IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 16, 2020, accepted April 24, 2020, date of publication May 4, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992219

Toward Unconstrained Palmprint Recognition on Consumer Devices: A Literature Review

ADRIAN-STEFAN UNGUREANU, SAQIB SALAHUDDIN, (Associate Member, IEEE), AND PETER CORCORAN, (Fellow, IEEE)
College of Engineering and Informatics, National University of Ireland, Galway, H91 TK33 Ireland

Corresponding author: Adrian-Stefan Ungureanu (a.ungureanu1@nuigalway.ie)

The research work presented here was funded under Industry/Academic Partnership 13/SPP/I2868 co-funded by Science Foundation Ireland (SFI) and FotoNation Ltd.

ABSTRACT As a biometric modality, palmprints have been largely under-utilized, but they offer some advantages over fingerprints and facial biometrics. Recent improvements in imaging capabilities on handheld and wearable consumer devices have re-awakened interest in the use of palmprints. The aim of this paper is to provide a comprehensive review of state-of-the-art methods for palmprint recognition including Region of Interest extraction methods, feature extraction approaches and matching algorithms along with an overview of available palmprint datasets in order to understand the latest trends and research dynamics in the palmprint recognition field.

INDEX TERMS Palmprint acquisition, feature extraction, matching, datasets, region of interest, template detection, deep learning, neural network, machine learning.

I. INTRODUCTION

The last decade has seen the migration of biometric recognition approaches onto mobile devices by using fingerprint [1], face [2] or iris [3] as an alternative to conventional authentication using PIN numbers or patterns.

Two-factor authentication, multi-modal and multi-biometrics are all considered to be viable options improving the security of a system, as they considerably increase the spoofing effort for an attacker [4].

Jain *et al.* [5] evaluate several biometric features and reach the conclusion that there is no ideal biometric. Alongside the previously mentioned features is another biometric which has not received as much attention: the palmprint. However, there are several advantages which palmprint recognition can offer regarding their deployment on consumer devices:

- The features contained in a palmprint are similar to fingerprints, but cover a much larger surface. For this reason they are generally considered to be more robust than fingerprints [5].
- Palmprints are more difficult to spoof than faces, which are public feature, or fingerprints, which leave traces on many smooth surfaces.
- There is no extra cost required for acquisition, as long as the device is fitted with a camera (optical sensor) and a flash source (LED or screen).
- It has potential for multi-biometric recognition, as it can be used with other hand-based features (fingerprints [6], finger knuckles [7], wrist [8])
- It can be seamlessly integrated into the use case of many consumer devices, such as AR/VR headsets [9], smartphones [10], gesture control systems, driver monitorin systems, etc.

A. CONTRIBUTIONS OF THIS PAPER

The aim of this paper is to provide a comprehensive review focusing on the pipeline of palmprint recognition in order to clarify the current trends and research dynamics in the palmprint recognition based biometric systems. The paper discusses in detail the available datasets of palmprint images in visible range and reviews the state-of-the-art methods for feature extraction.

B. DIFFERENCES FROM PREVIOUS WORKS

Compared to other recent reviews on palmprint recognition [11], [12] this review differs in the following way:

- the main focus is placed on visible-range datasets, as opposed to [11], [12], which also consider multi-spectral, 3D and latent datasets

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Ayoub Khan.

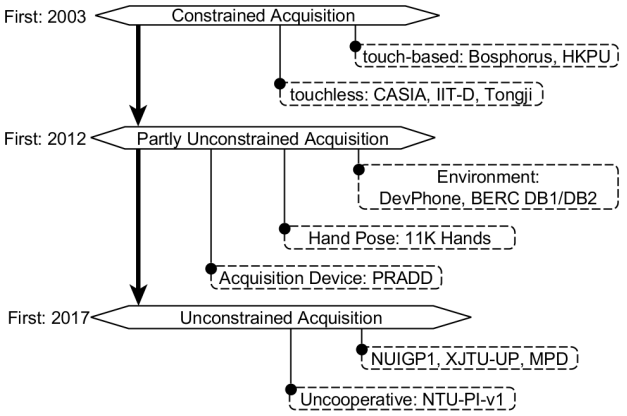


FIGURE 1. Timeline overview of palmprint datasets, based on how constrained their environment of acquisition is.

- the available palmprint datasets are categorized based on the level of unconstrainedness, and the device used for acquisition
- it does not focus on only one stage of the palmprint recognition pipeline. It includes an extensive overview of recent developments regarding datasets, Region of Interest (ROI) and feature extraction approaches. The only other review investigating ROI extraction techniques was made by Zhong *et al.* [12], which primarily focused on the conventional approaches (for constrained acquisition).

To the best of our knowledge this is the first review of palmprint recognition to focus on unconstrained acquisition and the challenges it brings.

C. ORGANIZATION OF THE PAPER

The rest of the paper is organized as follows. Section II describes existing datasets of palmprint images. Section III provides an overview of approaches developed for the palmprint ROI extraction from various palmprint datasets. Section IV presents an overview of approaches of feature extraction and matching algorithms. Section V discusses the previous sections and provides concluding remarks regarding research directions.

II. PALMPRINT DATASETS

This section presents an overview of palmprint datasets used for the recognition of palmprints in the visible spectrum (hyperspectral imaging at various wavelengths is not considered, nor 3D acquisition).

The currently available palmprint datasets can be split into three categories, based on the restrictions imposed to the user during the acquisition process (as represented in Fig. 1 and summarized in Table 1):

1) Constrained acquisition: This category includes the most popular palmprint datasets, which place the main focus on the feature extraction and matching stages, simplifying the acquisition as much as possible (for the recognition system). Images tend to display hands with a specific hand pose (fingers straight and separated) against a uniform background with no texture, usually black.
2) Partly unconstrained acquisition:
   - **Unconstrained** environment: The background is unconstrained, which corresponds to the use case of consumer devices. The hand pose is required to follow a specific protocol, generally consisting of presenting the fingers spread out in front of the sensor (preferably the center of the image).
   - **Unconstrained** hand pose: Allows the user to choose the pose of the hand during acquisition. This corresponds to the general expectations for consumer devices, which require a simplified (and intuitive) protocol of interaction.
   - **Multiple devices** used for acquisition: Matching biometric templates across several devices. Generally the other aspects of the acquisition process (hand pose and background) are constrained.
3) Unconstrained acquisition: **Unconstrained** environment and hand pose, this represents the most unconstrained scenario, where all conditions of acquisition are left to the choice of the user. A further step is closer to forensic recognition, where the acquisition is performed regardless of the user's cooperation (uncooperative).

A. CONSTRAINED PALMPRINT DATASETS

The Hong Kong Polytechnic University Palmprint dataset (HKPU) [13] was the first to provide a large-scale constrained palmprint dataset to compare recognition performance. The images were acquired using a scanner (**A1** in Table 1) having a cropped guide around the palm, reducing the impact of fingers' position. A similar approach for acquiring palmprints but including the entire hand can be found in the Bosphorus Hand dataset [14]. The earliest touch-less palmprint datasets (**A2** in Table 1) were the ones released by the Chinese Academy of Sciences (CASIA) [15] and by the Indian Institute of Technology in Dehli (IIT-D) [16]. Both used a digital camera for acquisition in an environment with uniform lighting. The main differences are the scale and color information contained in IIT-D. The hand images in CASIA are gray scale and have cropped fingers. The College of Engineering Pune (COEP) [17] released a touch-less dataset of palmprints, but the acquisition relied on pegs to direct the position of fingers relative to the camera. Another touch-less dataset was released by Las Palmas de Gran Canaria University under the name GPDS [18]. They used two webcams to acquire palmprint images in two sessions. One of the webcams was adapted to acquire NIR images by removing its IR filter and replacing it with an RGB filter. The dataset is split into images acquired in visible range (GPDS-CL1) and in NIR range (GPDS-CL2). HKPU Contact-free (HKPU-CF) [19] was released in 2011. Although it was acquired with a 3D scanner, it also provides 2D RGB images of hands

TABLE 1. Constrained palmprint datasets: (A1) touch-based and (A2) touch-less acquisition, having one hand pose (spread fingers). Partly unconstrained palmprint datasets: (B1) unconstrained environment/background, (B2) multiple devices used during acquisition and (B3) unconstrained hand pose. Unconstrained palmprint datasets (C1), as close as possible to the realistic deployment of a palmprint recognition system on smartphones (or similar consumer devices) and (C2) reflecting recognition in an uncooperative environment, closer to forensic recognition.

	Year	Name	Acq. Device	Hands	Images	Description
A1	2003	**HKPU** [13]	scanner	386	7,752	Cropped hand images, intended for performance comparison. Two sessions. **Graylevel** images with black background.
	2010	**Bosphorus** [14]	scanner	1,560	4,846	**RGB** images of the entire hand. Hand geometry also used.
A2	2005	**CASIA** [15]	digital camera	624	5,502	**Graylevel** images, black background. Uniform lighting. Cropped fingers.
	2006	**IIT-D v1** [16]	digital camera	470	3,290	**RGB** images, black background. Uniform lighting. Cropped wrist.
	2010	**COEP** [17]	digital camera	168	1,344	**RGB** images, black background. Pegs used for pose/scale control.
	2011	**GPDS-CL1** [18]	2 webcams	100	2,000	**RGB** images, acquired in both visible and 850 nm. Hand geometry also used.
	2011	**HKPU-CF** [19]	3D hand scanner	354	3,540	RGB and 3D images acquired at the same time. Black background, uniform lighting.
	2017	**Tongji** [20]	digital camera	1,200	12,000	**RGB** images, black background. Large-scale dataset.
	2018	**PolyU-IITD v3** [21]	2 digital cameras	700	-	**RGB** images, black background. Contains images from 2 ethnicities. Significant physical variation considered, and long interval acquisition (15 years).
	2019	**HFUT** [22]	digital camera	800	16,000	**RGB** images, black background. Whole hands: fingers and wrist.
B1	2013	**DevPhone** [23]	1 smartphone camera	30	600	Acquisition controlled with a square guide displayed on the screen; no information about the environment of acquisition.
	2015	**BERC** [24] **DB1** **DB2**	1 smartphone camera	120	8,957 9,224	Unconstrained environment; controlled hand orientation using a visual guide on the device screen; indoor (DB1) and outdoor (DB2) background.
	2016	**Tiwari** *et al.* [25]	1 smartphone camera	62	182 videos	Users positioned the palm in a circular guide displayed on the device's screen. Each user filmed 3 short videos of the palm.
	2019	**BMPD** [26]	1 smartphone camera	41	1,640	Hand images against black background. The 2nd acquisition session used stronger angles relative to the hands.
	2019	**SMPD** [26]	1 smartphone camera	110	4,400	Hand images with black background and flashlight turned on. Included 4 scenarios of hand orientation, including tilted away from the camera.
B2	2012	**Choras** *et al.* [27]	3 smartphone cameras	84	252	**RGB** hand images with black background.
	2012	**PRADD** [28]	1 digital camera, 2 smartphones	100	12,000	Hand images against black background. Two lighting cases considered. Acquisition not performed by user.
B3	2017	**11k Hands** [29]	1 digital camera	380	11,076	RGB images of hands (palm and dorsal) against white background. Variable hand pose.
	2019	**NTU-CP-v1** [30]	1 digital cameras	655	2,478	Unconstrained pose heterogeneous lighting, white background .
	2020	**NUIGP2** [31]	1 webcam	52	24,631	**RGB** images, with variable hand pose and scales, several backgrounds. Intended for training of palmprint ROI extraction algorithms.
C1	2017	**NUIGP1** [10]	5 smartphone cameras	81	1,816	Two lighting conditions, two backgrounds.
	2019	**XJTU-UP** [32]	5 smartphone cameras	200	30,000+	Unconstrained background, scenario with enabled flashlight included.
	2019	**MPD** [33]	2 smartphone cameras	200	16,000	Images include rotation but appear to have a single hand pose (spread fingers), to facilitate key-point detection.
C2	2019	**NTU-PI-v1** [30]	many devices	2,035	7,781	Collected from the Internet. Large variation in scale, pose and background.

