LDF-Net: Learning a Displacement Field Network for Face Recognition Across Pose

Lanqing Hu[1,2] and Meina Kan[1,3] and Shiguang Shan[1,3] and Xingguang Song[4]
and Xilin Chen[1]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology
[4] Huawei Technologies Co., Ltd

***Abstract*— Face recognition is an important problem in computer vision, however, it is still challenging due to a few wild factors, such as large variations caused by pose, expression, lighting, etc. In this work, we mainly focus on dealing with the pose variations for face recognition. The proposed method attempts to directly transform a non-frontal face image into frontal one by Learning a Displacement Field network (LDF-Net) and then recognizes with the transformed images. The existing methods, that follow the same scheme of transforming non-frontal faces into frontal ones, either transform by using 3D-model (3D methods) or transform by using 2D reconstructive methods (2D methods). The 3D methods may lead to the invisibility of some pixels in the transformed frontal images, while the 2D methods may lead to difference between the pixels in the transformed frontal images and the original non-frontal images. Our proposed LDF-Net method can handle these two problems by learning a morphable displacement field for each pixel in the transformed frontal image. Therefore, LDF-Net can achieve a frontal image where all pixels are from the original non-frontal image pixels and no invisible pixels exist, so as to maintain the informative information from the non-frontal images as much as possible. The experiments on MultiPIE dataset show that the proposed LDF-Net achieves state-of-the-art performance for face recognition across pose, especially for those large poses.**

I. INTRODUCTION

Face recognition has attracted more and more attentions, as it is widely used in the area of access control, law enforcement, surveillance, watch-list, and even electronic payment in recent years. Although significant progress of face recognition in the wild has been witnessed [23], it is still a challenging problem due to the extremely large within-class appearance variations in terms of pose, illumination, expression, etc. Among these factors, pose variation is notoriously difficult since the appearance of a face image can change significantly as pose changes, as shown in Fig. 1.

To address this problem, many methods have been proposed which can be roughly grouped into 3D methods and 2D methods based on the type of information they utilize.

A. 3D Methods

3D methods utilize 3D structure information to model the transformation between poses. They first use 3D face data to obtain morphable displacement fields and then apply them to obtain 2D face data in different pose angles. The work in [21] presents a fully automatic system for pose-invariant face recognition. It finds the accurate 2D facial feature points and then does 3D pose normalization on the face images. In [22], a 3D model is constructed for each subject by applying a 3D generic elastic model to the frontal face images and then novel 2D pose views are synthesized for matching. In [4], a morphable displacement field is obtained from 3D face model after alignment with their 2D face images. Then the morphable displacement field acts on 2D face images to get new frontal face images. Finally all images are compared in the frontal view. [25] proposes to represent the face image using the un-occluded facial texture that is automatically detected in the 3D pose normalized face image. These 3D methods keep more original information and usually have better performance than the methods without using 3D information. However, the transformed or frontalized images from these methods may have some invisible pixels (i.e., some black holes), especially for those large poses.

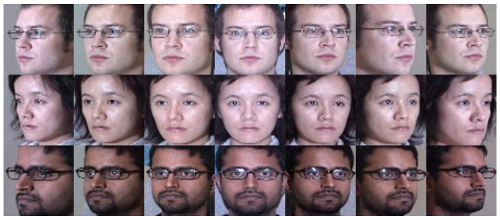


Fig. 1. An example of face images with variant poses (-45° to $+45^\circ$).

B. 2D Methods

2D methods only use the information from the 2D images rather than 3D structure model, therefore, most of them are learning-based methods. They usually attempt to learn pose-invariant feature representation or learn the 2D mapping between two poses. In [18], a discriminant coupled latent subspace framework is proposed to extract pose-invariant features. In [16], a Gaussian mixture model (GMM) is trained to decide if a pair of face tracks is matched or not. In the work of [17], a dynamic programming stereo algorithm is designed to deal with large occlusions, non-linear correspondences, and significant changes in appearance for face

978-1-5090-4023-0/17 $31.00 © 2017 IEEE
DOI 10.1109/FG.2017.12

recognition across pose. In [19], a deep neural network with random faces as target values is proposed. It extracts pose-invariant features via constraining unique target value of the same identity from different poses. In [20], the deep network classifies different poses and then explicitly tackles pose variation by using multiple pose specific models and rendering face images. In [26], a deep convolutional neural network learns the split node and CNN parameters at the same time to handle different poses separately. There are also some methods decoding their pose-invariant features into frontal face images. The work proposed in [9] designs a deep network that learns face identity-preserving (FIP) feature and then uses it to reconstruct the face images in canonical view. The work SPAE proposed in [10] transforms the non-frontal face images to frontal ones through a deep network in a progressive way. Each stacked part of the deep network learns the adjustment between small poses, and then the whole network can reconstruct corresponding frontal face images for the non-frontal face images. In both GMA [24] and MvDA [11], a supervised multi-view discriminant method is proposed to obtain pose-invariant features by optimizing multiple view-specific projection.

