Frankenstein: Learning Deep Face Representations using Small Data

Citation for published version:
Hu, G, Peng, X, Yang, Y, Hospedales, TM & Verbeek, J 2018, 'Frankenstein: Learning Deep Face Representations using Small Data', *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 293 - 303. https://doi.org/10.1109/TIP.2017.2756450

Digital Object Identifier (DOI):
10.1109/TIP.2017.2756450

Published In:
IEEE Transactions on Image Processing

Frankenstein: Learning Deep Face Representations using Small Data

Guosheng Hu, *Member, IEEE,* Xiaojiang Peng, *Member, IEEE,* Yongxin Yang, *Student Member, IEEE,* Timothy Hospedales, *Member, IEEE,* and Jakob Verbeek

***Abstract*—Deep convolutional neural networks have recently proven extremely effective for difficult face recognition problems in uncontrolled settings. To train such networks, very large training sets are needed with millions of labeled images. For some applications, such as near-infrared (NIR) face recognition, such large training datasets are, however, not publicly available and very difficult to collect. In this work, we propose a method to generate very large training datasets of synthetic images by compositing real face images in a given dataset. We show that this method enables to learn models from as few as 10,000 training images, which perform on par with models trained from 500,000 images. Using our approach we also improve the state-of-the-art results on the CASIA NIR-VIS2.0 heterogeneous face recognition dataset.**

***Index Terms*—face recognition, deep learning, small training data**

I. Introduction

In recent years, deep learning methods, and in particular convolutional neural networks (CNNs), have achieved considerable success in a range of computer vision applications including object recognition [22], object detection [10], semantic segmentation [34], action recognition [42], and face recognition [38]. The recent success of CNNs stems from the following facts: (i) big annotated training datasets are currently available for a variety of recognition problems to learn rich models with millions of free parameters; (ii) massively parallel GPU implementations greatly improve the training efficiency of CNNs; and (iii) new effective CNN architectures are being proposed, such as the Very Deep VGG Network [43], Google Inception Network [52] and Deep Residual Networks [13].

Good features are essential for object recognition, including face recognition. Conventional features include linear functions of the raw pixel values, including Eigenface (Principal Component Analysis) [55], Fisherface (Linear Discriminant Analysis) [3], and Laplacianface (Locality Preserving Projection) [14]. Such linear features were later replaced by hand-crafted local non-linear features, such as Local Binary Patterns [1], Local Phase Quantisation (LPQ) [2], and Fisher vectors computed over dense SIFT descriptors [41]. Note that the latter is an example of a feature that also involves unsupervised learning. These traditional features achieve promising face recognition rates in constrained environments, as represented in the CMU PIE dataset [40]. However, using these features face recognition performance may degrade dramatically in uncontrolled environments, as represented in the Labeled Faces in the Wild (LFW) benchmark [16]. To improve the performance in such challenging settings, metric learning can be used, see [5], [11], [57]. Metric learning methods learn a (linear) transformation of the features that pulls the objects that have the same label closer together, while pushing the objects that have different labels apart.

Although hand-crafted features and metric learning achieve promising performance for uncontrolled face recognition, it remains cumbersome to improve the design of hand-crafted local features (such as SIFT [25]) and their aggregation mechanisms (such as Fisher vectors [37]). This is because the experimental evaluation results of the features cannot be automatically fed back to improve the robustness to nuisance factors such as pose, illumination and expression. The major advantage of CNNs is that all processing layers, starting from the raw pixel-level input, have configurable parameters that can be learned from data. This obviates the need for manual feature design, and replaces it with supervised data-driven feature learning. Learning the large number of parameters in CNN models (millions of parameters are rather a rule than an exception) requires very large training datasets. For example, the CNNs, which achieve state-of-the-art performance on the LFW benchmark, are trained using datasets with millions of labeled faces: Facebook's DeepFace [53] and Google's FaceNet [38] were trained using 4 million and 200 million training samples, respectively.

For some recognition problems large supervised training datasets can be collected relatively easily. For example the CASIA Webface dataset [60] of 500,000 face images was collected semi-automatically from IMDb. However, in many other cases collecting large datasets may be costly, and possibly problematic due to privacy regulation. For example, thermal infrared imaging is ideal for low-light nighttime and covert face recognition applications [21], but it is not possible to collect millions of labeled training images from the internet for the thermal infrared domain. The lack of large training datasets is an important bottleneck that prevents the use of deep learning methods in such cases, as the models will overfit dramatically when using small training datasets [15].

To address this issue, the use of big synthetic training datasets has been explored by a number of authors [17], [30], [35]. There are two important advantages of using synthetic data (i) one can generate as many training samples as desired,

G. Hu, X. Peng and J. Verbeek are with THOTH team, INRIA Grenoble Rhone-Alpes, France. email: {guosheng.hu, xiaojiang.peng, jakob.verbeek}@inria.fr. G. Hu is also with Anyvision group.

Y. Yang is with Electronic Engineering and Computer Science, Queen Mary University of London, UK. email:yongxin.yang@qmul.ac.uk

T. Hospedales is with the University of Edinburgh, UK. email: t.hospedales@ed.ac.uk

and (ii) it allows explicit control over the nuisance factors. For instance, we can synthesize face images of all desired viewpoints, whereas data collected from the internet might be mostly limited to near frontal views. Data synthesis has successfully been applied to diverse recognition problems, including text recognition [17], scene understanding [30], and object detection [35]. Two very recent works [64], [9], [27] proposed 3D-aided face synthesis technique for facial landmark detection and face recognition in the wild respectively.

Data augmentation is another technique that is commonly used to reduce the data scarcity problem, see [32], [43]. This is similar to data synthesis, but more limited in that existing training images are transformed without affecting the semantic class label, e.g. by applying cropping, rotation, scaling, etc.

The main contribution of this paper is a solution for training deep CNNs using very small training data. To achieve this, we propose a data synthesis technique to expand very limited face datasets to very large ones that are suitable to train powerful deep CNNs. Specifically, we synthesize images of a 'virtual' subject c by compositing automatically detected face parts (eyes, nose, mouth) of two existing subjects a and b in the dataset in a fixed pattern. Images for the new subject are generated by composing a nose from an image of subject a with a mouth of an image of subject b. This is motivated by the observation that face recognition consists in finding the differences in the appearance and constellation of face parts among people. For a dataset with an equal number of faces per person, this method can increase a dataset of n images to one with n^2 images when using only 2 face parts (we use 5 parts in practice). A dataset like LFW can thus be expanded from a little over 10,000 images to a dataset of 100 million images.

Unlike the existing face synthesis methods [64], [9], [27] which use 3D models, our method is a pure 2D method which is much easier to implement. In addition, our method works on different tasks from [64], [9], [27]. Specifically, the methods [64], [9] are used for facial landmark detection, while ours for face recognition. The approach [27] assumes a relatively large training data (500,000 images) already exists, while we assumes the training data (10,000 images) is extremely small.

We experimentally demonstrate that the synthesized large training datasets indeed significantly improve the generalization capacity of CNNs. In our experiments, we generate a training set of 1.5 million images using an initial labeled dataset of only 10,000 images. We improve the face verification rates from 78.97% to 95.77% on LFW using CNNs trained on 10K images and 1.5 million synthetic images respectively. In addition, the proposed face synthesize is also used for NIR-VIS heterogeneous face recognition [29] and improve the rank-1 face identification rate from 17.41% to 85.05%. With the synthetic data, we achieve state-of-the-art performance on both (1) LFW under the "unrestricted, label-free outside data" protocol and (2) CASIA NIR-VIS 2.0 database under rank-1 face identification protocol.