Acq. Device = Acquisition Device; Hands = Number of hand classes (some datasets only have images from one hand per participant)
PP = Palmprint; HG = Hand Geometry

acquired in a constrained environment, with black background and uniform lighting. In 2017, Zhang *et al.* [20] released a large-scale dataset (12,000 images) of palmprints acquired with a dedicated device containing a digital camera (Tongji). The acquisition environment was dark with a controlled light source illuminating the palm area.

Recently, Kumar [21] released a large-scale dataset of palmprints entitled PolyU-IITD Contactless Palmprint Database v3, introducing a variety of challenges. Firstly, it contains hand images from two ethnicities (Chinese and Indian). Secondly, the palmprints were acquired from both rural and urban areas. The physical appearance of the hands varies significantly, there being instances of birth defects, cuts and bruises, callouses from manual labour, ink stains and writing, jewelry and henna designs. The dataset also contains a 2nd acquisition session after 15 years, for 35 subjects.

B. PARTLY UNCONSTRAINED PALMPRINT DATASETS

Moving away from constrained scenarios, several datasets introduced at least one challenging factor in the context of palmprint recognition systems.

Considering an unconstrained environment for acquisition (**B1** in Table 1) leads to both variable background and lighting conditions. An initial step was made for palmprint matching in the context of smartphones by Aoyama *et al.* [23] in 2013 with a small dataset of images (called DevPhone). Unfortunately, the conditions of acquisition are not clear (how many backgrounds considered, if flashlight was enabled), besides the fact that users were required to use a square guide to align the palm with the center of the acquired image. A much larger dataset was acquired by Kim *et al.* [24] both in-doors and out-doors (BERC DB1 and DB2). Both DB1 and DB2 included a scenario where the smartphone's flashlight was enabled. As in the case of DevPhone, the images in BERC DB1/DB2 contained hands with specific hand pose (open palm with spread fingers.

A different approach to acquisition was provided by Tiwari *et al.* [25] who recorded videos of palmprints with a smartphone, with the video centered on the user's palmprint.

Recently, Izadpanahkakhk *et al.* [26] introduced two palmprint datasets acquired with a smartphone camera - Birjand University Mobile Palmprint Database (BMPD) and Sapienza University Mobnile Palmprint Database (SMPD). The variation considered for investigation was the rotation of the hands (in both datasets), both in-plane and out-of-plane rotation.

The first dataset of palmprints acquired with multiple devices (**B2** in Table 1), albeit of reduced size, was developed by Choras and Kozik [27] using three smartphones.

Jia *et al.* [28] developed a large dataset of images entitled Palmprint Recognition Across Different Devices (PRADD) using two smartphones and one digital camera. The background used was a black cloth. The hand's posture was restricted. From the images provided in [28], it appears that the acquisition was performed by someone other than the participants. Unfortunately, the datasets developed by Choras and Kozik [27] and Wei *et al.* [34] are currently not available to the research community.

The first palmprint dataset to consider the hand pose variation (**B3** in Table 1), understood as open palms with spread fingers versus closed fingers, was collected by Afifi [29] and released under the name 11K Hands. It contains over 11,000 images of hand images - both palmar and dorsal (each has about 5,500 images). The images were acquired against a white background, using a digital camera. Matkowski *et al.* [30] also released a dataset of more conventional hand images where the hand pose varies significantly, with acquisition against white background. This dataset, entitled 'NTU-Contactless Palmprint Database' (NTU-CP-v1) contains a large number of hand classes (655), with 2,478 hand images in total. An auxiliary palmprint dataset exploring various hand poses was released in 2019 by the authors under the name NUIG_Palm2 (NUIGP2) [31]. NUIGP2 was designed to support the development of ROI extraction algorithms.

C. UNCONSTRAINED PALMPRINT DATASETS

This category of palmprint datasets attempts to bring to researchers conditions as close as possible to a realistic deployment of a palmprint recognition system on consumer devices. An overview is presented in Table 1 for categories **C1** and **C2**.

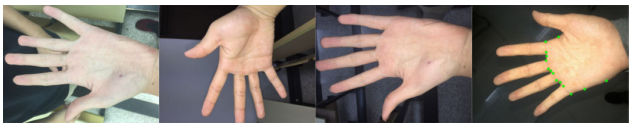
The first dataset to provide such palmprint images was released in 2017 by Ungureanu *et al.* [10] under the name NUIG_Palm1 (NUIGP1). It contains images from several devices in unconstrained scenarios (both background and hand pose, as presented in Fig. 2a).

Recently a large-scale dataset of palmprint images acquired in similar conditions to NUIGP1 was released by Shao *et al.*, entitled Xian Jiaotong University Unconstrained Palmprint database (XJTU-UP) [32]. The dataset contains 30,000+ images (200 hands) using five smartphones, making it the largest currently available palmprint dataset acquired with smartphone cameras. Several samples are provided for reference in Fig. 2b.

Another large-scale palmprint dataset acquired with smartphones was released recently by Zhang *et al.* [33]. They used two smartphones to collect 16,000 hand images in unconstrained conditions.

Representing the next step of this trend, the NTU-Palmprints from Internet (NTU-PI-v1) [30] was released in late 2019, where severe distortions in the hand pose represent the main challenge to palmprint recognition, together with heterogeneous lighting and acquisition devices. The dataset is especially large in terms of the number of hand classes (2,035), with a total of 7,781 images.

(a) NUIG_Palm1 [10]



(b) XJTU-UP [32] source: ©2019 IEEE

(c) MPD [33]

(d) NTU-PI-v1 [30] source: ©2019 IEEE

FIGURE 2. Hand image samples from Unconstrained datasets (C1 and C2) listed in Table 1.

III. ROI TEMPLATE DETECTION/EXTRACTION

This section presents a general overview of existing approaches for palmprint ROI extraction. This is an essential part of the palmprint recognition system, as any inconsistencies in ROI templates will affect the recognition task.

The detection/extraction are generally considered to be the same (as in [35]–[37]) whereas the alternative is to have two stages: one in which hand detection is performed, followed by key-point regression for palmprint extraction (as in [30], [32]). However, since palmprint recognition can benefit from a constrained acquisition protocol (where the hand/palm occupy most of the input image), a hand detection stage is not generally required (as in [33], [38]). Therefore the two terms are used interchangeably in this paper.

The existing ROI extraction techniques can be grouped in four categories, based on the cues contained in the hand images as shown in Fig. 3:

- Standard palmprint ROI extraction: algorithms based on separating the hand from the background (segmentation) and performing measurements to determine the landmarks (or palm region) required for ROI extraction [12]. This family of techniques relies on accurate segmentation, as well as a specific hand pose (open palm with spread fingers).

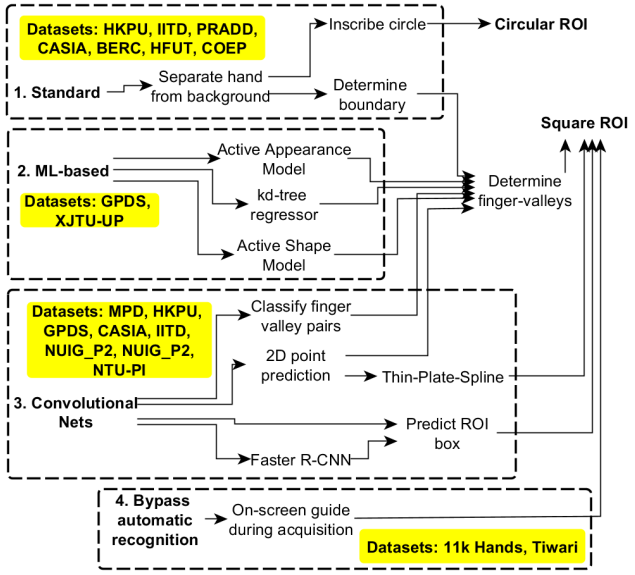


FIGURE 3. Overview of approaches for palmprint ROI extraction, with four categories based on how constrained the datasets are.

- ROI extraction based on conventional Machine Learning (ML) algorithms: ML approaches are used for the detection of palmprints or used for key-point regression. The key-point regression is a method that takes as input a hand image and returns a set of points used for ROI extraction.
- ROI extraction based on Deep Neural Networks (DNNs): Approaches relying on DNN soutions to perform detection or key-point regression task.
- Avoiding ROI detection altogether: based on specific acquisition protocols.

A. STANDARD PALMPRINT ROI EXTRACTION

Standard palmprint ROI extraction algorithms rely on accurate segmentation of the hand region from the background. The most used approaches include using Otsu's thresholding method [39] applied to grayscale images, or using a skin-color model [40]. The segmentation is a pre-processing stage that characterizes the shape of the hand and determines the key-points required for ROI extraction.

The most popular ROI extraction approach was introduced by Zhang *et al.* [41] in 2003, which relies on the constrained environment from images in databases (A1, A2) in Table 1, either touch-based or touch-less. Zhang *et al.* ROI extraction approach relies on determining the tangent line between the two side finger valleys in order to normalize the palmprint's rotation and provide a reference point from which to extract a square region. This step is made possible thanks to the constrained environment of acquisition (black background, constant lighting), characteristic of palmprint datasets (A1, A2) in Table 1.

Recently, Xiao *et al.* [22] proposed an approach based on the intersection of the binarized hand with lines of specific orientations, resulting in several candidate points for the finger valleys. They then used K-means clustering to obtain the center of each cluster.

A second category of approaches defines the contour of the extracted hand, and the distance from a point of reference (the geometric center [21], [42] or the wrist [43], etc) to the pixels found on the contour [23], [44]–[50]. Considering this distribution of distances, the peaks generally correspond to the tips of the fingers, while the local minimas correspond to the finger valleys. These type of approaches are extremely sensitive to segmentation artifacts and generally apply smoothing to the distribution of distances.

A third category traverses all the contour pixels and counts the pixels belonging to the hand region (a circle was considered for sampling). Balwant *et al.* [58] introduced specific rules to determine the finger valleys and finger tips, followed by the correct selection of finger valley points that form an isosceles triangle. Goh Kah Ong *et al.* [59] considered sampling with fewer points using 3 stages corresponding to circles with greater radius. The outliers resulting from segmentation artifacts were removed with specific rules. Franzgrote *et al.* [60] further developed the approach proposed by Goh Kah Ong *et al.* by classifying the angles of remaining lines in order to provide a rough rotation normalization step. The finger valley points were then determined with a horizontal/vertical line (depending on the orientation of the hand), having 8 points of transition from non-hand region to hand region. Zhou *et al.* [61] similarly filtered the hand contour points using intersecting circles. By evaluating the intersection of these circles with the other hand contour points and the angles they made, the finger valley points were determined. Their approach proved to be more robust to cases where the background was similar to skin color when compared to other methods [40], [59].

