakly-supervised learning for joint 3D face reconstruction and dense face alignment from a single face image. A new loss function is devised to comprehensively consider the information of face feature points, pixel spaces, and CNN deep feature spaces together, which alleviates the problem of local minima. The experimental results on AFLW2000-3D and AFLW-LFPA datasets confirm that our method is superior to previous methods in both 3D face reconstruction and dense face alignment. In future work, we will improve the 3DMM model to reconstruct more detailed face information, such as beard, mole and wrinkle, etc.

Table 1. Performance comparison on AFLW2000-3D (68 2D landmarks) and AFLW-LFPA (34 2D visible landmarks). The NME (%) for faces with different yaw angles are reported. The best results on each dataset are shown in bold, the lower is the better. "-" indicates the corresponding result is unavailable.

Method	AFLW2000-3D				AFLW-LFPA
	$0^\circ \sim 30^\circ$	$30^\circ \sim 60^\circ$	$60^\circ \sim 90^\circ$	mean	mean
SDM [32]	3.67	4.94	9.67	6.12	-
3DDFA [48]	3.78	4.54	7.93	5.42	-
3DDFA + SDM [48]	3.43	4.24	7.17	5.42	-
PAWF [25]	-	-	-	-	4.72
Yu et al. [47]	3.62	6.06	9.56	-	-
3DSTN [2]	3.15	4.33	5.98	4.49	-
DeFA [30]	-	-	-	4.50	3.86
Chang et al. [8]	3.11	3.84	6.60	4.52	-
Tran et al. [44]	-	-	-	4.12	-
Ours	**2.84**	**3.70**	**4.69**	**3.75**	**3.24**

Table 2. Performance comparison on AFLW2000-3D. The NME (%) for faces are reported. The lower is the better.

	3DDFA [48]	DeFA [30]	**Ours**
NME	2.43	4.33	**2.19**

ACKNOWLEDGEMENTS

We would like to thank Dr. Yudong Guo for the helpful discussions and comments. This work was supported in part by the National Key R&D Program of China (No. 2018YFC1504104), the National Natural Science Foundation of China (Nos. 71991464 / 71991460, and 61877056), and the Fundamental Research Funds for the Central Universities of China (No. WK6030000109).

REFERENCES

[1] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic, 'Robust discriminative response map fitting with constrained local models', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3444–3451, (2013).

[2] Chandrasekhar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides, 'Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3980–3989, (2017).

[3] Volker Blanz and Thomas Vetter, 'A morphable model for the synthesis of 3d faces', in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194, (1999).

[4] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway, 'A 3d morphable model learnt from 10,000 faces', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5543–5552, (2016).

[5] Adrian Bulat and Georgios Tzimiropoulos, 'Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge', in *European Conference on Computer Vision*, pp. 616–624. Springer, (2016).

[6] Adrian Bulat and Georgios Tzimiropoulos, 'How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1021–1030, (2017).

[7] Chen Cao, Qiming Hou, and Kun Zhou, 'Displaced dynamic expression regression for real-time facial tracking and animation', *international conference on computer graphics and interactive techniques*, **33**(4), 43, (2014).

[8] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard Medioni, 'Deep, landmark-free fame: Face alignment, modeling, and expression estimation', *International Journal of Computer Vision*, **127**(6-7), 930–956, (2019).

[9] Cao Chen, Weng Yanlin, Zhou Shun, Tong Yiying, and Zhou Kun, 'Facewarehouse: a 3d facial expression database for visual computing', *IEEE Transactions on Visualization & Computer Graphics*, **20**(3), 413–425, (2014).

[10] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor, 'Active appearance models', *IEEE Transactions on pattern analysis and machine intelligence*, **23**(6), 681–685, (2001).

[11] Flávio H de Bittencourt Zavan, Antônio CP Nascimento, Luan P e Silva, Olga RP Bellon, and Luciano Silva, '3d face alignment in the wild: A landmark-free, nose-based approach', in *European Conference on Computer Vision*, pp. 581–589. Springer, (2016).

[12] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong, 'Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (2019).

[13] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou, 'Joint 3d face reconstruction and dense alignment with position map regression network', in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 534–551, (2018).

[14] Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec, 'Driving high-resolution facial scans with video performance capture', *ACM Transactions on Graphics (TOG)*, **34**(1), 1–14, (2014).

[15] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman, 'Unsupervised training for 3d morphable model regression', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8377–8386, (2018).

[16] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter, 'Morphable face models-an open framework', in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 75–82. IEEE, (2018).

[17] Chao Gou, Yue Wu, Fei-Yue Wang, and Qiang Ji, 'Shape augmented regression for 3d face alignment', in *European Conference on Computer Vision*, pp. 604–615. Springer, (2016).

[18] Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng, 'Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41**(6), 1294–1307, (2019).

