International Journal of Computer Vision (2022) 130:344–362
https://doi.org/10.1007/s11263-021-01536-x

Personalized Convolution for Face Recognition

Chunrui Han[1,2] · Shiguang Shan[1,2,3] · Meina Kan[1,2] · Shuzhe Wu[1,2] · Xilin Chen[1,2]

Received: 3 April 2021 / Accepted: 27 September 2021 / Published online: 4 January 2022

Abstract
Face recognition has been significantly advanced by deep learning based methods. In all face recognition methods based on convolutional neural network (CNN), the convolutional kernels for feature extraction are fixed regardless of the input face once the training stage is finished. By contrast, we humans are usually impressed by some unique characteristics of different persons, such as one's blue eyes while another one's crooked nose, or even someone's naevus at specific location. Inspired by this observation, we propose a personalized convolution method which aims to extract special distinguishing characteristics of each person for more accurate face recognition. Specifically, given a face, we adaptively generate a set of kernels for him/her, named by us ordinary kernel, which is further analytically decomposed into two orthogonal components, i.e., the commonality component and the specialty component. The former characterizes the commonality among subjects which is optimized on a reference set. The latter is the residual part by filtering out the commonality component from the ordinary kernel, so as to capture those special characteristics, named by us personalized kernel. The CNNs with personalized kernels for convolution can highlight those specialty of a person's distinguishing characteristics while suppress his/her commonality with others, leading to better distinguishing of different faces. Additionally, as a by-product, the reference set also facilitates the adaptation of our method to different scenarios by simply selecting faces of a particular population. Extensive experiments on the challenging LFW, IJB-A and IJB-C datasets validate that our proposed personalized convolution achieves significant improvement over the conventional CNN, and also other existing methods for face recognition.

Keywords Face recognition · Personalized convolution · Personalized kernel

1 Introduction

Face recognition aims to identify or verify the identity of a person by using his/her face. As an effective method of biometric authentication, it has been deployed in various venues, ranging from public and finance security, to Face ID on mobile phone and conference registration. Generally, there are two types of face recognition tasks: face identification and face verification. The former classifies a given face to a specific identity in the gallery, while the latter aims at determining whether a pair of faces are of the same identity. In either cases, discriminative representation of face images is the key to high-accuracy face recognition.

In recent years, the most successful face recognition technology employs the powerful convolutional neural network (CNN). This is because a deep CNN model can extract informative face features with stable invariance to complex appearance variations of one's face, and high separability even between millions of faces, benefited from its excellent capability of modeling non-linearity. A few impressive

Communicated by Bumsub Ham.

✉ Meina Kan
kanmeina@ict.ac.cn

Chunrui Han
chunrui.han@vipl.ict.ac.cn

Shiguang Shan
sgshan@ict.ac.cn

Shuzhe Wu
shuzhe.wu@vipl.ict.ac.cn

Xilin Chen
xlchen@ict.ac.cn

[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, Shanghai, China

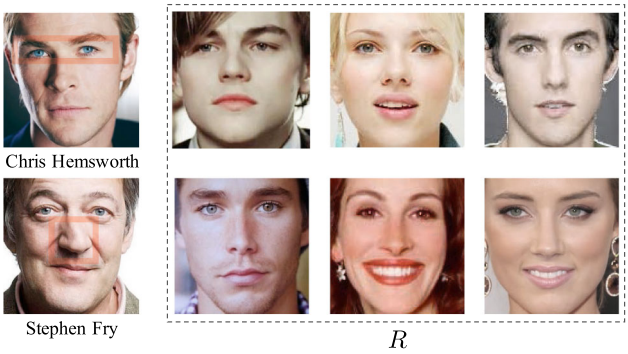


Fig. 1 Illustration that distinguishing characteristics differ from one person to another. For instance, one can easily notice Chris Hemsworth's blue eyes, while notice Stephen Fry's crooked nose relative to the reference set R

studies include DeepFace (Taigman et al., 2014), Deep ID series (Chen et al., 2014; Sun et al., 2014, 2015a, b), FaceNet (Schroff et al., 2015), VGGFace (Parkhi et al., 2015), SphereFace (Liu et al., 2017), CosFace (Wang et al., 2018), ArcFace (Deng et al., 2018), UniformFace (Duan et al., 2019), RegularFace (Zhao et al., 2019) etc. In these CNN-based approaches, firstly one needs to train a CNN model on the training set, and then extract the deep features of testing faces with this model for face recognition. The parameters in CNN are fixed once training is finished, so all testing face images are processed with identical kernels.

By contrast, we humans are usually impressed by distinct characteristics of different persons, such as one's blue eyes while another one's crooked nose, as shown in Fig. 1. Inspired by the above observation about human's face perception, this work proposes a ***personalized convolution*** method to extract the unique features of each person for better face recognition, in which personalized kernel is adaptively generated for each input face image to highlight his/her special distinguishing characteristics.

To achieve the above goal, for each individual, we first leverage a kernel generator to create a set of kernels for him/her, called by us ordinary kernel, which contains his/her compound information. To further filter out the common information and preserve the special information, the ordinary kernel is then decomposed into two orthogonal components, i.e. the commonality component and specialty component. The former (denoted as commonality kernel) depicts the commonality shared by all persons and is analytically computed with the aid of a reference set containing multiple face images as a proxy of all the persons. Then, the personalized kernel of the individual is obtained as the residual of orthogonal projection of his/her ordinary kernel on the commonality kernel. As a by-product, the proposed method can be used as a population-adaptive method by simply constructing the reference set with face images of a target population.

When there is only one face image in the reference set, the personalized kernel degenerates to a special case, i.e. the contrastive kernel proposed in our previous work (Han et al., 2018). Only in this sense, this work can be regarded as an extension of our conference version (Han et al., 2018). However, they are significantly different in three aspects. *Firstly*, the motivation is different. The conference version emphasizes the difference between two faces under comparison mainly for verification, while this paper emphasizes the distinctions of one face from the public faces. *Secondly*, the framework and formulation are different. This paper proposes a more general framework by introducing a reference set, while the conference version can be seen as a special variant of this framework. Moreover, the personalized kernel is mathematically optimized by decomposing the ordinary kernel into two orthogonal components, i.e. commonality component and specialty component, instead of the intuitive subtraction of the ordinary kernels of two faces in the conference version. *Thirdly*, as a by-product, the proposed method can be exploited as a population-adaptive method by simply constructing the reference set with face images of a target population. This paper achieves significant performance improvement over the conference version up to 5% on IJB-C with a 16-layer network, and also includes more experimental comparisons and analyses.

Briefly, the contributions of this work are in three folds.

- *Firstly* different from existing methods, our personalized convolution can adaptively extract the special distinguishing features of an input face for adaptive and better face recognition.
- *Secondly* a general framework decomposing one's ordinary kernel into commonality component and specialty component is proposed to generate the personalized kernel for extracting one's special distinguishing features. A reference set is introduced and serves as a strainer to filter out the commonality component from the ordinary kernel while leaving the specialty component as the personalized kernel. Different constitutions of the reference set derive distinct variants of this framework with potential for varying application scenarios.
- *Thirdly* our method achieves quite impressive face recognition accuracy on the challenging LFW, IJB-A and IJB-C datasets, demonstrating the effectiveness of personalized feature extraction for face recognition.

2 Related Work

In this section, we first briefly review the existing methods for face recognition, of which all adopt the same feature extraction for testing face images. Besides, we revisit some related

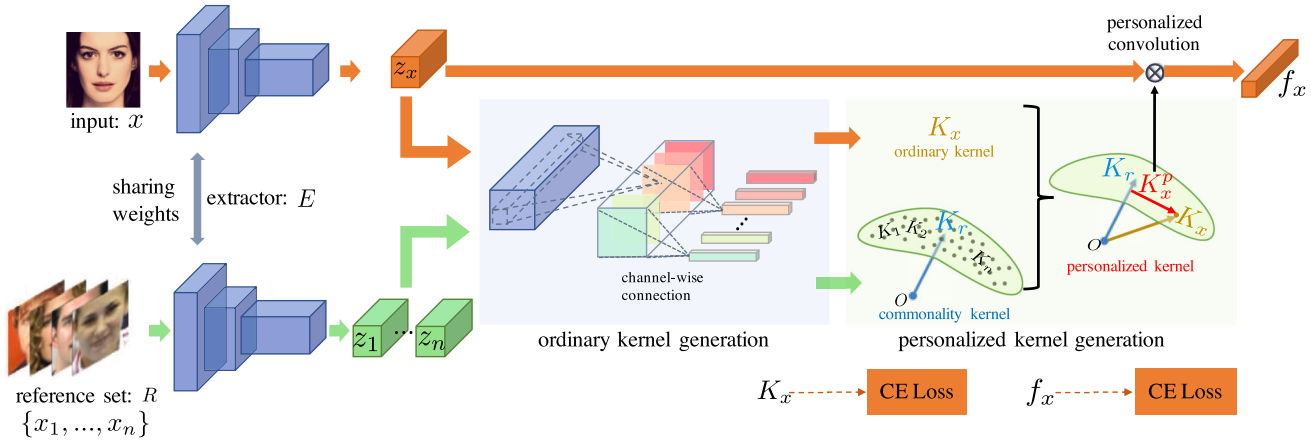


Fig. 2 The pipeline of our personalized CNN. Given an input face image x and a reference face set $R = \{x_1, ..., x_n\}$, a basic feature extractor E consisting of several cascaded convolutional layers is firstly used to obtain base feature representations z_x and $\{z_1, ..., z_n\}$ of them respectively. Then, the kernel generator G creates the ordinary kernels for the input face image and the reference face set respectively, based on which the commonality kernel K_r is optimized as the maximum commonality of ordinary kernels $\{K_1, ..., K_n\}$ in the reference set, and the personalized kernel K_x^p is achieved as the residual of orthogonal projection of one's ordinary kernel K_x on the commonality kernel K_r. Finally, personalized features f_x of input x are extracted by convolving z_x with its personalized kernel K_x^p, with which similarities between different face images are calculated. Note that the CE Loss in the figure means cross entropy loss

studies in other areas which are different but also follow the adaptive modeling thought.

2.1 Face Recognition

Face recognition is an important and long-standing problem in computer vision, whose accuracy heavily depends on expressive facial feature representations. In the early years, work on face recognition mainly obtains feature representations from the raw image pixels through linear or local linear projection (Wang & Deng, 2018), such as linear subspace (Belhumeur et al., 1997; Moghaddam et al., 1998; Turk & Pentland, 1991), manifold learning (He et al., 2005; Yan et al., 2005), and sparse representation (Deng et al., 2012; Wright et al., 2009; Zhang et al., 2011). However, those methods struggle to handle face recognition with complex appearance variations. At the same time, hand-crafted local features instead of raw pixels, such as Gabor (Liu & Wechsler, 2002), Local Binary Pattern (LBP) (Ahonen et al., 2006), as well as their extensions (Tan & Triggs, 2010; Xie et al., 2010; Wenchao et al., 2005; Zhang et al., 2007), are proposed and achieve favorable results in controlled environment. Joint Bayesian (Chen et al., 2012) has further improved the accuracy by formulating face embedding as the addition of the identity and the intra-class variation, which shares similar thoughts to our ordinary kernel decomposition in high level. However, the goal and basis of decomposition are different. Joint Bayesian decomposes the face information as the addition of identity and intra-class variation aiming for better estimating the covariance matrix of intra-personal pairs and extra-personal pairs, while our personalized CNN further decomposes the identity information of a face into the common identity-component and personal identity-component in order for more distinguishing identity information. The discrimination ability of hand-crafted features heavily depends on the design principles which may limit the accuracy of face recognition. Going a step further, a few learning-based approaches are proposed to learn more informative but mostly shallow representations (Cao et al., 2010; Duan et al., 2018; Lei et al., 2014). These learned shallow representations are favorable for controlled scenarios, but still hardly model those complex facial appearance variations in uncontrolled scenarios.

