



Deep Palmprint Recognition with Alignment and Augmentation of Limited Training Samples

Dane Brown¹ · Karen Bradshaw¹

Received: 5 December 2020 / Accepted: 6 September 2021 / Published online: 23 October 2021
© The Author(s) 2021

Abstract

This paper builds upon a previously proposed automatic palmprint alignment and classification system. The proposed system was geared towards palmprints acquired from either contact or contactless sensors. It was robust to finger location and fist shape changes—accurately extracting the palmprints in images without fingers. An extension to this previous work includes comparisons of traditional and deep learning models, both with hyperparameter tuning. The proposed methods are compared with related verification systems and a detailed evaluation of open-set identification. The best results were yielded by a proposed Convolutional Neural Network, based on VGG-16, and outperforming tuned VGG-16 and Xception architectures. All deep learning algorithms are provided with augmented data, included in the tuning process, enabling significant accuracy gains. Highlights include near-zero and zero EER on IITD-Palmprint verification using one training sample and leave-one-out strategy, respectively. Therefore, the proposed palmprint system is practical as it is effective on data containing many and few training examples.

Keywords Deep learning · Image augmentation · Palmprint alignment · Open-set identification · Verification

Introduction

Palmprints are contained within a region of human hands and consist of ridges, similar to fingerprints, forming unique patterns. Palmprints may contain more useful information about humans than fingerprints as the area of the pattern is bigger [1, 21, 41]. However, this means that palmprint sensors are consequently bulkier and costlier than fingerprint sensors.

Besides the similar ridge features of the fingerprint and palmprint, the palmprint also contains principal lines and wrinkles. This allows for capturing at lower resolutions and greater distances using cheaper sensors or DSLR cameras [1, 30]. This motivated the construction of a robust palmprint recognition system in previous work [7], which included effective region of interest (ROI) detection for

both constrained and ‘unconstrained’ palmprint image acquisition.

This paper *extends that system* by utilizing various image preprocessing techniques on traditional machine learning and adding hyperparameter tuning. The resulting best combinations are next compared with deep learning approaches—including popular convolutional neural network architectures and one that is empirically determined using the hyperparameter tuning library known as Keras-Tuner [26]. Finally, overfitting is reduced with improved generalization in a minimal training data subset.

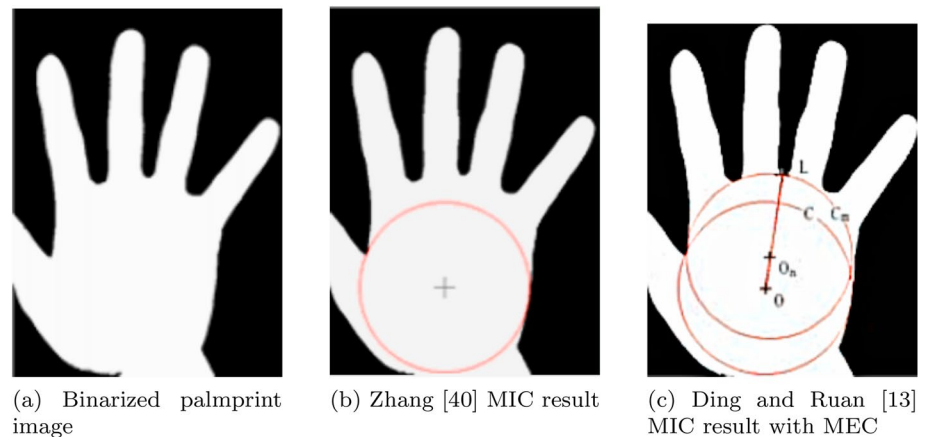
The document structure is as follows. Section “[Related Studies](#)” investigates palmprint systems in the literature and weighs their strengths and weaknesses. Section “[Overview of Palmprint Alignment System](#)” explains the proposed palmprint segmentation approach. This is followed by Section “[Traditional Classification](#)” and “[Classification using Deep Learning](#)” on traditional and deep learning, respectively. Test metrics for verification and identification system evaluation are provided in section “[Palmprint Identification Metrics](#)” and experimental results are analyzed. The proposed system is compared fairly to related systems that evaluated public datasets. The paper is discussed and

This article is part of the topical collection “Advances on Signal Image Technology and Internet based Systems” guest edited by Albert Dipanda, Luigi Gallo and Kokou Yetongnon.

✉ Dane Brown
d.brown@ru.ac.za

¹ Rhodes University, Grahamstown, South Africa

Fig. 1 Palmprint segmentation using the maximum inscribed circle



concluded in sections “[Discussion of Palmprint Recognition](#)” and “[Concluding Remarks](#)”.

Related Studies

A maximum inscribed circle (MIC) contained within the palm contains the unique features of palmprints [40]. The MIC is tangent to certain keypoints outside the solution. The MIC-based method is particularly popular due to its robust capturing of both contactless and contact sensor data and is thus explored for later adoption.

Figure 1a shows a typical binary thresholded hand image. The MIC of the palm can be calculated by locating the largest circle that can fit it. Palmprints can be aligned and subsequently segmented from the background this way without fingers, partial hands or other challenging scenarios. However, combining the MIC with the finger valley keypoints can lead to better accuracy, assuming that at least one finger valley is within the image.

Zhang [40] performed preprocessing in the form of silhouette detection before calculating the MIC.

Silhouette detection proceeded by first locating the centre-most white pixel. This initial centre was an anchor point for increasing the circle radius while shifting it in four directions until black pixels are reached. The MIC result is shown in Fig. 1b. State-of-the-art at the time, two studies used Zhang [40]’s coordinate system as a foundation for improved palmprint segmentation.

Ding and Ruan [13] stated that scale, translation, and rotation invariance could be achieved by modifying the resulting MIC’s location. They moved the mic towards the middle-ring finger valley as depicted in Fig. 1c. Their reasoning stemmed from the Zhang [40]’s MIC typically being close to the heel of the hand. This region contains redundant information. Hence, the remedy is to shift the mic upward to obtain a maximum effective circle (MEC).

Choge et al. [10] similarly found the centre of the palmprint such that the radius can pass through the ring-little and index-middle finger valleys.

The additional anchor point was argued to be more robust to changes in hand pose from contactless sensors. Another change is that the circular ROI, i.e. the MEC, is unwrapped as a rectangular ROI with a fixed size.

Both modified approaches do not adapt the circle centre or radius when the palmprint’s pose (fist or spread fingers) is changed. As such, there is room for more improvement, especially on data from contactless sensors. Our proposed solution to this is in section “[Overview of Palmprint Alignment System](#)”.

With respect to deep learning literature, Jalali et al. [18] implemented a four-layer Convolutional Neural Network (CNN) model. However, no palmprint ROI segmentation was performed, and it was trained on the entire hand. This yielded similar accuracies as wavelet and subspace-based methods involving PCA and LDA. It showed how robust CNNs are against shift and distortion and the general lack of palmprint alignment. The algorithm was effective on the controlled PolyU dataset, as expected. However, it was also effective on their own 10-subject 20-sample contactless dataset, captured with a digital camera. Testing CNNs on larger contactless datasets could thus yield interesting results.

Zhao et al. [44] used a two-layer Restricted Boltzmann Machine but trained in an unsupervised manner on 32×32 palmprint ROIs.

Minaee and Wang [24] performed palmprint recognition using a two-layer deep scattering convolutional network,¹ again demonstrating the effectiveness of CNNs but in supervised learning.

¹ Predefined wavelet transform instead of data-driven filter-based training.

Dian and Dongmei [12] used Alexnet CNN [20] on segmented palmprints from hand images and achieved promising results. Newer CNN architectures can potentially improve those results [11, 32].

Svoboda et al. [37] trained CNNs on palmprint ROI images and observed that loss functions significantly affect palmprint verification, especially with the presence of impostors.

CNNs are understudied in palmprint recognition. However, they have been used for auxiliary tasks such as palmprint ROI extraction instead of relying on image processing techniques. Other tasks include using CNNs to distinguish the left from the right hand while also relying on keypoints extraction for accurately obtaining the palmprint ROI for segmentation [3].

While the robustness of CNNs against misalignment and distortion in colour object detection, such as ImageNet [20] is well known, it can be particularly interesting to determine whether their feasibility extends to greyscale images of palmprints. This kind of data, like the fingerprint and iris, contain detailed patterns from images. The greyscale nature of the data, coupled with less control during data acquisition from a distance, will guide the extension of the automatic palmprint alignment and classification system.

Overview of Palmprint Alignment System

This section explains the proposed palmprint segmentation solution, based on the improved alignment of Zhang et al. [41]'s system. This solution is toward an effective palmprint recognition system that can be realized later in this paper. The process is visualized in the figures using PolyU image data.

Finger-valley keypoints are first used as anchors to lower intra-class variation of palmprint classes, especially but not limited to contactless sensors. Subsequently, Zhang et al. [41]'s MIC method is carried out to obtain an ROI.