II. Related work

Our work relates to three research areas that we briefly review below: face recognition using deep learning methods (Section II-A), face data collection (Section II-B), and data augmentation and synthesis methods (Section II-C).

A. Face recognition using deep learning

Here we briefly review recent state-of-the-art face recognition methods based on deep learning.

Since face recognition is a special case of object recognition, good architectures for general object recognition may carry over to face recognition. Schroff et al [38] explored networks that are based on that of Zeiler & Fergus [62] and Inception networks [52]. DeepID3 [46] uses aspects of both Inception networks [51] and the very deep VGG network [43]. Parkhi et al. [31] use the same architecture as the very deep VGG network [43], while Yi et al. [60] use 3×3 filters but fewer layers.

DeepFace [53] combines 3D face alignment and CNN for face recognition. Specifically, a 3D model is used for pose normalization, by which all the faces are rotated to the frontal pose. In this way, pose variations are removed from the faces. Then an 8-layer CNN is trained using four million pose-normalized images.

DeepID [49], DeepID2 [45], DeepID2+ [50] all train an ensemble of small CNNs. The input of one small CNN is an image patch cropped around a facial part (face, nose, mouth, etc.). The same idea is also used in [24]. DeepID uses only a classification-based loss to train the CNN, while DeepID2 includes an additional verification-based loss function. To further improve the performance, DeepID2+ adds the loss functions to all the convolutional layers rather than the topmost layer only.

All the above methods train CNNs using large training datasets (500,000 faces or more). To the best of our knowledge, only [15] uses small datasets to train CNNs (only around 10,000 LFW images) and achieves significantly worse performance on the LFW benchmark: 87% vs 97% or higher in [38], [50], [53]. Clearly, sufficiently large training datasets are extremely important for learning deep face representations.

B. Face dataset collection

Since big data is important for learning a deep face representation, several research groups have collected large datasets with 90,000 up to 2.6 million labeled face images [5], [28], [31], [47], [60]. To achieve this, they collect face images from the internet, by querying for specific websites such as IMDb or general search engines for celebrity names. This data collection process is detailed in [31], [60].

There are, however, two main weaknesses of the existing face data collection methods. First, and most importantly, internet-based collection of large face datasets is limited to visible spectrum images, and is not applicable to collect e.g. infrared face images. Second, the existing collection methods are expensive and time-consuming. It results from the fact that automatically collected face images are noisy, and manual filtering has to be performed to remove incorrectly labeled images [31].

The difficulty of collecting large datasets in some domains, e.g. for infrared imaging, motivates the work presented in

this paper. To address this issue we propose a data synthesis method that we describe in the next section.

C. Data augmentation and synthesis

The availability of large supervised datasets is the key for machine learning to succeed, and this is true in particular for very powerful deep CNN models with millions of parameters. To alleviate data scarcity in visual recognition tasks, data augmentation has been used to add more examples by applying simple image transformations that do not affect the semantic-level image label, see e.g. [7]. Examples of such transformations are horizontal mirroring, cropping, small rotations, etc. Since it is not always clear in advance which (combinations of) transformations are the most effective to generate examples that improve the learning the most, Paulin et al. [32] proposed to learn which transformations to exploit.

Data augmentation, however, it limited to relatively simple image transformations. Out-of-plane rotations, for example, are hard to accomplish since they would require some degree of 3D scene understanding from a single image. Pose variations of articulated objects are another example of transformations that are non-trivial to obtain, and generally not used in data augmentation methods.

Training models from synthetic data can overcome such difficulties, provided that sufficiently accurate object models are available. Recent examples where visual recognition systems have been trained from synthetic data include the following. Shotton et al. [39] train randomized decision forests for human pose estimation from synthesized 3D depth data. Jaderberg et al. [17] use synthetic data to train CNN models for natural scene text recognition. Su et al. [44] use synthetic images of objects to learn a CNN for viewpoint estimation. Papon and Schoeler [30] train a multi-output CNN that predicts class, pose, and location of objects from realistic cluttered room scenes that are synthesized on the fly. Weinmann et al. [58] synthesize material images under different viewing and lighting conditions based on detailed surface geometry measurements, and use these to train a recognition system using a SIFT-VLAD representation [18]. Ronzantsev et al. [36] use rough 3D models to synthesize new views of real object category instances. They show that this outperforms more basic data augmentation using crops, flips, rotations, etc.

Data synthesis techniques are also used for face analysis. To improve the accuracy of facial landmark detection in the presence of large pose variations [9], [64], a 3D morphable face models is used to synthesize face images in arbitrary poses. Similar data synthesis techniques are also used for pose-robust face recognition [27]. Unlike 3D solutions, we propose a 2D data synthesis method to solve the problem of training deep CNNs using very limited training data.

III. Synthetic data engine

Human faces are well structured in the sense that they are composed of parts (eyes, nose, mouth, etc.) which are organized in a relatively rigid constellation. Face recognition is conducted implicitly by finding the differences of one or more facial parts and possibly their constellation among people.

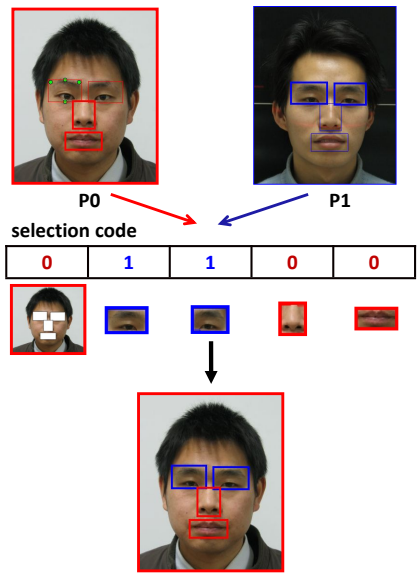

Fig. 1: Schematic illustration of the face synthesis process using five parts: left-eye, right-eye, nose, mouth and the rest. Parent images P0 and P1 (top) are mixed by using the eyes of P1 and the other parts of P0 (middle) to form the synthetic image (bottom).

Motivated by this, our synthetic face images are generated by swapping one or more facial parts among existing "parent" images. In our work we use five face parts: right eye (RE), left eye (LE), nose (N), mouth (M) and the rest (R). See Figure 1 for an illustration. For simplicity, we only consider the synthesis using only two parent images in this work. Our synthesis method can easily be extended, however, to the scenario of more than two parent images.

Suppose that we have an original dataset and let $\mathcal{S}$ denote the set subjects in the dataset, and let n_i denote the number of images of subject $i \in \mathcal{S}$. To synthesize an image, we select a tuple (i, j, c, s, t) where $i \in \mathcal{S}, j \in \mathcal{S}$ correspond to two subjects that will be mixed, and $s \in \{1, \ldots, n_i\}$ and $t \in \{1, \ldots, n_j\}$ are indices of images of i and j that will be used. The bitcode $c \in \{0, 1\}^5$ defines which parts will be taken from each subject. A zero at a certain position in b means that the corresponding part will be taken from i, otherwise it will be taken from j. There are only $2^5 - 2 = 30$ valid options for b, since the codes 00000 and 11111 would correspond to the original images of s and t respectively, instead of synthetic ones.