These 2D methods, especially those attempt to transform non-frontal face images to frontal images, are simpler and more efficient than those 3D methods. However, the retained information in the transformed frontal images is generally different from that in the non-frontal images, as the pixels in transformed images are usually combinations of the pixels in non-frontal images rather than the shifted original pixels in non-frontal images as in 3D methods. This difference results in information loss for the transformed images and performance degeneration of face recognition.

C. *The Proposed Method*

As illustrated above, the 3D methods have better performance than 2D methods but are usually much more time-consuming. These methods can preserve the values of pixels in original images but may lead to invisible pixels in the transformed image. On the contrary, the 2D methods are usually efficient, but are less powerful than the 3D methods. These methods can avoid black holes in the transformed images (i.e., more smooth visualization), but the values of pixels are varied compared to the original images leading to performance degeneration.

Considering the advantages and disadvantages of both methods, this work proposes a method that can combine the advantages of both methods and avoid the shortcomings of them. The proposed method endeavors to predict a displacement field for each pixel in the transformed frontal images by using a displacement field network with the non-frontal 2D face images as input, named as LDF-Net. As a result, all pixels in the transformed frontal image are all from the original non-frontal images without invisible parts, i.e., black holes, so the transformed images preserve the information as much as possible. Besides, the LDF-Net is a 2D method which is quite efficient and 3D information is an auxiliary for better effects.

Our contribution is three-fold. **(1)** We design a deep neural network named LDF-Net to recover frontal faces from non-frontal ones via learnable displacement field with which more details can be preserved in the recovered frontal faces. **(2)** LDF-Net can recover the pixels which are self-occluded in the non-frontal face images. **(3)** LDF-Net achieves the state-of-the-art performance on face recognition across pose.

The rest of this paper is organized as follows: section II presents the formulation of the proposed method and its optimization; section III evaluates LDF-Net and other methods on MultiPIE database, followed by the conclusion in the last section.

II. Method

A. *Overview*

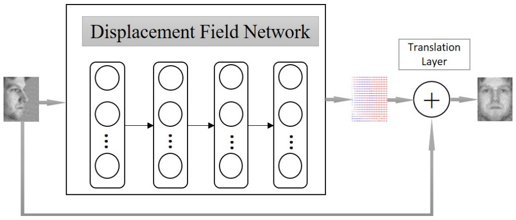


Fig. 2. Schema of our method, LDF-Net. LDF-Net is an end-to-end method to learn the transformation from a non-frontal face image to a frontal one, composing of a displacement field network F and a translation layer T.

The proposed method, LDF-Net, attempts to transform a non-frontal face image into a frontal one and uses the transformed images for recognition. The objective is that the output frontal face image of the LDF-Net should approximate the ground truth frontal one as much as possible. Each pixel in the transformed frontal face image is either directly the pixel in the original image, or linear interpolation of four neighboring pixels. In this way, we can preserve as much informative information as possible. Moreover, the LDF-Net implicitly employs the symmetry of human faces to obtain a transformed frontal face image without invisible pixels.

As shown in Fig. 2, the LDF-Net consists of two parts. The main part, the displacement field network F, whose details are shown in Fig. 3, learns the displacement field for the pixels between the target frontal face images and the input non-frontal ones. The other part, the translation layer T, transforms the input non-frontal face image into a frontal one with this displacement field given by F. The two parts are learnt end-to-end.

B. *Formulation of LDF-Net*

To train the LDF-Net, a set of n non-frontal face images, $I = I_1, I_2, ..., I_n \in \mathbb{R}^{h\times w}$ is used as input, and the corresponding frontal images $I^{gt} = I_1^{gt}, I_2^{gt}, ..., I_n^{gt} \in \mathbb{R}^{h\times w}$ as target frontal images, where I_k and I_k^{gt} are the non-frontal and frontal images of the same subject, respectively. The output is a set of estimated frontal face images denoted as $I^{est} = I_1^{est}, I_2^{est}, ..., I_n^{est} \in \mathbb{R}^{h\times w}$. The objective of the

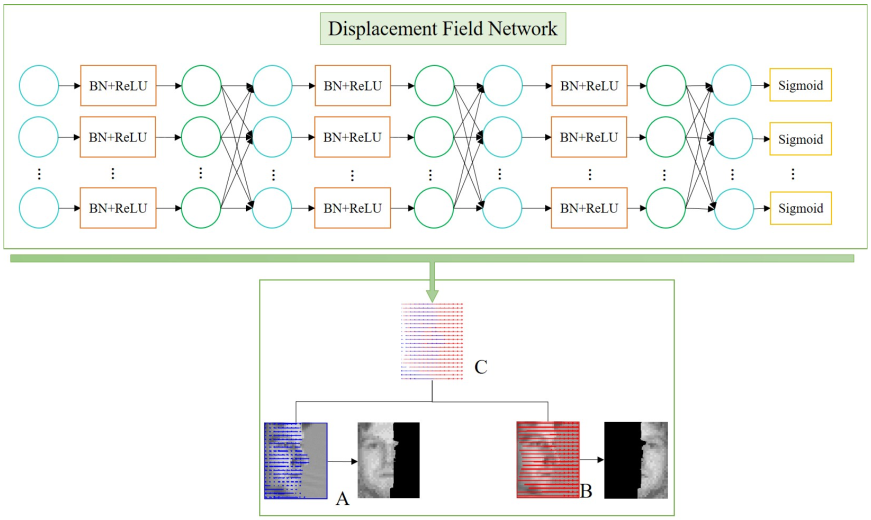


Fig. 3. Overview of the displacement field network. It composes of several fully connected layers with batch normalization and ReLU activation function and outputs a displacement field C for the input image. The displacement field is not only for the visible pixels (denoted as A) like 3D methods, but also for the self occluded part (denoted as B) shown in the bottom box.

