Palmprint recognition using realistic animation aided data augmentation

Pranjal, Swarup; Kong, Adams Wai Kin

2019

Pranjal, S. & Kong, A. W. K. (2019). Palmprint recognition using realistic animation aided data augmentation. 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). <https://dx.doi.org/10.1109/BTAS46853.2019.9186003>

<https://hdl.handle.net/10356/153466>

<https://doi.org/10.1109/BTAS46853.2019.9186003>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <https://doi.org/10.1109/BTAS46853.2019.9186003>.

Downloaded on 26 Apr 2023 19:29:21 SGT

Palmprint Recognition Using Realistic Animation Aided Data Augmentation

Swarup Pranjali

School of Computer Science and Engineering
Nanyang Technological University, Singapore

pswarup@ntu.edu.sg

Adams Wai Kin Kong

School of Computer Science and Engineering
Nanyang Technological University, Singapore

adamskong@ntu.edu.sg

Abstract

In this paper, a palmprint augmentation algorithm based on 3D animation is proposed for enhancing contactless palmprint recognition performance. Contactless palmprint varies in position, orientation and musculoskeletal deformations. As the existing contactless databases are small, they contain only a few such variations of a palm. Popular data augmentation approaches, including translation, rotation and scaling, have been used to increase the dataset size and its diversity, but these methods do not simulate non-linear deformation of the hand. Some researchers have used 3D and computer graphic techniques to generate more data for training deep networks. These techniques are application-specific. The proposed algorithm makes use of a 3D hand model to simulate muscular and skeletal deformations of the hand. The deformations from the 3D model are applied to 2D palmprint images to generate new palmprint images with the same identities. Four deep networks, Alexnet, VGG-16, Resnet-50 and Inception-V3 and two contactless palmprint databases, IITD and CA-SIA, are employed to evaluate the proposed algorithm. The proposed algorithm is compared with the standard augmentation methods. The experimental results show that the proposed augmentation algorithm reduces EER and Rank-1 error rate.

1. Introduction

Palm contains rich features, including flexions, secondary creases and ridges for personal identification [12]. Even genetically identical palms have different palmprints [14]. The previous palmprint studies can be categorized as latent, contact-based, and contactless. Figure 1 shows three types of images. Latent palmprints collected from crime scenes are high-resolution images with ridges and minutiae for criminal and victim identification. Figure 2 shows a high-resolution latent image with detailed features. Some methods have been developed to extract the features and match latent palmprints [9][3]. Contact-based palmprint

Figure 1: Examples of latent, contact-based and contactless palmprints

recognition uses tailor-made palmprint scanners to collect palmprint images. Since users are required to put their hand on the scanners, which have pegs for placing their hands, contact-based palmprints are always with high-quality [26]. With user cooperation and the pegs, aligning these palmprint images can be easily achieved by a traditional method. Coding methods such as CompCode[13], OrdinalCode[23], DOC[5], BOCV[7] designed for contact-based palmprint recognition have achieved high recognition performance.

Figure 2: Example of a high-resolution latent palmprint

These contact-based palmprint recognition methods are suitable for civil and commercial applications. For hygiene and high user acceptance, some researchers study contactless palmprint recognition. The differences between contact-based and contactless palmprint recognition are that in the contactless setting, the user is not required to touch the scanner, and there are no pegs for placing their hand. More clearly, contactless palmprint images are taken without any strict guidelines and the hand position, orientation and posture can vary.