To synthesize a new image, we distinguish the two parent images as the "base" image from which we use the R (the rest) part, and the "injection" image from which one or more parts will be pasted on this base image. Since the size of the facial parts of the two parent images are in general different, we re-size the facial parts of the injection image to that of the base image. The main challenge to implement the proposed synthesis method is to accurately locate the positions of the facial parts. Recently, many efficient and accurate landmark detectors have been proposed. We use four landmarks detected by the method of Zhang et al. [63] to define the rectangular

region that corresponds to each face part.

We refer to each choice of (i, j, c) with $i \neq j$ as a "virtual subject" which consists of a mix of two existing subjects in the dataset. In total we can generate $30|\mathcal{S}|(|\mathcal{S}|-1)/2$ different virtual subjects, and for each of these we can generate $n_i \times n_j$ samples. Note that if we set $i = j$ we can in the same manner synthesize $30n_i(n_i-1)/2$ new images for an existing subject.

Artefacts The synthetic faces present some artefacts which we discuss here. The artefacts can be categorised into two types: (I) 'hard' boundary, and (II) inconsistent/unnatural intra-personal variations (lighting, pose, etc) between facial patches. These are illustrated in Fig. 2. Note that the type I artefacts are generated by not only our method but also 3D synthesis methods such as [27], [54], [12]. As shown in the top-right side of Fig. 2, the artefacts created by 3D methods are due to inaccurate 3D model to 2D image fitting. The inaccurate fitting makes the synthetic faces extract the pixels from background rather than facial areas, leading to bad facial boundaries.

Despite the existence of these artefacts, this synthetic data is still useful for training strong face recognition models. This can be understood from a few perspectives: (1) Type I artefacts are common to all the synthetic faces in the training set, therefore the CNN does not learn to rely on artefacts as discriminative features coding for identity, i.e., it learns artefact invariance. This means its performance is not compromised when subsequently presented with artefact-free images at testing-time. Other studies have also shown that synthetic data still improves recognition performance, despite the presence of type I artefacts [27]. (2) The artefacts can be regarded as noise, which has been shown to improve model generalisation in a variety of settings by increasing robustness and reducing overfitting. For example in the case of CNNs, training with data augmentation in the form of specifically designed deformation noise is important to obtain good recognition performance [61]; and in the case of de-noising auto encoders, training on images with noise, corruption and artefacts has been shown to improve face classification performance [56]. (3) As a concrete example to understand how type II artefacts can improve performance, consider two synthetic images with the same identity label, but one a type II artefact on the mouth caused by illumination, e.g., in Fig. 2 (bottom left). Training to predict these images as having the same identity means that the CNN learns an illumination-invariant feature for the mouth. And similarly for other intra-personal variations (such as pose, expression). Thus while some artefact images look strange, they are actually a powerful form of data augmentation that helps the CNN to learn robustness to all these nuisance factors.

Discrimination How do synthetic faces improve the performance of CNNs? Although some works [27], [54] empirically verified the effectiveness of synthetic data, they did not give much insight into how. In our work, the synthetic data captures a dataset of richer intra-personal variations by generating a large number of images of the same identities, leading to a 'deeper' training set. Also, our engine can synthesise a large number of faces of new identities, generating a 'wider' training set. Thus the synthetic identities interpolate the whole space of pixel-identity mappings. Not surprisingly, a better CNN model can be trained using this deeper and wider training set. The methods of generating our deeper and wider training set are detailed in Section V-A. In addition, as discussed in previous paragraph, the texture inconsistence between local patches caused by illumination and pose is helpful to learn the invariance to those nuisance factors.

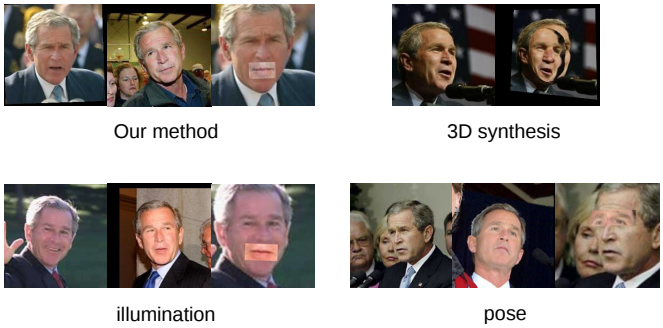

Fig. 2: Artefacts. Type I (Row 1) artefacts means hard boundaries generated by our method (top-left) and 3D synthesis method [27], [54], [12] (top right). Type II (Row 2) artefacts are inconsistent intra-personal variations such as those caused by inconsistent illumination (bottom left) and pose (bottom right).

IV. Face Recognition Pipeline

In this section we describe the different elements of our pipeline for face identification and verification in detail.

A. CNN architectures

Face recognition in the wild is a challenging task. As described in Section II-A, the existing deep learning methods highly depend on big training data. Very little research investigates training CNNs using small data. Recently, Hu [15] evaluated CNNs trained using small datasets. Due to the limited training samples, they found the performance of CNNs to be worse than handcrafted features such as high-dimensional features [6] (0.8763 vs 0.9318). In this work, we use a limited training set of around 10,000 images to synthesize a much larger one of around 1.5 million images for CNN training. The synthesized training data captures various deformable facial patterns that are important to improve the generalization capacity of CNNs.

We use two CNN architectures. The first one introduced in [15] has fewer filters and is referred as CNN-S, and the other introduced in [60] is much larger and therefore referred as CNN-L. These two architectures are detailed in Table I. Using the CNN-L model we achieve state-of-the-art performance on the LFW dataset [16] under 'unrestricted, label-free outside data' protocol.

B. NIR-VIS heterogeneous face recognition

NIR-VIS (near-infrared to visual) face recognition is important in applications where probe images are captured by NIR cameras that use active lighting which is invisible to the human eye [29]. Gallery images are, however, generally only

TABLE I: Our two CNN architectures

CNN-L	CNN-S
conv1	
32×3×3, st.1; 64 × 3 × 3, st.1 x2 maxpool, st.2	16×3×3, st.1; 16 × 3 × 3, st.1 x2 maxpool, st.2
conv2	
64×3×3, st.1; 128 × 3 × 3, st.1 x2 maxpool, st.2	32×3×3, st.1 x2 maxpool, st.2
conv3	
96×3×3, st.1 ; 192 × 3 × 3, st.1 x2 maxpool, st.2	48×3×3, st.1 x2 maxpool, st.2
conv4	
128×3×3, st.1 ; 256 × 3 × 3, st.1 x2 maxpool, st.2	-
conv5	
160×3×3, st.1 ; 320 × 3 × 3, st.1 x7 avgpool, st.1	-
fully connected	
Softmax-5000	FC-160 Softmax-5000

available in the visible spectrum. The existing methods for NIR-VIS face recognition include three steps: (i) illumination pre-processing, (ii) feature extraction, and (iii) metric learning. First, the NIR-VIS illumination differences cause the main difficulty of NIR-VIS face recognition. Therefore, illumination normalization methods are usually used to reduce these differences. Second, to reduce the heterogeneities of NIR and VIS images, illumination-robust features such as LBP are usually extracted. Third, metric learning is widely utilized, aiming at removing the differences of modalities and meanwhile keeping the discriminative information of the extracted features.