In recent years, deep convolutional neural network (CNN) has greatly improved the accuracy of face recognition owing to its excellent modeling capability of non-linear variations. DeepFace (Taigman et al., 2014) is the first deep method for face recognition. In DeepFace, all faces are aligned to be frontal through a general 3D shape model, with which CNN is trained for feature extraction. DeepFace outperforms many conventional non-deep face recognition methods, and inspires numerous follow-ups. Afterwards, Deep ID series (Chen et al., 2014; Sun et al., 2014, 2015a, b) use both identification and verification supervision to guide the learning of both intermediate and final feature extraction layers to obtain more discriminative feature representations. They even surpass the human's performance on the Labeled Face in the Wild (LFW) dataset (Huang & Learned-Miller, 2014). In FaceNet (Sankaranarayanan et al., 2016), triplet loss is employed to reduce/enlarge the distance between the positive/negative pair in a triplet on a large-scale face image set, achieving state-of-the-art results on multiple challeng-

ing benchmarks including LFW (Huang & Learned-Miller, 2014) and YouTubeFaces (Wolf et al., 2011) datasets.

Recent deep learning-based face recognition studies have made great progress from different aspects. Some studies focus on modifying the softmax loss function by explicitly adding a margin to the loss function. Among them, Large-Margin softmax (L-Softmax) loss (Liu et al., 2016) and SphereFace (Liu et al., 2017) propose multiplicative angular margin penalty to enforce extra intra-class compactness and inter-class discrepancy simultaneously. CosFace (Wang et al., 2018) and ArcFace (Deng et al., 2018) further add cosine margin and arc-cosine margin penalty to the target logit in Softmax loss respectively. UniformFace (Duan et al., 2019) further imposes an equidistributed constraint by uniformly spreading the class centers, so that the minimum distance between class centers can be maximized. Some other studies focus on improving the learning scheme. CurricularFace (Huang et al., 2020b) employs a curriculum learning loss to emphasize the easy samples in the early stage while the hard ones in the latter stage by adaptively adjusting the relative importance of easy and hard samples during different training stages. BroadFace (Kim et al., 2020b) proposes a learning process to comprehensively consider a massive set of identities in each iteration to increase the optimality of the classifier in the entire datasets, which can further globally optimize the encoder. The above studies have made great advances on face recognition accuracy.

Although the accuracy of face recognition on LFW and YTF datasets has reached human level, there still exists a gap between human performance and automatic face recognition with extreme pose, illumination, expression, age, or resolution variations in unconstrained environment, as reflected by the challenging IJB-A/B/C dataset. Therefore, some latest works focus on addressing face recognition robust to large face variations due to age, pose, expression and heavy occlusions, as exemplified in the following. Decorrelated Adversarial Learning (DAL) algorithm (Wang et al., 2019) proposes to factorize the mixed face feature into identity-dependent component and age-dependent component by adversarially minimizing the correlation between the two components. Age-Invariant Model (AIM) (Jian et al., 2019) unifies cross-age face synthesis and recognition in a mutual boosting way. To address the large change of the facial pose, expression, and illumination, Attentional Feature-pair Relation Network (AFRN) (Kang et al., 2019) represents a face by the relevant pairs of local appearance block features. To eliminate the influence of facial occlusion on face recognition, the study in Song et al. (2019) proposes a mask learning strategy to find and discard corrupted feature elements from recognition by exploiting the differences between the convolutional features of occluded and occlusion-free face pairs. GroupFace (Kim et al., 2020a) integrates group-aware representations into the embedding feature to narrow down the search space of the target identity. Although great progress has been made for face recognition, more accurate algorithms to address those complex scenarios are still in great demand.

2.2 Adaptive Convolution

Our work is also related to adaptive convolution, which has been employed for several problems in computer vision. In Chen et al. (2017), adaptive kernels are applied to image style transfer, where a given image is transformed from the original style to a target style through convolving the given image with kernels of the target style. In Zhang et al. (2017), scale-adaptive convolution is proposed to acquire flexible-size receptive fields for scene parsing to tackle the issue of inconsistent predictions of large objects and invisibility of small objects in conventional CNN. In Liao and Shao (2019), the authors employ adaptive convolution to achieve local matching for person re-identification, where the matching process and results become interpretable and generalizable. In Klein et al. (2015) and Jia et al. (2016), dynamically generated filters are used to predict the movement of pixels between frames for highway driving or weather image prediction. In Kang et al. (2017), the authors propose an adaptive convolutional neural network (ACNN), whose convolutional kernel is adapted to the current scene context via the side information (e.g. camera perspective, noise level, blur kernel parameters). The ACNN can disentangle the variations related to the side information, and extract discriminative features related to the current context, to address the supervised learning problem such as crowd counting, corrupted digit recognition, and image deblurring. The method in Bertinetto et al. (2016) proposes a second network, named *Learnet*, to predict parameters of a pupil network from a single exemplar, to handle the one-shot image classification task. CondInst (Tian et al., 2020) proposes to dynamically generate the filter parameters conditioned on the target instance in each mask head for effective instance segmentation.

All these methods explore the adaptive modeling schema. However, they are essentially different from ours. The main difference is the basis of creating kernels. The above mentioned adaptive kernels are generated only according to one input, in order for capturing the specific features of the input. Differently, our personalized kernel is created according to both the input and a reference set, where the reference set is used to filter out the commonality between distinct persons. Benefited from this design, those common features can be elaborately weakened and thus highlight one's characteristics more precisely. Another difference is that they have distinct purposes, i.e. solving different problems. As is described above, adaptive convolution in those previous studies mainly focuses on image generation or image prediction, while our personalized kernel aims at distinguishing between subjects, thus leading to different designs of the convolution operation.

3 Personalized Convolution

Given a face image, our personalized convolution extracts its special features with personalized kernel aiming at better distinguishing him/her from others. The overall structure is presented in Fig. 2, and the proposed CNN is referenced to as personalized CNN in the following for convenience.

Our personalized CNN mainly consists of three parts: a convolutional backbone E to extract basic features for all subjects, a kernel generator G to generate ordinary kernel for each subject, a module of personalized kernel optimization to decompose the ordinary kernel into commonality component and specialty component. Finally personalized kernel is used to convolve with the basic features of the input face image to obtain personalized feature representation. The personalized kernel generation is formulated as an analytical optimization process and thus the whole personalized CNN is optimized with two simple cross entropy losses imposed on personalized features and the ordinary kernels respectively in end-to-end manner. After the training stage is finished, the similarity of two test face images is calculated as the cosine similarity of their personalized features. The three parts are detailed in the following.

3.1 Basic Feature Extracting

As is shown in Fig. 2, initially a CNN backbone E shared by all persons is used to extract basic features for all input images. This feature extractor E can be implemented as any kind of network architecture, such as several stacked convolutional layers or the popular ResNet-50 backbone. For any input image x, its basic features are simply extracted as:

$$z_x = E(x) \in \mathbb{R}^{c_z \times h_z \times w_z}, \tag{1}$$

where c_z, h_z, and w_z are the number of channel, height, and width of the output feature maps respectively.

3.2 Ordinary Kernel Generating

With the basic feature extractor, features of all persons are computed in an identical manner, i.e. using the same kernel for feature extraction. By contrast, we humans naturally focus on distinct characteristics of different persons so as to better distinguish them. Following the same thought, we design personalized kernel that is adaptive to each subject to highlight his/her special features to distinguish him/her from others more accurately.

A direct way is to introduce a kernel generator G to adaptively generate the convolutional kernel for each input subject. The kernel generator takes basic feature maps z_x as input, and outputs a set of kernels as follows:

$$K_x = G(z_x). \tag{2}$$

Although the kernel generated from Eq. (2) is adaptive to each person, it usually captures compound features, including not only those special distinguishing characteristics of the input face but also some common distinguishing characteristics among different faces. In this way, although K_x is adaptive, while those special information may be not notable especially compared to the following proposed personalized kernel, so in this work K_x is named as ordinary kernel. To further obtain the personalized kernel of each person, the common characteristics should be removed from the ordinary kernel, as presented in Sect. 3.3.

Generally, the kernel generator G can be designed as any kind of network architectures, such as convolutional layers, fully connected layers, or a mixture of them. However, the number of parameters of kernel generator in the form of those architectures will multiplicatively increase with the dimension of the input feature map z_x and the output kernel K_x. So, to control the complexity, the kernel generator in this work is designed as a two-layer architecture, in which the first layer is a standard convolutional layer and the second layer is a ***channel-wise*** fully-connected layer to avoid parameters explosion. Supposing $K_x \in \mathbb{R}^{n_k \times c_k \times h_k \times w_k}$ with n_k kernels, c_k channels and filter size $h_k \times w_k$, the kernel generator can be specifically formulated in the following:

$$K_x = \phi(W_g^2 \odot \phi(W_g^1 * z_x + b_g^1) + b_g^2), \tag{3}$$

where W_g^1, b_g^1, and W_g^2, b_g^2 are the parameters of kernel generator G in the first layer and second layer respectively. ϕ is the non-linear activation function, $*$ means conventional convolution, and $\odot$ means channel-wise connection. In the channel-wise fully connected layer, the j-th ordinary kernel $K_x(j) \in \mathbb{R}^{c_k \times h_k \times w_k}$ is generated from the j-th channel of $\phi(W_g^1 * z_x + b_g^1)$ with the fully connection. Such an architecture of the kernel generator better trades off the accuracy and the efficiency.

3.3 Personalized Kernel Generating

As mentioned in the introduction part, to make the kernel of a given person focus on his/her special distinguishing characteristics, those common characteristics should be filtered out from his/her ordinary kernel. Therefore, we consider decomposing the ordinary kernel K_x into two orthogonal components, i.e. the commonality component and the specialty component, in order to isolate those most special characteristics. This decomposition is formulated as follows.

$$K_x = a_x K_r + K_x^p, \text{ with } ||K_r|| = 1, K_r \perp K_x^p. \tag{4}$$

Here, K_r denotes the commonality kernel that captures the common distinguishing characteristics among a group of subjects. K_x^p is orthogonal to the commonality kernel K_r, so it captures those uncommon i.e. the special characteristics of x. $a_x \in \mathbb{R} = (K_x)^T K_r$ is the magnitude of orthogonal projection of K_x on K_r, i.e. the magnitude of commonality.