proposed method is to minimize the difference between the output images I^{est} and the ground truth I^{gt}, i.e.,

$$\min \sum_{k=1}^{n} ||I_k^{gt} - I_k^{est}||_2^2. \tag{1}$$

1) Displacement Field Network: The displacement field models the shifting relationship of two pixels, a pixel in the transformed frontal image and its corresponding pixel in the original non-frontal image. The structure and workflow are shown in Fig. 3. Specifically, the displacement field network takes the non-frontal images I_k as input and outputs the displacement field D_k for all pixels in the transformed frontal images, i.e.,

$$D_k = F_W(I_k). \tag{2}$$

Here, the displacement field $D_k \in \mathbb{R}^{h \times w \times 2}$ consists of the displacement of two dimensions for each pixel in the transformed frontal images. Let $(\Delta_{kij}^h, \Delta_{kij}^w) \triangleq \big(D_k(i,j,1), D_k(i,j,2)\big)$ denote the translation distances of the pixel located at (i,j) in h and w axis. F_W is used to denote the displacement field network with model parameters W.

The displacement field network can be any kind of deep network structure, such as CNN, or fully connected network.

2) Translation Layer: With the displacement field D_k, the translation layer transforms the input image I_k into a frontal face image I_k^{est} by shifting the pixels in the input image, i.e.,

$$I_k^{est} = T(I_k, D_k) = T\big(I_k, F_W(I_k)\big). \tag{3}$$

If Δ_{kij}^h and Δ_{kij}^w are integers, each pixel $I_k^{est}(i,j)$ in the predicted frontal face image is directly shifted from one of the source pixels, calculated as follows:

$$I_k^{est}(i,j) = I_k(\hat{i}, \hat{j}). \tag{4}$$

where

$$\hat{i} \triangleq i + \Delta_{kij}^h, \tag{5}$$

and

$$\hat{j} \triangleq j + \Delta_{kij}^w. \tag{6}$$

However, Δ_{kij}^h and Δ_{kij}^w are computed from the displacement field network, which may be real value. So generally, the $I_k^{est}(i,j)$ can be obtained as the weighted sum of the four neighboring pixels not just rounded to integer:

$$I_k^{est}(i,j) = \sum_{m=\lfloor \hat{i} \rfloor}^{\lceil \hat{i} \rceil} \sum_{n=\lfloor \hat{j} \rfloor}^{\lceil \hat{j} \rceil} I_k(m,n)(1 - |\hat{i} - m|)(1 - |\hat{j} - n|), \tag{7}$$

Note that (4) is a special case of (7). Inferred from (4) and (7), with the displacement field, each pixel in the transformed frontal face image is either directly the pixel in the original image or linear interpolation of four pixels. Therefore, the LDF-Net can preserve the informative information of input images. Moreover, the displacement field outputted from the displacement field network is for the transformed images, i.e., $I_k^{est}(i,j) = I_k(i + \Delta_{kij}^h, j + \Delta_{kij}^w)$ rather than the input images as in most 3D methods like $I_k^{est}(i+\Delta_{kij}^h, j+\Delta_{kij}^w) = I_k(i,j)$. As a consequence, for the pixels of which the corresponding pixels in the input images are invisible, the LDF-Net can still recover these pixels from its symmetrical area of the face with a large $(\Delta_{kij}^h, \Delta_{kij}^w)$, even if there is small area of unsymmetrical appearance, e.g., nevus, scar and so on.

3) Overall Objective: In the training of LDF-Net, the overall objective of the LDF-Net is to minimize the difference between the transformed images and the ground truth frontal face images, formulated as below:

TABLE I
Attributes of Different Methods

Method	Linear vs. Non-linear	2D vs. 3D	Pose Estimation
PLS [5]	Linear	2D	Manually
MCCA [6]	Linear	2D	Manually
GMA [24]	Linear	2D	Manually
MvDA [11]	Linear	2D	Manually
MvDN [12]	Non-linear	2D	Manually
MDF [4]	Non-linear	3D	Automatically
FIP [9]	Non-linear	2D	No Need
SPAE [10]	Non-linear	2D	No Need
LDF-Net (Ours)	Non-linear	2D	No Need

$$\begin{aligned} W &= \underset{W}{\operatorname{argmin}} \sum_{k=1}^{n} ||I_k^{gt} - I_k^{est}||_2^2 \\ &= \underset{W}{\operatorname{argmin}} \sum_{k=1}^{n} ||I_k^{gt} - T(I_k, D_k)||_2^2 \\ &= \underset{W}{\operatorname{argmin}} \sum_{k=1}^{n} \left|\left|I_k^{gt} - T\Big(I_k, F_W(I_k)\Big)\right|\right|_2^2. \end{aligned} \tag{8}$$

C. Optimization

The whole network in (8) is optimized by using the gradient descent method. Just as most existing deep neural network, the gradient is firstly calculated following the chain rule, then the parameters are updated along the descent direction of the gradient. In each iteration of update, the loss of whole network is firstly calculated feed-forwardly, and then the gradients are back-propagated to minimize the loss during this iteration.