It is crucial that keypoints are detected and used reliably but adaptively to accommodate a process without guiding pegs and possible partials. The keypoints are thus used in an alignment procedure that performs affine transformations to align test images based on training examples. With this robust finger-valley keypoint alignment process, the MIC method is instead applied to warped images, using the centre valley point as a reference to obtain the final ROI.²

² The equivalent of the MEC

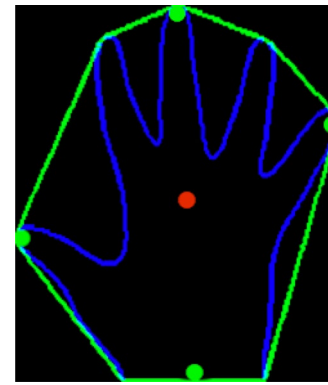


Fig. 2 Convex hull of hand in green and red centre used as the initial search space [taken from [7]]



Fig. 3 Wide captured input image

Hand Image Presegmentation

Data acquisition typically involves accepting an unprocessed hand image as input. Crude background segmentation is performed using the standard Otsu thresholding algorithm [27] after a 15×15 Gaussian blur. Silhouette detection follows by using contour detection to find the convex hull.³

Contour Detection

A vector of two-dimensional (2D) points is used to store the contours that represent the silhouette. Suzuki and Keiichi [36]'s contour detection algorithm is used for this purpose. To reduce memory overhead, contours that form straight lines are pruned, and only the endpoints are stored.

³ Extreme outer points on each side of the image.

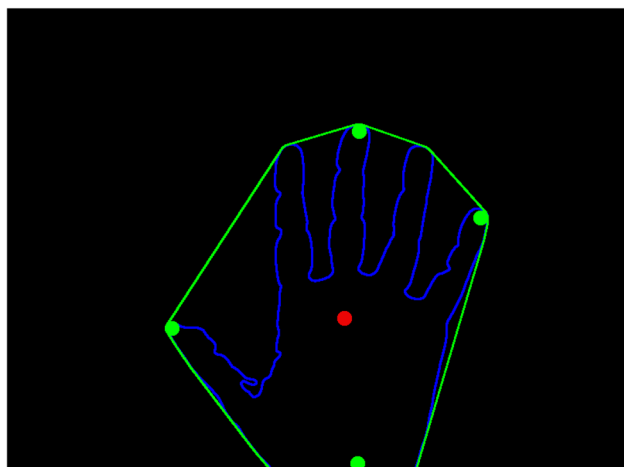


Fig. 4 Convex hull on a wide captured image after Otsu

Furthermore, the convex hull is computed on this 2D vector of extremal points.

An approximation of the palm centre is assumed to be the convex hull's centre. This allows segmentation regardless of the hand position in the image. The background is segmented again using the filled convex hull as a mask, as illustrated in Fig. 2. This is used as the initial search space for initial MIC detection to reduce the computational time. Figures 3 and 4 shows how this can be useful on a different dataset (IITD)—particularly, the more off-centre they are captured.

Shifting the MIC Effectively

The localization of finger valleys uses the nearest matching contour's y -coordinate with the initial MIC centre's y -coordinate for initialization. The 2D vector of contour points is traversed, and in doing so, the contours that form a u-shape are found. These arcs have a maximum distance of $\frac{1}{8}$ of the convex hull's width and include a ± 45 orientation tolerance in case of extreme data acquisition cases.

The initial MIC is shifted greedily by a one-pixel increment towards the middle valley point while varying its size again until all three finger valleys are found,⁴ as shown in Fig. 5. A square-shaped ROI is finally obtained by inscribing the circle with a square shape of $r\sqrt{2}$ length per side, where r is the radius.

Feature Extraction

Lighting normalization and feature extraction are applied to the square ROI. A popular lighting normalization methods

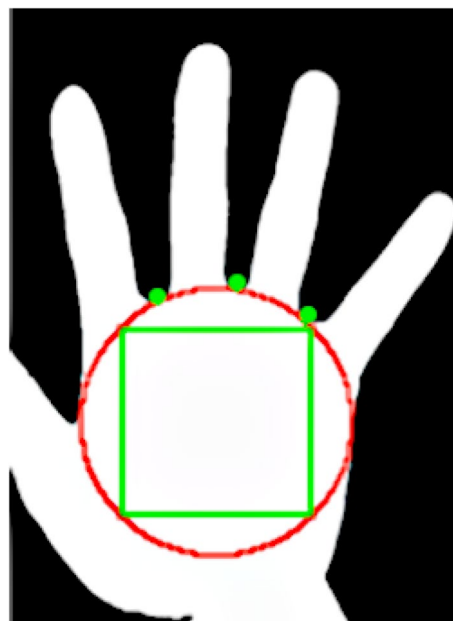


Fig. 5 Square ROI based on three finger valley point MEC [taken from [7]]

is histogram equalization (HE) algorithm [31]. However, a unique method that utilizes a modified local binary pattern variance algorithm (LBPV) [15] is used.

The original LBPV algorithm characterizes texture into a 1D LBP histogram. However, sporadic textures are captured this way due to (bilinear) interpolation. A sparsely tuned LBPV operator is used to circumvent this, which acts as a local lighting normalization algorithm. Moreover, the LBPV is further modified by subtracting the original image from the LBPV processed image. As LBP methods are not utilized this way in the literature, HE and contrast limited HE were also included in preliminary comparative tests.

Lighting normalization is used to lessen the side effects of Laplacian of Gaussian (LoG), Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) feature extraction methods. The LoG filter is less commonly used but helps remove high-frequency noise before enhancing the rest. Without lighting normalization, an unwanted side effect can exacerbate high intra-class variation in poorly aligned images [23].

The above methods were all applied as the final step before classification. Combining the modified LBPV and LoG filter reduced redundant features in image space and was especially effective on Eigen and Fisher approaches in the original study [7]. The suffix L or LBPVL is added to the classifier type to specify the feature extraction pipeline: LoG or LBPV followed by LoG, respectively.

⁴ Thumb-index valley is ignored.

Traditional Classification

Classification algorithms in image-based biometric systems aim to effectively express key extracted features and base a decision on them [9]. The dimensionality of texture features can be reduced to lower intra-class variation and potentially improve inter-class separation and generalization. However, the dimensionality of texture-based features is still generally high, and therefore, linear multithreaded implementations of classifiers are used.

Three image-based biometric classification algorithms are named: Eigen, Fisher and LBPH, and are all kNN-based, where $k = 1$. These methods stem from the classic ‘Eigen-face’ approach [4]. The fourth classifier is a linear SVM.

The Eigen, Fisher and LBPH classification algorithms and the SVM are evaluated on the final feature extracted result. Thus, these classifiers are used on the resized and postprocessed segmented texture features in all experiments for comparison, using a pipeline. Hyperparameters of all the classifiers were thus tuned.

The first three methods use the nearest neighbour distance to decide on the correct class when no threshold is applied. When a threshold is applied and the nearest neighbour distance falls above this requirement, the input image is rejected as an imposter in verification and open-set identification.

The linear SVM is used for its scalability over kernel-based versions and since it removes more data points that do not adhere to the maximum margin without requiring substantial parameter tuning [39]. Although the linear SVM might operate better for verification as it is a binary class problem, multiclass identification problems with imposter rejection are made possible through one-versus-rest using probability estimates.

This SVM uses a different measure to accept/reject a class. A logistic sigmoid is used to convert the deterministic decision weights into probability estimates using the formula of Platt [28]. Let S^+ and S^- denote the set of indices of positive (negative) polarity support vectors:

$$P(y|X) = 1 / (1 + \exp(S^+ * f(X) + S^-)) \quad (1)$$

Classification Using Deep Learning

A major limitation of the previous work was the lack of deep learning algorithms. This extension addressed this problem through careful consideration of model choice.

Convolutional Neural Networks in Biometrics

CNNs are a particularly successful deep learning algorithm on image analysis [35]. However, they are

relatively understudied when applied to palmprint biometric recognition.

Basic CNN architectures consist of three main layers. The first layer does convolution operations, where features are extracted with a sliding kernel on an image. Features of the first few blocks of layers are typically simple edges and blobs with contrast, whereas increasingly deeper layers are more abstract to humans. The convolved output is thus further processed in the second layer using a non-linear activation function that produces a feature map. The third layer performs pooling to reduce the dimensionality of neighbourhoods of the feature map with statistical information, e.g. mean, ceiling, etc.

The model architecture differs per application, but using a particular design is often motivated via practical and hardware limitations.