Given the ordinary kernel K_x and commonality kernel K_r, the personalized kernel can be obtained by removing the projection on K_r from K_x, i.e.:

$$\begin{aligned} K_x^p &= K_x - a_x K_r \\ &= K_x - K_r (K_x)^T K_r \end{aligned} \tag{5}$$

Therefore, the key becomes how to model the commonality kernel K_r. By its name, K_r captures the common distinguishing characteristics shared by a group of people. In order to optimize K_r, a reference set containing multiple images is introduced as the proxy of persons in a specific scenario, such as persons appearing in a movie or living in a community or just persons in the same dataset.

3.3.1 Optimization of Commonality Kernel

Formally, we denote the reference set with n face images as $R = \{x_1, \cdots, x_n\}$, where x_i means the i-th face image in the reference face set. The basic features and ordinary kernels of face images in the reference set are computed as below:

$$z_i = E(x_i) \tag{6}$$

$$K_i = G(z_i). \tag{7}$$

With the ordinary kernels $\{K_i \in \mathbb{R}^{n_k \times c_k \times h_k \times w_k} |_{i=1}^n\}$ of the reference set, we explicitly model the commonality kernel K_r as the maximum commonality between them. For the convenience of notation, we take a 3D kernel in $\mathbb{R}^{c_k \times h_k \times w_k}$ from K_i as example for presentation, which is first reshaped into a vector in $\mathbb{R}^{L \times 1}$ and still denoted as K_i in the following. Here, $L = c_k \times h_k \times w_k$. Similar as Eq. (4), each K_i is decomposed to two orthogonal components, the commonality component $a_i K_r$ and the specialty component R_i as below:

$$K_i = a_i K_r + R_i \text{ with } ||K_r|| = 1, K_r \perp R_i, \tag{8}$$

where $K_r \in \mathbb{R}^{L \times 1}$ represents the commonality kernel of kernels $\{K_i|_{i=1}^n\}$, and $a_i \in \mathbb{R}$ is the projection magnitude of K_i on K_r with $a_i = (K_i)^T K_r$.

As the commonality kernel K_r represents the commonality between kernels $\{K_i|_{i=1}^n\}$, the projection magnitude of each kernel K_i on K_r is expected to be identical, i.e. the variance of $\{a_i|_{i=1}^n\}$ is expected to be zero. Only with the variance constraint, the commonality kernel K_r may degenerate to trivial solution, e.g. $a_i = 0, i \in \{1, ..., n\}$ in extreme case. Thus, the projection magnitude a_i is expected to be as large as possible to make K_r contain the maximum commonality. Therefore, an objective is formulated to fulfill these expectations:

$$\begin{aligned} \min_{K_r} \alpha \cdot \frac{1}{n} \sum_i (a_i - \overline{a})^2 - (1-\alpha) \cdot \frac{1}{n} \sum_i (a_i)^2 \\ = \frac{\alpha}{n} \sum_i ((K_i)^T K_r - \overline{K}^T K_r)^2 - \frac{1-\alpha}{n} \sum_i ((K_i)^T K_r)^2 \\ = \frac{\alpha}{n} ||K^T K_r - \mathbb{1} \overline{K}^T K_r||_2^2 - \frac{1-\alpha}{n} ||K^T K_r||_2^2 \end{aligned} \tag{9}$$

where $(K_r)^T K_r = 1, \overline{a}$ and $\overline{K}$ are the mean of $\{a_1, ..., a_n\}$ and $\{K_1, ..., K_n\}$ with $\overline{a} = \frac{1}{n} \sum_i a_i$ and $\overline{K} = \frac{1}{n} \sum_i K_i$ respectively, K is the matrix concatenating $\{K_i|_{i=1}^n\}$ by column, and $\mathbb{1}$ is a all-one matrix with shape $n \times 1$. In the Eq. (9), the first term enforces the projection of the ordinary kernel K_i on the commonality kernel K_r to be identical by minimizing their variance, and the second term aims to maximize the magnitude, i.e. make K_r contain commonality as much as possible. The parameter $\alpha \in [0, 1]$ is to balance the commonality and the magnitude terms as they may contradict each other in complex applications.

This objective can be analytically optimized with the Lagrange multiplier method. First, considering the constraint condition $(K_r)^T K_r = 1$, in Lagrange multiplier method, the objective in Eq. (9) can be written as:

$$\begin{aligned} J &= \frac{\alpha}{n} ||K^\top K_r - \mathbb{1} \bar{K}^\top K_r||_2^2 - \frac{1-\alpha}{n} ||K^\top K_r||_2^2 \\ &\quad - \mu((K_r)^\top K_r - 1) \\ &= \frac{2\alpha - 1}{n} (K_r)^\top K K^\top K_r - \frac{\alpha}{n^2} K_r^\top K \mathbb{1} \mathbb{1}^\top K^\top K_r \\ &\quad - \mu((K_r)^\top K_r - 1), \end{aligned}$$

where μ is the multiplier. The derivative of J on K_r is:

$$\frac{\partial J}{\partial K_r} = \frac{2(2\alpha - 1)}{n} K K^T K_r - \frac{2\alpha}{n^2} K \mathbb{1} \mathbb{1}^T K^T K_r - 2\mu K_r.$$

Let the derivative be equal to 0. The equation can be written as:

$$(\frac{2\alpha - 1}{n} K K^T - \frac{\alpha}{n^2} K \mathbb{1} \mathbb{1}^T K^T) K_r = \mu K_r. \tag{10}$$

Thus, the objective of Eq. (9) achieves its minimal value as the least eigenvalue of matrix $\frac{2\alpha-1}{n} K K^T - \frac{\alpha}{n^2} K \mathbb{1} \mathbb{1}^T K^T$ when K_r is taken as the corresponding eigenvector, i.e.:

$$K_r = \text{LeastEigVec}(\frac{2\alpha - 1}{n} K K^T - \frac{\alpha}{n^2} K \mathbb{1} \mathbb{1}^T K^T). \tag{11}$$

In special, when $\alpha = 0.5$, i.e. taking the two terms equally important, K_r degenerates into the normalized mean kernel of $\{K_i|_{i=1}^n\}$. When $\alpha = 1.0$, i.e. only keeping the commonality term, K_r is the kernel lying in the null-space of the scatter matrix of $\{K_i\}$ if $\frac{1}{n}KK^T - \frac{1}{n^2}K\mathbb{1}\mathbb{1}^TK^T$ is singular, i.e.

$$K_r = \begin{cases} \frac{\overline{K}}{||\overline{K}||}, & \text{if } \alpha = 0.5, \\ \text{Null}(\frac{1}{n}KK^T - \frac{1}{n^2}K\mathbb{1}\mathbb{1}^TK^T), & \text{if } \alpha = 1.0, \end{cases} \tag{12}$$

where Null$(\cdot)$ means the null-space of a matrix.

After obtaining the commonality kernel K_r, the personalized kernel of the given face x is achieved as the residual of his/her ordinary kernel K_x on K_r, formulated in Eq. (5). The personalized kernel is then used to convolve with the basic features of the given face image x to finally obtain the corresponding personalized feature representation as follow:

$$f_x = K_x^p * z_x, \tag{13}$$

where $*$ means the standard convolution operation.

The whole architecture to calculate the personalized features f_x is optimized with two types of cross entropy loss detailed in Sect. 3.4.

After the training is finished, given any two face images x_1 and x_2 at testing stage, their features are firstly extracted according to Eq. (13), and then the similarity of them is calculated as the cosine similarity as follows:

$$s(x_1, x_2) = \left(\frac{f_{x_1}}{\|f_{x_1}\|}\right)^T \cdot \left(\frac{f_{x_2}}{\|f_{x_2}\|}\right). \tag{14}$$

3.4 Objective Function

By using Eq. (1), Eq. (5), and Eq. (13), the personalized feature of an input face image is calculated as f_x. For face recognition, the feature f_x is expected to distinguish x from others. So a cross entropy loss for identity classification is necessary to enforce f_x to be discriminative. Besides, another cross entropy loss is introduced to enforce the ordinary kernels generated from different images of the same subject to be identical. This can make the ordinary kernel and personalized kernel well capture the distinguishing characteristics (i.e. identity information) of a person.

Given a face image x with label l, its personalized features f_x are expected to be classified into the l-th class like the conventional CNN does. Supposing W_f is the parameter of classifier, and each column of W_f is the classifying vector for a class, the probability p_x^l that personalized feature f_x is classified into class l can be calculated by the softmax function:

$$p_x^l = \frac{e^{u_l}}{\sum_{j=1}^C e^{u_j}}, \tag{15}$$

$$u = (u_1, u_2, ..., u_C) = \beta(f_x)^T W_f, \tag{16}$$

where C is the number of subjects in the training set. Here, the length of f_x and each column vector in W_f is normalized. To make the training converge, we multiply a scale value β on the cosine similarity $(f_x)^T W_f$ like (Wang et al., 2018) does. The classification loss in our method is the typical cross entropy loss:

$$L_f = \frac{1}{N}\sum_{i=1}^N -log(p_{x_i}^{l_i}), \tag{17}$$

where N is the number of images in each training batch, x_i and l_i are the i-th image and its label in a batch respectively.

Moreover, those ordinary kernels K_x created by kernel generator G are expected to capture the intrinsic characteristics of a person, which means that ordinary kernels of face images from the same person should be the same even with various poses, illuminations or expressions, forming another cross entropy loss over the ordinary kernels as follows:

$$\begin{aligned} L_K &= \frac{1}{N}\sum_{i=1}^N -log(p_{K_{x_i}}^{l_i}) \\ &= \frac{1}{N}\sum_{i=1}^N -log\frac{e^{v_{l_i}}}{\sum_{j=1}^C e^{v_j}}, \end{aligned} \tag{18}$$

$$v = (v_1, v_2, ..., v_C) = (K_{x_i})^T W_k. \tag{19}$$

Here, K_{x_i} is the ordinary kernel of x_i generated by using Eq. (2), and $p_{K_{x_i}}^{l_i}$ denotes the probability of kernel K_{x_i} being classified into class l_i. W_k is the classification weight for those generated kernels. Generally, one more layer can be added to decrease the dimension of kernels K_{x_i} before calculating the probability $p_{K_{x_i}}^{l_i}$.