Morales *et al.* [62] fitted a circle inside the binarized hand, with its center found equidistantly from the finger valleys (previously determined with the center-to-contour distances).

A fourth category uses the convex hull to describe the concavity of the binarized hand map and finger valleys [63], [64].

The following are methods that are hard to classify into one category or another, as they either employ very different or combine several of the previously mentioned approaches together.

Khan *et al.* [65] determined the finger tips and the start of the palm by counting the hand-region pixels along the columns. After determining the pixels corresponding to finger valleys, several 2nd order polynomials were used to extrapolate the middle of the finger valleys. The palm's width was used to determine the size of the ROI (70% of palm size). This approach requires specific hand pose, with hands always rotated towards the left with spread fingers.

Han *et al.* [66] successively cropped the binarized hand image regions corresponding to fingers (after rotation normalization with PCA) by determining the number of transitions from background to hand area. Leng *et al.* [40] determined the finger valleys by computing differential maps

upward, to the right and the left. The AND operator was applied on these maps, resulting in 4 regions corresponding to all finger valleys. Ito *et al.* [47] considered an approach based on line detection after determining the binarized hand region, and subtracting the major lines corresponding to finger edges. Then a distance was computed from center of the palm, allowing the detection of finger valleys even with closed fingers (not relying on spread fingers). Ito *et al.* compared the effectiveness of their approach with three other algorithms [40], [41], [66].

Liang *et al.* [67] used an ROI extraction approach loosely based on [41] and [68], where the tip of the middle finger was determined and then extended to the center of the palm 1.2 times. This point was then used as a reference to determine the distance to all contour points, allowing the detection of both finger valleys and tips.

Jia *et al.* [28] exploited the constrained nature of acquisition (hand position pose, scale and rotation) to base the ROI extraction on the accurate detection of the heart line's intersection with the edge of the hand (using the MFRAT defined in [69]), performing specific pixel operations to decide on the ROI's center and size.

Kim *et al.* [24] combined several elements for ROI extraction, such as the use of a distance based on a YCbCr model, a specific hand pose (fingers spread) indicated by a guide displayed during acquisition, as well as validating finger valley points by sampling 10 pixels from the determined hand region.

Shang *et al.* [70] modified the original Harris corner detection algorithm [71] in order to locate the points at the middle of finger valleys. However, this approach relied on constrained acquisition, as the background was not overly complex. Another approach using Harris corners was proposed by Javidnia *et al.* [72]. After obtaining an initial candidate for the hand region based on skin segmentation, the palm region was located using an iterative process based on the strength of the Harris corners.

However, none of the standard approaches for palmprint ROI extraction can be used in circumstances where the background's color remotely resembles skin color or the hand's pose is not constrained (such as the **C1**, **C2** datasets in Table 1). Furthermore, one can point out the limitation of skin color segmentation regardless of the chosen color space, based on the inherent inability of classifying a pixel into skin or non-skin [73].

B. PALMPRINT ROI EXTRACTION BASED ON CONVENTIONAL ML ALGORITHMS

There are few approaches using ML algorithms for ROI extraction regressing either a predefined shape or a set of points.

Initially, Doublet *et al.* [51] considered to fit an Active Shape Model (ASM) to a number of points describing the shape of a hand (with spread fingers). The model regressed the output of a skin segmentation step, after which the centers of the two finger valleys were used to normalize the hand's rotation. Ferrer *et al.* [18] used a similar ASM to extract the hand from the background in the GPDS-CL1 dataset.

Aykut *et al.* [53] considered an Active Appearance Model (AAM), which also considered the texture information from the hand's surface. They also provided the first evaluation of predicted key-points. Because the acquisition of images was performed in a considerably constrained environment, no normalization was required relative to the palmprint's scale. Aykut *et al.* preferred to report the error in terms of pixels (from the ground truth points).

Recently, Shao *et al.* [32] employed a complex pipeline for ROI extraction for unconstrained palmprint recognition. The approach included an initial stage of hand detection using Histogram of Oriented Gradients (HOG) and a sliding window providing candidate regions at several scales to a pre-trained SVM classifier. An Ensemble of Regression Trees (ERT) [74] (initially developed for face key-point detection) was then used for the landmark regression task applied to all 14 key-points placed around the palm and base of fingers. Unfortunately, Shao *et al.* did not provide details regarding the performance of their ROI extraction, how its accuracy influences the recognition task, or any comparison with prior algorithms. An overview of these methods (II) is presented in Table 2.

TABLE 2. Overview of ROI extraction approaches that are (II) ML-based, (III) DNN-based and (IV) Bypassing automatic extraction.

	Approach	Year	Ref.	Dataset
II	ASM	2006	[51]	Own
		2011	[18]	GPDS-CL1 [18]
		2018	[52]	Own
	AAM	2015	[53]	Own
		2017	[54]	Own
	ERT	2019	[32]	XJTU-UP [32]
III	Key-point prediction	2018	[38]	CASIA [15]
		2019	[30]*	NTU-CP0-v1 [30] NTU-PI-v1 [30] HKPU [13] IIT-D [16] CASIA [15]
	PP Detector	2018	[36]	HKPU [13] HKPU-CF [19] IIT-D [16]
		2018	[55]	Own
		2018	[35]	HKPU [13] CASIA [15] GPDS-CL1 [18]
	FV Detector	2019	[33]	MPD [33]
IV	On-screen guide	2016	[25]	Own
		2018	[56]	Own
		2018	[57]	Own
	Whole image	2017	[29]	11KHands [29]

PP=Palmprint, **FV**=Finger Valley, *also includes a Spatial Network Transform [30]

C. PALMPRINT ROI EXTRACTION BASED ON NEURAL NETWORKS

There have been only a handful of attempts to use Convolutional Neural Networks (CNNs) for the ROI extraction, and most have consisted solely on experimenting on gray-level images. Bao *et al.* [38] used the CASIA palmprint database [15] to determine the positions of a hand's finger valley points. They used a shallow network composed

of 4 Convolutional and 2 Fully-Connected layers (FC), including several Dropout and MaxPooling layers. The CNN architecture achieved results comparable to Zhang *et al.* [41] in stable conditions, but surpassed it when noise was added. Since, a CNN can adapt to noisy or blurred images, the pixel-based approach used by Zhang *et al.* is vulnerable to any kind of image quality degradation.

Izadpanahkakhk *et al.* [36] trained a similar shallow network based on an existing model proposed by Chatfield *et al.* [75]. The network determined a point in the hand image and the corresponding width/height of the palmprint ROI. The network was composed of 5 Convolutional and 2 FCs, including several MaxPooling layers and one Local Response Normalization Layer (LRN). The reported results are good for constrained images from HKPU [13], but the case of in-plane rotated hands was not considered.

Jaswal *et al.* [35] trained a Faster R-CNN [76] model based on Resnet-50 (87 layers) on three palmprint datasets (HKPU, CASIA and GPDS-CL1). They reported lower Accuracy and Recall rates for CASIA (up to 5% less) than for HKPU and GPDS-CL1. This can be explained by slightly larger variation in rotation. Similar to [36], the predicted bounding boxes (considered as ROIs) do not include measures for rotation normalization, which considerably affects the recognition rate for the scenario using images from CASIA, as they contain significant rotation variation. Comparatively, images from HKPU and GPDS-CL1 are already normalized rotation-wise.

Recently, Liu and Kumar [37] also considered a Fast R-CNN [77] for palmprint ROI detection. They acquired several videos of palmprints in 11 environments (no other details provided) where the hand pose was varied (from spread to closed fingers, with several hand orientations). These acquisition sessions resulted in 30,000 images that were used for training and testing. For evaluation, Liu *et al.* only considered the percentage of images above a given threshold for Intersection over Union (IoU). However, several important aspects were not covered in Liu *et al.* work: the number of subjects in the training set, the ROI being aligned with the hand (it is maintained vertical regardless of the hand's orientation) or how much an ROI having 60% IoU (with the ground truth) affects the recognition task.

An especially promising approach was proposed by Matkowski *et al.*, which integrated a Spatial Transformer Network (STN) into ROI-LAnet, an architecture performing the palmprint ROI extraction. The STN was initially proposed by Jaderberg *et al.* [78] to improve the recogniton of distorted digits. This is achieved by learning a thin plane spline transform based on a collection of points, a Grid generator and a bilinear sampler. The STN learns a transformation T_θ that is differentiable with respect to the predicted coordinates $\hat{\theta}$ based on the input feature map.

ROI-LAnet uses a feature extraction network (based on the first 3 MaxPooling stages from the VGG16 network [79]) to obtain the feature map, followed by a regression network providing estimates for the 9 points used for describing the palmprint region (trained initially using L2 loss). The output of ROI-LAnet is a palmprint ROI of fixed size, which is normalized w.r.t. the hand's pose. The authors then include ROI-LAnet into a larger architecture to train it end-to-end using cross-entropy for loss function. An overview of these methods is presented in Table 2 (III).

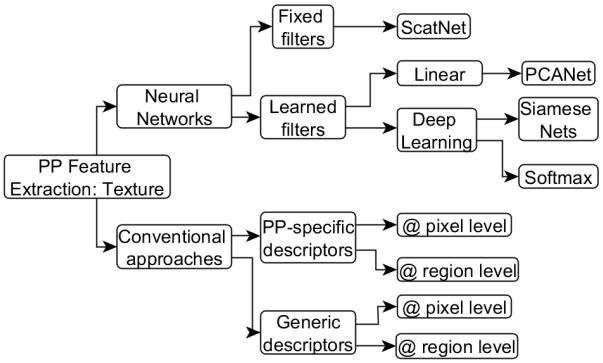


FIGURE 4. Overview of palmprint feature extraction techniques.

D. AVOIDING THE ROI DETECTION ALTOGETHER

Tiwari *et al.* [25] provided a guide on the screen of the smartphone during acquisition, avoiding the need for an ROI step. Tiwari then used an algorithm to determine the best frames for feature extraction. Similar to Tiwari's approach, Leng *et al.* [56] presented a guide on the smartphone's screen, indicating a specific hand pose and orientation for the hand. Gao *et al.* [57] used two guide-points of reference to normalize the hand's rotation during acquisition.

Afifi *et al.* [29] considered a different approach, having the entire image as the input to a CNN, thus removing any need for an ROI extraction phase. This approach is only feasible because all other parameters in the acquisition environment (background, lighting and hand orientation/scale) are not constant. An overview of these methods (IV) is presented in Table 2.

IV. PALMPRINT FEATURE EXTRACTION AND MATCHING

This section presents a general overview of approaches used for palmprint feature extraction, with emphasis being placed on the more recent advancements. In this section, the algorithms are split into two categories, based on how the kernels used for feature extraction were obtained (as visualized in Fig. 4):

1) Conventional approaches:
   a) Encoding the line orientation **at pixel-level** with:
      i) Generic texture descriptors
      ii) Palmprint-specific descriptors.
   b) Encoding the line orientation **at region-level**, with:
      i) Generic texture descriptors, a special category including descriptors such as SIFT, SURF and ORB, which are treated separately
      ii) Palmprint-specific descriptors.