CNN Architecture Considerations

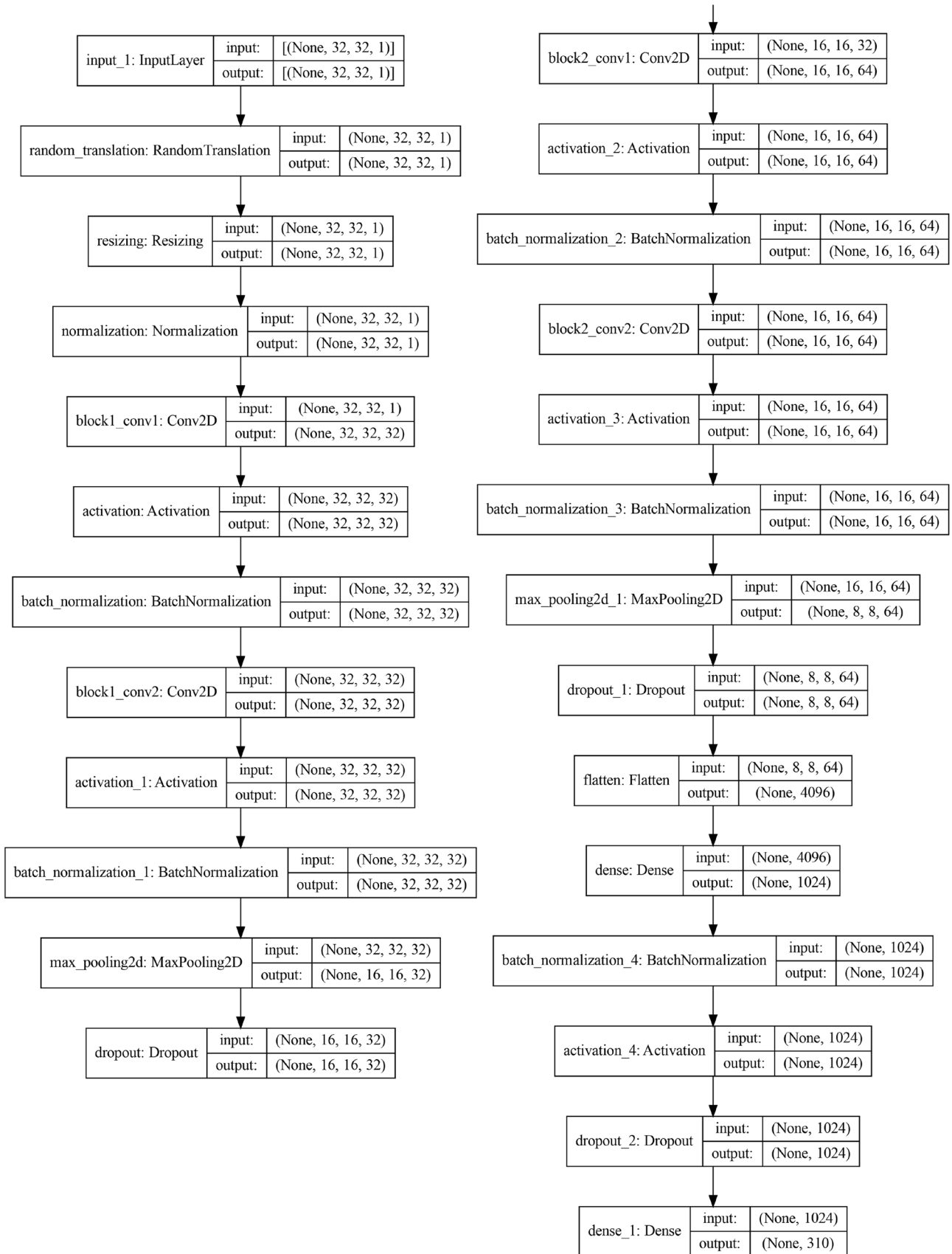
Two popular CNN architectures were considered: VGG-16 [32] and Xception [11]. However, they were mainly designed for problems such as ImageNet [20], which contains a vast number of objects within colour images. The ImageNet weights and top layers were thus discarded and trained from scratch, as modelling proved ineffective otherwise. Therefore, a custom architecture is proposed for comparative purposes using VGG-16 as a basis and utilizing Keras-Tuner to iteratively determine the optimal number of blocks of layers by a hyperparameter search.⁵ The high-level pruned structure is the first 2/5 blocks of the VGG-16, and one Fully Connected (FC) flattened layer and finally, a softmax classification layer.

Proposed CNN

Convolutional layers map inputs to output feature maps using a 2D filter. Each filter’s weights are updated during supervised learning to extract relevant discriminant features from the data [43]. The result is input to a softmax activation layer for multiclass classification. This result is compared to the known labels of validation data, and the validation loss is computed to guide how the weights are updated per epoch.

Since the proposed architecture consists of only two blocks of layers, FC and softmax classification, it is simply visualized as a block diagram (Fig. 6) with layers highlighted in bold when referring to them in text. Input size depends on the segmented resolution from the palmprint alignment system explained in section “[Overview of Palmprint Alignment System](#)”. It should be noted that Keras-Tuner is used to help

⁵ Keras-Tuner was used with for loops based on <https://keras-team.github.io/keras-tuner/tutorials/subclass-tuner/>.



◀**Fig. 6** Proposed CNN Architecture: Starting with the input layer (top left) until the first dropout layer at the end of Block 1. This continues (top right) with Block 2 until 'dense_1', the softmax layer (310 classes)

determine the architectural choices with the aim of maximum validation accuracy. Hyperparameter tuning results are provided in section “[Hyperparameter Tuning Results](#)” for all classifiers.

Image Augmentation and Other Processing

As the three evaluation datasets contain few samples per subject, it is crucial to investigate few-sample learning, as is the case in real-world biometric systems. A typical method to reduce overfitting on image data is to enlarge the training dataset [33] artificially. CNNs require many training examples so that they can extract more features at each layer. Image augmentation is a helpful image processing technique for generating additional images by applying operations, such as random translations, rotations, shear, flips, etc.

Keras uses the *ImageDataGenerator* function, which supports several operations. The proposed approach used Keras-Tuner to determine which operations are appropriate during training. The most effective operations were determined as rotation, shearing, zooming, and horizontal and vertical shifts. Furthermore, nine augmented images per original training image were sufficient, e.g. augmenting three training images results in a total of 30 training images that are used as input. The Resizing layer involves bilinear resizing the resolution and is tuned for values from 16×16 to typical ImageNet size of 224×224 , in steps of 16 (fixed aspect ratio). Over and above those augmentations, the model is also compiled with a tunable RandomTranslation layer, which provides additional variation in training data. However, since it is a randomized operation, training loss/accuracy may appear to be unstable in graphs per epoch, shown later in section “[Deep Learning Parameters](#)”.

The above operations may prove particularly useful when using only a single sample image for training and avoiding overly high learning rates. Single sample learning is also evaluated during experiments. As such, overfitting and generalization on ‘untuned’ training data is investigated further in Section [Experiments](#).

The Normalization layer refers to the application of scaling to unit variance. This was applied using the Keras Pre-processing module as the final image processing step before using the result as input for the CNN.

Convolution Layer

The input passed through a stack of convolutional (conv1/conv2) layers, where a very small receptive field⁶ 3×3 with a 1-pixel stride is used [32].

Non-linear Layer with Batch Normalization

Krizhevsky et al. [20] explain that modelling a neuron’s output f as a function of its input x is with $f(x) = \tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$. However, training time is substantially slower with those saturating than non-saturating nonlinearity $f(x) = \max(0, x)$. Neurons with the latter nonlinearity are known as Rectified Linear Units (ReLU). The vanishing gradient problem is addressed using ReLU activation function [16] and is used.

The proposed approach first applied Batch Normalization followed by ReLU in both blocks as it achieved better validation accuracy. However, in the FC layer, the order was reversed.

Pooling Layer

Although the role of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to summarize semantically similar features in the same kernel map [22]. A Max pooling 2D layer replaces $n \times n$ neighbourhoods with their highest activation result.

In the proposed approach, spatial pooling is carried out by a 2×2 max-pooling layer, with a 2-pixel stride, as the second last layer per block. Additional feature extraction is performed in the final layer per block using Dropout.

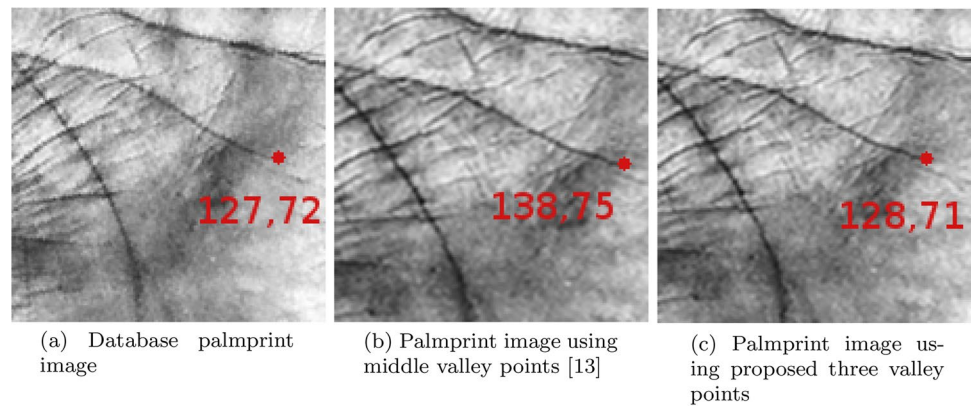
Dropout

Dropout regularisation randomly sets the output of each hidden neuron to zero based on a certain probability (typically 0.5) [34]. The neurons which are removed do not contribute to the forward pass or backpropagation.