In this work, we also follow these three steps that are detailed in Section V-B3. Unlike the existing work that extracts handcrafted features, we learn face representations using two CNN architectures introduced in Section IV-A. To our knowledge, we are the *first* to use deep CNNs for NIR-VIS face recognition. The main difficulty of training CNNs results from the lack of NIR training images which are not available from the Internet. To solve this, we synthesize a big NIR dataset for CNN training.

C. Network Fusion

Fusion of multiple networks is a widely used strategy to improve the performance of deep CNN models. For example, in [43], an ensemble of seven networks is used to improve the object recognition performance due to complementarity of the models trained at different scales. Network fusion is also successfully applied to learn face representations. DeepID and its variants [45], [49], [50] train multiple CNNs using image patches extracted from different facial parts.

The heterogeneity of NIR and VIS images is intrinsically caused by the different spectral bands from which they are acquired. The images in both modalities, however, are reflective in nature and affected by illumination variations. Illumination normalization can be used to reduce such variability, at the risk of loosing identity-specific characteristics. In this work, we fuse two networks that are trained using the original and illumination-normalized images respectively. This network fusion significantly boosts the recognition rate.

D. Metric Learning

The goal of metric learning is to make different classes more separated, and instances in the same class closer. Most approaches learn a Mahalanobis metric

$$d_A^2(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j) \tag{1}$$

which maximizes inter-class discrepancy, while minimizing intra-class discrepancy. Some methods, instead, learn a generalized dot-product of the form

$$d_B^2(x_i, x_j) = x_i^T B x_j \tag{2}$$

Metric learning methods are widely used for face identification and verification. Because identification and verification are two different tasks, different loss functions should be optimized to learn the metric. Joint Bayesian metric learning (JB) [5] and Fisher linear discriminant analysis (LDA) are probably the two most widely used metric learning methods for face verification and identification respectively. In particular, LDA can be seen as a method to learn a metric of the form of Eq. (1), while JB learns a verification function that can be written as a weighted sum of Eq. (1) and (2). In our work we use JB and LDA to improve the performance of face verification and identification respectively.

V. Experiments

A. Data synthesis methods

Given some face images and their IDs, we define three synthetic strategies: *Inter-Synthesis*, *Intra-Synthesis*, and *Self-Synthesis*. *Inter-Synthesis* synthesizes a new image using two parents from different IDs as shown in Fig. 1. The facial components of an *Intra-Synthesized* face are from different images with the same ID. *Self-synthesis* is a special cause of *Intra-Synthesis*. Specifically, one given image synthesizes new images by swapping facial components of itself and its mirrored images. By virtue of *Self-Synthesis*, one input image can become maximum 32 images which have complementary information. The fusion of features extracted from these 32 images has stronger face representation capacity which is validated in Section V-B2. In the view of NIR-VIS cross-modality, we also define 'cross-modality synthesis' which uses images from different modalities to synthesize a new one. Some synthetic images from the CASIA NIR-VIS 2.0 dataset with LSSF [59] illumination normalization are shown in Fig. 3. The reasons of using LSSF illumination normalization is detailed in Section V-B3. As shown in Fig. 3, the results of *Intra-Synthesis* method are usually more natural than *Inter-Synthesis* method since the *Intra-Synthesis* method uses the same ID. However, as shown in the right of Fig. 3, some samples from *Intra-Synthesis* can also be very strange due to large pose variations.

B. Face recognition

1) Implementation details: Our implementation is based on the Caffe open source deep learning toolbox. Before face synthesis, all the raw images are aligned and cropped to size 100×100 as in [60] on both datasets. We train our

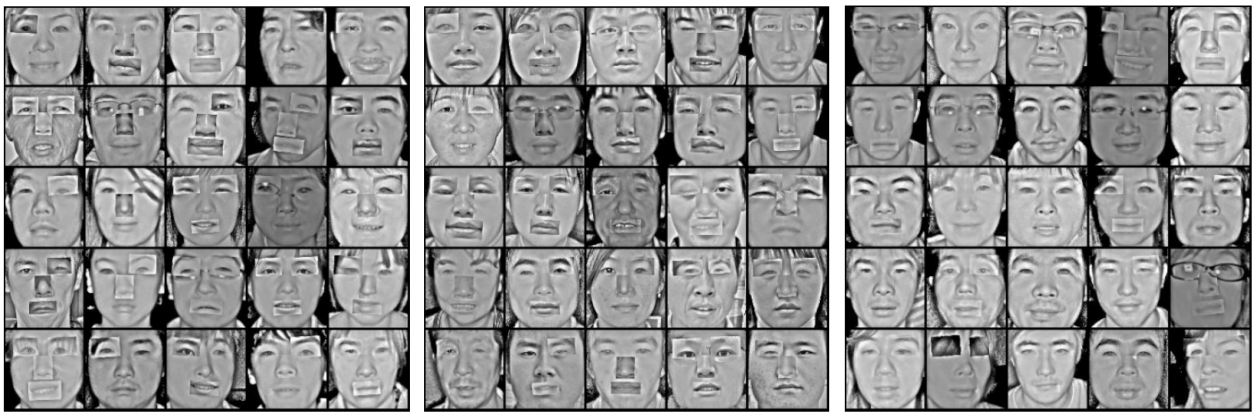

Fig. 3: Left: Inter-Synthesis. Middle: cross-modality Intra-Synthesis. Right: Intra-Synthesis.

models using images only from LFW and CASIA NIR-VIS2.0 databases. For the CNN-S model on both datasets, we set the learning rate as 0.001, and decrease it by 10 times every 4000 iterations, and stop training after 10K iterations. We practically find dropout is not helpful for the small network, therefore, we train the CNN-S model without dropout. For the CNN-L model on the NIR-VIS dataset, we set the learning rate as 0.01, and decrease it by 10 times every 8000 iterations, and stop training after 20K iterations. For the CNN-L model on the LFW dataset, we set the learning rate as 0.01, and decrease it by 10 times every 120K iterations, and stop training after 200K iterations. We set dropout rate as 0.4 for the pool5 layer of the CNN-L model. For both CNN-S and CNN-L models, the batch size is 128, momentum is 0.9, and decay is 0.0005. Softmax loss function is used to guide CNN training. The features used in our experiments of CNN-S and CNN-L are FC-160 (160D) and Pool5 (320D), respectively.