2) Neural Networks approaches:
   a) Having fixed kernels, such as ScatNet [80]
   b) Kernels learned based on a training distribution:
      i) With no non-linearities, such as PCANet [81]
      ii) Deep Learning approaches:
         A) Classifying with Softmax
         B) Using Siamese network architectures.

An overview of the more conventional approaches to palmprint feature extraction is presented in Table 3, whereas an overview of the more recent approaches based on Neural Networks is presented in Table 4.

A. PALMPRINT FEATURE EXTRACTION - CONVENTIONAL APPROACHES

Conventional palmprint recognition approaches are mainly focused on line-like feature detection, subspace learning or texture-based coding. Of these, the best performing approaches have been the texture-based ones [82], which will represent the main focus of this overview. For a broader description of the other groups, please refer to the work of Zhang *et al.* [82], Kong *et al.* [83] and Dewangan *et al.* [84].

Jia *et al.* [85] defined a framework that generalized the palmprint recognition approaches. The stages of feature encoding are broken down and populated with various approaches. The following sub-sections describe these approaches and provide results in the form of either Equal Error Rate (EER) or Recognition Rate (RR) corresponding to popular palmprint datasets such as HKPU [13], CASIA [15] or IITD [16].

1) EXTRACTING PALMPRINT FEATURES WITH TEXTURE DESCRIPTORS

Chen *et al.* [86] used a 2D Symbolic Aggregate approximation (SAX) for palmprint recognition. The SAX represents a real valued data sequence using a string of discrete symbols or characters. Applied to grayscale images, it encodes the pixel values, essentially performing a form of compression. The low complexity and high efficiency of SAX make it suitable for resource-constrained devices.

Leng *et al.* [87] extract Double Discrimination Power Analysis (DPA), representing the fusion of the 2D-Discrete Cosine Transform from both palms to encode the features present.

Ramachandra *et al.* [88] employed a series of BSIF filters that were trained for texture description on a large dataset of images. The ROI is convolved with the bank of filters and then binarized (using a specific threshold value), allowing for an 8-bit encoding.

Jia *et al.* [89] investigated the potential use of HOG [90], which were successfully used in the past for robust object detection, especially pedestrians and faces. Furthermore, the Local Directional Pattern (LDP) [91] was evaluated in the context of palmprint feature extraction.

Zheng *et al.* [92] described the 2D palmprint ROI with a descriptor recovering 3D information, a feature entitled Difference of Vertex Normal Vectors(DoN). The DoN represents the filter response of the palmprint ROI to a specific filter containing several sub-regions (of 1 or -1) intersecting in the center of the filter (borders are made up of 0s), with various orientations. In order to match two DoN templates, a weighted sum of AND, OR and XOR operators was used.

Li *et al.* [93] extracted the Local Tetra Pattern (LTrP) [94] from a palmprint image that was initially filtered with a Gabor [95] or MFRAT [69] filter. Only the real component from the Gabor convolution was taken into consideration, after the winner-take-all rule of arg_{min} was applied at pixel level between all filter orientations. Then, block-wise histograms of the LTrP values were concatenated in order to determine the final vector describing a palmprint image.

Wang *et al.* [96] used the Local Binary Pattern (LBP), which encodes the value of a pixel based on a neighborhood around it [97]. Generally, the 3×3 kernel is used, allowing codes that range in value from 0 to 255.

An overview of these approaches is detailed in Table 3 under category (A0).

2) ENCODING PALMPRINT LINE ORIENTATION AT PIXEL LEVEL

One of the first approaches to extract the palmprint features from an ROI relied on only one Gabor filter oriented at $\frac{\pi}{4}$, entitled PalmCode [41]. Three values were used in the matching stage of PalmCode, namely the real, imaginary, as well as a segmentation mask to reduce the influence of poor ROI segmentation. Several approaches following a similar rationale were proposed in the following years after PalmCode, with the introduction of Competitive Code (CompCode) [95] and Robust Line Orientation Code (RLOC) [69]. Both CompCode and RLOC used a competitive rule (arg_{min}) between a bank of filters having 6 orientations. Every pixel from the palmprint ROI was considered to be part of a line, and as the lines in the palmprint correspond to black pixels, the minimum response was chosen. Whereas CompCode used the filter response from Gabor filters, RLOC used the filter response from a modified filter Jia *et al.* called MFRAT because it was inspired from the RADON transform. In the case of CompCode only the real component was used.

Gaussian filters were also used, either the derivative of two 2D Gaussian distributions (DoG [101]) or as the difference between two 2D orthogonal Gaussian filters (OLOF [100]).

Guo *et al.* [102] introduced Binary Orientation Co-occurrence Vector (BOCV), obtained the filter response of a Gabor filterbank and encoded every pixel relative to a specific threshold (0 or another threshold, chosen based on the distribution of values after convolution with a specific filter). Every filter response was L1 normalized prior to the encoding, after which the thresholded values from each orientation were used to encode an 8-bit number corresponding to every pixel. An extension of this approach was introduced by Zhang *et al.* [104] with EBOCV, which included masking the 'fragile' bits obtained after convolution with the Gabor filter-bank (as performed previously on IrisCode [119] in the

TABLE 3. Overview of (A1) approaches encoding the orientation at pixel level, (A2) approaches encoding the orientation at region level, and (B) approaches based on rotation/scale invariant image descriptors.

	Year	Acronym	Short description	Classifier	DB(s)	Best Result (EER/RR)
A0	2010	SAX [86]	Discretization of a 2D grayscale image		HKPU [13] CASIA [15]	0.3% 0.9%
	2014	BSIF [88]	Encoding filter responses from several BSIF filters	Sparse Repr. Classifier	HKPU [13] IITD (L, R) [16]	6.19% 0.42%; 1.31%
	2014	HOG [89]	Histogram of Oriented Gradients	Euclidean distance	HKPU [13]	98.03%
	2015	DDPA [87]	2D-DCT components, from both palms extracted and fused	Euclidean distance	Own	99.88%
	2016	LDP [98]	Convolution with Kirsch edge masks	Manhattan + Chi-square	HKPU [13] IITD [16]	6.10% 10.42%
	2016	DoN [92]	3D recovered descriptor from 2D image	weighted sum of 3 scores	HKPU [13] IITD [16] CASIA [15]	0.033% 0.68% 0.53%
	2017	LBP [93]	Local Binary Pattern		HKPU [13] IITD [16]	4.92% 9.71%
	2017	LTrP [93]	Local Tetra Pattern		BERC1, BERC2 [24] IITD [16]	1.49%; 1.83% 0.94%
	2020	SDDLM [99]	Salient and Discriminative Descriptor Learning Method	Euclidean distance, 1-NN	IITD [16] CASIA [15], n-CASIA* GPDS [18],n-GPDS*	97.82% 98.87%, 98.91% 99.56%, 93.75%
A1	2003	PalmCode [41]	Real and Im. components of convolution with Gabor filter $\pi/4$	Norm. Hamming	HKPU-v1 [13]	0.60%
	2004	CompCode [95]	Real components of convolution with Gabor filters	Angular Distance	HKPU-v1 [13]	98% at 10^{-6} FAR
	2005	OLOF [100]	Convolution with difference of orthogonal Gaussians	Hamming	HKPU-v1 [13]	0.0%
	2006	DoG [101]	Convolution with Derivative of Gaussians	Hamming	HKPU [13]	0.19%
	2008	RLOC [69]	Convolution with 6 MFRAT filters		HKPU [13]	0.40%
	2009	BOCV [102]	Thresholding Gabor filter response. Binary encoding	Hamming	HKPU [13]	0.019%
	2011	Contour-Code [65]	Two-sequence convolution, followed by hashing	Hash Table	HKPU-MS [103] CASIA-MS [104]	0.006% 0.3%
	2012	E-BOCV [105]	Removing 'fragile' bits from matching	Fragile-bit pattern	HKPU [13]	0.0316%
	2013	PalmHash, PalmPhasor [106]	Hash functions applied to Gabor filter response. Score-level fusion	Norm. Hamming	HKPU [13]	0.038%, 0.039%
	2016	DOC [107]	Include the 2 strongest orientations at pixel-level	Non-linear Angular Distance	HKPU [13] IITD [16]	0.0092% 0.0622%
	2016	Fast-RLOC [108]	Convolution with orthogonal pairs of Gabor/MFRAT filters	Hamming	HKPU [13]	0.041%
	2016	Half-orientation Code [109]	Convolution with 2 pairs of half-Gabor filters. Using both halves during matching		HKPU [13] IITD [16]	0.0204% 0.0633%
	2017	COM [110]	Convolution with filters describing concavity	Angular Hamming	HKPU [13]	0.14%
A2	2013	HOL [89]	Block-wise histogram of strongest orientation	Euclidean distance	HKPU [13] HKPU-MS(B) [103]	0.31% 0.064%
	2015	[24]	Modified CompCode (+HOG). Run-time of 466.65 ms on 1.40 GHz device (smartphone)	Chi-square distance	BERC1, BERC2 [24] HKPU [13] IITD [16]	2.88%, 3.15% 0.11% 5.19%
	2016	NDI [111]	Neighboring direction indicator		HKPU [13] IITD [16]	0.0225% 0.0626%
	2016	LLDP [98]	Extended encoding strategies to Gabor/MFRAT	Manhattan, Chi-square distance	HKPU [13] IITD [16]	0.021% 4.07%
	2016	LMDP [112]	Block-wise encoding of multiple dominant orientations		HKPU-v2 [13] IITD [16] GPDS [18]	0.0059% 0.0264% 0.1847%
	2016	DRCC [113]	CompCode with side orientations in weighted manner	Modified Angular Distance	HKPU [13] IITD [16]	0.0189% 0.0626%
	2017	LMTrP [93]	Local micro-tetra pattern		BERC1, BERC2 [24] HKPU-MS [103] IITD [16]	1.11%, 1.68% 0.0006% 0.87%
	2017	CR-CompCode [20]	Block-wise histogram of CompCode	CRC	Tongji [20]	98.78%
	2018	CDR [85]	Convolution with MFRAT at several scales (15) with 12 orientations. 6 overlapping regions	BLPOC	HKPU-v2 [13] HFUT [22]	0.001% 0.0868%
	2019	DDBPD [114]	Hash functions learned based on CDVs, obtained using Gabor filters.	Chi-square distance	CASIA [15] IITD [16] Tongji [20] HFUT [22]	96.40% 96.43% 98.73% 97.86%
	2019	LDDBP [115]	Dominant orientation, along with secondary orientation, encoded block-wise	Chi-square distance	HKPU [13], n-HKPU* IITD [16], n-IITD* GPDS [18], n-GPDS* CASIA [15], n-CASIA*	99.85% 97.81% 97.70% 97.27%
B	2008	[116]	SIFT + SAX. Rank-level fusion		HKPU [13]	0.37%
	2011	[117]	modified SIFT (OLOF)	Similarity + Hamming	IITD [16] GPDS [18]	0.21% 0.17%
	2013	[118]	SIFT + Iterative RANSAC	I-RANSAC	IITD [16]	(L) 0.513% (R)0.552%
	2014	[119]	RootSIFT	hierarchical	CASIA-MS [104]	1.00%
	2016	[25]	SIFT and ORB descriptors	disimilarity index	Tiwari [25]	5.55%
	2016	[50]	Sparse representation, fused at rank-level with SVM	SRC + SVM	REST [50] CASIA [15]	98.33% 99.72%

*n-HKPU, n-IITD, n-CASIA, n-GPDS are HKPU, IITD, CASIA and GPDS with Gaussian noise added.