1) Step 1-Gradient of loss layer: Let l be the loss of the whole network, then it is formulated as below:

$$l = \sum_{k=1}^{n} ||I_k^{gt} - I_k^{est}||_2^2. \tag{9}$$

We need to compute the gradient of the loss w.r.t. the image $\frac{\partial l}{\partial I_k^{est}}$. Each element in the gradient can be calculated separately w.r.t. each pixel of I_k^{est} as below:

$$\frac{\partial l}{\partial I_k^{est}(i,j)} = -2\Big(I_k^{gt}(i,j) - I_k^{est}(i,j)\Big). \tag{10}$$

2) Step 2-Gradient of translation layer: The gradient of translation layer is computed similarly with spatial transformer network [1] as follows. As seen in (8), the input consists of the input image I_k and the displacement field D_k, so we need to obtain the gradient of the predicted frontal face image I_k^{est} w.r.t. the variable D_k denoted as $\frac{\partial I_k^{est}}{\partial D_k}$. As the elements in the translation layer T are independently transformed, so each element $\frac{\partial I_k^{est}(i,j)}{\partial D_k(i,j)}$ in $\frac{\partial I_k^{est}}{\partial D_k}$ can be separately computed as $\left(\frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^h}, \frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^w}\right)$. $\frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^h}$ is calculated as follows:

$$\begin{aligned} \frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^h} &= \frac{\partial I_k^{est}(i,j)}{\partial \hat{i}} \cdot \frac{\partial \hat{i}}{\partial \Delta_{kij}^h} = \frac{\partial I_k^{est}(i,j)}{\partial \hat{i}} \\ &= \sum_{m=\lfloor \hat{i} \rfloor}^{\lceil \hat{i} \rceil} \sum_{n=\lfloor \hat{j} \rfloor}^{\lceil \hat{j} \rceil} I_k(m,n)(1-|\hat{j}-n|) \left\{ \begin{array}{cc} 1 & \hat{i} \leq m \\ -1 & \hat{i} > m \end{array} \right\} \end{aligned} \tag{11}$$

Similarly, $\frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^w}$ can be calculated as below:

$$\begin{aligned} \frac{\partial I_k^{est}(i,j)}{\partial \Delta_{kij}^w} &= \frac{\partial I_k^{est}(i,j)}{\partial \hat{j}} \cdot \frac{\partial \hat{j}}{\partial \Delta_{kij}^w} = \frac{\partial I_k^{est}(i,j)}{\partial \hat{j}} \\ &= \sum_{m=\lfloor \hat{i} \rfloor}^{\lceil \hat{i} \rceil} \sum_{n=\lfloor \hat{j} \rfloor}^{\lceil \hat{j} \rceil} I_k(m,n)(1-|\hat{i}-m|) \left\{ \begin{array}{cc} 1 & \hat{j} \leq n \\ -1 & \hat{j} > n \end{array} \right\}. \end{aligned} \tag{12}$$

3) Step 3-Gradient of displacement field network: After calculating the gradient of all elements in D_k, the gradient of displacement field network w.r.t parameters $\frac{\partial D_k}{\partial W}$ can be easily achieved like any existing deep neural network.

4) Step 4-Overall parameter update via gradient descent: The gradient of overall LDF-Net is calculated with chain rule. With each part's gradient, the gradient of LDF-Net is

$$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial I_k^{est}} \cdot \frac{\partial I_k^{est}}{\partial D_k} \cdot \frac{\partial D_k}{\partial W}. \tag{13}$$

With the gradient, in each iteration t, the parameters W of the whole network can be updated with a learning rate η as below:

$$W_t = W_{t-1} + \eta \frac{\partial l}{\partial W}. \tag{14}$$

After a number of updates with (14), the LDF-Net in (8) can be optimized.

D. Differences with the Existing Methods

1) Difference with FIP and SPAE: FIP [9], SPAE [10] and our LDF-Net are all 2D methods that attempt to transform a non-frontal image into a frontal one. FIP and SPAE are end-to-end networks learning the value of each pixel directly. The pixels in the transformed frontal images in FIP and SPAE are obtained from a complex transformation and combination of the pixels of the original non-frontal image. They usually result in a smoothed frontal image. Differently, our method, LDF-Net learns the translating relationship of pairs of pixels. The pixels in the transformed frontal images in LDF-Net are directly shifted from the original non-frontal image, so the pixels in the transformed frontal image are almost the same as those in the original non-frontal image. In this way, LDF-Net can preserve more informative information to achieve better performance.