The Dropout layer at the end of each block was initially set to 0.5, similarly to Simonyan and Zisserman [32]. However, this was found to reduce the validation accuracy. Instead, Keras-Tuner provided different optimal dropout values per block of layers.

⁶ The smallest filter that captures left/right and up/down from a centre pixel’s perspective.

Fig. 7 Improved palmprint alignment when using the modified MEC method [taken from [7]]



Fully Connected

For classification problems involving $K \geq 2$ classes, the softmax function is popular [14]. At this stage, the flattened stack of the FC layer contains 4096 channels before a dense layer with 1024 filters. As seen in Fig. 6, an additional Dropout layer is added. This made an insignificant difference during parameter tuning⁷ and can thus be discarded. The result is used to backpropagate the parameters for training the network using the ADAM optimizer [19]. The softmax function is used for multiclass classification with a channel per class.

Experiments

The palmprint segmentation methodology is first evaluated on the train and validation set via visual inspection followed by empirical means for parameter tuning. Open-set identification and verification experiments are subsequently conducted on the unseen test of three datasets. Typically, the top performer is shown while others are mentioned when interesting for conciseness. All tests were run on a AMD Ryzen 3950x with Nvidia RTX 2080 Ti. Python: OpenCV and Tensorflow (including Keras API and CUDA) were used to implement the systems. While verification and closed-set metrics are well-known, open-set identification metrics are further explained.

Palmprint Identification Metrics

Open-set identification, a “watchlist” task, differs from closed-set identification which lacks impostors or assumes no attacks. The accuracy measures for open-set identification can be summarized as follows [17, 29, 38]:

1. Detection and identification rate (DIR)—percentage of correctly predicted class images out of the total class images.
2. FNIR—or miss rate ($1 - \text{recall}$), is the percentage of incorrectly predicted class images out of the total class images.
3. False-positive identification-error rate (FPIR)—or false alarm rate $1 - \text{precision}$, is the percentage of non-class images that are incorrectly detected out of the total non-class images.

The open-set identification results are visualized in special ROC curves that have DIR on the y-axis and FPIR on the x-axis. Of note, the DIR on the y-axis takes into account both positive identification and impostor detection. FNR and FNIR are hence used interchangeably. This also allows reporting of EERs for open-set identification—the sum of the rates at which classes are misidentified as the wrong class and impostors. The accuracy score is shown in tables without the F1-score as the datasets have balanced classes.

Alignment Validation

Figure 7a provides a sample image for visual inspection by illustrating the middle principal line ending’s coordinate. Figure 7c shows that affine transformation warping enables the proposed segmentation approach to qualitatively attain better results than Ding and Ruan [13]’s approach in Fig. 7b. The improved segmentation consistency makes it robust to images containing hand pose variations, especially those obtained from contactless sensors.

Hyperparameter Tuning Results

Hyperparameter tuning proceeded using Random Search with 100 trials on the 5-fold cross-validation (CV) CASIA-Palmprint right hand dataset using a parallel for loop for deep learning models. This was also performed in previous work on traditional machine learning classifiers using the

⁷ Very low probabilities ≤ 0.04 worked best.

Table 1 Best parameters for 5-fold CV on CASIA-Palmprint right hand

Classifier	Resize	Parameter	Accuracy
EigenLBPVL	64 × 64	Comp = 200	0.972 ± 0.05
EigenLBPVL	32 × 32	Comp = 150	0.972 ± 0.06
EigenLBPVL	64 × 64	Comp = 250	0.969 ± 0.03
FisherLBPVL	64 × 64	Comp = 250	0.922 ± 0.09
FisherLBPVL	64 × 64	Comp = 200	0.918 ± 0.07
FisherLBPVL	64 × 64	Comp = 150	0.888 ± 0.05
LBPHL	160 × 160	Radius = 4	0.936 ± 0.07
LBPHL	160 × 160	Radius = 6	0.927 ± 0.08
LBPHL	128 × 128	Radius = 8	0.767 ± 0.11
SVMLBPVL	32 × 32	C = 10 ⁴	0.981 ± 0.03
SVMLBPVL	32 × 32	C = 10 ³	0.977 ± 0.05
SVMLBPVL	32 × 32	C = 10 ²	0.973 ± 0.02

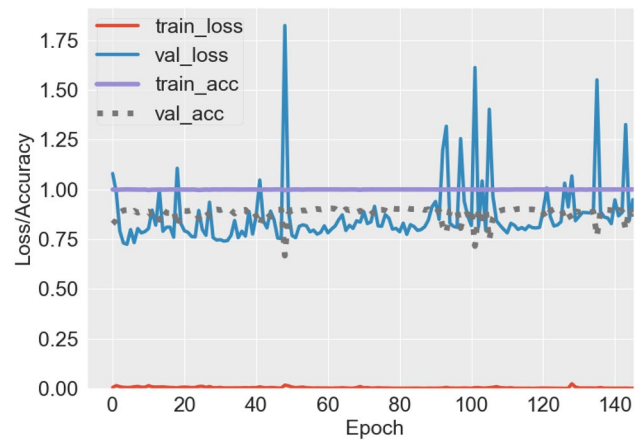


Fig. 9 Augmentation: training loss on one sample

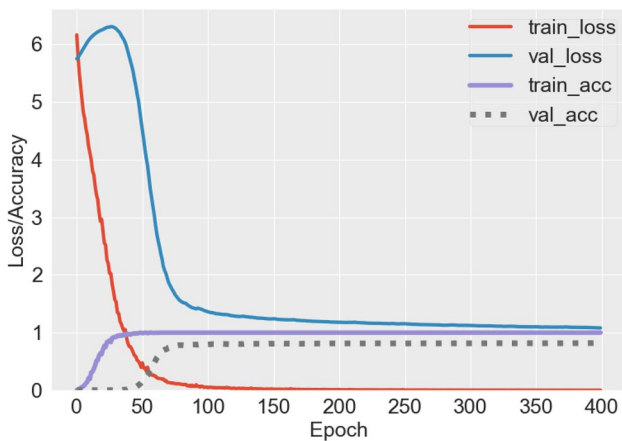


Fig. 8 No Augmentation: training loss on one sample

built-in Scikit function, and results are included in the next subsection for completeness.

Of note, since the original ImageNet weights of VGG-16 and Xception were found to be ill-effective during preliminary testing, their hyperparameters were also tuned but with parameters very similar to those used in the original studies. Logarithmic base-two parameter stepping was used for VGG-16 and Xception to save time, while 8-step increments were used for the proposed CNN, although within narrower ranges.

Traditional Machine Learning Parameters

LoG parameters were determined empirically from kernel size 3 × 3 to 19 × 19, in steps of three. Large Gaussian but small Laplacian kernels (17 × 7) yielded the best accuracies

Table 2 Best CNN parameters for 10 repeats on CASIA-Palmprint right hand

Classifier	Trans.	Resize	LR	Neurons (B1, B2)	Dropout (per block)	Accuracy
VGG-16	0.32	224 × 224	10 ⁻⁴	32, 64, 128, 256, 256	0.2, 0.2, 0.4, 0.4, 0.2	0.859 ± 0.09
VGG-16	0.1	128 × 128	50 ⁻⁴	32, 64, 128, 128, 128	0.1, 0.1, 0.2, 0.2, 0.1	0.839 ± 0.12
VGG-16	0.1	128 × 128	10 ⁻³	64, 128, 128, 256, 256	0.16, 0.16, 0.32, 0.32, 0.1	0.839 ± 0.14
Xception	0.2	224 × 224	10 ⁻⁴	64, 128, 256, 512, 728	0.32	0.799 ± 0.13
Xception	0.32	224 × 224	10 ⁻⁴	32, 64, 128, 256, 512	0.32	0.799 ± 0.11
Xception	0.04	64 × 64	10 ⁻³	16, 32, 64, 128, 256	0.24	0.797 ± 0.09
Proposed CNN	0.2	32 × 32	10 ⁻⁴	32, 64	0.1, 0.2, 0	0.929 ± 0.15
Proposed CNN	0.1	16 × 16	50 ⁻⁴	24, 72	0.1, 0.32, 0.04	0.929 ± 0.11
Proposed CNN	0.04	32 × 32	10 ⁻⁴	24, 64	0.1, 0.2, 0.04	0.927 ± 0.03

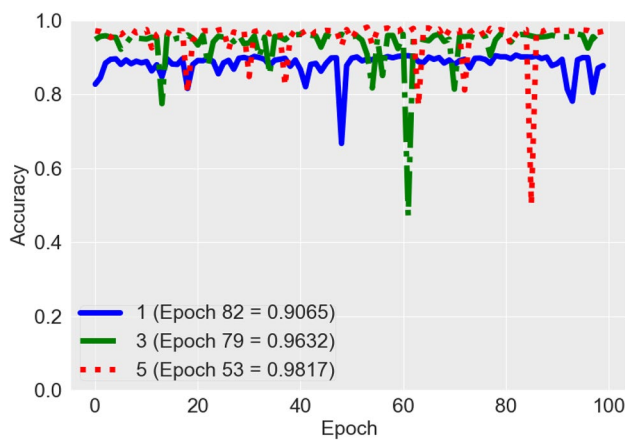


Fig. 10 Early Stopping: validation accuracy for augmented 1, 3, 5 training samples at the ‘best’ epoch

within that scope [6]. This was similar to well-tuned Gabor filters but with substantially less computation overhead.