TABLE 4. Pre-trained networks (C1), or linear Neural Networks (C2). Training CNNs for palmprint feature extraction (C3A). Siamese approaches (C3B) to training CNNs for palmprint feature extraction.

	Year	Acronym	Short description	Classifier	DB(s)	Best Result (EER/RR)
C1	2016	[131]	Log-its layer of AlexNet, pre-trained on ImageNet	Hausdorff dist.	HKPU [13] CASIA [15] IITD [16]	0.044% 0.0803% 0.1113%
	2018	[132]	Output of FC6 layer from AlexNet, VGG16/19, pre-trained on ImageNet	SVM	MOHI [133] COEP [17]	95.50% 98.00%
	2018	[134]	Transfer learning: AlexNet pre-trained on ImageNet	Softmax+SVM	CPNB [134]	0.310%
C2	2016	**ScatNet** [80]	Deep Scattering Network, fixed weights	lin-SVM K-NN	HKPU [13]	100% 94.40%
	2016	**PCANet** [81]	Obtaining filters based on PCA and training distribution	SVM	HKPU-MS [103] CASIA-MS [104]	0.0% 0.12%
	2019	**PalmNet** [135]	Modification of PCANet, with the 2nd layer composed of Gabor filters (selected adaptively, based on a training distribution)	1-NN, L2 dist.	CASIA [15] IITD [16] REST [50] Tongji [136]	0.72% 0.52% 4.50% 0.16%
C3-A	2015	[137]	Shallow net	Softmax	HKPU-MS [103] Own	99.97% 93.4%
	2017	[29]	Two-stream CNN: low-frequency and high-frequency, then trained to classify the image according to its class.	SVM	11KHands [29] IITD [16]	96.00% 94.80%
	2018	[36]	Transfer-learning: CNN [75] pre-trained on ImageNet, re-trained with cross-entropy	KNN, SVM, RFC	HKPU [13] IITD [16]	100% 96.9%
	2018	**Palm-RCNN** [138]	Inception-ResNetV1, with Cross-entropy and Center loss (combined loss)	SVM, Euclidean dist.	Tongji-MS [138]	100%* 2.74%***
	2019	[11]	Transfer-learning: AlexNet, VGG16, InceptionV3 and ResNet50, retrained with cross-entropy loss	Softmax	CASIA [15] IITD [16] GPDS-CL1 [18]	3.78% 4.79% 4.69%
	2019	**JDCFR** [139]	Shallow CNNs trained on each spectral band (53)	CRC	Own (MS)	0.01%
	2019	[26]	Transfer learning: VGG16, GoogLeNet, [75] pre-trained on ImageNet; trained using Cross-entropy loss	Softmax	SMPD [26]	93.40%
	2019	**FERnet**, **EE-PRnet** [30]	Architecture based on pre-trained VGG16 (pruned after 3rd maxpool)," with **D** and **FC** (FERnet). EE-PRnet is trained end-to-end, together with the ROI extraction architecture. Trained with cross-entropy loss.	PLS [140]	NTU-PI [30] NTU-CP [30]; IITD [16] HKPU [13]; CASIA [15]	41.92%;64.73* 0.76%; 0.26% 0.15%; 0.73%
C3-B	2016	[141]	Siamese network trained with d-prime loss		CASIA [15] IITD [16]	1.86% 1.64%
	2018	**RFN** [37]	Soft-shifted Triplet loss		IITD [16] PolyU-IITD [21]	0.68% 0.15%
	2018	[142]	VGG16 retrained last layers		HKPU [13] XJTU-UP [32]	0.2819% 4.559%
	2019	**DHCN** [32]	Binary Hashing, with Knowledge Distillation [143]		XJTU-UP [32] XJTU-kd [32]	0.60% 5.83%
	2019	**Deep-MV** [33]	MobileNetV2 with secondary net	-	MPD [33]	89.91%
	2019	**PalmGAN** [144]	Cyclic GAN used for cross-domain transformation		HKPU-MS [103] SemiU [144] Uncontr. [144]	0.0% 0.005% 5.55%
	2019	[145]	Siamese with secondary network. Few-shot training	Softmax	HKPU-MS [103] Pa, Pb Pc, Pd	99.4% 95.4%, 93.4% 98.8%, 96.4%

D = dropout; **FC** = fully connected layer; * and ** refer to the identification results expressed in Recognition Rate, Rank-1 and Rank-30. *** refers to verification results, as opposed to identification.

context of iris recognition). In this context, a 'fragile' bit is interpreted as being the pixels close to 0 (after convolution).

Leng *et al.* [105] introduced PalmHash code and PalmPhasor code (phasor domain), obtained with hash functions after the convolution of a palmprint template with a family of Gabor filters. Several orientations were used for score-level fusion to improve the robustness of the feature extraction. The use of hash functions in biometrics allows the generation of cancelable biometric tokens.

Khan *et al.* [65] introduced ContourCode, obtained by convolving the input ROI in two distinct stages. Initially, the filter response corresponding to a Non-subsampled Contourlet Transform (uniscale pyramidal filter) was obtained, after which the ROI was convolved with a directional filter bank. The strongest sub-band was determined (arg_{max}) and the resulting code was binarized into a hash table structure.

Fei *et al.* [106] introduced the Double-orientation Code (DOC) which encodes the two lowest responses (to a Gabor filter bank). In order to compute the distance between two ROIs, a non-linear angular distance, measuring the dissimilarity of the two responses was determined.

Zheng *et al.* [107] investigated the effect of number of filter orientations on the efficiency of CompCode [95] and RLOC [69]. A single orthogonal pair of Gabor and MFRAT filters was found to perform better than when using 6 orientations. This encoding approach was called Fast-Compcode/Fast-RLOC due to its increase in speed, mostly due to a reduction in complexity.

An interesting approach was introduced by Tabejamaat *et al.* [109], who described the concavity of a 2D palmprint ROI by convolving it with several Banana wavelet filters [120]. Three pairs of filters (positive and

negative concavity) were convolved with the ROI and a competitive rule (arg_{min}) was used for encoding. The joint representation was called Concavity Orientation Map (COM). An angular hamming distance was then used for matching COMs.

Recently Zhao *et al.* [99] determined a projection matrix (SDD) using Least Square Regression (LSR), where only salient and discriminative features of palmprints are learned, removing the influence of noise. Once the SDD is determined, it can be used to encode any palmprint image.

An overview of these approaches is detailed in Table 3 under category (A1).

3) REGION-BASED PALMPRINT LINE ORIENTATION ENCODING

Jia *et al.* [89] introduced an analysis of region-based methods applied to palmprint recognition. They extended the RLOC encoding capabilities to the region-level by using the histogram of dominant orientations (after the arg_{min} rule). The histograms of orientations were then concatenated. This approach essentially replaced the gradient information used in HOG with the dominant MFRAT filter response. For matching two palmprint templates, the L2 distance was used.

Zhang *et al.* [20] used a similar approach to retrieve the block-wise histograms of CompCode orientations, but a Collaborative Representation Classifier (CRC) was used to perform the classification.

Kim *et al.* [24] used a modified version of CompCode, where a segmentation map was first determined by using the real values of the filter responses. This segmentation map was then used to compute the strongest gradients and compute the corresponding HOG. The Chi-square distance was used for matching palmprint templates.

Li *et al.* [93] extended the general approach of Local Tetra Patterns [94] by replacing the derivative along the width and length with the filter response to MFRAT [69] or Gabor [95] filter banks. Furthermore, the encoding method was modified to take into account the thickness of the palm lines. The image was then separated into regions and histograms were computed for each region. Finally, they were concatenated and passed through a Kernel PCA filter to reduce the dimensionality of the template.

Luo *et al.* [98] introduced the Local Line Directional Pattern (LLDP), which represented an extension of general region encoding approaches (LDP [91], ELDP [121] and LDN [122]). The convolution stage replaced the use of Kirsch filters with Gabor or MFRAT filter banks. This step corresponds to replacing the general gradient information in a region with palmprint-specific line information. A similar approach was employed by Fei *et al.* [123] to encode the 2D information in the context of a 3D palmprint recognition system. The response to the Gabor bank of filters was encoded using the LBP [97] strategy. The system used a feature-level fusion technique.

Fei *et al.* [111] introduced the Local Multiple Directional Pattern (LMDP) as a way of representing two strong line orientations when these were present, instead of choosing only the dominant line orientation. The block-wise histograms of LMDP codes were computed and matching was performed using the Chi-square distance. In a similar manner, Xu *et al.* [112] introduced SideCode as a robust form of CompCode, representing a combination of the dominant orientation with the side orientations in a weighted manner.

Fei *et al.* [110] used the Neighboring Direction Indicator (NDI) to determine the dominant orientation for each pixel, along with its relation to the orientations of the neighboring regions in the image.

Jia *et al.* [85] introduced the Complete Directional Representation (CDR) code, encoding the line orientation information at 15 scales with 12 MFRAT filters. From these images 6 overlapping regions were extracted, resulting in 1080 regions. These features were then matched using Band Limited Phase-only Correlation (BLPOC) [124]. This approach was based on the average cross-phase spectrum of the 2D Fast Fourier Transforms (FFT) corresponding to two palmprint templates. The impulse centered on (x_0, y_0) corresponds to the probability of the two templates belonging to the same class (large if intra-class, low if inter-class).

Fei *et al.* [113] first extracted a Convolution Difference Vector (CDV) for every pixel in a palmprint image, based on the filter response to a family of Gabor filters with 12 orientations. Then using these CDVs, six hash functions are determined, which learn a discriminative projection of the CDVs such that intra-class distance is minimized and inter-class distance is maximized. The binary output is then split into non-overlapping blocks and histograms are used to create the final feature vector by concatenating.

Similarly, Fei *et al.* [114] compute a feature called Local Discriminant Direction Binary Pattern (LDDBP), which encodes the dominant orientation at every pixel.

An overview of these approaches is detailed in Table 3 under category (A2).

4) IMAGE DESCRIPTORS USED FOR PALMPRINT FEATURE EXTRACTION

Image descriptors such as the Scale Invariant Feature Transform (SIFT) [125] represented a major breakthrough for object detection in unconstrained conditions because of the rotation and scale invariance of SIFT key-points. This brought much interest to SIFT descriptors, which were either applied directly to palmprint images, such as in [25], [117], [126] or with certain modifications brought to one of its stages.

Morales *et al.* [116] replaced the DoG with the Ordinal Line Oriented Feature (OLOF) in the stage associated to key-point detection. Furthemore, the score determined from matching SIFT descriptors was fused with the OLOF matching prediction, making the prediction more robust. Zhao *et al.* [117] improved the initial key-point detection stage by filtering the palmprint image with a circular Gabor

filter. Then the corresponding SIFT descriptors were matched using a modified version of the RANSAC algorithm which used several iterations.

Kang *et al.* [118] introduced a modified SIFT which is more stable, called RootSIFT. Furthermore, histogram equalization of the graylevel image was added as a pre-processing stage. A mismatching removal algorithm (of SIFT descriptors) based on neighborhood search and LBP histograms further reduced the number of out-liers.