[19] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos, 'Large pose 3d face reconstruction from a single image via direct volumetric cnn regression', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1031–1039, (2017).

[20] László A Jeni, Jeffrey F Cohn, and Takeo Kanade, 'Dense 3d face alignment from 2d videos in real-time', in *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, pp. 1–8. IEEE, (2015).

[21] Boyi Jiang, Juyong Zhang, Bailin Deng, Yudong Guo, and Ligang Liu, 'Deep face feature for face alignment and reconstruction', *CoRR*, **abs/1708.02721**, (2017).

[22] Luo Jiang, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu, '3d face reconstruction with geometry details from a single image', *IEEE Transactions on Image Processing*, **27**(10), 4756–4770, (2018).

[23] Xin Jin and Xiaoyang Tan, 'Face alignment in-the-wild: A survey', *Computer Vision and Image Understanding*, **162**, 1–22, (2017).

[24] Amin Jourabloo and Xiaoming Liu, 'Pose-invariant 3d face alignment', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3694–3702, (2015).
[25] Amin Jourabloo and Xiaoming Liu, 'Large-pose face alignment via cnn-based dense 3d model fitting', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4188–4196, (2016).
[26] Ira Kemelmacher-Shlizerman and Ronen Basri, '3d face reconstruction from a single image using a single reference face shape', *IEEE transactions on pattern analysis and machine intelligence*, **33**(2), 394–405, (2010).
[27] Diederik P. Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', *CoRR*, **abs/1412.6980**, (2014).
[28] Zhujin Liang, Shengyong Ding, and Liang Lin, 'Unconstrained facial landmark localization with backbone-branches fully-convolutional networks', *CoRR*, **abs/1507.03409**, (2015).
[29] Feng Liu, Ronghang Zhu, Dan Zeng, Qijun Zhao, and Xiaoming Liu, 'Disentangling features in 3d face shapes for joint face reconstruction and recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5216–5225, (2018).
[30] Yaojie Liu, Amin Jourabloo, William Ren, and Xiaoming Liu, 'Dense face alignment', in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1619–1628, (2017).
[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, 'Deep learning face attributes in the wild', in *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, (2015).
[32] John McDonagh and Georgios Tzimiropoulos, 'Joint face detection and alignment with a deformable hough transform model', in *European Conference on Computer Vision*, pp. 569–580. Springer, (2016).
[33] Xi Peng, Rogerio S Feris, Xiaoyu Wang, and Dimitris N Metaxas, 'A recurrent encoder-decoder network for sequential face alignment', in *European conference on computer vision*, pp. 38–56. Springer, (2016).
[34] Elad Richardson, Matan Sela, and Ron Kimmel, '3d face reconstruction by learning from synthetic data', in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 460–469. IEEE, (2016).
[35] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel, 'Learning detailed face reconstruction from a single image', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1259–1268, (2017).
[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, 'U-net: Convolutional networks for biomedical image segmentation', in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, (2015).
[37] Joseph Roth, Yiying Tong, and Xiaoming Liu, 'Adaptive 3d face reconstruction from unconstrained photo collections', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4197–4206, (2016).
[38] Zsolt Sánta and Zoltan Kato, '3d face alignment without correspondences', in *European Conference on Computer Vision*, pp. 521–535. Springer, (2016).
[39] R. T. Seeley, 'Spherical harmonics', *American Mathematical Monthly*, **73**(4), 115–121, (1966).
[40] Tianyang Shi, Yi Yuan, Changjie Fan, Zhengxia Zou, Zhenwei Shi, and Yong Liu, 'Face-to-parameter translation for game character auto-creation', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 161–170, (2019).
[41] Karen Simonyan and Andrew Zisserman, 'Very deep convolutional networks for large-scale image recognition', *CoRR*, **abs/1409.1556**, (2014).
[42] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt, 'Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2549–2559, (2018).
[43] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt, 'Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction', in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1274–1283, (2017).
[44] Luan Tran and Xiaoming Liu, 'On learning 3d face morphable model from in-the-wild images', *IEEE transactions on pattern analysis and machine intelligence*, (2019).
[45] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni, 'Regressing robust and discriminative 3d morphable models with a very deep neural network', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5163–5172, (2017).
[46] Nannan Wang, Xinbo Gao, Dacheng Tao, Heng Yang, and Xuelong Li, 'Facial feature point detection: A comprehensive survey', *Neurocomputing*, **275**, 50–65, (2018).
[47] Ronald Yu, Shunsuke Saito, Haoxiang Li, Duygu Ceylan, and Hao Li, 'Learning dense facial correspondences in unconstrained images', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4723–4732, (2017).
[48] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li, 'Face alignment across large poses: A 3d solution', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 146–155, (2016).
[49] Xiangyu Zhu, Junjie Yan, Dong Yi, Zhen Lei, and Stan Z Li, 'Discriminative 3d morphable model fitting', in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pp. 1–8. IEEE, (2015).