PCA components were varied with 80% explained variance in steps of ‘5%’ increments until 99%, and yielded best results at 100–250 principal components (simply rounded to 50).

LBPV was tuned in steps of 2-pixels. The top-performing radius and neighbourhood size of (4, 31) were optimal. LBPVL performed significantly better than other lighting normalization⁸ methods, and thus other methods are not in the top three results. The LoG filter did not alter LBPH’s accuracy. Moreover, using 4-pixel radius with six neighbours was optimal.

The above results were consistently achieved and were part of a bigger processing pipeline of parameters that were searched, including bilinear-interpolated resize,⁹ data scaling,¹⁰ and lighting and classifier-feature extractor combination. The top three results per classifier are shown in Table 1. When applying PCA and LDA to SVM, the first 200 components were optimal for Eigen and Fisher. The linear SVM at $C = 10^4$ outperformed all kernel methods due to training time and model convergence issues of the latter. However, LoG was preferred over PCA/LDA.¹¹ A lower $C = 10^2$ was used to reduce overfitting.

Using one fixed training sample instead of cross-validation, LBPH achieved the best closed-set identification result by a significant margin. Eigen and SVM scaled better with more data, while Fisher yielded the worst results, with more

⁸ Pixel Norm, HE and CLAHE variants.

⁹ 16×16 to 224×224 , in steps of 16.

¹⁰ Standard scaler—unit variance normalization, MinMax—0 to 1.

¹¹ This may only be the case when using LBPVL and requires further research.

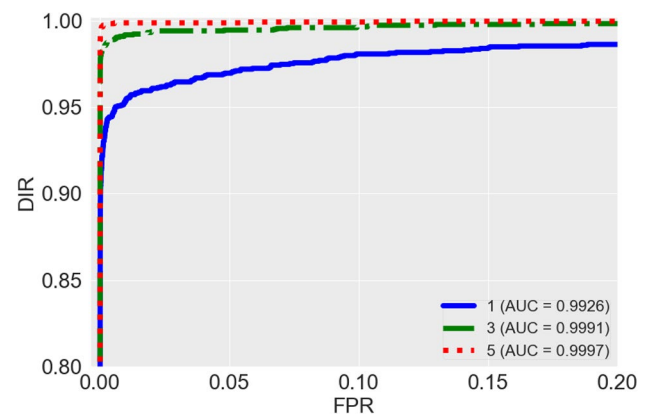


Fig. 11 Inference on the left palmprint identification using augmented 1, 3, 5 training samples

data, presumably due to some intra-class variance from the varied/low control hand poses.

Deep Learning Parameters

Since 5-fold CV resulted in perfect accuracy, a fixed split of one training sample was used and the rest (7) for validation. This was averaged over ten repeats to account for stochastic behaviour. A batch size of 32 was sufficient in all experiments. Parameters tuned include random translation (Trans.), resize, LR, number of neurons and dropout rate per block. Early stopping [8] was applied during the tuning process to reduce overfitting and speed up training.

First, the proposed architecture peaked in validation accuracy well before the 400 epoch set limit.¹² Early stopping was thus used to improve generalization on new data and other datasets. Furthermore, the datasets have a low number of images that are not enough for training the various weights within the CNN without an aggressive learning rate. Overfitting problems are, therefore, expected. Figures 8 and 9 illustrate non-augmented and augmented validation results, respectively.

The results on the best parameters of the proposed CNN in Table 2 is shown in Fig. 10. One training sample per class was enough to achieve peak accuracy at the 82nd epoch. Similarly, 79 epochs were required to reach peak accuracy when using three training samples. The significant fluctuations may be attributed to the stochastic behaviour of the augmented model, i.e. due to adjusting the random translation factor during parameter tuning. Early stopping may not always yield the ‘best’ epoch on validation data, primarily since a subjective value of 10 epochs was used for the maximum allowed ‘bad’ epochs before stopping training. Of note, the validation accuracy improvement when increasing

¹² Worked out based on time limitations.

Table 3 Comparative performance for palmprint verification on the CASIA-Palmprint dataset

Approach	EER (%)
Proposed CNN	0.5
VGG-16	1.1
Xception	1.0
SVMLBPVL	1.1
Badrinath and Gupta [2]	1.2

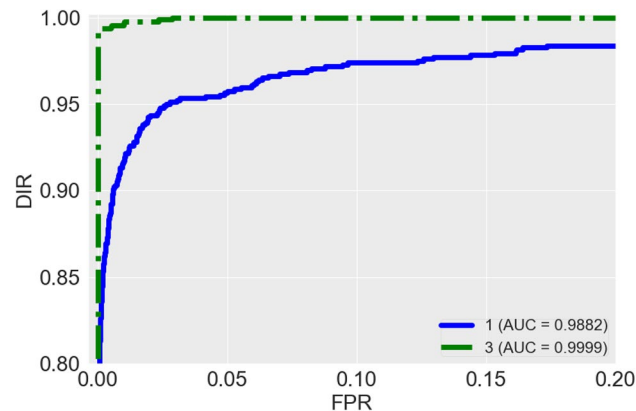


Fig. 12 Inference on the left palmprint identification using augmented 1, 3 training samples

the number of training samples from one to three is highly significant, and it will be interesting to see how well it generalizes on new data during inference. Five training samples required the least epochs before early stopping.

It was noted (not shown in the figure) that the validation accuracy of both VGG-16 and Xception continued to (slowly) improve at the set limit. Training times during tuning were infeasible to pursue additional epochs. No augmentation vs augmentation results on IITD and PolyU produced the same trends.

Three open-set identification experiments on different palmprint datasets were conducted. The classification algorithms used the best parameters identified earlier in this section on new data—the left hand’s palmprint as class data and the right-hand palmprint as impostor data.

CASIA-Palmprint Open-Set Identification

The CASIA-Palmprint database contains 5502 palmprints of left and right-hand images, collected from 312 subjects. Figure 11 shows the zoomed-in ROC curve on the unseen data (left hand). While one training sample is noticeably lower,

Table 4 Leave-one-out error rates for palmprint verification on the IITD-Palmprint dataset

Approach	EER (%)
Proposed CNN	0
VGG-16	0.16
Xception	0.16
LBPHL	0.23
Morales et al. [25]	0.21
Kumar [21]	1.31

Table 5 Inference on the left palmprint identification using augmented 1 training samples

Approach	DIR at FPR=FNR=0
Proposed CNN	0.99
VGG-16	0.98
Xception	0.98
SVM (not augmented)	0.96

the AUC values illustrate that the proposed CNN performs exceptionally well with only one training sample. This is attributed to data augmentation as, without it, accuracy was reduced from 89.4%, 95.3%, 97.1% to 79.1%, 87.3%, 96.0% for one, three and five training samples, respectively, using the proposed CNN. VGG-16 and Xception were significantly outperformed by the proposed CNN on this dataset—12% and 11%, on average. Augmentation similarly affected VGG-16 and Xception, albeit by an insignificantly greater deficit. The corresponding results on the (outperformed) traditional classifiers obtained in previous work can be found in “Appendix”.

The palmprint verification results of Badrinath and Gupta [2]’s system compared with the proposed approaches show its versatility, as the proposed identification systems were effective on the same verification data that the study used. Both the left and right palmprints were evaluated for 624 classes in total. While Table 3 shows that the proposed SVM and Badrinath and Gupta’s system both achieve good EERs of 1.1% and 1.2%, respectively, CNN approaches are close to par, and the proposed CNN approach performs even better. Moreover, Dian and Dongmei [12]’s system achieved an EER of 0.0803%. However, they used an unknown sampling strategy and hand-picked 225 classes, making a direct comparison impossible. Svoboda et al. [37] performed two-fold cross-validation and achieved a top EER of 1.86%, which outperformed VGG-16 (3.35%) and Xception (3.2%), but the proposed CNN achieved 1.67%.

IITD-Palmpoint Open-Set Identification

The IITD-Palmpoint dataset contains 2601 images captured from the left and right hands of 230 subjects. It was captured using a touchless hand sensor with low control and is known to be challenging.

Figure 12 shows that near-perfect accuracy is achieved using three training samples. Note that differences appear larger than they are due to zooming. AUC values illustrate that the proposed CNN performs exceptionally well with only one training sample. Without augmentation, accuracy reduced by about 2% for one or three training samples, regardless of the deep learning classifier.