Charfi *et al.* [50] used a sparse representation of the SIFT descriptors to perform the matching, as well as rank-level fusion with an SVM. Similarly, a rank-level fusion was performed by Chen *et al.* [115] matching SAX and SIFT descriptors.

Tiwari *et al.* matched SIFT and ORB [127] descriptors acquired using smartphone cameras. As with most other approaches using SIFT descriptors, a dissimilarity function was defined, counting the number of in-lier matches performed between two images. Srinivas *et al.* [128] used Speeded Up Robust Features (SURF) [129] to match two palmprint ROIs. They further improved the matching speed by only matching the SURF descriptors extracted from specific subregions of the ROI, instead of the entire surface of the ROI.

An overview of these approaches is detailed in Table 3 under category (B).

B. CNN-BASED APPROACHES

One of the great advantages of using CNNs is that the filters are learned from a specific training distribution, which makes them relevant to the task of palmprint recognition. As opposed to traditional (crafted) features, the learned features are trained to describe any distribution. The main disadvantage of this approach lies in the requirement of abundant and accurately labeled training data, which generally is a problem.

The existing approaches for palmprint feature extraction relying on CNNs, can be split into three categories:

- Using pre-trained models (on ImageNet), the network's output is considered to be the extracted feature. Also relies on a classifier such as SVM.
- Networks of filters optimised using various approaches.
- Training from scratch (or using transfer-learning) of DNNs to determine embeddings that minimize intra-class distance and maximize inter-class distance.

1) USING PRE-TRAINED DNNs

Dian *et al.* [130] used AlexNet [145] pre-trained on ImageNet to extract deep features. These were then matched using the Hausdorff distance. In a similar fashion, Tarawneh *et al.* [131] used several networks pretrained on ImageNet (AlexNet, VGG16 [79] and VGG19). The extracted deep features from the images in two hand datasets (COEP [17] and MOHI [132]) were then matched using a multi-class SVM.

Ramachandra *et al.* [133] used transfer-learning (AlexNet) to match palmprints acquired from infants. The class decision was obtained through a fusion rule, which took into consideration the prediction from an SVM, as well as the Softmax prediction of the network.

An overview of these approaches is presented in Table 4 under category (C1).

2) PCANet, ScatNet AND PalmNet

Minaee and Wang [80] employed a scattering network (ScatNet) that was first introduced by Bruna and Mallat [146] for pattern recognition tasks, especially because of its invariance to transformations such as translation and rotation. ScatNet uses Discrete Wavelet Transforms (DWT) as filters and considers the output(s) at each layer as the network outputs (not just the last layer), providing information regarding the interference of frequencies in a given image [146]. Meraoumia *et al.* used a filter bank of 5 scales and 6 orientations, the network having an architecture composed of 2 layers. The palmprint ROIs were split into blocks of 32×32 pixels and passed through the network, resulting in 12,512 scattering features. PCA was applied to reduce the dimensionality, reducing it to the first 200 components. A linear SVM was then used for the classification task.

Chan *et al.* [147] initially introduced PCANet for general pattern recognition applications. Unlike DNNs which make use of the Rectified Linear Unit (ReLU), the PCANet does not contain any non-linearity. Instead, the filters are determined from a distribution of training images. Specifically, a series of overlapping blocks are extracted from every input image, after which the mean is removed. Based on the derived covariance matrix a number of Eigen vectors are extracted (after being sorted, the top 8) and considered as filters belonging to the first layer. The input to the second layer is the distribution of input images to the 1st layer, but convolved with the computed filters in layer 1. This process is repeated for any given number of layers, but generally architectures with 2 layers are commonplace.

PCANet was used for palmprint feature extraction by Meraoumia *et al.* [81] on two datasets - CASIA Multispectral [43] and HKPU-MS [103]. For classification, both SVM and KNN reported 0% EER across all spectral bands for HKPU-MS and 0.12% EER for CASIA-MS. However, after applying a score-fusion scheme where the first 3 bands are used, the EER drops to 0%.

Recently, Genovese *et al.* [134] expanded the PCANet approach to include convolutions with fixed-size and variable-sized Gabor filters in the 2nd layer. The described architecture entitled 'PalmNet' determines the Gabor filters with the strongest response, followed by a binarization layer. An alternative architecture is considered, entitled 'PalmNet-GaborPCA', where the filters of the first layer are configured using the PCA-based tuning procedure used in PCANet, whereas the kernels in the 2nd layer are configured using the Gabor-based tuning procedure. For classification, a simple KNN classifier is used.

PalmNet represents an interesting approach for quickly training on large datasets of palmprints, at the same time requiring fewer resources than DNNs.

An overview of these approaches is presented in Table 4 under category (C2).

3) TRAINING DNNs

The main distinction separating approaches in this category is the training strategy being used.

If the classification task is borrowed from the standard pattern recognition problem (like the ImageNet challenge), then the CNN is required to predict the class to which an input palm print belongs to. The network's last layer is FC with a number of units corresponding to the number of classes (in the form of a one-hot vector, depending on the size of the dataset), with the activation function being Softmax (expressing the probability of that input image to belong to either class). In this case, the loss function is the cross-entropy. Example implementations include [11], [26], [29], [36], [136], [138].

Fei *et al.* [11] compared the performance of several networks like AlexNet, VGG16, InceptionV3 and ResNet50. Izadpanahkakhk *et al.* [26] trained and evaluated four networks (GoogLeNet, VGG16, VGG19 and a CNN developed by Chatfield *et al.* [75] for the ImageNet challenge) on two novel palmprint datasets.

Alternatively, after training with cross-entropy loss, the output from the log-its layer (the layer preceding the Softmax layer) can be considered as the extracted feature, which is then used to train a classifier such as SVM [29], CRC [138] or Random Forest Classifier (RFC) [36]. Zhang *et al.* [137] used a combination of cross-entropy and center-loss functions during training for multi-spectral palmprint matching. After learning a representation of palmprints, they then fed the embeddings (output of log-its layer) to an SVM.

Afifi *et al.* also take into consideration separating the input image's information into either high-frequency and low-frequency, thus having a two-stream CNN. The two branches later concatenate, to allow the training based on classification. Several of these layers' outputs are then concatenated, and then classified using an SVM which employs a SUM rule for fusion.

Matkowski *et al.* [30] provided the first CNN-based solution for palmprint recognition which was trained End-to-End (EE-PRnet) for palmprint feature extraction. This architecture was composed of the previously mentioned ROI-LAnet and FERnet, which was also based on a pre-trained VGG16 (pruned after the 3rd Maxpool) architecture. This was followed by two FC layers benefiting from Droput regularization. The network is trained using Cross-entropy (a 3rd FC layer was added to the network, corresponding to palmprint classes), but the authors explore several training scenarios regarding the Dropout layers, or fine-tune specific blocks in FERnet. Furthermore, a color augmentation protocol consisting of randomly shifting the saturation and contrast of images, was performed on-the-fly during training.

After obtaining the palmprint embeddings (from the 2nd FC layer), they are matched using Partial Least Squares regression (PLS) [139], linear SVM, KNN-1 and Softmax. The best results were obtained using PLS.

Overall, the EE-PRnet provides the best results, showing that training both networks (ROI-LAnet and FERnet) together allows the architecture to reach a better understanding of the features contained in the palmprint, as well as the distortions brought by the hand's pose. Furthermore, this setup provides a considerable advantage, as the input to the network is the full image, not a cropped image of the hand.

An overview of these approaches is presented in Table 4 under category (C3-A).

Another training approach is to use the Siamese architecture (overview presented in Table 4), characterized by two inputs (or several) resulting in two embeddings (usually 128 units corresponding to the last FC) that are then compared with a loss function to determine how similar they are versus how similar they should be. This architecture, where the same network outputs the two embeddings, relies on a similarity estimation function, such as the Contrastive loss [148], or the Center loss [149], where the distance between inputs is minimized (intra-class) or increased (inter-class). When the three inputs (triplets) are considered, the distance between the anchor and the positive sample is reduced while increasing the distance between the anchor and the negative sample [150].

Svoboda *et al.* [140] introduced a loss function called 'discriminative index', aimed at separating genuine-impostor distributions. Zhong *et al.* [141] used transfer-learning based on VGG16 (initially trained on ImageNet) and Contrastive loss.

Zhang *et al.* [33] used a Siamese architecture of two MobileNets [151] outputting feature vectors that are then fed to a sub-network tasked with the intra-class probability (0 for inter-class and 1 for intra-class, with 0.5 as a decision threshold). It is not clear, however, what loss function they used (most likely contrastive loss). Du *et al.* [144] used a similar architecture trained using the few-shot strategy. Shao *et al.* [152] used the output of a 3-layer Siamese network, and matched the palmprints from two datasets (HKPU-Multispectral and a dataset collected with a smartphone camera) with a Graph Neural Network (GNN). Unfortunately, the training details of the Siamese network are not clear.

Liu *et al.* [37] introduced the soft-shifted triplet loss as a 2D embedding specifically developed for palmprint recognition (instead of a 1D embedding). Furthermore, translations on x and y axes were used to determine the best candidates for triplet pairs (at batch level). Recently, Shao *et al.* [32] introduced an approach based on hashing coding, where the embeddings used to encode the palmprint classes are either 0 or 1. Furthermore, similar matching performances were obtained using a much smaller network, obtained via Knowledge Distillation [142]. These are worthwhile directions for development, as they represent solutions to the limitations of resource-constrained devices.

A promising strategy for cross-device palmprint matching was recently proposed by Shao *et al.* [143] with PalmGAN, where a cycle Generative Adversarial Network (cycle GAN) [153] was used to perform cross-domain transformation between palmprint ROIs. A proof of concept was evaluated on the HKPU-Multispectral (HKPU-MS) palmprint dataset containing palm images acquired at several wavelengths, as well a semi-unconstrained dataset acquired with several devices.

An overview of these approaches is presented in Table 4 under category (C3-B).

C. RUN-TIME EVALUATION ON CONSUMER DEVICES

Despite of the interest in smartphone-based palmprint recognition, the number of papers evaluating ROI and feature extraction pipelines on consumer devices is limited. To the best of our knowledge, only Kim *et al.* [24] reported the run-time required for their proposed system, as implemented on a 1.40 GHz smartphone. They have determined the acquisition and ROI extraction to require 218.76 ms, whereas the feature extraction was timed at 466.65 ms.

Over the years considerable effort has been invested into migrating deep architectures that show high performance onto devices found 'at the edge' of computing [154]–[156].

At the time of writing this review there were no papers in the field of palmprint recognition that evaluate the inference time required by CNNs when done on smartphones. However, Ignatov *et al.* [156], [157] considered the problem of face recognition (with feature embedding, based on Inception-Resnet-v1) and evaluated on a large collection of smartphones/Systems on a Chip (SoC). When running on the CPU, the average run-time for one image was around 400 ms, whereas enabling the use of the GPU reduced the run-time by up to 10 times. Therefore, the recently launched smartphones/SoCs support the integration of CNNs (as demonstrated with Inception-ResNet-V1) into real-time applications of biometric recognition systems, such as palmprint templates.

V. DISCUSSION AND CONCLUSIONS

A. PALMPRINT DATASETS

The advancement of palmprint recognition relies on the release of relevant datasets which reflect specific sets of requirements. Initially the main focus was placed on recognition, allowing little to no flexibility in terms of interaction with the system (e.g. HKPU [13]).