2) Face recognition in the wild:

a) Database and protocol: Labeled Faces in the Wild (LFW) [16] is a public available dataset for unconstrained face recognition study. It contains 5,749 unique identities and 13,233 face photographs. The training and test sets are pre-defined in [16]. For evaluation, the full dataset is divided into ten splits, and each time nine of them are used for training and the left one for testing. Our work falls in the protocol of 'Unrestricted, Label-Free Outside Data' as we use the identity information to train the neural network (softmax loss). Meanwhile, all face images are aligned using a model trained on unlabeled outside data. As a benchmark for comparison, we report the mean and standard deviation of classification accuracy.

b) Synthetic data generation: Under LFW protocol, the training set in each fold is different. Therefore, the size of synthetic data and the original raw LFW data in Table II is averaged over 10 folds. We generate 1.5 million training images including 1 million 'Inter-Syn' ones and 0.5 million 'Intra-Syn' ones. 'Inter-Syn' and 'Intra-Syn' are defined in Section V-A.

c) Analysis of CNN model and synthetic data: We here analyse the trained model by visualising the synthetic images in feature space. We choose the images of two subjects

TABLE II: Training data synthesized from LFW

		IDs	Images	Images/ID
Synthetic	Intra-Syn	5K	500K	100
	Inter-Syn	5K	1M	200
	Total	10K	1.5M	150
Raw		5K	10K	2

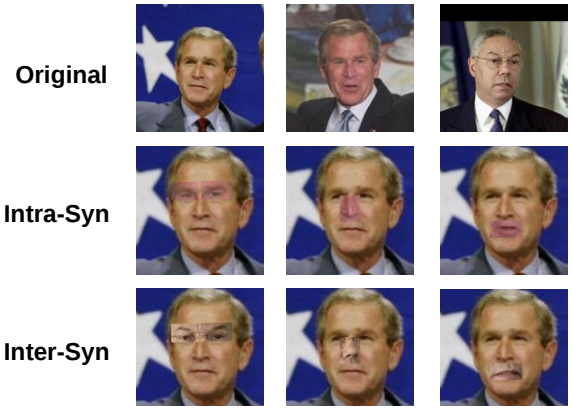

Fig. 4: Samples of Inter-Synthesis and Intra-Synthesis

(George W. Bush and Colin Powell), which have the largest number of images in LFW, and the synthetic images derived from the two subjects for analysis. To analyse the effects of each single facial component, we only replace Bush's one particular patch with the one from himself (Intra-Synthesis) or Powell (Inter-Synthesis) as shown in Fig. 4. For each case, 100 images are synthesised. Therefore, there are 8 groups of images: 2 groups of original images (Bush and Powell), 3 Intra-Synthesis (one of three components is replaced by images of Bush) and 3 Inter-Synthesis (by images of Powell). These images are fed into one CNN-L to extract features, which are then projected to a PCA space. The first 2 PCA components of each feature are shown in Fig. 5.

Fig. 5(a)-(c) show the face distributions if one particular facial component is replaced. In Fig. 5(a), 3 identities, 'GeorgeWBush+intra-Eyes', 'ColinPowell' and 'inter-Eyes' are well separated. It means that the identity information is kept if Bush's eyes are replaced by those from himself. In

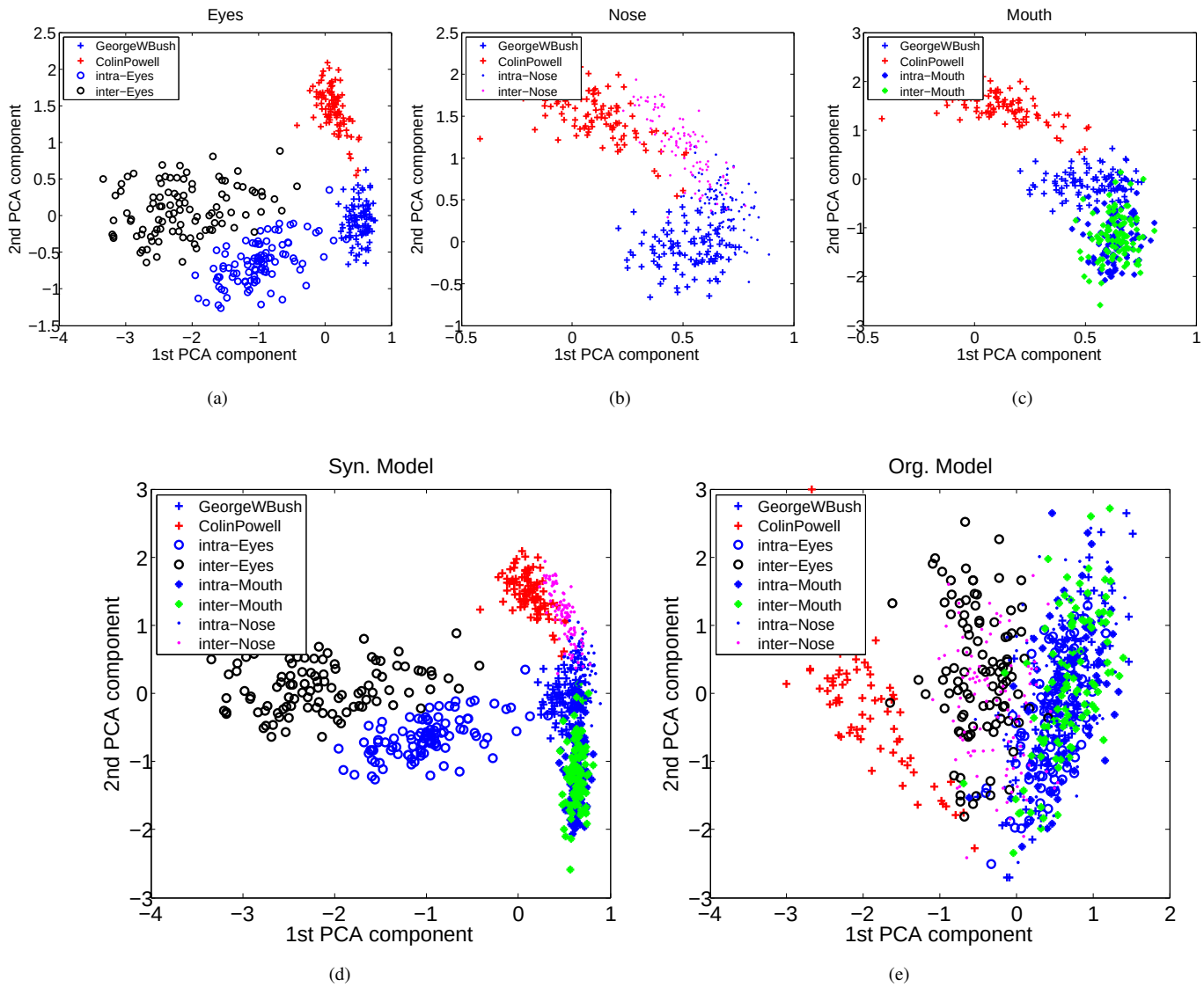

Fig. 5: Face distributions. 'GeorgeWBush' and 'ColinPowell' denote the original images from these 2 subjects. 'intra-X' and 'inter-X' denote the synthetic images with component X replaced by another example from Bush himself and from Powell respectively, as in Fig. 4. Features of (a)-(d) are extracted using a CNN trained using original and synthetic data, while (e) using original data only. One colour represents one (real or synthetic) subject.

contrast, a new identity space is generated by the synthetic images that replace Bush's eyes with Powell's. The same conclusion can be drawn from nose in Fig. 5(b). In Fig. 5(c), however, 'intra-Mouth' and 'inter-Mouth' are not well separated. It means that the mouth component only is not very discriminative between people. Fig. 5(d) and (e) contrast the results when training with original and synthetic (Fig. 5(d)) versus original data only (Fig. 5(e)). Fig. 5(d) is relatively more discriminative for identity, particularly the synthetic identities. Thus we can see that training with synthetic data is important to interpolate the identity space, and thus achieve good results for unseen identities – as required to obtain good performance at testing time.