2) Difference with MDF: MDF [4] is a 3D method that employs an auxiliary 3D dataset to estimate the displacement field for each image, while our LDF-Net is 2D method that employs a deep neural network to predict the displacement field for each image. Besides, in MDF, some pixels are invisible in the transformed frontal image for those large poses, while in our LDF-Net, every pixel is visible benefitting from that the displacement field network can implicitly employ the symmetry of the face to predict the displacement field even for those invisible pixels. Moreover, as a 2D method, LDF-Net is faster than MDF for testing.

III. EXPERIMENTS

A. Experimental Settings

1) Dataset: This work mainly focuses on the pose variations of face images, and the MultiPIE [7] dataset is employed for the evaluations as this dataset contains rich pose variations. The MultiPIE dataset contains images of 337 subjects under various poses, illuminations and expressions. These images were captured in four sessions during different periods. The detailed setting and result of three different experiments are showed below.

Setting-I was introduced in [12]. 337 subjects are chosen from 4 sessions with 13 poses (from -90° to 90°), neutral illumination and expression(708 images of the first 229 subjects for training and 213 images of the remaining 108 for testing in each pose).

Setting-II was also the same with experiment setting in [12] but with smaller pose variance. Still, 337 subjects are chosen from 4 sessions with 7 poses (from -45° to 45°), neutral illumination and expression (708 images of the first 229 subjects for training and 213 images of the remaining 108 for testing in each pose).

Setting-III was introduced in [4] and [9]. The images of the first 200 identities in all the 4 sessions with 7 poses (from -45° to 45°) are chosen for training, and the images of the remaining 137 identities for test.

In all of the three experiments, the images are cropped into size of $40 * 32$. Some exemplar images are shown in Fig. 4. In the stage of testing, the images from the pose in 0° (i.e., frontal images) are used as the gallery, and the images from the rest poses are used as the probe. The rank-1 recognition rate is used as the measurement of performance for all methods.

2) Methods for Comparison: In this work, the methods of PLS [5], MCCA [6], GMA [24], MvDA [11], MvDN [12], FIP [9], SPAE [10], MDF [4] are compared. Among them, the PLS, MCCA, GMA, and MvDA are linear methods that attempt to extract pose-invariant features. MvDN is a deep method for extracting pose-invariant features. The FIP, SPAE and MDF are also deep methods but they endeavor to directly transform the non-frontal face image into the frontal one like our proposed method in this work. The MDF is a 3D method, while the rest are 2D methods. The characteristics of each method is detailed in Table I. As PLS, MCCA, SPAE, FIP and our LDF-Net are unsupervised methods for recognition, the Linear Discriminant Analysis (LDA) [3] is applied for supervised feature extraction for face recognition across pose. Besides, for all methods, the Principal Component Analysis (PCA) [2] is used for dimension reduction, and the reduced dimension is determined by preserving 95% energy at least.

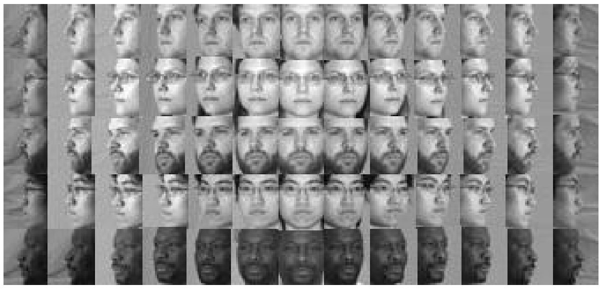


Fig. 4. The face images in MultiPIE are in 13 poses from -90° to $+90^\circ$.

For all these methods, we try our best to tune their parameters to report the best results. Specifically, in PLS, MCCA and MvDA, the number of the projection matrix is tuned; in GMA, the λ and γ are tuned; in SPAE, 6 hidden layers are used instead of 3 layers for recognition of large poses, i.e., from 0° to 90°; in FIP, batch normalization layer [15] is added to each convolution layer and fully connected layer. Besides, the pixel values of input images are scaled to $[0, 1]$ for better performance.

3) The Structure of Our Method: Our LDF-Net consists of a displacement field network and a translation layer, among which only the displacement field network includes variables. Specially, the displacement field network composes of four fully connected layers. But note that convolutional layers are also applicable. The bottom three layers are followed by a batch normalization layer [15] and a ReLU activation function [14]. The last fully connected layer outputting the displacement field has $2560 \quad (= 40 \times 32 \times 2)$ output units with the sigmoid activation function $\frac{1}{1+e^{-f(x)}}$. The activation function of the last layer scales the output into a fixed range to make it convenient to map the output into arbitrary height and width. Different activation functions and number of nodes of the bottom three layers of the displacement field network are tried respectively, and the recognition results are shown in Table II and Table III, which imply that using structure like the Pyramid and batch normalization with ReLU has better effect for feature extracting. Among them, we choose the parameters with the best performance. The details of the chosen structure of the displacement field network are summarized in Table IV. LDF-Net is implemented by using the Caffe platform [13].