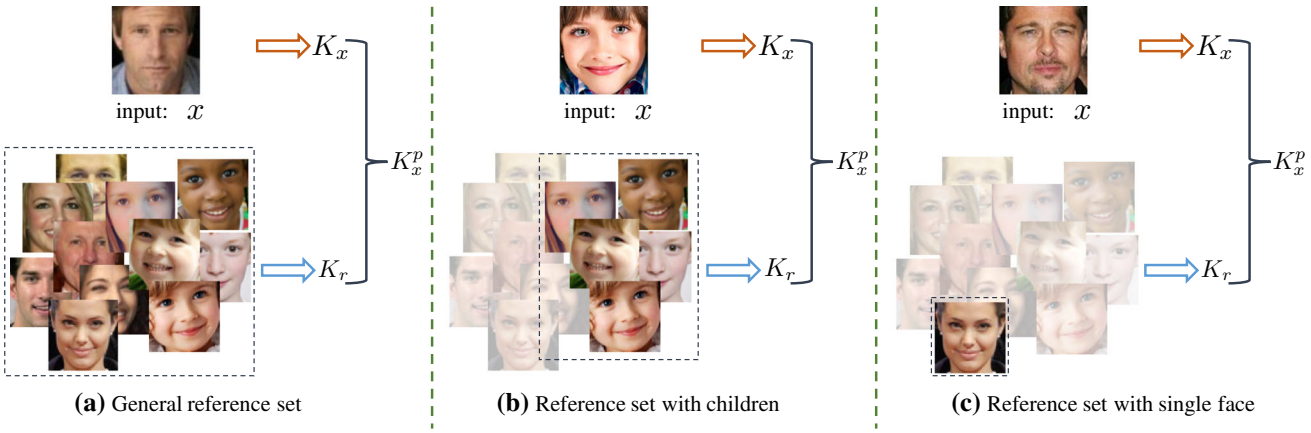
Overall, the objective function of our personalized CNN is formulated as simply combining the above two losses as below:

$$\min_{E,G} L_f + \lambda L_K. \tag{20}$$

Here, λ is a balance parameter. The first term of the objective function aims to make those personalized features be discriminative, while the second term endeavors to make ordinary kernels be intra-class invariant. This objective can be easily optimized in an end-to-end manner by using stochastic gradient decent (SGD) like most CNN-based methods do.

3.5 Variants

In our proposed personalized convolution, the reference face set is introduced as a proxy of persons in a specific scenario,



(a) General reference set **(b)** Reference set with children **(c)** Reference set with single face

Fig. 3 Variants of the personalized kernel with different reference sets. **a** Reference set with images of abundant persons to represent the public. **b** Reference set with a special group of persons (children in the figure). **c** Reference set only with only one face image that the input image is compared with

which can filter out the commonality while leaving the specialty. Therefore, to adapt to different scenarios, the reference set can be constructed by including different faces, deriving different variants for better capturing one's distinguishing features in different scenarios.

As is shown in Fig. 3 (a), generally, the reference set can consist of images of many general persons so that it can represent the commonality of general public. However, in some scenarios we may only want to recognize persons from a specific group, such as recognizing children in primary school, or recognizing girls in girl's apartment. In these cases, the commonality of the specific group to recognize is usually different from that of general public. To stress a child or a girl's discriminative features, the reference set should be set as a collection of children or girls' images. Thus, the reference set can be specially constructed as a particular group of face images for better distinguishing different subjects for various scenarios, as shown in Fig. 3 (b).

More specially, the reference set can contain only one face image. This usually happens in the case of face verification. For example, when we compare Audrey Hepburn with Renée Zellweger, we focus more on Audrey Hepburn's eyes, while when we compare Audrey Hepburn with Chrissy Metz, we may rather pay attention to the contour of Audrey Hepburn's face. This means that, the compared person, Renée Zellweger or Chrissy Metz, is used as the 'reference system' instead of the general public or a specific group. In this case, the proposed personalized kernel becomes a special variant, i.e. the contrastive kernel proposed in our conference work (Han et al., 2018) to highlight those distinct characteristics between a pair of face images being compared.

Overall, when aiming for different scenarios, the reference set can be constructed as faces of different subjects, deriving distinct variants of the personalized kernel.

4 Experiments

In this section, we evaluate the proposed personalized CNN for both face identification and verification tasks. We first give the experimental settings, and also the implementation details. Then, we develop ablation study to analyze the proposed personalized kernel via comparing our method with baselines, and evaluate different variants of reference set. Next, we study the influence of hyper-parameter α and λ respectively. Last, we compare our method with some existing methods on LFW, IJB-A and IJB-C, and visualize feature response of vanilla CNN and our personalized CNN.

4.1 Experimental Settings

4.1.1 Dataset

In all experiments, five datasets are used for training or testing. The CASIA-WebFace (Yi et al., 2014), and VGGFace2 (Cao et al., 2018) datasets are used for training, and the LFW (Huang et al., 2008), IJB-A (Klare et al., 2015), and IJB-C (Maze et al., 2018) datasets are used for testing.

The **CASIA-WebFace** dataset (Yi et al., 2014) is a large scale face dataset containing about 500,000 face images of about 10,000 different individuals collected from Internet by Institute of Automation, Chinese Academy of Sciences (CASIA). It is the default training dataset if unspecified. To fairly compare with existing methods, we also develop experiments on VGGFace2 dataset. The **VGGFace2** dataset (Cao et al., 2018) consists of 3.3 million faces from 9,131 subjects, which covers a large range of poses, ages, ethnicities and professions. The two datasets are commonly used datasets to train a deep network for unconstrained face recognition, such as in Yi et al. (2014), Ding and Tao (2015), Liu et al.

Table 1 Architectures of the CNN used in our method with 4, 10, 16 layers respectively. There are 4 stages in each CNN, and the shape of feature map is the same in each stage. Each stage contains multiple convolution units. $conv[c, k, s, g]$ denotes convolution with c filters of size $k \times k$, stride s, and group g. The $max[3, 2]$ denotes the max pooling within pooling window size 3×3 and stride 2. In CNNs with 10 and 16 layers, the residual network structure is used for better performance and the residual units are shown in the double-column brackets. In the last layer of personalized convolution, the convolution is the same as conventional convolution except that its kernels are dynamically generated during testing

Layer	4-layer CNN	10-layer CNN	16-layer CNN
input	$112 \times 112 \times 3$		
Stage1	Conv[64, 3, 1, 1]	Conv[64, 3, 1, 1]	Conv[64, 3, 1, 1] $\begin{bmatrix}\text{Conv}[64, 3, 1, 1]\\ \text{Conv}[64, 3, 1, 1]\end{bmatrix} \times 1$
Pool1	max[3, 2]		
Stage2	Conv[128, 3, 1, 1]	Conv[128, 3, 1, 1] $\begin{bmatrix}\text{Conv}[128, 3, 1, 1]\\ \text{Conv}[128, 3, 1, 1]\end{bmatrix} \times 1$	Conv[128, 3, 1, 1] $\begin{bmatrix}\text{Conv}[128, 3, 1, 1]\\ \text{Conv}[128, 3, 1, 1]\end{bmatrix} \times 2$
Pool2	max[3, 2]		
Stage3	Conv[256, 3, 1, 1]	Conv[256, 3, 1, 1] $\begin{bmatrix}\text{Conv}[256, 3, 1, 1]\\ \text{Conv}[256, 3, 1, 1]\end{bmatrix} \times 2$	Conv[256, 3, 1, 1] $\begin{bmatrix}\text{Conv}[256, 3, 1, 1]\\ \text{Conv}[256, 3, 1, 1]\end{bmatrix} \times 3$
Pool3	Max[3, 2]		
Stage4	Conv[512, 3, 1, 1]	Conv[512, 3, 1, 1]	Conv[512, 3, 1, 1]
Pool4	Max[3, 2]		
Personalized Conv	Conv[512, 3, 1, 512]	Conv[512, 3, 1, 512]	Conv[512, 3, 1, 512]
Global Pooling	Global Max Pooling		
Features	512 dimensions		

(2016), Liu et al. (2017), Weidi et al. (2018), Huang et al. (2020a).

During training on CASIA-WebFace, the general **Reference Set** R in our method is constructed with 46K images of 1000 individuals randomly selected from CASIA-WebFace. When training on VGGFace2, 169K images of 500 individuals are randomly selected from this dataset as the reference set R. In the stage of network optimizing, 20 images are randomly chosen from the general reference set as the batch reference images in each iteration to compute the commonality kernel K_r in Eq. (9). When the network optimizing is finished, the commonality kernel K_r can be simply calculated over the entire reference set before seeing any testing image, and then used to extract features of testing images at test time. This strategy offers us a good way to calculate the commonality kernel K_r when a new or more elaborated reference set is available. However, it will cause additional computational overhead. Thus, the moving average strategy is further proposed to avoid the computational overhead, where the statistics including scatter matrix KK^T and mean kernel $K\mathbb{1}$ in Eq. (11) of the reference set are updated on each mini-batch in the moving average way during network optimizing. Therefore, the commonality kernel is well calculated when the network optimizing is finished and can be directly used in the testing phase. Note that class labels of faces in the reference set is not needed, which means the reference set can be generally developed by using an additional unlabeled set.

The **LFW** dataset (Huang et al., 2008) is a standard benchmark for the research of unconstrained face recognition. It includes 13,233 face images from 5,749 different identities with large variations in pose, expression and illumination. On this dataset, we follow the standard evaluation protocol of 'unrestricted with labeled outside data', and report the mean verification accuracy (mAcc) of 10-folder verification sets.

The **IJB-A** dataset (Klare et al., 2015) is an open challenging benchmark containing 5,712 images and 2,085 videos from 500 subjects captured from the wild environment. Because of the extreme variation in head pose, illumination, expression and resolution, so far IJB-A is regarded as one of the most challenging datasets for both verification and identification. The standard protocol on this dataset performs evaluations in the template-based manner, for both identification(1:N face search) and verification (1:1 comparison). A template may include images and/or videos of a subject.

The **IJB-C** face dataset (Maze et al., 2018) advances the goal of robust unconstrained face recognition by increasing dataset size and variability, which is developed upon the previous public IJB-A and IJB-B (Whitelam et al., 2017)

datasets. It includes 31,334 images and 11,779 videos from 3,531 subjects with variations in head orientation, facial expression, illumination, and also occlusion and reduced resolution. The protocol of this dataset also operates in template-based manner.

4.1.2 Implementation Details

Preprocessing For all five datasets, MTCNN (Zhang et al., 2016) is first used to detect faces and landmarks. Then all detected faces are aligned to a canonical one according to the five landmarks (2 eyes centers, 1 nose tip, and 2 mouth corners). Finally, all aligned images are resized into 112×112 for training and testing. Note that, for those undetectable faces on IJB-A and IJB-C, bounding boxes offered by corresponding dataset are used to crop them, and the mean of landmarks of all detected faces in corresponding dataset is used to align them. This preprocessing is the same as most existing work such as Yin et al. (2017), Liu et al. (2017), Wang et al. (2018), Deng et al. (2018).

Architecture All the experiments are implemented with pytorch. For extensive investigation, the proposed personalized CNN with three backbone architectures of 4-layer, 10-layer, and 16-layer are evaluated respectively. Those backbone structures are similar with those shallower networks used in SphereFace (Liu et al., 2017). The detailed architectures of the three backbones are listed in Table 1. The structure of kernel generator is the same for all the three backbone architectures, which is simply designed as a small 2-layer subnetwork, and the detailed design is introduced in Sect. 3.2. Note that, in testing stage, kernels of the last convolutional layer of our personalized CNNs are dynamically generated by the kernel generator learned in the training stage.