d) Impact of synthetic data: Table III analyzes the importance of using the synthetic data. First, CNN-S trained using synthetic data ('Intra-Syn' and 'Inter-Syn') outperforms greatly that trained using raw LFW images, showing the importance of data synthesis. Second, 'Inter-Syn' works slightly better than 'Intra-Syn' because 'Inter-Syn' can capture richer facial variations. Third, combining 'Inter-Syn' and 'Intra-Syn' works better than either of them because both inter- and intra-personal variations can be captured. Fourth, averaging the features of 32 'Self-Syn' ('32-Avg' in Table III and defined in Section V-A) images works consistently better than that of one single test image ('single' in Table III). Fifth, CNN-L works consistently better than CNN-S using either raw LFW or synthetic images because deeper architecture has stronger generalization capacity. Last but not least, the metric learning (JB) can further enhance the face recognition performance.

e) Impact of Image Blending: In Section III, we claim the existence of artifacts ('hard boundaries') can improve the robustness of the model. However, it is interesting to

TABLE III: Comparison of synthetic data methods on LFW

Architecture	Metric learning	Training data	single (%)	32-Avg (%)
CNN-S	-	Raw	78.97 ± 0.78	-
		Intra-Syn+Raw	83.03 ± 0.56	83.93 ± 0.49
		Inter-Syn+Raw	83.18 ± 0.74	84.35 ± 0.65
		Intra-Syn+Inter-Syn+Raw	85.61 ± 0.71	86.98 ± 0.57
CNN-L	-	Raw	85.03 ± 0.98	-
	JB [5]	Raw	87.03 ± 0.69	-
	-	Intra-Syn+Inter-Syn+Raw	94.88 ± 0.66	95.13 ± 0.53
	JB [5]	**Intra-Syn+Inter-Syn+Raw**	$\mathbf{95.32 \pm 0.38}$	$\mathbf{95.77 \pm 0.38}$
	-	Intra-Syn+Inter-Syn+Raw (blending)	94.27 ± 0.65	94.46 ± 0.51
	JB [5]	Intra-Syn+Inter-Syn+Raw (blending)	94.61 ± 0.35	95.05 ± 0.34

know whether it is useful to remove or smooth those 'hard boundaries' by image blending technique. In this work, we implemented poisson image editing [33] to smooth these boundaries. In Fig. 6, we show some results of poisson image editing. From Fig. 6, as expected, Poisson blending does make the boundaries much smoother compared with our synthesis method. In Table III, we compare the results with and without poisson image editing. We can see that the accuracy of our synthesis with 'hard' boundaries is higher than that with Poisson blending. It results from the fact that the 'hard' boundaries working as noises make the network training more robust as analysed in Section III.

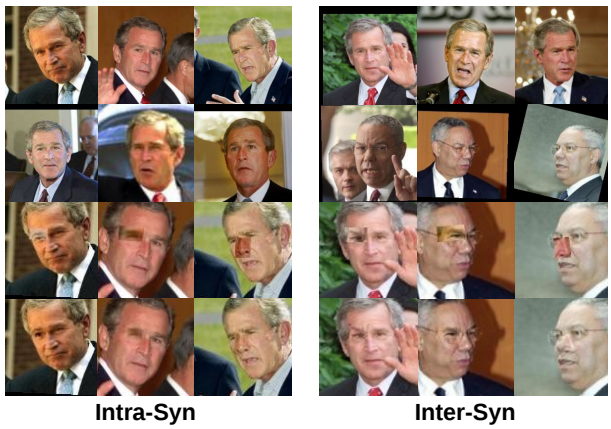

Fig. 6: Visualisation of Poisson blending. Row 1-2: Input image pairs, Row 3: Our synthesis with 'hard boundaries', Row 4: Poisson blending

f) Comparison with the state-of-the-art: Table IV compares our method with state-of-the-art methods. All methods listed in Table IV except ours use hand-crafted features, and this again indicates the hardness of training deep CNNs with small data. In fact, the best deep learning solution [48] recorded in official benchmark achieves 91.75%, and ours is 4% better. In addition, most of state-of-the-art solutions rely on an extremely high dimensional feature vector because they fundamentally employ dense sampling on the face image, in contrast, we just use a 320-dimensional feature vector, which is much more compact than the others.

g) Non-CNN methods using synthetic data: The previous results show the effectiveness of synthetic data with deep learning methods. It is also interesting to investigate the usefulness of the synthetic big data in improving methods based on traditional hand-crafted features. One typical hand-crafted feature used for unconstrained face recognition problem is high dimensional LBP (HD-LBP) [6]. To further improve performance, it is common practice to train metric learners, such as Joint Bayesian (JB) [5], based on the extracted HD-LBP features [6]. We extract the HD-LBP feature using the open source code [4]. From Table V, we see that HD-LBP with JB trained using original LFW images works much better that without JB (89.02% vs 84.13%), showing the expected effectiveness of metric learning. More interestingly, we see that training JB using both original and synthetic images outperforms that using original images only, 91.03% vs 89.02%. This shows that our synthetic data approach is useful in a conventional hand-crafted feature+metric learning pipeline, as well as in a deep feature learning context.

TABLE IV: Comparison with state-of-the-art methods on LFW

Methods	Accuracy (%)
Fisher vector faces [41]	93.03 ± 1.05
HPEN [65]	95.25 ± 0.36
MDML-DCPs [8]	95.58 ± 0.34
The proposed	$\mathbf{95.77 \pm 0.38}$

TABLE V: Hand-crafted features on LFW

Methods	Accuracy (%)
HD-LBP	84.13 ± 1.76
HD-LBP+JB (raw)	89.02 ± 1.11
HD-LBP+JB (raw + synthetic)	91.03 ± 1.06

h) The effectiveness of the synthetic images for bigger training set: In Table III, we already show the effectiveness of synthetic data derived from a small training set (LFW, 5K identities and 10K images). Currently, there are some bigger training set of face images in the wild, such as CASIA WEBFACE [60] (10K identities and 0.5M images), it is interesting to know whether our data synthesis strategy is useful for such bigger training data. Therefore, we conduct the experiment using original CASIA WEBFACE images and synthetic data derived from them. Similar to Table II, we keep the ratio 2:1 of 'Inter-Syn' and 'Intra-Syn' synthetic images. Our CNN-L network is used for this evaluation. From Fig 7, we can see that the recognition rates on LFW increase with the increase of the size of synthetic data. It means our data

synthesis strategy is still very effective even when the training data is much bigger.

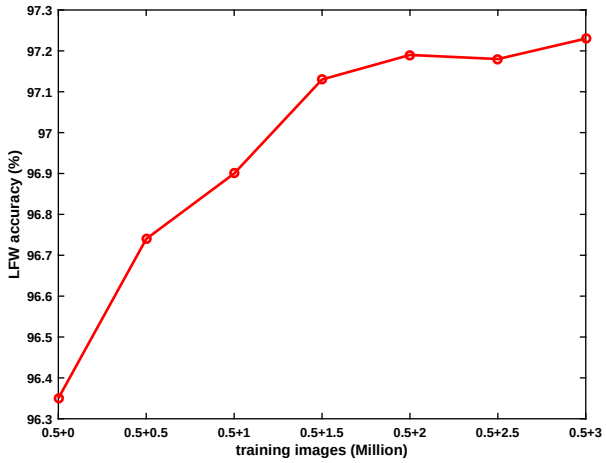

Fig. 7: Recognition rate (%) on LFW using CASIA WEBFACE training data (0.5M) and synthetic data (0-3M).