As the sensor technology progressed (and new consumer devices appeared on the market), there was more room for various aspects, i.e. contactless systems (IITD [16], CASIA [15]). Then invariance to various factors of the acquisition encouraged the introduction of datasets like BERC [24] (background), or 11K Hands [29] (hand pose) and PRADD [28] (devices used for acquisition). Unfortunately there are several datasets that are no longer available to researchers, such as PRADD [28] or DevPhone [23]. Some recently introduced datasets are yet to be released to the research community (e.g. HFUT [22], MPD [33] or XJTU-UP [32]).

Following the general trend of biometric recognition migrating to consumer devices, the last years have seen the introduction of several large-scale palmprint datasets (e.g. XJTU-UP [32]) reflecting the challenging operating conditions brought by a mobile environment.

A new category of unconstrained palmprint datasets was recently introduced with NTU-PI-v1 [30], including the palmprint acquired with conventional cameras to the list of forensic applications. This collection of palmprints gathered from the Internet proved to be especially challenging, given the low resolution of images, the high degree of distortion, as well as the large number of hand classes.

It is our opinion that these will be the most meaningful palmprint datasets for the upcoming 5-10 years, anticipating the adoption of palmprint recognition on smartphones and other devices. An overview of this transition was presented, the culmination of which is represented by the unconstrained datasets class (C in Table 1), initiated with the introduction of NUIG_Palm1 [10] in 2017.

B. PALMPRINT ROI EXTRACTION

The approaches used for palmprint region of interest extraction are linked directly with the operating conditions of devices used for acquisition. In palmprint datasets where the background is fixed (e.g. HKPU, CASIA, IITD, COEP) the task of segmentation is a straightforward procedure. However, when the background is unconstrained such as is the case with images from BERC, skin color thresholding provides limited results, even when the skin model is computed for every image based on a distribution of pixels [24].

With the migration of palmprint recognition onto consumer devices, the general pipeline for ROI extraction needs to take into consideration more challenging factors such as lighting conditions, hand pose and camera sensor variation. It is in this context that more powerful approaches based on machine learning or deep learning can provide robust solutions without imposing strict protocols for acquisition onto the user of consumer devices.

A complete evaluation of these approaches is yet to be made in terms of:

1) The prediction error of the key-points used for ROI extraction/alignment. This seems to have been a commonly overlooked step in most research papers, with some exceptions (e.g. [65]).
2) Recognition rate and the main sources of error (from the ROI extraction) affecting recognition.
3) Running time and resource requirements, especially for CNN-based approaches. Low inference time is expected from all solutions running on consumer devices.

Furthermore, at the time of writing of this literature review, there are currently no CNN-based solutions to detect the

palmprint in unconstrained environments, besides the Fast R-CNN approach demonstrated by Liu *et al.* [37], which is a Fast-RCNN.

The recent use of a CNN for the normalization of palmprint ROIs regarding hand pose by Matkowski *et al.* [30] has opened up exciting new possibilities for unconstrained palmprint ROI extraction (they do not address the task of palmprint detection). The Spatial Transform Network learns a non-affine transform applied to the ROI, defined by the palmprint's labeled key-points. Alternatively, pose correction could be made using 3D information, similar the work of Kanhangad *et al.* [158]. Although a special 3D sensor is used in [158], the hand's 3D structure can be recovered from the 2D image with hand pose estimation algorithms (as was developed by Mueller *et al.* [159]).

C. PALMPRINT FEATURE EXTRACTION

Although palmprint recognition took off in early 2000's with the introduction of HKPU [13] dataset, the pipeline stage that received the most attention from the research community has been the palmprint feature extraction.

As was the case for iris and face recognition, CNNs have become the current state of the art in palmprint recognition (Section IV-B). The general trend is to either train a network using Cross-entropy or Center-loss (e.g. [11], [26], [29], [30], [137]), Siamese networks (e.g. [33], [37], [140], [143]), but there are or also entirely linear networks (PCANet [81] and PalmNet [134]).

It is important to note that most of these works use in their training/evaluation scenarios images acquired with smartphones (on datasets such as XJTU-UP [32] and MPD [33]). The cross-device training and matching will become a main focus especially for device-independent palmprint recognition solutions, as demonstrated by [30]. This is first investigated in [10], with impressive results being obtained in [37] and [30]. The cross-domain conversion of a palmprint ROI using a generative approach [143] also represents a promising direction of research. A GAN-based architecture might benefit from the ROI pose-normalization approach introduced by Matkowski *et al.* [30], where the ROI extraction network contains a Spatial Transform Network [78].

D. ON-DEVICE RUN-TIME OPTIMIZATION

Based on recent work [155], [156], it is possible to estimate the inference time for neural networks (used for feature embedding) to be done in real-time on the newer smartphones, released in 2019/2020.

The complexity of architectures becomes an important factor to optimize, as in [32], where the network is distilled (number of layers is reduced) and the network's output is a discrete hash code (binary values). This not only reduces the processing requirements (including matching), but also the storage space necessary when dealing with a large number of classes. An alternative approach would be to consider the ternarization of networks [160].

As in the case of ROI extraction algorithms, the feature extraction approaches (especially the CNN-based solutions) require an evaluation in terms of processing time, as this aspect is only touched in few papers (e.g. [24] and [37], [55]).

REFERENCES

[1] P. Corcoran and C. Costache, "Biometric technology and smartphones: A consideration of the practicalities of a broad adoption of biometrics and the likely impacts," in *Proc. IEEE Int. Symp. Technol. Soc. (ISTAS)*, Jun. 2015, pp. 1–7.
[2] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," *CMU School Comput. Sci.*, vol. 6, p. 2, Jun. 2016.
[3] S. Thavalengal, P. Bigioi, and P. Corcoran, "Iris authentication in handheld devices–considerations for constraint-free acquisition," *IEEE Trans. Consum. Electron.*, vol. 61, no. 2, pp. 245–253, May 2015.
[4] C. Roberts, "Biometric attack vectors and defences," *Comput. Secur.*, vol. 26, no. 1, pp. 14–25, Feb. 2007.
[5] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.
[6] A. Genovese, V. Piuri, F. Scotti, and S. Vishwakarma, "Touchless palmprint and finger texture recognition: A deep learning fusion approach," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Jun. 2019, pp. 1–6.
[7] A. Meraoumia, S. Chitroub, and A. Bouridane, "Fusion of finger-knuckle-print and palmprint for an efficient multi-biometric system of person recognition," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.
[8] W. M. Matkowski, F. K. S. Chan, and A. W. K. Kong, "A study on wrist identification for forensic investigation," *Image Vis. Comput.*, vol. 88, pp. 96–112, Aug. 2019.
[9] (May 2018). *Epson and Redrock Biometrics Bring First Biometric Authentication Solution to Consumer Augmented Reality Headsets*. [Online]. Available: https://www.businesswire.com/news/home/20180508005612/en/Epson-Redrock-Biometrics-Bring-Biometric-Authentication-Solution
[10] A.-S. Ungureanu, S. Thavalengal, T. E. Cognard, C. Costache, and P. Corcoran, "Unconstrained palmprint as a smartphone biometric," *IEEE Trans. Consum. Electron.*, vol. 63, no. 3, pp. 334–342, Aug. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8103383/
[11] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, "Feature extraction methods for palmprint recognition: A survey and evaluation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 2, pp. 346–363, Feb. 2019.
[12] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," *Neurocomputing*, vol. 328, pp. 16–28, Feb. 2019.
[13] The Hong Kong Polytechnic University. *PolyU Palmprint Palmprint Database*. Accessed: Jan. 16, 2020. [Online]. Available: http://www.comp.polyu.edu.hk/~biometrics/
[14] Bogazici University, Istanbul, Turkey. *Bosphorus Hand Database*. Accessed: Jan. 16, 2020. [Online]. Available: http://bosphorus.ee.boun.edu.tr/hand/Home.aspx
[15] The Chinese Academy of Sciences, Automation Institute. *CASIA Palmprint Database*. Accessed: Jan. 16, 2020. [Online]. Available: http://biometrics.idealtest.org/
[16] Indian Institute of Technology Delhi. (2014). *IIT Delhi Touchless Palmprint Database (Version 1.0)*. Accessed: Jan. 16, 2020. [Online]. Available: https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm
[17] College of Engineering, Pune-411005 (An Autonomous Institute of Government of Maharashtra). *Palmprint Dataset*. Accessed: Jan. 16, 2020. [Online]. Available: http://www.coep.org.in/resources/coeppalmprintdatabase
[18] M. A. Ferrer, F. Vargas, and A. Morales, "BiSpectral contactless hand based biometric system," in *Proc. CONATEL*, May 2011, pp. 1–6.
[19] V. Kanhangad, A. Kumar, and D. Zhang, "A unified framework for contactless hand verification," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1014–1027, Sep. 2011.
[20] L. Zhang, L. Li, A. Yang, Y. Shen, and M. Yang, "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach," *Pattern Recognit.*, vol. 69, pp. 199–212, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317301681