Morales et al. [25] outperformed the IITD dataset authors [21] when using a single test sample—*n*-fold cross-validation, per 235 subjects. It is expected that more training data allows excellent EERs. However, since three training samples in the previous experiment yielded a near-perfect score in identification, it is unsurprising yet encouraging that the proposed CNN achieves zero EER in a verification problem. The proposed LBPH performs similarly with the non-texture-based system of Morales et al., as shown in Table 4. Eigen and SVM also achieved below 1% EER (not shown). Moreover, the proposed CNN also outperformed Dian and Dongmei [12]’s system, which achieved an EER of 0.1113% with an unknown sampling strategy. Svoboda et al. [37] performed two-fold cross-validation and achieved an EER of 1.64%, which outperformed VGG-16 (1.95%) and Xception (1.9%), but the proposed CNN achieved 1.2%.

PolyU Open-Set Identification

The PolyU database contains a total of 7752 greyscale left and right palmprint images from 193 individuals. Deep learning classification results on PolyU were impressive, as seen in Table 5. One training sample yielded 0.99, 0.97 and 0.97 accuracy rates on CNN, VGG-16 and Xception, respectively, while the linear SVM yielded 0.96. VGG-16 and Xception yielded near-perfect accuracy when using three and five training samples, but the proposed CNN achieved a perfect score. However, the same was not the case for the traditional classifiers. This comparison was arguably not fair as the deep learning classifiers achieved those high scores with augmentation, not made available to the traditional classifiers at the time of testing. Without augmentation, one training sample yielded only 0.88, 0.95 and 0.96 accuracy rates on the proposed CNN. In other words, augmentations enabled up to 11% improvement on this dataset.

Zhang et al. [42]’s palmprint verification performance was compared to the proposed systems on 250 classes of the PolyU palmprint dataset. The first sample is used for training as the data split was unspecified in the study. Six samples from session one were used in the test.

Zhang et al.’s system achieved an excellent EER of 0.0257%. However, all proposed approaches achieved zero error on this relatively trivial verification problem. This was expected as a very high identification accuracy was achieved on this dataset. Minaee and Wang [24]’s system also achieved a perfect score but only when six training samples were used. Moreover, their system achieved 99.84% when using two training samples, i.e. underperforming compared with Zhang et al. and all the proposed systems, highlighting the importance of accurate palmprint segmentation. Finally, all the proposed approaches also outperformed Dian and Dongmei [12]’s system, which achieved an EER of 0.0443% using an unknown sampling strategy.

Discussion of Palmprint Recognition

The traditional classification approaches, including the *k*NN-based Eigen, Fisher, LBPH, and the linear SVM, were evaluated during previous work and summarized.

LBPH achieved the best DIR on a single sample training out of these traditional classifiers but with high FPR. The SVM achieved a similar DIR on a single training sample but the lowest FPR. When specifically considering challenging data, Fisher has poor discrimination against impostors but occasionally achieves the best DIR. The SVM does not benefit from additional training samples as much as Eigen, Fisher and LBPH. On the other hand, both SVM and Fisher scale well with more training data on datasets with low intra-class variations—typically controlled and unchallenging. Overall, SVM was the best traditional classifier. Kernel-based SVMs did not offer an advantage over linear on this type of data. Limitations of the traditional classifier evaluation include no augmentations applied and the omission of several other classifiers, such as random forests and logistic regression and more.

Following the previous study, the palmprint recognition system was extended to include CNN-based classification. The overall results indicate that a smaller designed CNN can achieve impressive results on challenging datasets, especially when combating overfitting in one or low-sampled learning. State-of-the-art CNN architectures VGG-16 and Xception did not perform effectively, which may be attributed to a lack of data, inappropriate application, or the type of palmprint segmentation approach. Furthermore, additional epochs may prove beneficial as the validation accuracy slowly increased to the set epoch limit. Another reason may be due to the very narrow parameter tuning range—unfortunately, time restrictions did not allow for a greater range. Despite this, they still outperformed traditional classifiers, although not always significantly.

Running inference on the palmprint datasets, in general, showed that data augmentation could provide highly

significant accuracy gains, especially when only a single training sample is available. The ‘standard’ dropout rate of 0.5 was too aggressive for the three CNNs but particularly bad for the proposed CNN. VGG-16 and Xception preferred dropout rates in the range of 0.24–0.4 per block. The proposed CNN’s structure was largely based on VGG-16, and ‘good’ dropout rates were about 0.1 for the first block and about double for the second block.¹³ Including any of the remaining three blocks of the original VGG-16 resulted in an accuracy drop of 2–9% on CASIA-Palmprint (similarly on others). The rest of the architectural choices made when constructing the proposed CNN were in line with VGG-16, as dropout on the FC layer was found to be unnecessary, and Max Pooling was effective.¹⁴

Concluding Remarks

A deep learning palmprint recognition system was constructed based on a previous robust palmprint segmentation algorithm and various techniques that tailored the model for palmprint data. Although a lesser studied area, this was implemented to show the supervisor discrimination ability of CNN classifiers when tuned properly.

The robust segmentation process first removed background noise. Two additional finger-valleys adjacent to the middle finger were validated as more precise than other MEC and keypoint approaches. The result was a palmprint

segmentation algorithm that can work on both contactless and contact sensors—allowing for palmprint acquisition in unconstrained conditions.

The traditional and deep learning classifier parameters were tuned. A CNN architecture was proposed based on VGG-16 but was tuned to be effective on greyscale palmprints. The traditional classifiers were generally not as effective as the proposed CNN but achieved similar recognition performance as VGG-16 and Xception. This pends further investigation since the difference in computational expense is substantial. On the other hand, the proposed CNN proved extremely effective and outperformed the related studies and all the other proposed approaches.

Open-set identification accuracy was the focus, because the original palmprint system was constructed for that purpose. However, with the positive verification results, future work may include a detailed comparison of verification datasets with more classes and additional tuning trials and parameters. The applicability of the proposed CNN on other image-based biometrics may also be a promising investigation.

Appendix

CASIA-Palmprint Traditional Results

See Figs. 13, 14.

¹³ The second block preferring double the dropout rate was a consistent trend.

¹⁴ Average Pooling was not effective.

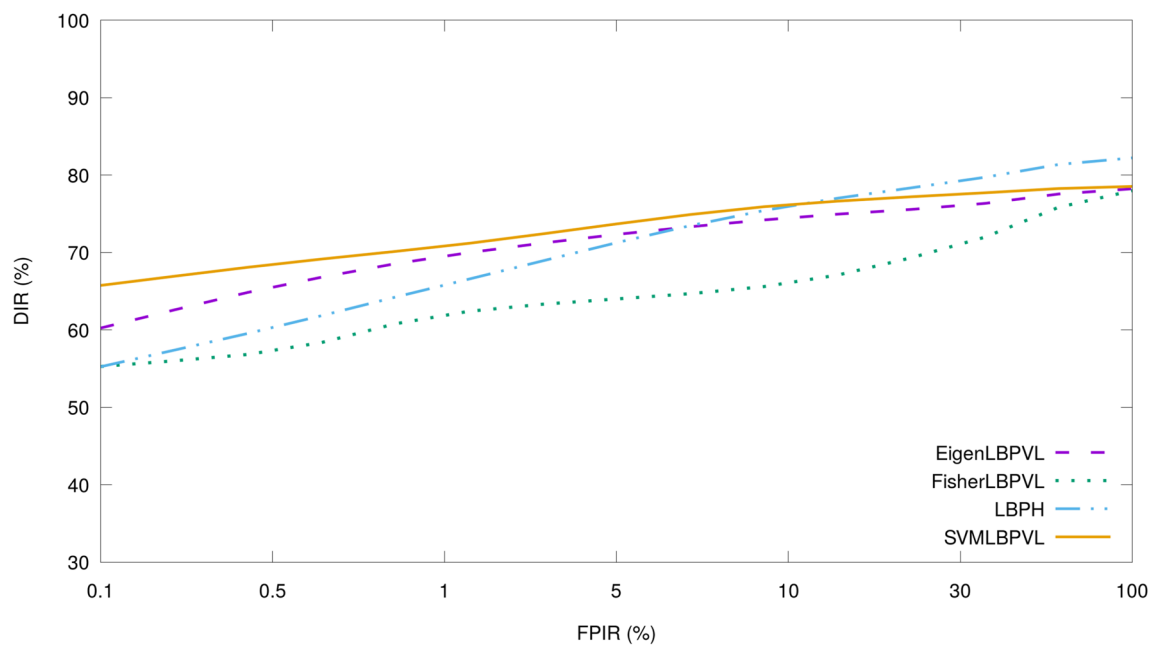


Fig. 13 One training sample: the optimal resolution was 32×32 to 64×64 , except LBPH which preferred the segmented resolution (about 200×200). LBPH yield better DIR, but at poor FPIR values compared with the rest [taken from [5]]