Thus, in these images, regions of interests, i.e. the cen-

Figure 3: Example frames of the animated hand used to extract the distorted palm ROI's.

tral part of a palm, are not well aligned and the methods that offer high performance for contact-based images do not produce similar results on contactless images. To alleviate these problems, deep learning based methods are applied. However, the power of deep learning cannot be fully exploited in contactless palmprint recognition because the existing contactless palmprint databases are small. The largest contactless palmprint database has only 5502 images from 602 palms. Comparing with ImageNet and COCO datasets for object classification and detection and face recognition databases, e.g., LFW face database, palmprint databases are very small. Furthermore, in real-world applications, requiring users to provide more images in registration stages would reduce user acceptance. Although traditional data augmentation methods, including translation, rotation, and scaling, have been applied to train deep networks for contactless palmprint recognition, they cannot simulate realistic deformations of hand. In this paper, a 3D hand model is used to simulate hand deformations, and the deformation fields are applied to 2D palmprint images such that palmprints with different realistic deformations can be synthesized. The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the proposed data augmentation algorithm. Section 4 reports the experimental results. Section 5 discusses the optimal number of synthetic images. Section 6 gives some conclusive remarks.

2. Related works

As mentioned before, there are three types of palmprint research, contact-based, contactless and latent. Since this study is for contactless palmprint recognition, only previous research related to contactless palmprint recognition is given in this section. Before deep learning era, some popular hand-crafted features were applied to contactless palmprints. Wu et al. and Morales et al. [25, 18] used SIFT and Kang et al. [10] used local binary patterns (LBP) as feature extractors. Kanhangad et al. [11] attempted to extract 3D information from 2D images to increase robustness against pose variation. Fei et al. [6] proposed a double half-orientation feature extraction method, which was modified CompCode [13] by replacing its Gabor filter bank with a bank of half-Gabor filters and their method was de-

signed to capture information from curly palm lines or palm lines two dominant orientations. In addition to traditional hand-crafted feature methods, Fei et al. evaluated popular deep networks on contactless palmprint databases [4]. It shows that the deep networks outperform the hand-crafted feature extraction methods. In this evaluation, traditional data augmentation methods were applied. In addition to palmprint, deep learning has been applied to other biometric traits such as the face, fingerprint, iris, vein, signature, gait and palmprint recognition. Some tailor-made data augmentation techniques have been designed for them. Using 3D face models to synthesize face images with different poses and viewpoints is a well-known one [17]. Charalambous et al. [2] used skeletal animation of 3D models to augment their gait recognition dataset. They used the MakeHuman software to generate human models that match the subjects in their dataset and then created a script to change the poses of the models and rendered images with different poses using the Blender software. In addition to biometrics, data augmentation is also important for other domains. Simard et al. [20] used elastic distortion to generate additional data for improving network performance on the MNIST dataset. In general, these data augmentation methods are very application-specific.

3. A method for non-linear data augmentation

3.1. Capturing deformations of the hand

A 3D hand model illustrated in Figure 4 is downloaded from the Internet. Applying bone and muscle rig on the hand model, common hand poses are animated. Using the Unity 3D game engine and rendering software, a sequence of images from a realistic hand animation is rendered. 10 frames with the most variations are selected from the rendered images. Figure 3 shows six frames with different hand poses from the animation sequence. The animation sequence is rendered twice with the exact same camera position, one with normal hand texture and the other with tracking points, i.e., red dots marked in the ROI region of the palm. Using the tracking points as a reference, the regions of interest (ROI) in the selected frames with normal texture are cropped.

Figure 4: [Top] Hand model mesh, [Bottom] 2 frames of bone and muscle rig.

The ROIs are then scaled and rotated to match the height, width and orientation of the ROIs of a given dataset. To determine the deformations among different frames, smoothing and edge detection algorithms are applied to the selected frames. The edge points are used as control points in the next subsection for computing the correspondences between a reference frame and the other frames and finally computing their deformations. Figure 5 illustrates the muscle deformations on the surface of the palm.

Figure 5: Illustration of the muscle deformations on the surface of the palm.