Training Parameters When training on CASIA-WebFace dataset (Yi et al., 2014), the batch size (doesn't include reference images) is set as 128. All models are trained with 100K iterations. The learning rate is set as 0.2 at the beginning, and decays as 0.02 at the 50k-th iteration, and as 0.002 at the 80k-th iteration. For experiments on VGGFace2 dataset (Cao et al., 2018), the batch size is set as 512, the learning rate is divided by 10 at 80K-th, 120K-th, and 150K-th iteration respectively starting with 0.2, and the training process is finished at 160K iterations. Random flapping and cropping face images are employed in the training stage for data augmentation as most existing methods do. In our implementation, the size of generated ordinary kernel is $512 \times 1 \times 3 \times 3$. Thus, the personalized convolution is conducted as the group convolution with 1 channel in each group.

Evaluation Details: For both identification and verification, the evaluation involves similarity computation and evaluation protocols. Thus, we introduce the evaluation details from the two aspects in the following.

The similarity computation includes two levels, i.e., image-level similarity on LFW, and template-level similarity on IJB-A and IJB-C introduced as follows.

- The image-level similarity means cosine similarity of features extracted by a CNN of two face images like Eq. (14) in the paper.
- Following (Masi et al., 2016b), the SoftMax operator is used to calculate the similarity between two templates. Specifically, for two templates $\mathcal{P}$, and $\mathcal{Q}$, the SoftMax operator for calculating their similarity can be seen as a weighted average of similarities of all pairs (p, q) with $p \in \mathcal{P}, q \in \mathcal{Q}$. The weight for the weighted average is calculated as $\frac{e^{\beta s(p,q)}}{\sum_{a\in\mathcal{P},b\in\mathcal{Q}} e^{\beta s(a,b)}}$ for a pair (p, q), where $s(p, q)$ means the image-level similarity. The SoftMax hyper-parameter β controls the trade-off between averaging the similarity or taking the maximum (or minimum). That is:

$$s_\beta(\cdot,\cdot) = \begin{cases} \max(\cdot), & \text{if } \beta \to \infty \\ \text{mean}(\cdot), & \text{if } \beta = 0 \\ \min(\cdot), & \text{if } \beta \to -\infty \end{cases}, \tag{21}$$

and

$$s_\beta(\mathcal{P}, \mathcal{Q}) = \frac{\sum_{p\in\mathcal{P},q\in\mathcal{Q}} s(p, q) e^{\beta s(p,q)}}{\sum_{p\in\mathcal{P},q\in\mathcal{Q}} e^{\beta s(p,q)}}, \tag{22}$$

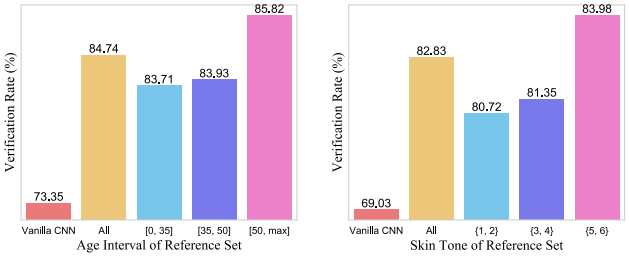
where p and q are images in $\mathcal{P}$ and $\mathcal{Q}$ respectively. The final similarity between two templates is the average of SoftMax response over multiple values of $\beta \in \{0, 1, ..., 20\}$ as follow:

$$S(\mathcal{P}, \mathcal{Q}) = \frac{1}{21} \sum_{\beta=0}^{20} s_\beta(\mathcal{P}, \mathcal{Q}). \tag{23}$$

After the similarity of two face images or two templates is computed, mean accuracy (mAcc) or receiver operating characteristic (ROC) is used for face verification, and Rank-1/Rank- 5 recognition rate is used for face identification for performance evaluation following the standard evaluation protocols. Detailed introductions are as follows.

- The mean accuracy (mAcc) is used for the evaluation of face verification on LFW. Specifically, all verification pairs on this dataset are split into 10 folds, and the mAcc is calculated as the average percentage of accurate verification on 10 separate folds in a leave-one-out cross verification scheme, i.e. nine of the folds are used to search an optimal threshold and the left one is used for testing.

– The receiver operating characteristic (ROC) is used for the evaluation of face verification on IJB-A and IJB-C. At a given threshold (the independent variable) ROC analysis measures the true accept rate (TAR), which is the fraction of genuine comparisons that correctly exceed the threshold, and the false accept rate (FAR), which is the fraction of impostor comparisons that incorrectly exceed the threshold.
– Rank-k accuracy is used for the evaluation of face identification on IJB-A. Specifically, rank-k accuracy means the percentage of correctly recognized probe templates where a probe template is considered to be recognized correctly if the gallery template with the same identity appears in the top k gallery templates ranked according to their similarity.



(a) Testing set with age in [50, max] **(b)** Testing set with skin tone in {5, 6}

Fig. 4 Evaluation on the same testing set but with different reference sets on IJB-A, in terms of verification rate at FAR=1e-2. **a** Testing set with age in [50, max], while the reference sets are with ages in [0, max], [0, 35], [35, 50], [50, max] respectively. **b** Testing set with skin tone in {5, 6}, while the reference sets are with skin tone in {1, 2, 3, 4, 5, 6}, {1, 2},{3, 4},{5, 6} respectively

4.2 Ablation Study

Effectiveness of personalized kernel To investigate the effectiveness of our proposed personalized kernel, we compare our personalized CNN with the vanilla CNN, the self-attention CNN, and the contrastive CNN with the same backbone architecture trained on CASIA-WebFace. As our personalized kernel is generated from an additional kernel generator with two layers, two additional layers are added to the vanilla CNN (referred to as L-Vanilla CNN), so that they have the same number of parameters for fair comparison. The self-attention CNN is constructed by removing the reference set and using an attention map generator with the same structure with our kernel generator to generate the attention map to multiply the original feature z_x for an input face x. The self-attention CNN is designed to evaluate the influence of removing reference set. Moreover, We also retrain our contrastive CNN proposed in the conference version (Han et al., 2018) with the same setting for fair comparison. For comprehensive investigation, the comparison is performed with three types of backbone shown in Table 1 on three testing sets.

The results on LFW, IJB-A and IJB-C are shown on Table 2, 3, and 4. As can be seen, contrastive CNN, self-attention CNN, and our personalized CNN consistently outperform the vanilla CNN. Especially, on the verification task, our personalized CNN with 16-layer backbone improves the vanilla CNN over 9% at FAR=1e-3 on IJB-A, and over 20% at FAR=1e-5 on IJB-C, which safely shows the effectiveness of personalized feature extraction. Moreover, compared to the self-attention CNN, and the Contrastive CNN which has no reference set or uses single image as the reference,

Table 2 Comparison with L-vanilla CNN, self-attention CNN, and contrastive CNN on LFW with three different backbones. The architecture is detailed in Table 1. †denotes directly calculating the commonality kernel over the reference set, and ‡denotes calculating it with the moving average strategy during network optimizing.

# Layers of backbone CNN	Model	mAcc on LFW(%)
4	L-vanilla CNN	98.01
	Contrastive CNN (Han et al., 2018)	98.32
	Self-attention CNN	98.78
	Personalized CNN†	**98.78**
	Personalized CNN‡	**98.77**
10	L-vanilla CNN	98.93
	Contrastive CNN (Han et al., 2018)	98.93
	Self-attention CNN	99.21
	Personalized CNN†	**99.27**
	Personalized CNN‡	**99.38**
16	L-vanilla CNN	99.18
	Contrastive CNN (Han et al., 2018)	99.12
	Self-attention CNN	99.23
	Personalized CNN†	**99.40**
	Personalized CNN‡	**99.45**

Best results are shown in bold

Table 3 Comparison with L-vanilla CNN, self-attention CNN, and contrastive CNN on IJB-A with three different backbones. The architecture is detailed in Table 1

# Layers of backbone CNN	Model	Verification rate (%) on IJB-A		Identification rate (%) on IJB-A	
		TAR@FAR=1e-2	TAR@FAR=1e-3	Rank-1	Rank-5
4	L-vanilla CNN	85.57	63.70	89.62	96.01
	Contrastive CNN (Han et al., 2018)	84.47	61.78	-	-
	Self-attention CNN	84.54	67.08	90.86	96.59
	Personalized CNN†	**86.08**	**71.71**	**91.68**	**96.77**
	Personalized CNN‡	**85.96**	**69.88**	**91.90**	**96.57**
10	L-vanilla CNN	88.36	64.79	91.76	96.55
	Contrastive CNN (Han et al., 2018)	90.15	74.64	-	-
	Self-attention CNN	**91.25**	76.24	93.86	97.39
	Personalized CNN†	91.03	**80.94**	**94.23**	**97.72**
	Personalized CNN‡	**90.59**	**80.60**	**94.34**	**97.62**
16	L-vanilla CNN	89.97	72.47	93.32	97.05
	Contrastive CNN (Han et al., 2018)	91.20	78.14	-	-
	Self-attention CNN	91.18	78.30	94.12	97.57
	Personalized CNN†	**91.25**	**82.27**	**94.69**	**97.80**
	Personalized CNN‡	**92.19**	**82.67**	**94.39**	**97.54**

Best results are shown in bold
†denotes directly calculating the commonality kernel over the reference set, and ‡denotes calculating it with the moving average strategy during network optimizing

Table 4 Comparison with L-vanilla CNN, self-attention CNN and contrastive CNN on IJB-C with three different backbones. The architecture is detailed in Table 1

# Layers of backbone CNN	model	Verification rate (%) on IJB-C			
		TAR@FAR=1e-2	TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
4	L-vanilla CNN	88.97	71.91	42.84	19.38
	Contrastive CNN (Han et al., 2018)	89.45	74.21	55.42	32.69
	Self-attention CNN	89.94	75.66	57.63	39.08
	Personalized CNN†	**89.98**	**77.39**	**61.10**	**44.75**
	Personalized CNN‡	**89.39**	**76.02**	**59.16**	**42.53**
10	L-vanilla CNN	91.80	74.81	48.03	23.47
	Contrastive CNN (Han et al., 2018)	92.81	81.12	65.22	47.26
	Self-attention CNN	93.22	84.06	69.12	48.67
	Personalized CNN†	**93.52**	**84.83**	**72.76**	**57.02**
	Personalized CNN‡	**93.29**	**84.53**	**72.31**	**58.19**
16	L-vanilla CNN	93.18	80.39	60.76	38.48
	Contrastive CNN (Han et al., 2018)	94.07	84.54	70.27	54.59
	Self-attention CNN	93.83	83.62	70.87	53.56
	Personalized CNN†	**94.02**	**85.77**	**74.61**	**59.41**
	Personalized CNN‡	**94.42**	**86.58**	**75.00**	**59.83**

Best results are shown in bold
†denotes directly calculating the commonality kernel over the reference set, and ‡denotes calculating it with the moving average strategy during network optimizing

our personalized CNN with a general reference set achieves improvement up to 4% at FAR=1e-3 on IJB-A, and 6% at FAR=1e-5 on IJB-C. These improvements demonstrate that removing the commonality can better highlight one's special characteristics for higher recognition accuracy, verifying the effectiveness of the personalized convolution. Table 2, 3, and 4 also compare the two ways of calculating the commonality kernel, where the accuracy of using moving average strategy for calculating the commonality kernel during network optimizing is comparable to that of directly calculating over the reference set. This indicates that we can calculate the commonality kernel with the moving-average strategy when the general reference set is used, or directly calculate it over the reference set when more elaborated reference set is available.