TABLE II
AVERAGE RECOGNITION RATES IN SETTING-I WITH DIFFERENT ACTIVATION FUNCTION OF THE BOTTOM THREE LAYERS IN THE DISPLACEMENT FIELD NETWORK

Activation Function	Sigmoid	TanH	bn+ReLU
Average Accuracy	0.880	0.887	0.898

TABLE III
AVERAGE RECOGNITION RATES OF IN SETTING-I WITH DIFFERENT NUMBER OF NODES OF THE DISPLACEMENT FIELD NETWORK

Number of Nodes	(4096, 4096, 4096)	(4096, 2048, 1024)	(4096, 1024, 512)	(1024, 1024, 1024)	(1024, 256, 128)
Average Accuracy	0.890	0.890	0.901	0.887	0.898

TABLE IV
COMPONENTS OF LDF-NET

Layer 1	Layer 2	Layer 3	Layer 4
fc(4096) bn ReLU	fc(1024) bn ReLU	fc(512) bn ReLU	fc(2560) Sigmoid

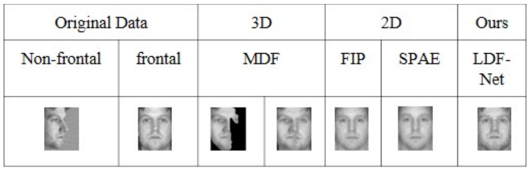
Our proposed LDF-Net is 2D method, and the outputted displacement field is also 2D dimensional. However, we can pre-train the 2D displacement field with the displacement fields estimated from any 3D methods. In this work, to obtain a better performance of LDF-Net and reduce the training time, the displacement field network is pre-trained before the end-to-end training by minimizing $\sum_{k=1}^{n}||D_k - D_k^{est3D}||_2^2$, where the displacement field D_k^{est3D} are estimated from BJUT-3D [8] dataset. From BJUT-3D, only single displacement field can be obtained for all images in one pose which is quite coarse. The occluded pixels in the estimated displacement field are filled by their symmetrical ones. Though the displacement field is coarse, our displacement field network will refine it and obtain the displacement field for all target pixels including self occluded ones in the process of end-to-end training.

B. Face Recognition Across Pose on MultiPIE

All the methods are evaluated on MultiPIE in terms of rank-1 recognition rate for face recognition across pose. For all methods, the training dataset and testing data are the same as illustrated in section III-A. The evaluation results are shown in Table V and Table VII.

As seen from Table V, the PLS and MCCA perform worse than other compared methods as they are both unsupervised. By incorporating supervised information, the PLS+LDA, MCCA+LDA, GMA and MvDA perform much better. Furthermore, the deep method MvDN performs even better benefitting from the characteristic of non-linearity in deep network. The methods of SPAE, FIP and our LDF-Net are all transformation based methods, i.e., firstly transform a non-frontal image into a frontal image followed by supervised method LDA [3] for the final recognition. In recognizing phase, for fair comparison, we only use the transformed frontal face images of FIP and SPAE as input of LDA. All these three methods are slightly worse than the MvDN in some poses, however the advantage of them is that they do not need to know the pose of the testing images, while MvDN must know the pose of testing images (either manually labeled or estimated automatically). As seen, LDF-Net significantly outperforms all methods especially for large poses, which can be attributed to the combination of the advantages of both 2D methods and 3D methods demonstrating the effectiveness of the proposed method.

Furthermore, we evaluate several representative methods on MultiPIE following the protocol used in MDF for fair comparison as we can not fully re-implement the performance of MDF. The evaluation results are shown in Table VII. In this protocol, face recognition is evaluated only within $45°$. Similar observations can be obtained as that the LDF-Net outperforms all methods even the 3D method MDF, benefitting from the end-to-end LDF-Net which has the advantages of preserving more of the original face information as well as with no invisible pixels.



Original Data		3D		2D		Ours
Non-frontal	frontal	MDF		FIP	SPAE	LDF-Net

Fig. 5. The frontal image obtained from MDF, FIP, SPAE and our method, LDF-Net.

To visually compare the results of different methods that transform the non-frontal face images into frontal images, Fig. 5 shows the transformed frontal images of different methods. As seen from Fig. 5, the transformed frontal image from the 3D method MDF is almost as clear as the original frontal image, but has some invisible pixels. The transformed images from the 2D methods of SPAE and FIP are without invisible pixels but look more smooth and average than the original image. This indicates the loss of information in the transformed image. The transformed frontal image from our LDF-Net can avoid the shortcomings of them, which implies that the transformed frontal image of LDF-Net can preserve the information of original image as much as possible with no invisible pixels and obtain the image that looks more alike the ground truth frontal image. More exemplar images transformed from SPAE, FIP and LDF-Net are shown in Fig. 6. The results of SPAE and FIP are more smooth and blur, and some details of the hair, beard, glasses are missing. On the contrary, these details are well preserved in the transformed frontal images from our LDF-Net.