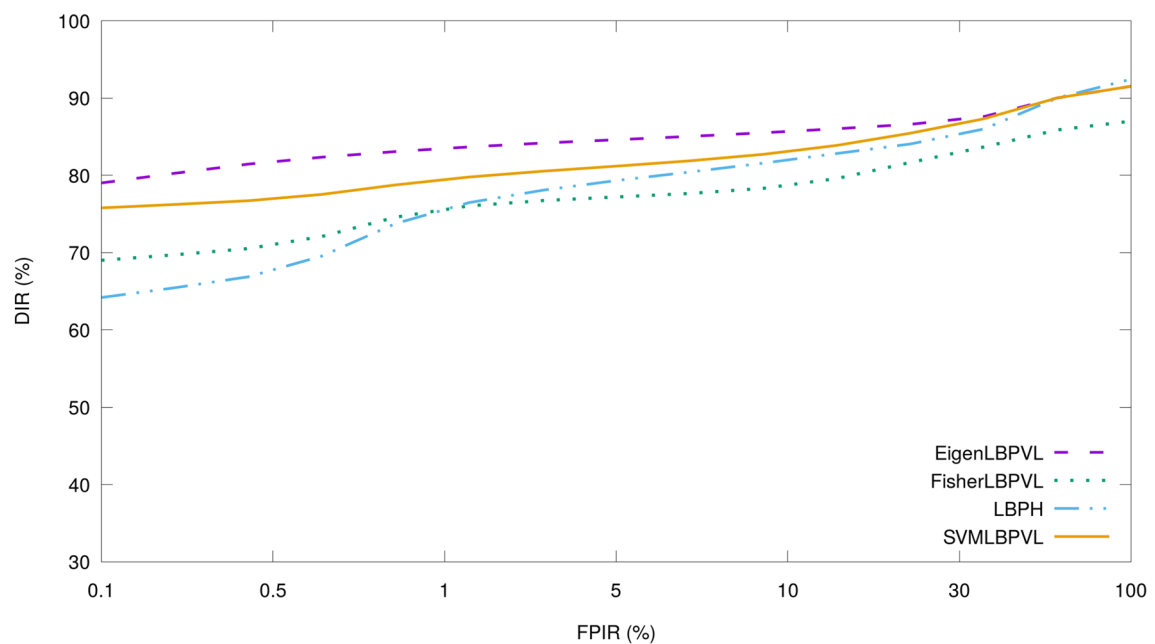


Fig. 14 Three training samples: significantly highest AUC by Eigen when using three training samples. Eigen seems to discriminate between real classes and impostors the best. SVM yields higher AUC

than LBPH and an identical maximum DIR as Eigen. Fisher is the worst at discriminating against impostors

IITD-Palmprint Traditional Results

See Figs. 15, 16.

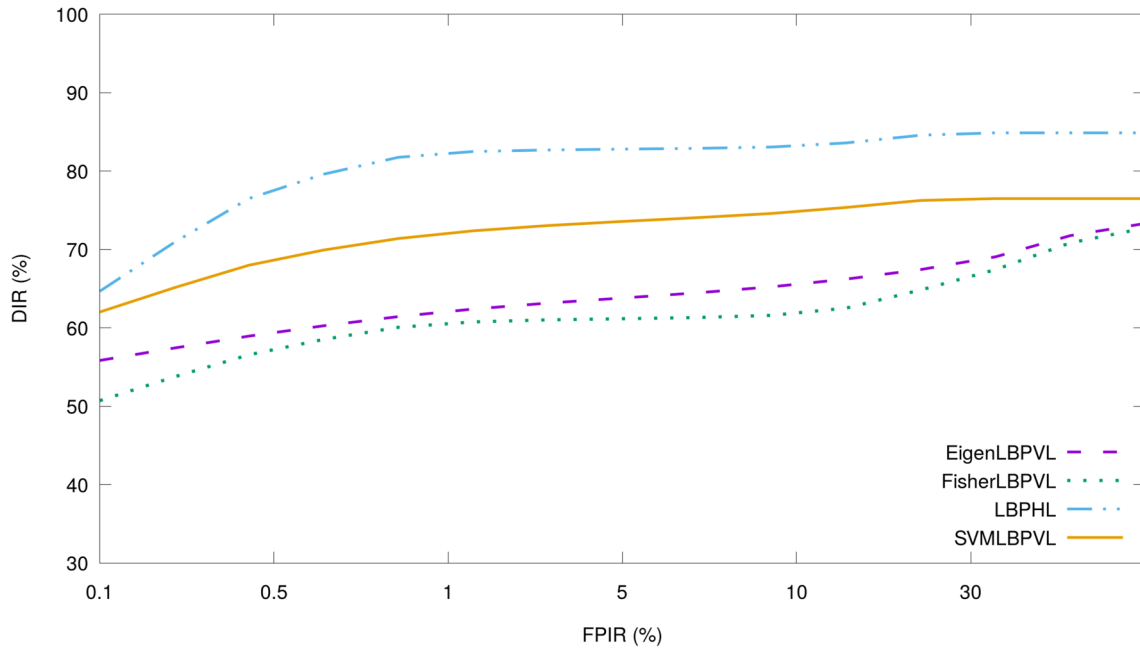


Fig. 15 One training sample: the segmentation resolution for this dataset is approximately 160×160 , preferred by LBPH, while the optimal resolution remains low for the other classifiers. LBPH significantly outperforms the other proposed methods for one training sample

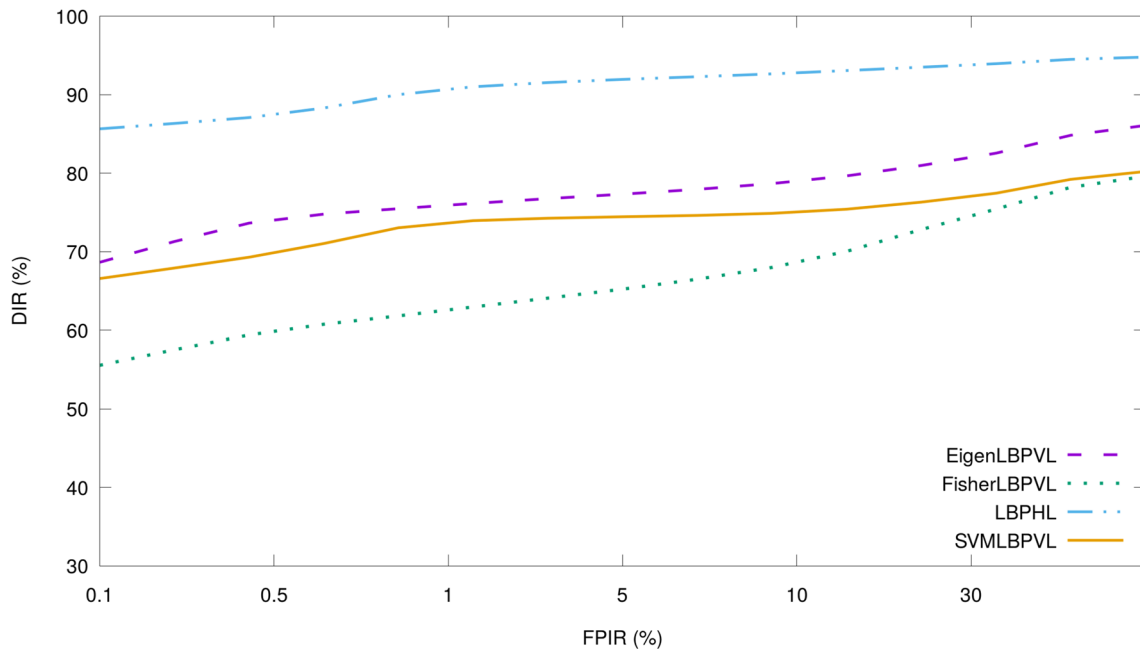


Fig. 16 Three training samples: Eigen and LBPH improve by 10% and 13%, respectively, with the additional two training samples. Moreover, LBPH yields better DIR values than Eigen by a 9% followed by SVM, and Fisher continues its poor run

PolyU Palmprint Traditional Results

See Fig. 17.