3.2. Non-rigid point cloud registration using CPD

Myronenko et al.[19] introduced the Coherent Point Drift algorithm, which is a probabilistic method for non-rigid point set registration. The CPD algorithm assumes that the points in one dataset correspond to the points in the other dataset and that the correspondence can be inferred using a probabilistic approach. The ROI extracted from the first frame of the animation is considered as the reference Image mI , and the ROI extracted from the other frame is considered as the target Image tI . Let v be a transformation

function, which describes the movement of the two images:

$$\hat{m}I = mI + v(mI) \quad (1)$$

Where $\hat{m}I$ is an approximation of tI . Given two sets of points detected by the edge detector (Section 3.1), one from mI and the other from tI , the transformation function v can be obtained through CPD. Let y_m and x_n be the points respectively from mI and tI . CPD uses the expectation maximization (EM) algorithm to estimate v . In the E-step, the point correspondences between y_m and x_n are established and in the M-step, the transformation v is updated. The algorithm considers y_m as GMM centroids and uses maximum likelihood estimation to fit y_m to x_n , which is implemented by minimizing the energy function:

$$E(\hat{m}I) = - \sum_{n=1}^N \log \sum_{m=1}^M e^{-\frac{1}{2} \|\frac{x_n - y_m}{\sigma}\|^2} + \frac{\lambda}{2} \phi(v) \quad (2)$$

Let $\hat{y}_m$ be the corresponding points of y_m in mI . Note that $\hat{y}_m$ and y_m have the same number of points and the points in $\hat{y}_m$ are close to x_m . CPD is a non-rigid point registration method and the numbers of points in y_m and x_n can be different. These properties make it suitable for capturing the deformation of hands because the numbers points extracted from two frames are not always the same, and hand deformation is non-rigid. Algorithm 1 gives the pseudo code of the application of CPD to capture the deformation on hand. In the experiments, ten frames are selected from the animation sequence. Using Algorithm 1, ten point sets pairs, denoted as $[p_0, \hat{p}_0], [p_1, \hat{p}_1], \dots [p_9, \hat{p}_9]$ are generated. p_i is the edge points in the reference frame, and $\hat{p}_i$ is the corresponding new positions in one of the selected frame.

3.3. Approximate thin plate spline mappings

Using the detected point sets i.e. $[p_k, \hat{p}_k]$, given by Algorithm 1 and Thin plate spline (TPS) interpolation proposed by Bookstein [1], ROIs in real palmprint images can be warped. Let $(\hat{x}_i, \hat{y}_i)$ be a point in $\hat{p}_k$ and (x_i, y_i) be a point in p_k , where $k = 0, \dots, 9$ and $i = 1, \dots, n$. Given a pixel location (x, y) in a ROI, its warped position can be determined by

$$f(x, y) = a_1 + a_2x + a_3y + \sum_{i=1}^n \omega_i U(\|(\hat{x}_i, \hat{y}_i) - (x, y)\|) \quad (3)$$

where a_1, a_2, a_3 and ω_i are parameters and $U(r) = r^2 \log r$. The parameters a_1, a_2, a_3 and ω_i can be obtained by minimizing the non-negative bending energy

$$E_f = \iint_{R^2} (f^2_{xx} + 2f^2_{xy} + f^2_{yy}) dx dy \quad (4)$$

More precisely, they are determined by

$$W = [\omega_1, \omega_2 \cdots \omega_n, a_1, a_2, a_3] = L^{-1}Y \quad (5)$$

where L and Y are defined as

$$Y = [v_1 \ v_2 \ \cdots \ v_n \ 0 \ 0 \ 0]^T, \ n+3 \quad (6)$$

$$L = \begin{bmatrix} K & P \\ P^T & O \end{bmatrix}, \ (n+3) \times (n+3) \quad (7)$$

The sub-matrices K , P and O in L are given below and U_{ij} in K is defined as $U(\|(x_i, y_i) - (\hat{x}_j, \hat{y}_j)\|)$.

$$K = \begin{bmatrix} 0 & U_{12} & U_{13} & \cdots & U_{1n} \\ U_{21} & 0 & U_{23} & \cdots & U_{2n} \\ U_{31} & U_{32} & 0 & \cdots & U_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{n1} & U_{n2} & U_{n3} & \cdots & 0 \end{bmatrix}, \ n \times n \quad (8)$$

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}, \ 3 \times n \quad (9)$$

$$O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ 3 \times 3 \quad (10)$$

A regularization parameter λ is introduced to control the amount of smoothing in TPS. That is the submatrix K in Eq. 7 is replaced by $K + \lambda I$, where I is an $n \times n$ identity matrix. When the regularization parameter λ becomes larger, the estimated warp becomes more rigid [16]. The TPS coefficients can be estimated by solving the linear system mentioned in Eq. 5. Once the TPS coefficients are obtained, points on ROIs are transformed to new locations.