To better understand and analyze the sources of the improvement of our method, we also evaluate the same testing set but with different reference sets organized according to the attributes of age and skin tone on IJB-A dataset. Specifically, in the first evaluation, the testing set only contains images with age in [50, max], while four reference sets are organized with ages respectively in [0,max], [0, 35], [35, 50], [50, max]. The results are shown in Fig. 4 (a). As can be seen, when reference set is similar with the testing set, i.e. both with age in [50, max], best accuracy is achieved with significant improvement over the baseline. On the contrary, when the reference set is not so similar with the testing set, such as reference set with age in [0, 35], the performance improvement is limited. Even in the worst case that the reference set (e.g. the pupils) is completely irrelevant to the testing scenario (e.g. the elders), personalized kernel just degenerates to a kind of self-attention CNN (better than vanilla-CNN), because the commonality kernels would capture nothing about the persons in testing scenario under our orthogonal projection formulation. Similarly, in the second evaluation, the testing set only contains images with skin tone in {5, 6}, while four reference sets are organized with skin tone in {1, 2, 3, 4, 5, 6}, {1, 2}, {3, 4} and {5, 6} respectively. The same observation can be obtained, i.e. the higher the similarity between reference set and testing set, the better the recognition accuracy. This verifies that the main source of our improvement comes from the elegant design of personalized kernel.

Variants of reference set As discussed in Sect. 3.5, generally the reference set is used as a proxy of persons in a specific scenario. In special, the reference set can contain a particular group of faces, or even only one face image for different scenarios. To evaluate the variants of personalized CNN, besides the general reference set, three additional kinds of reference set are developed. The first is designed with face images grouped with age, the second is constructed with face images grouped with their skin tone, and the third is our conference work (Han et al., 2018), i.e. containing only one face

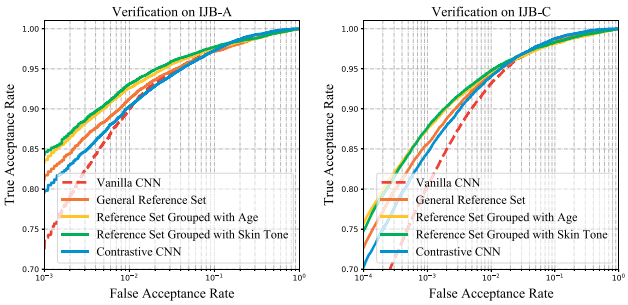


Fig. 5 Performance of personalized CNN w.r.t different reference sets on the IJB-A and IJB-C verification. Best viewed in color

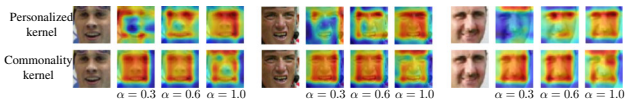


Fig. 6 Visualization of feature map from the personalized kernel and the commonality kernel w.r.t. different α. The first row is the feature maps from the personalized kernel, and the second row is the feature maps from the commonality kernel. In each of the three groups, the first column is the input image, and the rest three ones are the feature maps from the model trained with different α

image in the reference set according to who one face is compared with. Specifically, in the experiments with grouped reference set, all face images in testing set and reference set are divided into three groups respectively according to their age/skin tone. For age as criterion, three groups are with age in interval [0, 35], [35, 50], and [50, max] respectively, while for skin tone as the criterion, they are with skin tone {1, 2}, {3, 4}, and {5, 6} respectively, according to the label on IJB-A and IJB-C datasets. For face images in a specific face group, one's personalized features are extracted with corresponding reference set to highlight the special characteristics.

The personalized CNN with the general reference set and three variants are evaluated on IJB-A and IJB-C datasets as shown in Fig. 5. As can be seen, the personalized CNN significantly outperforms the vanilla CNN benefiting from focusing more on those personalized characteristics. Besides, personalized CNN with more delicate reference set, such as different age groups, different skin tone groups, performs better than that with general reference set. Taking the recognition of children face as an example, if the reference set more accurately depicts the commonality of all children, the common component of children faces will be removed from the input child face. Therefore, the most distinguishing facial characteristics that can distinguish the child from other children are highlighted. However, as can be seen, in the conference version where the reference set contains only one face from the pair faces being compared, the accuracy is slightly declined. This is because with one single face in the reference, the commonality calculated only between two faces may be inaccurate due to the divers inter-person variations.

Table 5 Influence of the balance parameter α in our personalized CNN on LFW, IJB-A, and IJB-C

α	mAcc on	Veri. rate (%) on IJB-A		Iden. rate (%) on IJB-A		Veri. rate (%) on IJB-C			
	LFW(%)	@FAR=1e-2	@FAR=1e-3	@Rank-1	@Rank-5	@FAR=1e-2	@FAR=1e-3	@FAR=1e-4	@FAR=1e-5
0.3	98.58	84.50	68.57	90.93	96.83	89.41	75.70	58.93	41.50
0.4	98.63	84.98	67.50	91.14	96.67	89.67	76.05	60.05	42.71
0.5	**98.80**	85.29	68.64	91.42	96.88	**89.93**	76.68	59.48	43.34
0.6	98.78	**86.08**	**71.71**	91.68	96.77	89.89	**77.39**	**61.10**	**44.75**
0.8	98.85	85.93	71.02	91.73	96.67	89.90	76.20	58.88	41.34
1.0	98.67	84.94	69.16	**91.89**	**96.89**	89.01	75.41	57.98	40.05

Best results are shown in bold

Table 6 Influence of the balance parameter λ in our personalized CNN on LFW, IJB-A, and IJB-C

λ	mAcc on	Veri. rate (%) on IJB-A		Iden. rate (%) on IJB-A		Veri. rate (%) on IJB-C			
	LFW(%)	@FAR=1e-2	@FAR=1e-3	@Rank-1	@Rank-5	@FAR=1e-2	@FAR=1e-3	@FAR=1e-4	@FAR=1e-5
0.0	98.60	85.78	67.98	91.07	**96.90**	**89.96**	75.48	58.06	41.41
0.1	98.77	85.03	68.23	91.13	96.81	89.77	75.72	58.52	41.52
0.3	98.68	85.39	68.36	91.60	96.78	89.95	76.13	59.17	42.62
0.5	**98.78**	**86.08**	**71.71**	**91.68**	96.77	89.89	**77.39**	**61.10**	**44.75**
0.7	98.62	85.33	69.05	91.38	96.62	89.71	75.89	59.30	42.57
0.9	98.45	85.23	68.17	91.62	96.82	89.12	75.78	59.93	42.39

Best results are shown in bold

4.3 Influence of Hyper-Parameters

Influence of parameter α. In Eq. (9), α is used to control the balance between the variance and magnitude of commonality component in the process of optimizing the commonality kernel. Here, we investigate the influence of α by fixing λ as 0.5 since α and λ are independent to each other . The experiments are conducted on our personalized CNN with 4-layer backbone. Specifically, we set the α as 0.3, 0.4, 0.5, 0.6, 0.8, 1.0 respectively. As shown in Table 5, when increasing α, the performance first rises and then descends, with highest accuracy achieved at $\alpha = 0.6$. This shows that both constraints of the small variance and large magnitude for optimizing the commonality kernel are necessary. Only small variance constraint would make the commonality kernel only contain few characteristics, while excessive large magnitude would make the commonality kernel capture many unshared characteristics, both leading to decreased accuracy.

Furthermore, we also visualize the feature maps from the personalized kernel w.r.t. the commonality kernel obtained from different α. As is shown in Fig. 6, with bigger α, the high response region of feature maps from the personalized kernel expands, while that from commonality kernel shrinks as expected, and vice versa.

Influence of parameter λ. In Eq. (20), parameter λ is used to balance the two types of cross entropy loss for feature classification and kernel's intra-person invariance respectively. We investigate the influence of λ by fixing the α as 0.6, and setting λ as 0, 0.1, 0.3, 0.5, 0.7, and 0.9 respectively. The experiments are conducted on our personalized CNN with 4-layer backbone. As shown in Table 6, the performance also first rises and then descends. The highest accuracy is achieved at $\lambda = 0.5$, which means the cross entropy loss over generated kernels is beneficial but can not be weighted too large. Thus, in all experiments, we set $\alpha = 0.6$ and $\lambda = 0.5$ unless otherwise specified.

4.4 Comparison with Existing Methods

The above ablation studies investigate the factors that are involved in our personalized convolution and analyzes the advantage of our method.

Furthermore, we also compare our proposed method with a few state-of-the-art methods on LFW, IJB-A, and IJB-C. In this experiment, personalized CNN with ResNet-50 backbone trained on VGGFace2 is developed for fair comparison as most existing methods are equipped with large architectures and training datasets. Among these compared methods, FaceNet (Schroff et al., 2015), LargeMargin (Liu et al., 2016), SphereFace (Liu et al., 2017), CosFace (Wang et al., 2018), and ArcFace (Deng et al., 2018), UniformFace (Duan et al., 2019) propose new loss functions, PAM (Masi et al., 2016a), 3DMM (Tuan Tran et al., 2017), DR-GAN (Luan et al., 2017), Deep Multi-pose (AbdAlmageed et al., 2016), and FFGAN (Yin et al., 2017) design novel archi-

Table 7 Comparison with existing methods on LFW in terms of mean accuracy (mAcc)

Methods	#Models	Depth of Backbone CNN	#Training Data	mAcc on LFW
DeepFace (Taigman et al., 2014)	3	7	4M*	97.35
DeepID2+ (Sun et al., 2015b)	1	5	300K*	98.70
Deep FR (Parkhi et al., 2015)	1	15	2.6M	98.95
FaceNet (Schroff et al., 2015)	1	14	200M*	99.65
CosFace (Wang et al., 2018)	1	64	5M*	99.73
UniformFace (Duan et al., 2019)	1	30	5.8M	99.70
RegularFace (Zhao et al., 2019)	1	20	3.3M	99.61
ArcFace (Deng et al., 2018)	1	100	5.8M	99.83
DemoID (Gong et al., 2020)	1	100	5.8M	99.50
Yi et al. (2014)	1	10	0.5M	97.73
Ding and Tao (2015)	1	14	0.5M	98.43
LargeMargin (Liu et al., 2016)	1	17	0.5M	98.71
SphereFace (Liu et al., 2017)	1	64	0.5M	99.42
CosFace (Wang et al., 2018)	1	64	0.5M	99.33
Contrastive CNN (Han et al., 2018)	1	16	0.5M	99.12
Personalized CNN	1	16	0.5M	**99.40**
Personalized CNN with ResNet-50	1	50	3.3M	**99.63**
Personalized CNN§ with ResNet-50	1	50	3.3M	**99.68**

Our results are shown in bold
*Denotes the training data is private (not publicly available). §denotes the ArcFace loss is used to train the model

Table 8 Comparison with existing methods on IJB-A, and top-1, top-5 accuracy for identification