3) NIR-VIS face recognition:

a) Database and protocol: The largest face database across NIR and VIS spectrum so far is the CASIA NIR-VIS 2.0 face database (CASIA NIR-VIS2.0) [23]. It contains 17,580 images of 725 subjects which exhibit intra-personal variations such as pose and expression. This database includes two views: view 1 for parameter tuning and view 2 including 10 folds for performance evaluation. During test, the gallery and probe images are VIS and NIR images respectively, simulating the scenario of face recognition in the dark environment. The rank 1 identification rate including the mean accuracy and standard deviation of 10 folds are reported.

b) Synthetic data generation: We synthesize training samples using the existing images in CASIA NIR-VIS2.0. Because the images of CASIA NIR-VIS2.0 are from two modalities (NIR and VIS), we applied 'cross-modality synthesis' to synthesize new images. The size of synthesized data is detailed in Table VI.

TABLE VI: Synthetic data using CASIA NIR-VIS2.0 database

		IDs	Images	Images/ID
Synthetic	Intra-Syn	357	90K	250
	Inter-Syn	1K	150K	150
	Total	1.4K	240K	170
Raw		357	8.5K	23

c) Illumination normalization and feature extraction: Illumination Normalization (IN) methods are usually used to narrow the gab between NIR and VIS images. To investigate the impact of IN, we preprocessed images using three popular IN methods: illumination normalization based on large- and small-scale features (LSSF) [59], DoG filtering-based normalization (DOG) and single-scale retinex (SSR) [19]. We train CNN-S and CNN-L using illumination normalized and non-normalized images. For simplicity, only the images from CASIA NIR-VIS2.0 excluding synthetic ones are used. Fig. 8 shows the face recognition rates at different training iterations using different input images. In Fig. 8 and 9, three IN methods outperform 'GRAY' which does not use IN method, showing the effectiveness of IN. Note that LSSF achieves the best performance due to its strong capacity of removing illumination but keep identity information. Same as the LFW results in Section V-B2, CNN-L works better than CNN-S.

TABLE VII: Evaluation of the impact of synthetic data

	Training Data		Accuracy(%)
	CASIA NIR-VIS2.0	LFW	
Baseline	Raw	-	17.41 ± 3.76
Synthetic Data	Raw+Syn	-	34.13 ± 2.13
	-	Raw	38.45 ± 2.08
	-	Raw+Syn	66.37 ± 1.45
	Raw+Syn	Raw+Syn	$\mathbf{68.97 \pm 1.24}$

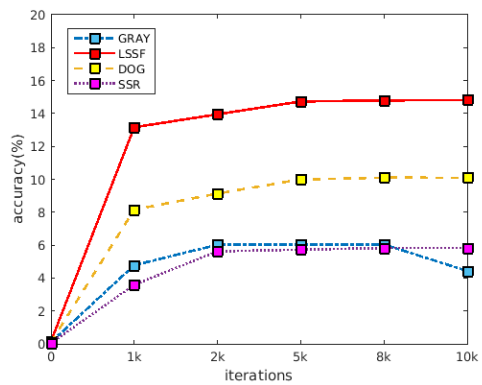

Fig. 8: Illumination normalization methods using CNN-S

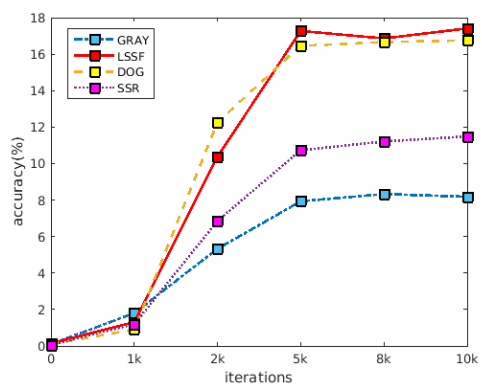

Fig. 9: Illumination normalization methods using CNN-L

The use of illumination-robust features can effectively narrow the gap between NIR and VIS images. The most commonly used hand-crafted feature is LBP. Since LSSF achieves the best performance, we extract LBP features from LSSF-normalized images and achieve 12.48 ± 3.1 in comparison with 17.41 ± 3.76 by CNN-L learned feature. It shows the superior performance of CNN learned features.

d) Effects of synthetic data: To evaluate the effects of synthetic data on a common ground, LSSF is used to preprocess the illumination and CNN-L is applied to learn face

representations for all (raw and/or synthetic) input images. In the practice, we find two problems with the synthetic data generated from CASIA NIR-VIS2.0 database: (1) It cannot capture enough facial variations because it only has 357 subjects as shown in Table VI. (2) There are much fewer VIS images than NIR ones. To solve these two problems, we also use the synthetic data generated from LFW images defined in Table II. Table VII compares the results achieved by these two sources of synthetic data. First, the accuracy achieved by using the synthetic data generated from CASIA NIR-VIS2.0 database is 34.13 ± 2.13, in comparison with 17.41 ± 3.76 without synthetic data. The significant improvement shows the effectiveness of data synthesis. Second, the model trained using raw and synthetic LSSF-normalized LFW images greatly outperforms those synthetic CASIA NIR-VIS2.0 images: 66.37 ± 1.45 vs 38.45 ± 2.08, although NIR images are completely unseen during training. The reasons are 2-fold: (1) LFW images contains more subjects which can capture more facial variations as analyzed above. (2) LSSF can greatly reduce the gap between NIR and VIS, therefore, LSSF-normalized LFW synthetic images can generalize well to LSSF-normalized NIR images. To further improve the face recognition performance, we trained the network using both raw and synthetic data from both sources (LFW+CASIA NIR-VIS). The face recognition rate is improved from 66.37 ± 1.45 to 68.97 ± 1.24, showing the value provided by our of bigger synthetic dataset.

e) Comparison with the state-of-the-art: The CNN-Ls in Table VIII are all trained using synthetic LFW data. First, LSSF-normalized and Original LFW synthetic data achieve very comparable performance: 68.97% vs 69.11%. However, the fusion (averaging) of these 2 features can significantly improve the face recognition rates. It shows the fusion can keep the discriminative facial information but remove the illumination effects. Second, not surprisingly, metric learning can further improve the performance. The metric learning method used here is LDA, which is the most widely used one for face identification. Last, Table VIII compares the proposed method against the state-of-the-art solutions [26], [20]. [26] uses a designed descriptor that performs better in this dataset compared with other general hand-crafted features, and LDA can further improve the accuracy. Our method significantly outperforms [26] in the case that LDA is not employed, while it earns 4% advantage with the help of LDA. [20] tries to solve the domain shift between two data sources (NIR and VIS) by a cross-modal metric learning: it assumes that a pair of NIR and VIS images shares the same sparse representation under two jointly learned dictionaries. Our method beats [20] with a 7% margin without such an extra step of dictionary learning.

VI. Conclusions and Future Work

Recently, convolutional neural networks have attracted a lot of attention in the field of face recognition. However, deep learning methods heavily depend on big training data, which is not always available. To solve this problem in the field of face recognition, we propose a new face synthesis method which swaps the facial components of different face images to generate a new face. With this technique, we achieve state-of-the-art face recognition performance on LFW and CASIA NIR-VIS2.0 face databases.