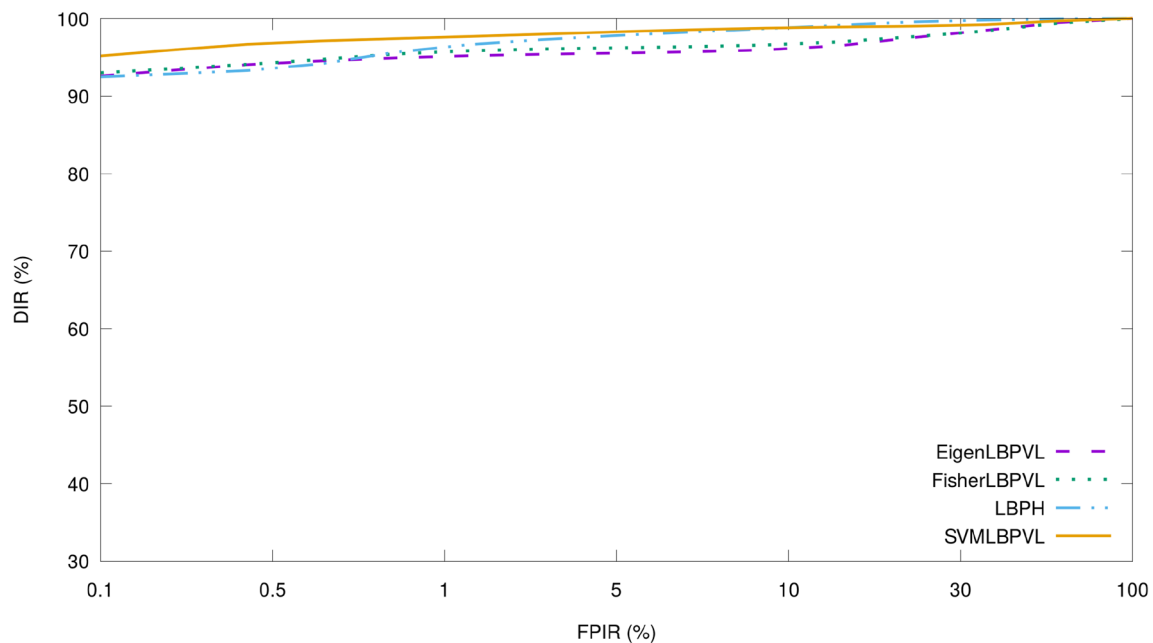


Fig. 17 One training sample: a 130×130 segmentation resolution and 50×50 optimal resolution. A 10% DIR reduction when using 100×100 resolution. Classifiers perform similarly on this dataset as samples are unchallenging—highly constrained. DIR values at 30%

FPIR for Eigen, Fisher, LBPH and SVM are 98.10%, 98.10%, 99.70% and 99.10%, respectively, at zero FNIR (miss rate). Results were achieved using only a single training sample

Since all classifiers achieved a 100% DIR above 30% FPIR, DIR results at 0% FPIR and 0% FNIR were evaluated. SVM achieved the best results (98.70%), and it is again apparent that LBPH(95.20%) does not scale as well as the other classifiers when given more training samples. Fisher (97.70%) achieved slightly better accuracy than Eigen (97.00%), and this is attributed to less stretch and distortion in this dataset, hence lower intra-class variation.

Funding This study was funded by National Research Foundation (120654). This work was undertaken in the Distributed Multimedia CoE at Rhodes University.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

The authors thank the authors for permission to use PolyU, CASIA and IITD Palmprint datasets.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Amraoui A, Fakhri Y, Kerroum MA. Multispectral palmprint recognition based on fusion of local features. In: 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), IEEE; 2018. pp. 1–6.
2. Badrinath G, Gupta P. Stockwell transform based palm-print recognition. *Appl Soft Comput*. 2011;11(7):4267–81.

3. Bao X, Guo Z. Extracting region of interest for palmprint by convolutional neural networks. In: 2016 Sixth international conference on image processing theory, tools and applications (IPTA), IEEE; 2016. pp. 1–6.
4. Belhumeur PN, Hespanha JP, Kriegman D. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell.* 1997;19(7):711–20.
5. Brown D, Bradshaw K. An investigation of face and fingerprint feature-fusion guidelines. In: Beyond databases, architectures and structures (BDAS 2016), communications in computer and information science, vol 613, Springer, Cham; 2016. pp. 585–99. https://doi.org/10.1007/978-3-319-34099-9_45
6. Brown D, Bradshaw K. A multi-biometric feature-fusion framework for improved uni-modal and multi-modal human identification. In: Technologies for Homeland Security (HST), IEEE; 2016.
7. Brown D, Bradshaw K. Improved palmprint segmentation for robust identification and verification. In: 2019 15th international conference on signal-image technology & internet-based systems (SITIS), IEEE; 2019. pp. 1–7.
8. Caruana R, Lawrence S, Giles CL. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: Advances in neural information processing systems; 2001. pp. 402–408.
9. Chapelle O, Haffner P, Vapnik VN. Support vector machines for histogram-based image classification. *IEEE Trans Neural Netw.* 1999;10(5):1055–64.
10. Choge HK, Oyama T, Karungaru S, Tsuge S, Fukumi M. A circle-based region-of-interest segmentation method for palmprint recognition. In: Proceedings of the international conference on control, automation, and systems and society of instrument and control engineers joint conference (ICCAS-SICE), IEEE; 2009. pp. 4993–4997.
11. Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 1251–58.
12. Dian L, Dongmei S. Contactless palmprint recognition based on convolutional neural network. In: 2016 IEEE 13th international conference on signal processing (ICSP), IEEE; 2016. pp. 1363–67.
13. Ding B, Ruan Q. The localization of the palmprint images based on the maximal effective circle. In: Proceedings of the 8th international conference on signal processing, IEEE, vol 4; 2006.
14. Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. [arXiv:1706.04599](https://arxiv.org/abs/1706.04599). 2017.
15. Guo Z, Zhang L, Zhang D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recogn.* 2010;43(3):706–19.
16. Ide H, Kurita T. Improvement of learning for cnn with relu activation by sparse regularization. In: 2017 international joint conference on neural networks (IJCNN), IEEE; 2017. pp. 2684–91.
17. Jain AK, Li SZ. Handbook of face recognition. Berlin: Springer; 2011.
18. Jalali A, Mallipeddi R, Lee M. Deformation invariant and contactless palmprint recognition using convolutional neural network. In: Proceedings of the 3rd international conference on human-agent interaction. 2015. pp. 209–12.
19. Kingma DP, Ba J. Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980). 2014.
20. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM.* 2017;60(6):84–90.
21. Kumar A. Incorporating cohort information for reliable palmprint authentication. In: Proceedings of the sixth indian conference on computer vision, graphics & image processing (ICVGIP), IEEE; 2008. pp. 583–90.
22. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature.* 2015;521(7553):436–44.
23. Li P, Zhang T, et al. Proposal of novel histogram features for face detection. In: Proceedings of the international conference on pattern recognition and image analysis. Springer; 2005. pp. 334–43.
24. Minaee S, Wang Y. Palmprint recognition using deep scattering convolutional network. [arXiv:1603.09027](https://arxiv.org/abs/1603.09027). 2016.
25. Morales A, Ferrer MA, Kumar A. Towards contactless palmprint authentication. *IET Comput Vis.* 2011;5(6):407–16.
26. O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L, et al. Keras Tuner. 2019. <https://github.com/keras-team/keras-tuner>
27. Otsu N. A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern.* 1979;9(1):62–6.
28. Platt J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv Large Margin Classifiers.* 1999;10(3):61–74.
29. Quinn GW, Grother P, Matey J. IREX IX. Tech. rep., NIST Interagency/Internal Report (NISTIR)-7836, Image Group, Information Access Division, Information Technology Laboratory; 2016.
30. Ross AA, Nandakumar K, Jain A. Handbook of multibiometrics, vol. 6. Berlin: Springer Science & Business Media; 2006.
31. Sepasian M, Balachandran W, Mares C. Image enhancement for fingerprint minutiae-based algorithms using CLAHE, standard deviation analysis and sliding neighborhood. In: Proceedings of the world congress on engineering and computer science. 2008. pp. 22–4.
32. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556). 2014.
33. Song JM, Kim W, Park KR. Finger-vein recognition based on deep densenet using composite image. *IEEE Access.* 2019;7:66845–63.
34. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res.* 2014;15(1):1929–58.
35. Sundararajan K, Woodard DL. Deep learning for biometrics: a survey. *ACM Comput Surv (CSUR).* 2018;51(3):1–34.
36. Suzuki S, Keiichi A. Topological structural analysis of digitized binary images by border following. *Comput Vis Graph Image Process.* 1985;30(1):32–46.
37. Svoboda J, Masci J, Bronstein MM. Palmprint recognition via discriminative index learning. In: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE; 2016. pp. 4232–37.
38. Tabassi E, Watson C, Fiumara G, Salamon W, Flanagan P, Cheng SL. Performance evaluation of fingerprint open-set identification algorithms. In: Proceedings of the IEEE international joint conference on biometrics (IJCB), IEEE; 2014. pp. 1–8.
39. Tang Y. Deep learning using support vector machines. *Clin Orthopaedics Relat Res.* 2013;2013:2.
40. Zhang D. Palmprint segmentation by key point features. In: Palmprint authentication. Springer US; 2004. pp. 73–83. https://doi.org/10.1007/1-4020-8097-2_6
41. Zhang D, Kong WK, You J, Wong M. Online palmprint identification. *IEEE Trans Pattern Anal Mach Intell.* 2003;25(9):1041–50.
42. Zhang D, Guo Z, Gong Y. An online system of multispectral palmprint verification. In: Multispectral Biometrics, Springer; 2016. pp. 117–37.
43. Zhang L, Zhang L, Du B. Deep learning for remote sensing data: a technical tutorial on the state of the art. *IEEE Geosci Remote Sens Mag.* 2016;4(2):22–40.
44. Zhao D, Pan X, Luo X, Gao X. Palmprint recognition based on deep learning. In: 6th international conference on wireless, mobile and multi-media (ICWMMN 2015), IET; 2015. pp. 214–6.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.