Methods	Data	Veri. rate (%) on IJB-A			Iden. rate (%) on IJB-A	
		@FAR=1e-1	@FAR=1e-2	@FAR=1e-3	@Rank-1	@Rank-5
OPENBR	-	43.3	23.6	10.4	24.6	37.5
GOTS	-	62.7	40.6	19.8	44.3	59.5
ReST (Wu et al., 2017)	0.5M	-	63.0	54.8	-	-
LSFS (Dayong et al., 2017)	0.5M	89.3	72.9	51.0	82.0	92.9
PAM (Masi et al., 2016a)	0.5M	-	73.3	55.2	77.1	88.7
3DMM (Tuan Tran et al., 2017)	0.5M	87.0	60.0	-	76.2	89.7
Deep Multi-pose (AbdAlmageed et al., 2016)	0.5M	91.1	78.7	-	84.6	92.7
Triplet Similarity (Sankaranarayanan et al., 2016)	0.5M	94.5	79.0	59.0	88.0	95.0
Joint Bayesian (Chen et al., 2016)	0.5M	96.1	81.8	-	88.2	95.7
FaceID-GAN (Shen et al., 2018)	0.5M	-	87.6	69.2	-	-
FFGAN (Yin et al., 2017)	0.6M	-	85.2	66.3	90.2	95.4
DR-GAN (Luan et al., 2017)	1.2M	-	75.5	51.8	85.5	94.7
PRN (Kang et al., 2018)	3.3M	-	96.5	91.9	98.2	99.2
UniformFace (Duan et al., 2019)	5.8M	-	96.9	92.3	97.9	98.8
DemoID (Gong et al., 2020)	5.8M	-	-	92.2	-	-
Contrastive CNN (Han et al., 2018)	0.5M	98.04	91.20	78.14	-	-
Personalized CNN	0.5M	**98.32**	**91.25**	**82.27**	**94.69**	**97.80**
Personalized CNN with ResNet-50	3.3M	**98.76**	**96.66**	**91.45**	**97.24**	**98.88**
Personalized CNN§ with ResNet-50	3.3M	**98.92**	**96.85**	**93.32**	**97.72**	**98.89**

Our results are shown in bold
Results of GOTS and OPENBR are from (Klare et al., 2015). §denotes the ArcFace loss is used to train the model. It is worth noting that our personalized CNN is not fine-tuned on the training splits of IJB-A to adapt the datasets

Table 9 Comparison with state-of-the-arts on IJB-C

Methods	Data	Verification rate (%) on IJB-C (%)			
		TAR@FAR=1e-2	TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
GOTS (Maze et al., 2018)	-	62.0	33.0	14.7	6.6
FaceNet (Schroff et al., 2015)	200M	81.7	66.5	48.7	33.0
VGGFace (Parkhi et al., 2015)	2M	87.1	74.8	59.8	43.7
VGGFace2 (Cao et al., 2018)	3.3M	94.7	90.9	84.1	74.7
SENet50 (Parkhi et al., 2015)	3.3M	96.0	91.0	84.0	-
DCN (Xie et al., 2018)	3.3M	98.3	94.7	88.5	-
DemoID (Gong et al., 2020)	5.8M	-	92.9	89.4	83.2
Contrastive CNN (Han et al., 2018)	0.5M	94.07	84.54	70.27	54.59
Personalized CNN	0.5M	**94.02**	**85.77**	**74.61**	**59.41**
Personalized CNN with ResNet-50	3.3M	**97.83**	**95.52**	**91.32**	**87.69**
Personalized CNN§ with ResNet-50	3.3M	**98.03**	**96.93**	**94.48**	**91.62**

Our results are shown in bold
Results of GOTS, FaceNet and VGGFace are read from ROC curve in (Maze et al., 2018). §denotes the ArcFace loss is used to train the model

tectures to especially address the large pose variations. Our method can be also regarded as a novel architecture, which is orthogonal to those methods with new loss function (Deng et al., 2018; Liu et al., 2016, 2017; Schroff et al., 2015; Wang et al., 2018), and can be combined for further improvement. Here, the ArcFace loss is employed to investigate that our personalized CNN can be further improved with more advanced loss function.

The comparison on LFW is shown in Table 7. As can be seen, this dataset is almost saturated, and most recent proposed methods achieve performance higher than 99% in terms of mAcc, including Facenet (Schroff et al., 2015), SphereFace (Liu et al., 2017), CosFace (Wang et al., 2018), ArcFace (Deng et al., 2018), RegularFace (Zhao et al., 2019), and UniformFace (Duan et al., 2019). When training on the Webface dataset, our method achieves comparable accuracy to them even with a much shallower backbone. When training on bigger dataset with deeper backbone and more advanced loss, the accuracy of our personalized CNN is further boosted, and even outperforms the recently proposed DemoID (Gong et al., 2020). Those clearly demonstrate the effectiveness of our personalized convolution.

Furthermore, the comparison is performed on the more challenging IJB-A and IJB-C dataset, and the results are shown in Table 8 and 9 respectively. On IJB-A, unlike PAM (Masi et al., 2016a), 3DMM (Tuan Tran et al., 2017), DR-GAN (Luan et al., 2017), Deep Multi-pose (AbdAlmageed et al., 2016), and FFGAN (Yin et al., 2017), which are proposed pose-aware face recognition methods, our method isn't especially designed for dealing with extreme pose variations of faces. But surprisingly, our method still obtains higher accuracy than those specially designed methods when trained on the same dataset, i.e. CASIA-WebFace (Yi et al., 2014).

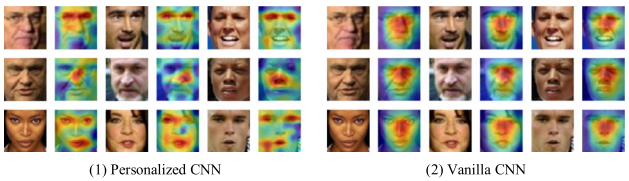


Fig. 7 Comparison of feature maps between our personalized CNN and vanilla CNN. (1) In our personalized CNN, distinct features are focused for different face images. The first row mainly focuses on the regions around eyes, the second row highlights features of nose, and the third row focuses not only eyes but also mouth. (2) In the vanilla CNN, features of all face images are focused almost equally

What's more, our Personalized CNN trained on VGGFace2 dataset even achieves comparable results with UniformFace (Duan et al., 2019) which are trained on a much bigger dataset. On the IJB-C, our personalized CNN is mainly compared to methods trained on VGGFace2 (Cao et al., 2018). As can be seen in Table 9, our method outperforms the recent proposed DemoID (Gong et al., 2020). Those comparisons further show that the extracted personalized features extracted with our personalized CNN can better distinguish different persons even in challenging scenario.

4.5 Visualization and Failure Cases

To intuitively verify whether personalized kernel captures one's personalized characteristics, we visualize the averaged feature maps across channel in the last convolutional layer of our personalized CNN and vanilla CNN as it is easy to see which regions are with higher responses in Fig. 7. However, this does not imply other regions with lower responses are not exploited in the feature representation, though loosely speaking smaller average responses generally mean less important

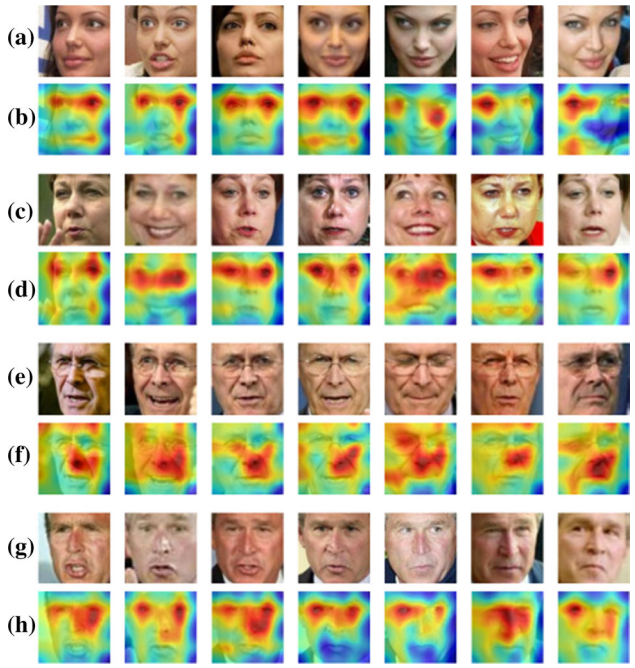


Fig. 8 Illustration of feature map of different images of the same person from personalized kernel. The rows **a**, **c**, **e**, **g** show the images of one person with variance on pose, expression respectively, and the rows **b**, **d**, **f**, **h** show corresponding feature maps from personalized kernel

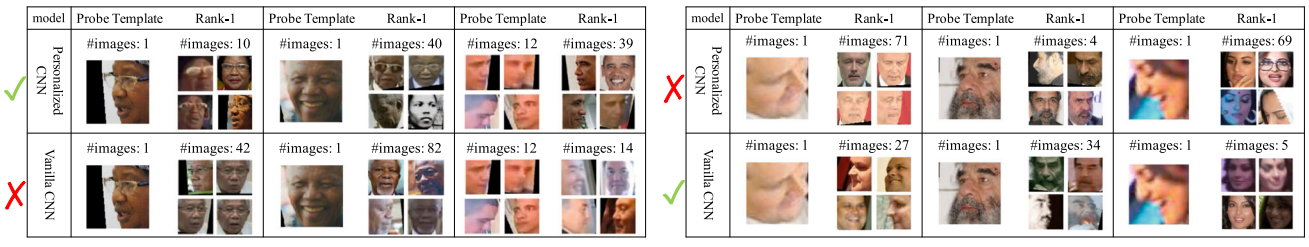
features. As can be seen, our personalized CNN focuses on different features of distinct persons such as one's eyes and another one's mouth, while the vanilla CNN focuses almost the same center features leading to inferior discrimination.

To show the robustness of extracted personalized features of one person to the variance of poses and expressions, we visualize the feature maps of different images for the same person output from the personalized convolutional layer. As shown in Fig. 8, for the same person in each row, the regions with high response of images from the same person are similar even with variance on pose, and expression, e.g, eye regions for persons in the rows (a) (c), nose regions for the person in the row (e), and eye and nose regions for the person in the row (g).

We also show several success and failure cases of our personalized CNN on IJB-A dataset for face identification in Fig. 9. As can be seen from Fig. 9 (a), the personalized CNN can successfully recognize some face images with extreme poses, occlusion, or blur variations while the vanilla CNN can not. However, there are also some failure cases that our method can hardly handle, while the L-vanilla CNN can still recognize correctly as shown in the Fig. 9 (b). The failure cases are substantially less than the success cases. On IJB-A for face verification, at FAR=0.001, the success cases that our personalized CNN can correctly verify while the vanilla CNN can't are up to 12.94%; and the failure cases that are wrongly verified by our personalized CNN but still correctly verified by the vanilla CNN only take up 2.38%. Those failure cases are mainly because our personalized kernel highlights one's specialty that is different from others for better recognition, while at the same time it may also enlarges some within-class variations between the probe and gallery if it is extreme enough to be more like between-class variations (like $a^2 > a$ if $a > 1$). This leaves interesting room for further study.