3.4. Transfer of non-linear deformation to dataset images

Using TPS obtained in the previous subsection, the ROIs in a given database can be non-linearly deformed according to the hand gesture of the hand model. It is assumed that input ROIs and the reference frame have similar hand gesture. Each ROIs in the databases are processed by the Algorithm 2 to generate multiple deformed ROIs. These deformed images form an augmented dataset. The parameter γ in Algorithm 2 helps accentuate linear deformations. In the experiment, it is set to 3.0 and the TPS regularization parameter λ is set to 2.0 and in Algorithm 1 α, β are set to 2.0.

input : A set of l animated ROI frames as jpeg
output: A set of point sets on detected edges LEP and a set of their corresponding deformed point sets LDP

```
Reference frame  $f = I_0$ ;
 $EP_f \leftarrow \text{DetectPointsOnEdges}(f)$ ;
 $LEP$ , define an empty set to store detected point sets;
 $LDP$ , define an empty set to store deformed point sets;
for  $i \leftarrow 1$  to  $l$  do
     $I_i$ , is the rendered image of the  $i$ th frame;
     $EP_i \leftarrow \text{DetectPointsOnEdges}(I_i)$ ;
     $\alpha \leftarrow 2.0, \beta \leftarrow 2.0$ ;
     $DP_i \leftarrow \text{DeformableRegistration}(EP_f, EP_i, \alpha, \beta)$ ;
     $LEP.insert(EP_i)$ ;
     $LDP.insert(DP_i)$ ;
```

end

Algorithm 1: Point set detection and deformation

input : A set of point sets on detected edges LEP and a set of their corresponding deformed point sets LDP ; Set of palm ROI images from the datasets S

output: Augmented dataset of ROI images

```
 $\gamma$ , amplitude of distortion;
 $\lambda$ , regularization parameter;
 $X$ , source point set;
 $Y$ , destination point set;
 $I_i$ , is the  $i$ th image of the dataset  $S$ ;
 $m \leftarrow \text{LengthOf}(S)$ ;
 $n \leftarrow \text{LengthOf}(LEP)$ ;
for  $i \leftarrow 1$  to  $m$  do
     $I_i \leftarrow S[i]$  for  $j \leftarrow 1$  to  $n$  do
         $\gamma \leftarrow 3.0$ ;
         $\lambda \leftarrow 2.0$ ;
         $X \leftarrow LEP[j]$ ;
         $Y \leftarrow LDP[j]$ ;
         $Y \leftarrow X + (Y - X) \times \lambda$ ;
         $I_{ij} \leftarrow \text{TpsWarpImage}(Y, X, I_i, \lambda)$ ;
         $\text{Save}(I_{ij})$ ;
```

end

end

Algorithm 2: TPS warp algorithm to apply non-linear deformation on the dataset images

4. Results

4.1. Palmprint databases

The IITD contactless palmprint dataset consists of 2,300 images taken from 460 different palms of 230 individuals.

Figure 6: Illustration of the points movement from Algorithm 1. The red points are from the reference frame and the blue points are their new positions.