In the future, we will apply this technique to more applications of face analysis. For example, the proposed data synthesis method can easily be used in training CNN-based face detection, facial attribute recognition, etc. More generally, the method applies to any objects which are well structured. For example, the human body is well structured and human images can be synthesised using this method. The synthetic images can be used to train deep models for pose estimation, pedestrian detection, and person re-identifiation.

Acknowledgment

This work is supported by the ANR project 'Physionomie', the European Unions Horizon 2020 research and innovation program under grant No 640891, and Natural Science Foundation of China (No. 61502152). This work has also been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01). We also gratefully acknowledge NVIDIA Corporation for the donation of the GPUs for this research.

TABLE VIII: Comparisons with State-of-the-art on CASIA NIR-VIS2.0 Database

Method			Accuracy (%)
CNN-L	Training Data	Original	69.11 ± 1.21
		LSSF	68.97 ± 1.24
	Network Fusion	Original+LSSF	79.96 ± 1.18
	Metric Learning	LDA (Original+LSSF)	$\mathbf{85.05} \pm \mathbf{0.83}$
State-of-the-art	C-CBFD [26]		56.6 ± 2.4
	C-CBFD+LDA [26]		81.8 ± 2.3
	Dictionary Learning [20]		78.46 ± 1.67

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Computer vision-eccv 2004*, pages 469–481. Springer, 2004.

[2] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila. Recognition of blurred faces using local phase quantization. In *International Conference on Pattern Recognition*, 2008.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *PAMI*, 19(7):711–720, 1997.

[4] W. H. H. Bor-Chun Chen, Chu-Song Chen. Review and implementation of high-dimensional local binary patterns and its application to face recognition. 2014.

[5] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Computer Vision–ECCV 2012*, pages 566–579. Springer, 2012.

[6] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3025–3032. IEEE, 2013.

[7] D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46:161–190, 2002.

[8] C. Ding, J. Choi, D. Tao, and L. S. Davis. Multi-directional multi-level dual-cross patterns for robust face recognition. *arXiv preprint arXiv:1401.5311*, 2014.

[9] Z. Feng, G. Hu, J. Kittler, W. Christmas, and X. Wu. Cascaded collaborative regression for robust facial landmark detection trained using a mixture of synthetic and real images with dynamic weighting. 24(11):3425–3440, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[11] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *ICCV*, 2009.

[12] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. *arXiv preprint arXiv:1411.7964*, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[14] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using Laplacianfaces. *PAMI*, 27(3):328–340, 2005.

[15] G. Hu, Y. Yang, D. Yi, J. Kittler, W. J. Christmas, S. Z. Li, and T. M. Hospedales. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. *CoRR*, abs/1504.02351, 2015.

[16] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[18] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[19] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE Trans. Image Processing*, 6(3):451–462, 1997.

[20] F. Juefei-Xu, D. Pal, and M. Savvides. Nir-vis heterogeneous face recognition via cross-spectral joint dictionary learning and reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 141–150, 2015.

[21] S. Kong, J. Heo, B. Abidi, J. Paik, and M. Abidi. Recent advances in visual and infrared face recognition – a review. *CVIU*, 97(1):103 – 135, 2005.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] S. Z. Li, D. Yi, Z. Lei, and S. Liao. The casia nir-vis 2.0 face database. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 348–353. IEEE, 2013.

[24] J. Liu, Y. Deng, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015.

[25] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[26] J. Lu, V. E. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.

[27] I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? *ECCV*, 2016.

[28] D. Miller, E. Brossard, S. M. Seitz, and I. Kemelmacher-Shlizerman. Megaface: A million faces for recognition at scale. *arXiv preprint arXiv:1505.02108*, 2015.

[29] S. Ouyang, T. Hospedales, Y.-Z. Song, and X. Li. A survey on heterogeneous face recognition: Sketch, infra-red, 3d and low-resolution. *arXiv preprint arXiv:1409.5114*, 2016.

[30] J. Papon and M. Schoeler. Semantic pose using deep networks trained on synthetic rgb-d. *arXiv preprint arXiv:1508.00835*, 2015.

[31] O. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[32] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. Transformation pursuit for image classification. In *CVPR*, 2014.

[33] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*. ACM, 2003.

[34] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[35] A. Rozantsev, V. Lepetit, and P. Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 2015.

[36] A. Rozantsev, V. Lepetit, and P. Fua. On rendering synthetic images for training an object detector. *CVIU*, 137:24 – 37, 2015.

[37] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[38] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[39] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[40] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression (PIE) database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[41] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *British Machine Vision Conference*, 2013.

[42] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[44] H. Su, C. Qi, Y. Li, and L. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015.

[45] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[46] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[47] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *ICCV*, 2013.

[48] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1489–1496. IEEE, 2013.

[49] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition*

(CVPR), 2014 IEEE Conference on, pages 1891–1898. IEEE, 2014.
[50] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *arXiv preprint arXiv:1412.1265*, 2014.
[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.
[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.
[53] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.
[54] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.
[55] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591, 1991.
[56] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.
[57] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.
[58] M. Weinmann, J. Gall, and R. Klein. Material classification based on training data synthesized using a BTF database. In *ECCV*, 2014.
[59] X. Xie, W.-S. Zheng, J. Lai, P. C. Yuen, and C. Y. Suen. Normalization of face illumination based on large-and small-scale features. *IEEE Trans. Image Processing*, 20(7):1807–1821, 2011.
[60] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
[61] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision*, 2016.
[62] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.
[63] Z. Zhang, P. Luo, C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.
[64] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. *arXiv preprint arXiv:1511.07212*, 2015.
[65] 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.

Guosheng Hu is a postdoctoral researcher in THOTH team, INRIA Grenoble Rhone-Alpes, France. He received his PhD degree in the Centre for Vision, Speech and Signal Processing, University of Surrey, UK in 2015. His research interests include pattern recognition, biometrics, machine learning and graphics. His PhD topic is 'face analysis using 3D morphable models. He is an IEEE member.

Xiaojiang Peng is a postdoctoral researcher in THOTH team, INRIA Grenoble Rhone-Alpes, France. He received his PhD degree in Computer Science at Southwest Jiaotong University in 2014. His research focus is in the areas of action recognition and deep learning.

Yongxin Yang is a PhD student at the department of Electronic Engineering and Computer Science, Queen Mary University of London (QMUL). His research interests include machine learning (Transfer Learning, Domain Adaptation, Multi-Task Learning and Deep Learning) and computer vision.

Timothy Hospedales received the PhD degree in neuroinformatics from the University of Edinburgh in 2008. He is currently a Reader (associate professor) at The University of Edinburgh. His research interests include probabilistic modelling and machine learning, particularly life-long, transfer, and active learning with applications to computer vision and beyond.

Jakob Verbeek received a PhD degree in computer science in 2004 from the University of Amsterdam, The Netherlands. After being a postdoctoral researcher at the University of Amsterdam and at INRIA Rhone-Alpes, he has been a full-time researcher at THOTH group, INRIA, Grenoble, France, since 2007. His research interests include machine learning and computer vision, with special interest in applications of statistical models in computer vision.