5 Conclusion and Future Work

Inspired by the observations about humans' face recognition perception, this work proposes a personalized convolution method with personalized kernel to extract one's special features for more accurate face recognition. To achieve this goal, ordinary kernel of each person created by a kernel generator is decomposed into two orthogonal components, i.e. the commonality component and the specialty component. The commonality component is filtered out by a reference set, and the personalized component is retained as the personalized kernel. Especially, the reference set can be constructed as different types for various scenarios. Extensive experiments demonstrate that our method achieves quite promising performance of face recognition, implying that personalized convolution can extract more discriminative features than a fixed convolution.



(a) Success cases **(b)** Failure cases

Fig. 9 Success and failure cases of our personalized CNN. **a** faces that our personalized CNN can correctly recognize, while the vanilla CNN can not. **b** failure cases of our personalized CNN. The notation #images means the number of images in the probe template or Rank-1 template

In the future, we would like to explore more generic structure to deal with those extreme variations of pose, expression, and etc.

Acknowledgements This work was partially supported the National Key Research and Development Program of China (No. 2017YFA0700800), the Natural Science Foundation of China (No. 61772496), and the Beijing Nova Program (Z191100001119123).

References

AbdAlmageed, W., Wu, Y., Rawls, S., Harel, S., Hassner, T., Masi, I., Choi, J., Lekust, J., Kim, J., Natarajan, P., et al. (2016). Face recognition using deep multi-pose representations. In *Winter conference on applications of computer vision (WACV)*.

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 2037–2041.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997) Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 711–720

Bertinetto, L., Henriques, J. F., Valmadre, J., Torr, P., & Vedaldi, A. (2016). Learning feed-forward one-shot learners. In *Advances in neural information processing systems (NeurIPS)*.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018) Vggface2: A dataset for recognising faces across pose and age. In *International conference on automatic face and gesture recognition (FG)*.

Cao, Z., Yin, Q., Tang, X., & Sun, J. (2010). Face recognition with learning-based descriptor. In *Conference on computer vision and pattern recognition (CVPR)*.

Chen, D., Cao, X., Wang, L., Wen, F., & Sun J. (2012). Bayesian face revisited: A joint formulation. In *European Conference on Compute Vision (ECCV)*.

Chen, D., Yuan, L., Liao, J., Yu, N., & Hua, G. (2017). StyleBank: An explicit representation for neural image style transfer. In *Conference on computer vision and pattern recognition (CVPR)*.

Chen, J. C., Patel, V. M., Chellappa, R. (2016). Unconstrained face verification using deep CNN features. In *Winter conference on applications of computer vision (WACV)*.

Chen, Y., Chen, Y., Wang, X., Tang, X. (2014). Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems (NeurIPS)*.

Dayong, W., Charles, O., Jain, A. K. (2017). Face search at scale. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 1122–1136.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2018). Arcface: Additive angular margin loss for deep face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Deng, W., Hu, J., & Guo, J. (2012). Extended src: Undersampled face recognition via intraclass variant dictionary. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 1864–1870.

Ding, C., & Tao, D. (2015). Robust face recognition via multimodal deep face representation. *Transactions on Multimedia (TMM)* 2049–2058.

Duan, Y., Lu, J., Feng, J., & Zhou, J. (2018). Context-aware local binary feature learning for face recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 1139–1153.

Duan, Y., Lu, J., & Zhou, J. (2019). Uniformface: Learning deep equidistributed representation for face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Gong, S., Liu, X., & Jain, A. K. (2020). Jointly de-biasing face recognition and demographic attribute estimation. In *European Conference on Computer Vision (ECCV)*.

Han, C., Shan, S., Kan, M., Wu, S., & Chen, X. (2018). Face recognition with contrastive convolution. In *European conference on computer vision (ECCV)*.

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. J. (2005). Face recognition using laplacianfaces. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 328–340.

Huang, G. B., & Learned-Miller, E. (2014). Labeled faces in the wild: Updates and new reporting procedures. In *Department of Computer Science, Univ. Massachusetts Amherst, Tech. Rep*.

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*.

Huang, Y., Shen, P., Tai, Y., Li, S., Liu, X., Li, J., Huang, F., & Ji, R. (2020a). Improving face recognition from hard samples via distribution distillation loss. In *European Conference on Computer Vision (ECCV)*.

Huang, Y., Wang, Y., Tai, Y., Liu, X., Shen, P., Li, S., Li, J., & Huang, F. (2020b). Curricularface: Adaptive curriculum learning loss for deep face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Jia, X., De Brabandere, B., Tuytelaars, T., & Gool, L. V. (2016). Dynamic filter networks. In *Advances in neural information processing systems (NeurIPS)*.

Jian, Z., Yu, C., Yi, C., Yang, Y., Fang, Z., Jianshu, L., Hengzhu, L., Shuicheng, Y., & Jiashi, F. (2019). Look across elapse: Disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition. In *Conference on artificial intelligence (AAAI)*.

Kang, B. N., Kim,Y., & Kim, D. (2018). Pairwise relational networks for face recognition. In *European Conference on Computer Vision (ECCV)*.

Kang, B. N., Kim, Y., Jun, B., Kim, D. (2019). Attentional feature-pair relation networks for accurate face recognition. In *International conference on computer vision (ICCV)*.

Kang, D., Dhar, D., & Chan, A. (2017). Incorporating side information by adaptive convolution. In *Advances in neural information processing systems (NeurIPS)*.

Kim, Y., Park, W., Roh, M. C., Shin, J. (2020a). Groupface: Learning latent groups and constructing group-based representations for face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Y., Park, W., & Shin, J. (2020b). Broadface: Looking at tens of thousands of people at once for face recognition. In *European Conference on Computer Vision (ECCV)*.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Conference on computer vision and pattern recognition (CVPR)*.

Klein, B., Wolf, L., & Afek, Y. (2015). A dynamic convolutional layer for short range weather prediction. In *Conference on computer vision and pattern recognition (CVPR)*.

Lei, Z., Pietikäinen, M., & Li, S. Z. (2014). Learning discriminant face descriptor. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 289–302.

Liao, S., & Shao, L. (2019). Interpretable and generalizable deep image matching with adaptive convolutions. Computing Research Repository (CoRR).

Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Transactions on Image processing (TIP)* 467–476.

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *International conference on machine learning (ICML)*.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Luan, T., Yin, X., & Liu, X. (2017). Disentangled representation learning GAN for pose-invariant face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Masi, I., Rawls, S., Medioni, G., & Natarajan, P. (2016a). Pose-aware face recognition in the wild. In *Conference on computer vision and pattern recognition (CVPR)*.

Masi, I., Tran, A. T., Leksut, J. T., Hassner, T., & Medioni, G. G. (2016b). Do we really need to collect millions of faces for effective face recognition? In *European conference on computer vision (ECCV)*.

Maze, B., Adams, J. C., Duncan, J. A., Kalka, N. D., Miller, T., Otto, C., Jain, A. K., Niggel, W. T., Anderson, J., Cheney, J., & Grother, P. (2018). Iarpa janus benchmark - c: Face dataset and protocol. In *International conference on biometrics (ICB)* (pp. 158–165).

Moghaddam, B., Wahid, W., & Pentland, A. (1998). Beyond eigenfaces: Probabilistic matching for face recognition. In *International conference on automatic face and gesture recognition (FG)*.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *British machine vision conference (BMVC)*.

Sankaranarayanan, S., Alavi, A., & Chellappa, R. (2016). Triplet similarity embedding for face verification. Preprint arXiv:160203418

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Conference on computer vision and pattern recognition (CVPR)*.

Shen, Y., Luo, P., Yan, J., Wang, X., & Tang, X. (2018). Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *Conference on computer vision and pattern recognition (CVPR)*.

Song, L., Gong, D., Li, Z., Liu, C., & Liu, W. (2019). Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *International conference on computer vision (ICCV)*.

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Conference on computer vision and pattern recognition (CVPR)*.

Sun, Y., Liang, D., Wang, X., & Tang, X. (2015a). Deepid3: Face recognition with very deep neural networks. Preprint arXiv:150200873

Sun, Y., Wang, X., & Tang, X. (2015b). Deeply learned face representations are sparse, selective, and robust. In *Conference on computer vision and pattern recognition (CVPR)*.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *Conference on computer vision and pattern recognition (CVPR)*.

Tan, X., & Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing (TIP)*.

Tian, Z., Shen, C., & Chen, H. (2020). Conditional convolutions for instance segmentation. In *European Conference on Computer Vision (ECCV)*.

Tuan Tran, A., Hassner, T., Masi, I., & Medioni, G. (2017). Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Conference on computer vision and pattern recognition (CVPR)*.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 71–86.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., & Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Wang, H., Gong, D., Li, Z., & Liu, W. (2019). Decorrelated adversarial learning for age-invariant face recognition. In *Conference on computer vision and pattern recognition (CVPR)*.

Wang, M., & Deng, W. (2018). Deep face recognition: A survey. Computing Research Repository (CoRR).

Wenchao., Z, Shiguang., S, Wen., G, Xilin., C, & Hongming., Z. (2005) Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *International conference on computer vision (ICCV)*.

Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A. K., Duncan, J. A., & Allen, K., et al. (2017). Iarpa janus benchmark-b face dataset. In *Conference on computer vision and pattern recognition workshops (CVPRW)*.

Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *Conference on computer vision and pattern recognition (CVPR)*.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 210–227.

Wu, W., Kan, M., Liu, X., Yang, Y., Shan, S., & Chen, X. (2017). Recursive spatial transformer (ReST) for alignment-free face recognition. In *International conference on computer vision (ICCV)*.

Xie, S., Shan, S., Chen, X., & Chen, J. (2010). Fusing local patterns of gabor magnitude and phase for face recognition. *Transactions on Image Processing (TIP)* 1349–1361.

Xie, W., Shen, L., & Zisserman, A. (2018). Comparator networks. In *European conference on computer vision (ECCV)*.

Yan, S., Xu, D., Zhang, B., & Zhang, H. J. (2005). Graph embedding: A general framework for dimensionality reduction. In *Conference on computer vision and pattern recognition (CVPR)*.

Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. Preprint arXiv:14117923

Yin, X., Yu, X., Sohn, K., Liu, X., & Chandraker, M. (2017). Towards large-pose face frontalization in the wild. In *International conference on computer vision (ICCV)*.

Zhang, B., Shan, S., Chen, X., & Gao, W. (2007) Histogram of gabor phase patterns (HGPP): A novel object representation approach for face recognition. *Transactions on Image Processing (TIP)* 57–68.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *Signal Processing Letters* 1499–1503.

Zhang, L., Yang, M., & Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *International conference on computer vision (ICCV)*.

Zhang, R., Tang, S., Zhang, Y., Li, J., & Yan, S. (2017). Scale-adaptive convolutions for scene parsing. In *International conference on computer vision (ICCV)*.

Zhao, K., Xu, J., & Cheng, M. M. (2019). Regularface: Deep face recognition via exclusive regularization. In *Conference on computer vision and pattern recognition (CVPR)*.