Five images are taken from each palm. The palmprint images in this database have variations in pose, orientation and position. This dataset also provides regions of interest (ROI) of palmprint images for evaluating feature extraction and classification methods. In the experiment, the ROIs are inputted to deep networks with different data argument methods. The CASIA database consists of over 5500 palmprint images collected from 312 individuals. Eight to seventeen palmprint images were captured from both the left and right palms for an individual. Due to the fact that there were no guidelines on the posture and position placement of the hands during image acquisition, the palmprint images generally have significant variations on the pose, ori-

entation and position. As the ROIs are not provided with the CASIA datasets, we used a method to extract the ROIs. Figure 7 shows some sample original images and ROI from these two databases. In the experiments, the first four images of each palm are used for training and the rest are used for testing, same as Fei et al. [4].

Figure 7: (a) and (b) are respectively a sample palmprint from CASIA dataset and its ROI and (c) and (d) are respectively a sample palmprint from IITD dataset and its ROI.

4.2. Networks and parameters

The deep learning architectures used to compare the results of the non-linearly transformed augmented datasets are Alexnet, VGG-16, Resnet-50 and Inception-V3. Krizhevsky et al.[15] proposed a popular CNN based architecture called Alexnet that significantly improved the image classification results on the ImageNet dataset. Other popular CNN based architectures namely VGGNet[21], GoogleNet[24] and ResNet[8] are primary classification models used in this domain. Alexnet contains five convolutional layers and three fully connected layers. ReLu function is used for non-linearity, and the softmax function is used to calculate the normalized probability distribution of K possibilities. VGG-16 uses 3×3 filters with a stride of 1 in the convolution layer, and it has five pooling layers and thirteen convolutional layers. The GoogLeNet deep convolutional architecture introduced in 2015 has later revised to add batch normalization, gradient clipping and other factorization ideas to form its later generation Inception architectures. The main module of this architecture is an Incep-

tion block, which passes the input to multiple convolutions and pooling towers of different size filters and concatenates the outputs together. The Inception-V3 architecture comprises of a stem layer which has 5 convolutional layers, followed by 3 inception modules, a grid size reduction module and then 4 more inception modules. Then, the features are passed to an auxiliary classifier module and in parallel to a grid size reduction module followed by 2 more inception modules and then to the primary classifier. The network is trained by applying loss functions on both the primary and auxiliary classifiers. The Residual Networks (Resnet) architecture adds identity shortcut connections to bypass signals from one layer to the next layer. The Highway Network [22] introduced gated shortcut connections. It is different from a plain feed-forward neural network because the input of each block is convolved with the layers of the current block and then added with its original input before passing to the next block in the hierarchy. The Resnet-50 model uses blocks that have 2 convolution layer of filter size of 1×1 , and a layer with filter size 3×3 with padding of 1, by keeping the stride at 1 the input is not down sampled, and it can be added to the identity input; it also performs batch normalization after each convolution layer. The Resnet-50 model starts with a convolution and pool layer followed by 4 layers with 3, 4, 6, 3 blocks each and a global average pooling layer, followed by a classifier. The input image size is 224×224 pixels for all the networks, except for the Inception-V3 network whose input size is 299×299 pixels. The batch size was set to 32. The pre-trained models on ImageNet are employed for all the networks. They are retrained on the palmprint databases using the hyper-parameters. All the networks are trained with 1000 epochs. The initial learning rate of 0.01 is used for all the networks, except for VGG-16, whose initial learning rate is 0.001. A cosine annealing learning rate scheduler was used along with an SGD optimizer with a weight decay of 0.0005. All the networks are trained using a Ubuntu 18.04 workstation running on an Intel Xeon(R) E5-1650 v4 CPU installed with an NVidia GTX 1080 Ti GPU card. All the models used are defined in the torchvision library and are trained with Pytorch 1.0 and CUDA version 9.0. For Alexnet, all the weights till the second max pooling layer are frozen and all the weights above, up to the last fully connected layer, are retrained. For VGG-16, the weights till the second max pooling layer are frozen, and all the weights above it are retrained. All the layers in Resnet-50 are retrained for our experiments. For Inception V3, the weights of all the basic convolution layers are frozen while the weights of all the Inceptions blocks and other weights up to the final linear layer are retrained.