TABLE V

FACE RECOGNITION ACROSS POSE IN SETTING-I

Method	Pose of Testing Images												Average
	$-90°$	$-75°$	$-60°$	$-45°$	$-30°$	$-15°$	$+15°$	$+30°$	$+45°$	$+60°$	$+75°$	$+90°$	
PLS [5]	0.319	0.775	0.892	0.934	0.883	0.981	0.981	0.934	0.906	0.873	0.723	0.268	0.789
MCCA [6]	0.409	0.742	0.822	0.723	0.685	0.920	0.906	0.798	0.747	0.779	0.714	0.376	0.718
PLS [5]+LDA	0.380	0.798	0.869	0.944	0.920	0.995	0.986	0.967	0.883	0.850	0.709	0.319	0.802
MCCA [6]+LDA	0.488	0.662	0.817	0.887	1.000	**1.000**	**1.000**	0.995	0.831	0.803	0.676	0.568	0.811
GMA [24]	0.526	0.732	0.845	0.901	1.000	**1.000**	**1.000**	**1.000**	0.906	0.859	0.718	0.573	0.838
MvDA [11]	0.568	0.723	0.845	0.920	0.967	**1.000**	**1.000**	0.991	0.897	0.864	0.714	0.559	0.837
MvDN [12]	**0.704**	0.822	0.883	0.911	0.991	**1.000**	**1.000**	0.991	0.930	0.911	0.798	**0.709**	0.887
FIP [9]+LDA	0.578	0.775	0.859	0.953	**0.995**	**1.000**	**1.000**	**1.000**	0.962	0.826	0.756	0.573	0.856
SPAE [10]+LDA	0.516	0.751	0.892	0.948	0.986	0.991	0.995	0.995	0.939	0.864	0.770	0.545	0.849
LDF-Net (Ours)+LDA	0.639	**0.873**	**0.930**	**0.981**	0.986	0.972	**1.000**	0.991	**0.986**	**0.944**	**0.850**	0.667	**0.901**

TABLE VI

FACE RECOGNITION ACROSS POSE IN SETTING-II

Method	Pose of Testing Images						Average
	$-45°$	$-30°$	$-15°$	$+15°$	$+30°$	$+45°$	
MCCA [6]	0.774	0.930	1.000	0.997	0.976	0.850	0.759
MvDA [11]	0.800	0.948	0.997	0.994	0.985	0.872	0.933
FIP [9]+LDA	0.977	0.995	1.000	1.000	1.000	0.967	0.990
SPAE [10]+LDA	0.967	0.991	1.000	1.000	1.000	0.967	0.988
LDF-Net(Ours)+LDA	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	**0.981**	**0.996**

TABLE VII

FACE RECOGNITION ACROSS POSE IN SETTING-III

Method	Pose of Testing Images						Average
	$-45°$	$-30°$	$-15°$	$+15°$	$+30°$	$+45°$	
MDF [4]	0.930	0.987	0.997	0.997	0.983	0.936	0.972
FIP [9]+LDA	0.956	0.985	**1.000**	0.993	0.985	**0.978**	0.983
LDF-Net(Ours)+LDA	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.970	**0.995**

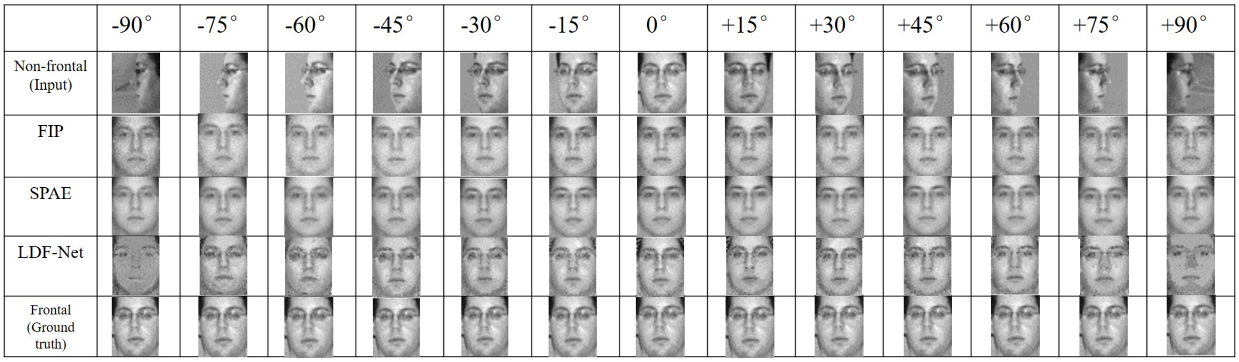


Fig. 6. Exemplar frontal images obtained from FIP, SPAE and LDF-Net in all 13 poses. As seen, image details and identity features such as glasses can be better reserved in LDF-Net.