4.3. Comparisons and results

In these experiments, rank-1 identification accuracy, equal error rates and ROC curves are used as performance

indexes to compare different augmentation methods. To compare the proposed data augmentation algorithm and the previous data augmentation methods applied on palmprints, four settings are evaluated on the two datasets. The first setting is the original palm ROI datasets without using any data augmentation techniques. It is denoted as NoAug in the results Tables 1 and 2.

Table 1: Comparisons with different augmentation methods on IITD. (a) Rank-1 error (%) and (b) EER (%).

(a)				
Model	NoAug	Affine	Nlt	NltTrans
Alexnet	7.734	6.099	6.005	3.104
VGG-16	4.536	2.974	4.760	2.250
Resnet-50	14.708	1.599	1.729	1.172
Inception-V3	7.307	1.469	1.302	0.781

(b)				
Model	NoAug	Affine	Nlt	NltTrans
Alexnet	0.801	0.403	0.881	0.263
VGG-16	0.394	0.505	0.418	0.169
Resnet-50	1.271	0.131	0.131	0.129
Inception-V3	0.891	0.202	0.208	0.177

Table 2: Comparisons with different augmentation methods on CASIA. (a) Rank-1 error (%) and (b) EER (%).

(a)				
Model	NoAug	Affine	Nlt	NltTrans
Alexnet	23.298	13.825	19.840	9.647
VGG-16	14.577	8.520	12.904	7.796
Resnet-50	5.461	11.32	4.671	4.079
Inception-V3	6.217	6.283	4.408	4.572

(b)				
Model	NoAug	Affine	Nlt	NltTrans
Alexnet	3.242	1.753	3.242	1.422
VGG-16	1.985	1.323	1.819	1.095
Resnet-50	0.863	2.084	0.633	0.662
Inception-V3	0.926	1.089	0.863	0.767

The second one applies rotation and translation on the ROIs, and then, central regions of the ROIs are copped. This setting is denoted as Affine in the tables. The rotation angles are in the range of -15 to +15 degrees, and the images are randomly translated in the range of -15 to +15 pixels in both X and Y directions. The third setting applies the proposed algorithm on the ROIs to generate 10 non-linearly

Figure 8: Comparisons of different augmentation methods on IITD and CASIA databases and on different networks.

deformed palmprint images from each of the original images. The central parts are cropped. This setting is denoted as NIt. The last setting uses the proposed algorithm and linear translation together. The central parts are cropped after translation. It is denoted as NItTrans. The ROIs from IITD database are centre cropped to 90% of the height and width of the original ROIs., while the ROIs from CASIA database are cropped to 87% of the height and width of the original ROIs. Then they are resized to the input image size for the respective networks. Table 1(a) and (b) respectively list rank-1 error and EER from IITD database. Table 2(b) and (b) respectively list rank-1 error and EER from the CASIA database. Comparing with the original ROIs (NoAug), the proposed augmentation algorithm reduces the rank-1 error and EER in all the experiments, except for VGG-16 on IIT database. For the IITD database, the proposed algorithm and affine augmentation perform similarly. However, for the CASIA database, they perform differently. For the low-performance networks, i.e., Alexnet and VGG-16, the affine method outperforms the proposed algorithm but for the high-performance networks, i.e., Resnet-50 and Inception V3, the proposed algorithm performs better. Combining the affine method and the proposed algorithm (NItTrans) achieves the best results in all the experiments, except for Resnet-50 on CASIA database. It pinpoints that the proposed algorithm and the affine method offer comple-

mentary information to enhance network performance.

5. Number of generated images

The previous experiments show that the proposed algorithm can improve the network performance. Is it possible to further improve the performance by generating more images per original palmprint? In this experiment, 5, 10 and 20 images are generated from each original image. The deformed frames are equally sampled from the animation sequence (Section 3.1) for generating the images. CASIA database and Inception-V3 network are employed in this experiment. No affine translation is applied. Their ROCs are given in Figure 9. Comparing with the ROCs from the training datasets with 5 and 10 augmented images per palmprint, there is significant improvement by increasing the number of augmented images. However, there is no significant difference between the two ROCs from the training datasets with 10 and 20 augmented images per palmprint.

6. Conclusion

To simulate palmprint images with realistic non-linear deformation; in this paper, an algorithm makes use of a 3D hand model to simulate muscular and skeletal deformations of the hand, is proposed. Four deep networks, Alexnet, VGG-16, Resnet-50 and Inception-V3 and two contactless palmprint databases, IITD and CASIA, are employed to

Figure 9: Comparisons of the performance from the training datasets with different numbers of augmented images.

evaluate the proposed algorithm. The proposed algorithm is compared with the standard augmentation methods. The experimental results demonstrate the effectiveness of the proposed algorithm in improving the recognition performance of the networks.

Acknowledgement

This work is partially supported by the Ministry of Education, Singapore through Academic Research Fund Tier 2, MOE2016-T2-1-042(S).

References

- [1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
- [2] C. C. Charalambous and A. A. Bharath. A data augmentation methodology for training machine/deep learning gait recognition algorithms. *arXiv preprint arXiv:1610.07570*, 2016.
- [3] J. Dai and J. Zhou. Multifeature-based high-resolution palmprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):945–957, 2011.
- [4] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang. Feature extraction methods for palmprint recognition: A survey and evaluation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(2):346–363, Feb 2019.
- [5] L. Fei, Y. Xu, W. Tang, and D. Zhang. Double-orientation code and nonlinear matching scheme for palmprint recognition. *Pattern Recognition*, 49:89–101, 2016.
- [6] L. Fei, Y. Xu, and D. Zhang. Half-orientation extraction of palmprint features. *Pattern Recognition Letters*, 69:35–41, 2016.
- [7] Z. Guo, D. Zhang, L. Zhang, and W. Zuo. Palmprint verification using binary orientation co-occurrence vector. *Pattern Recognition Letters*, 30(13):1219–1227, 2009.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [9] A. K. Jain and J. Feng. Latent palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1032–1047, 2009.
- [10] W. Kang and Q. Wu. Contactless palm vein recognition using a mutual foreground-based local binary pattern. *IEEE Transactions on Information Forensics and Security*, 9(11):1974–1985, 2014.
- [11] V. Kanhangad, A. Kumar, and D. Zhang. Contactless and pose invariant biometric identification using hand surface. *IEEE Transactions on Image Processing*, 20(5):1415–1424, 2011.
- [12] A. Kong, D. Zhang, and M. Kamel. A survey of palmprint recognition. *Pattern Recognition*, 42(7):1408–1418, 2009.
- [13] A. K. Kong and D. Zhang. Competitive coding scheme for palmprint verification. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 520–523. IEEE, 2004.
- [14] A. W.-K. Kong, D. Zhang, and G. Lu. A study of identical twins palmprints for personal verification. *Pattern Recognition*, 39(11):2149–2156, 2006.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, pages 1097–1105, USA, 2012. Curran Associates Inc.
- [16] J. Lim and M.-H. Yang. A direct method for modeling non-rigid motion with thin plate spline. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1196–1202. IEEE, 2005.
- [17] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*, pages 579–596. Springer, 2016.
- [18] A. Morales, M. A. Ferrer, and A. Kumar. Towards contactless palmprint authentication. *IET Computer Vision*, 5(6):407–416, 2011.
- [19] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, Dec 2010.
- [20] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963, Aug 2003.
- [21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [22] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.
- [23] Z. Sun, T. Tan, Y. Wang, and S. Z. Li. Ordinal palmprint representation for personal identification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 279–284. IEEE, 2005.

- [24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [25] X. Wu, Q. Zhao, and W. Bu. A sift-based contactless palmprint verification approach using iterative ransac and local palmprint descriptors. *Pattern Recognition*, 47(10):3314–3326, 2014.
- [26] D. Zhang, W.-K. Kong, J. You, and M. Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1041–1050, 2003.