IV. CONCLUSIONS AND FUTURE WORKS

To deal with the face recognition across pose problem, we introduce an end-to-end deep neural network (LDF-Net) to transform a non-frontal face image into a frontal one. To achieve the transformation, the proposed method, LDF-Net learns the displacement field, which reflects the shifting relationship of pixels from the non-frontal face image and the transformed frontal face images. LDF-Net can achieve a frontal image which preserves the informative information from the original image as much as possible and has no invisible pixels. Benefited from the end-to-end training, our LDF-Net can achieve a relatively smooth displacement field even without smoothness constraint, which is considered beneficial for better visual quality. Besides, LDF-Net can also generalize to handle other variations. As evaluated on MultiPIE, LDF-Net achieves quite promising performance for face recognition across pose, especially for those large poses.

Although LDF-Net is an end-to-end method for transforming non-frontal-face images into frontal-face images, it is not an end-to-end method for recognition. In the future, we will improve this work to an end-to-end one for both the transformation and recognition.

V. ACKNOWLEDGMENTS

This work was partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61650202, 61402443, 61272321, and the Strategic Priority Research Program of the CAS (Grant XDB02070004).

REFERENCES

[1] Jaderberg M, Simonyan K, Zisserman A, "Spatial transformer networks", *in Advances in Neural Information Processing Systems (NIPS)*, 2015, pp 2017-2025.

[2] Turk M A, Pentland A P, "Face recognition using eigenfaces", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991, pp 586-591.

[3] Lu J, Plataniotis K N, Venetsanopoulos A N, "Face recognition using LDA-based algorithms", *in IEEE Transactions on Neural Networks (TNN)*, 2003, 14(1): 195-200.

[4] S. Li, X. Liu, X. Chai, H. Zhang, S. Lao, and S. Shan, "Morphable displacement field based image matching for face recognition across pose", *in European Conference on Computer Vision (ECCV)*, 2012, pp 102-115.

[5] H. Hotelling, "Relations between two sets of variates", *in Biometrika*, 1936, 28(3/4):321-377.

[6] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis", *in Conference on Data Mining and Data Warehouses (SiKDD)*, 2010, pp 1-4.

[7] R. Gross, I. Matthews, J. Cohn, T. kanada, and S. Baker, "The cmu multi-pose, illumination, and expression (multipie) face database", Technical report, Carnegie Mellon University Robotics Institute, 2007, TR-07-08.

[8] Baocai, Y., Yanfeng, S., C. W, Yun, G., "Bjut-3d large scale 3d face database and information processing", *in Journal of Computer Research and Development (JCRD)*, 2009.

[9] Z. Zhu, P. Luo, X. Wang, et al, "Deep learning identity-preserving face space", *in IEEE International Conference on Computer Vision (ICCV)*, 2013, pp 113-120.

[10] M. Kan, S. Shan, H. Chang, et al, "Stacked progressive auto-encoders (spae) for face recognition across poses", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp 1883-1890.

[11] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis", *in European Conference on Computer Vision (ECCV)*, 2012, pp 808-821.

[12] M. Kan, S. Shan, X. Chen, "Multi-view Deep Network for Cross-view Classification", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Jia Y, Shelhamer E, Donahue J, et al, "Caffe: Convolutional architecture for fast feature embedding", *in ACM international conference on Multimedia (ACMMM)"*, 2014, pp 675-678.

[14] Krizhevsky A, Sutskever I, Hinton G E, "Imagenet classification with deep convolutional neural networks", *in Advances in Neural Information Processing Systems (NIPS)*, 2012, pp 1097-1105.

[15] Ioffe S, Szegedy C, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *in International Conference on Machine Learning (ICML)*, 2015.

[16] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp 3499-3506.

[17] C. D. Castillo and D. W. Jacobs. "Wide-baseline stereo for face recognition with large pose variation", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp 537-544.

[18] A. Sharma, M. A. Haj, J. Choi, L. S. Davis, and D. W. Jacobs, "Robust pose invariant face recognition using coupled latent space discriminant analysis", *in Computer Vision and Image Understanding (CVIU)*, 2012, pp 1095-1110.

[19] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu, "Random faces guided sparse many-to-one encoder for pose-invariant face recognition", *in IEEE International Conference on Computer Vision (ICCV)*, 2013, pp 2416-2423.

[20] I. Masi, S. Rawls, G. Medioni, P. Natarajan, "Pose-Aware Face Recognition in the Wild", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp 4838-4846.

[21] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith, "Fully automatic pose-invariant face recognition via 3d pose normalization", *in IEEE International Conference on Computer Vision (ICCV)*, 2011, pp 937-944.

[22] U. Prabhu, J. Heo, and M. Savvides, "Unconstrained poseinvariant face recognition using 3d generic elastic models", *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2011, 33(10):1952-1961.

[23] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", *in IEEE International Conference on Computer Vision (ICCV)*, 2015

[24] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space", *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012

[25] C. Ding, C. Xu, D. Tao, "Multi-task pose-invariant face recognition", *in IEEE Transactions on Image Processing (TIP)*, 2015, 24(3):980-993.

[26] C. Xiong, X. Zhao, D. Tang, et al, "Conditional convolutional neural network for modality-aware face recognition", *in IEEE International Conference on Computer Vision (ICCV)*, 2015: 3667-3675.