[21] A. Kumar, "Toward more accurate matching of contactless palmprint images under less constrained environments," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 34–47, Jan. 2019.
[22] Q. Xiao, J. Lu, W. Jia, and X. Liu, "Extracting palmprint ROI from whole hand image using straight line clusters," *IEEE Access*, vol. 7, pp. 74327–74339, 2019.
[23] S. Aoyama, K. Ito, T. Aoki, and H. Ota, "A contactless palmprint recognition algorithm for mobile phones," in *Proc. Int. Workshop Adv. Image Technol.*, Nagoya, Japan, 2013, pp. 409–413.
[24] J. S. Kim, G. Li, B. Son, and J. Kim, "An empirical study of palmprint recognition for mobile phones," *IEEE Trans. Consum. Electron.*, vol. 61, no. 3, pp. 311–319, Aug. 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7298090
[25] K. Tiwari, C. J. Hwang, and P. Gupta, "A palmprint based recognition system for smartphone," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 577–586.
[26] M. Izadpanahkakhk, S. M. Razavi, M. T. Gorjikolaie, S. H. Zahiri, and A. Uncini, "Novel mobile palmprint databases for biometric authentication," *Int. J. Grid Utility Comput.*, vol. 10, no. 5, p. 465, 2019.
[27] M. Choras and R. Kozik, "Contactless palmprint and knuckle biometrics for mobile devices," *Pattern Anal. Appl.*, vol. 15, no. 1, pp. 73–85, Feb. 2012.
[28] W. Jia, R.-X. Hu, J. Gui, Y. Zhao, and X.-M. Ren, "Palmprint recognition across different devices," *Sensors*, vol. 12, no. 6, pp. 7938–7964, 2012.
[29] M. Afifi, "11K hands: Gender recognition and biometric identification using a large dataset of hand images," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 20835–20854, Aug. 2019.
[30] W. M. Matkowski, T. Chai, and A. W. K. Kong, "Palmprint recognition in uncontrolled and uncooperative environment," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1601–1615, Oct. 2019.
[31] (Feb. 2020). *NUIG-PALM2 Dataset of Hand Images*. [Online]. Available: https://github.com/AdrianUng/NUIG-Palm2-palmprint-database
[32] H. Shao, D. Zhong, and X. Du, "Efficient deep palmprint recognition via distilled hashing coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 714–723.
[33] Y. Zhang, L. Zhang, X. Liu, S. Zhao, Y. Shen, and Y. Yang, "Pay by showing your palm: A study of palmprint verification on mobile platforms," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 862–867.
[34] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4724–4732.
[35] G. Jaswal, A. Kaul, R. Nath, and A. Nigam, "DeepPalm–A unified framework for personal human authentication," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jul. 2018, pp. 322–326.
[36] M. Izadpanahkakhk, S. Razavi, M. Taghipour-Gorjikolaie, S. Zahiri, and A. Uncini, "Deep region of interest and feature extraction models for palmprint verification using convolutional neural networks transfer learning," *Appl. Sci.*, vol. 8, no. 7, p. 1210, Jul. 2018. [Online]. Available: http://www.mdpi.com/2076-3417/8/7/1210
[37] Y. Liu and A. Kumar, "Contactless palmprint identification using deeply learned residual features," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 2, no. 2, pp. 172–181, Apr. 2020.
[38] X. Bao and Z. Guo, "Extracting region of interest for palmprint by convolutional neural networks," in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Dec. 2016, pp. 1–6.
[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[40] L. Leng, G. Liu, M. Li, M. K. Khan, and A. M. Al-Khouri, "Logical conjunction of triple-perpendicular-directional translation residual for contactless palmprint preprocessing," in *Proc. 11th Int. Conf. Inf. Technol., New Generat.*, Apr. 2014, pp. 523–528.
[41] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.
[42] Y. Zhou and A. Kumar, "Human identification using palm-vein images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1259–1274, Dec. 2011.
[43] Y. Hao, Z. Sun, T. Tan, and C. Ren, "Multispectral palm image fusion for accurate contact-free palmprint recognition," in *Proc. 15th IEEE Int. Conf. Image Process.*, Jun. 2008, pp. 281–284.
[44] G. S. Badrinath and P. Gupta, "Palmprint based recognition system using phase-difference information," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 287–305, Jan. 2012.
[45] K. Tiwari, D. K. Arya, G. S. Badrinath, and P. Gupta, "Designing palmprint based recognition system using local structure tensor and force field transformation for human identification," *Neurocomputing*, vol. 116, pp. 222–230, Sep. 2013.
[46] M. Hammami, S. B. Jemaa, and H. Ben-Abdallah, "Selection of discriminative sub-regions for palmprint recognition," *Multimedia Tools Appl.*, vol. 68, no. 3, pp. 1023–1050, Feb. 2014.
[47] K. Ito, T. Sato, S. Aoyama, S. Sakai, S. Yusa, and T. Aoki, "Palm region extraction for contactless palmprint recognition," in *Proc. Int. Conf. Biometrics (ICB)*, May 2015, pp. 334–340.
[48] A. Poinsot, F. Yang, and M. Paindavoine, "Small sample biometric recognition based on palmprint and face fusion," in *Proc. 4th Int. Multi-Conf. Comput. Global Inf. Technol.*, Aug. 2009, pp. 118–122.
[49] W.-S. Chen, Y.-S. Chiang, and Y.-H. Chiu, "Biometric verification by fusing hand geometry and palmprint," in *Proc. 3rd Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, vol. 2, Nov. 2007, pp. 403–406.
[50] N. Charfi, H. Trichili, A. M. Alimi, and B. Solaiman, "Local invariant representation for multi-instance toucheless palmprint identification," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 3522–3527.
[51] J. Doublet, O. Lepetit, and M. Revenu, "Contact less hand recognition using shape and texture features," in *Proc. 8th Int. Conf. Signal Process.*, vol. 3, 2006.
[52] F. Gao, K. Cao, L. Leng, and Y. Yuan, "Mobile palmprint segmentation based on improved active shape model," *J. Multimedia Inf. Syst.*, vol. 5, no. 4, pp. 221–228, 2018.
[53] M. Aykut and M. Ekinci, "Developing a contactless palmprint authentication system by introducing a novel ROI extraction method," *Image Vis. Comput.*, vol. 40, pp. 65–74, Aug. 2015.
[54] Ö. Bingöl and M. Ekinci, "Stereo-based palmprint recognition in various 3D postures," *Expert Syst. Appl.*, vol. 78, pp. 74–88, Jul. 2017.
[55] Y. Liu and A. Kumar, "A deep learning based framework to detect and recognize humans using contactless palmprints in the wild," 2018, *arXiv:1812.11319*. [Online]. Available: http://arxiv.org/abs/1812.11319
[56] L. Leng, F. Gao, Q. Chen, and C. Kim, "Palmprint recognition system on mobile devices with double-line-single-point assistance," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 93–104, Feb. 2018.
[57] F. Gao, L. Leng, and J. Zeng, "Palmprint recognition system with double-assistant-point on IOS mobile devices," in *Proc. 29th Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–8.
[58] M. K. Balwant, A. Agarwal, and C. R. Rao, "Online touchless palmprint registration system in a dynamic environment," *Procedia Comput. Sci.*, vol. 54, pp. 799–808, 2015.
[59] G. K. O. Michael, T. Connie, and A. B. J. Teoh, "Touch-less palm print biometrics: Novel design and implementation," *Image Vis. Comput.*, vol. 26, no. 12, pp. 1551–1560, Dec. 2008.
[60] M. Franzgrote, C. Borg, B. J. Tobias Ries, S. Bussemaker, X. Jiang, M. Fieleser, and L. Zhang, "Palmprint verification on mobile phones using accelerated competitive code," in *Proc. Int. Conf. Hand-Based Biometrics*, Nov. 2011, pp. 1–6.
[61] Z. Zhou, Q. Chen, and L. Leng, "Key point localization based on intersecting circle for palmprint preprocessing in public security," *J. Defense Acquisition Technol.*, vol. 1, no. 2, pp. 24–31, 2019.
[62] A. Morales, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Multisampling approach applied to contactless hand biometrics," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2012, pp. 224–229.
[63] T. Chai, S. Wang, and D. Sun, "A palmprint ROI extraction method for mobile devices in complex environment," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 1342–1346.
[64] X. Sun, Q. Xu, C. Wang, W. Dong, and Z. Zhu, "ROI extraction for online touchless palm vein based on concavity analysis," in *Proc. 32nd Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2017, pp. 1123–1126.
[65] Z. Khan, A. Mian, and Y. Hu, "Contour code: Robust and efficient multispectral palmprint encoding for human recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1935–1942.
[66] Y. Han, Z. Sun, F. Wang, and T. Tan, "Palmprint recognition under unconstrained scenes," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2007, pp. 1–11.
[67] X. Liang, D. Zhang, G. Lu, Z. Guo, and N. Luo, "A novel multicamera system for high-speed touchless palm recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Mar. 12, 2019, doi: 10.1109/TSMC.2019.2898684.
[68] E. Yoruk, E. Konukoglu, B. Sankur, and J. Darbon, "Shape-based hand recognition," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1803–1815, Jul. 2006.

[69] W. Jia, D.-S. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognit.*, vol. 41, no. 5, pp. 1504–1513, May 2008.
[70] L. Shang, J. Chen, P.-G. Su, and Y. Zhou, "Roi extraction of palmprint images using modified harris corner point detection algorithm," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2012, pp. 479–486.
[71] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, vol. 15, no. 50, p. 5244.
[72] H. Javidnia, A. Ungureanu, and P. Corcoran, "Palm-print recognition for authentication on smartphones," in *Proc. IEEE Int. Symp. Technol. Soc. (ISTAS)*, Nov. 2015, pp. 1–5.
[73] A. Albiol, L. Torres, and E. J. Delp, "Optimum color spaces for skin detection," in *Proc. Int. Conf. Image Process.*, vol. 1, Oct. 2001, pp. 122–124. [Online]. Available: http://ieeexplore.ieee.org/document/958968/
[74] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.
[75] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014, *arXiv:1405.3531*. [Online]. Available: http://arxiv.org/abs/1405.3531
[76] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[77] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[78] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, pp. 1–14, Sep. 2014.
[80] S. Minaee and Y. Wang, "Palmprint recognition using deep scattering convolutional network," 2016, *arXiv:1603.09027*. [Online]. Available: http://arxiv.org/abs/1603.09027
[81] A. Meraoumia, F. Kadri, H. Bendjenna, S. Chitroub, and A. Bouridane, "Improving biometric identification performance using pcanet deep learning and multispectral palmprint," in *Biometric Security and Privacy*. Cham, Switzerland: Springer, 2016, pp. 51–69.
[82] D. Zhang, W. Zuo, and F. Yue, "A comparative study of palmprint recognition algorithms," *ACM Comput. Surveys*, vol. 44, no. 1, pp. 1–37, Jan. 2012.
[83] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1408–1418, Jul. 2009.
[84] D. P. Dewangan and A. Pandey, "A survey on security in palmprint recognitio: A biometric trait," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 1, p. 347, Oct. 2012.
[85] W. Jia, B. Zhang, J. Lu, Y. Zhu, Y. Zhao, W. Zuo, and H. Ling, "Palmprint recognition based on complete direction representation," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4483–4498, Sep. 2017.
[86] J. Chen, Y.-S. Moon, M.-F. Wong, and G. Su, "Palmprint authentication using a symbolic representation of images," *Image Vis. Comput.*, vol. 28, no. 3, pp. 343–351, Mar. 2010.
[87] L. Leng, M. Li, C. Kim, and X. Bi, "Dual-source discrimination power analysis for multi-instance contactless palmprint recognition," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 333–354, Jan. 2017.
[88] R. Raghavendra and C. Busch, "Robust palmprint verification using sparse representation of binarized statistical features: A comprehensive study," in *Proc. 2nd ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, 2014, pp. 181–185.
[89] W. Jia, R.-X. Hu, Y.-K. Lei, Y. Zhao, and J. Gui, "Histogram of oriented lines for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 385–395, Mar. 2014.
[90] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 886–893.
[91] T. Jabid, "Robust facial expression recognition based on local directional pattern," *ETRI J.*, vol. 32, no. 5, pp. 784–794, Oct. 2010. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314207001555
[92] Q. Zheng, A. Kumar, and G. Pan, "A 3D feature descriptor recovered from a single 2D palmprint image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1272–1279, Jun. 2016.
[93] G. Li and J. Kim, "Palmprint recognition with local micro-structure tetra pattern," *Pattern Recognit.*, vol. 61, pp. 29–46, Jan. 2017.
[94] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2874–2886, May 2012.
[95] A. W.-K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, 2004, pp. 520–523.
[96] X. Wang, H. Gong, H. Zhang, B. Li, and Z. Zhuang, "Palmprint identification using boosting local binary pattern," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2006, pp. 503–506.
[97] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[98] Y.-T. Luo, L.-Y. Zhao, B. Zhang, W. Jia, F. Xue, J.-T. Lu, Y.-H. Zhu, and B.-Q. Xu, "Local line directional pattern for palmprint recognition," *Pattern Recognit.*, vol. 50, pp. 26–44, Feb. 2016.
[99] S. Zhao and B. Zhang, "Learning salient and discriminative descriptor for palmprint feature extraction and identification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 30, 2020, doi: 10.1109/TNNLS.2020.2964799.
[100] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint represention for personal identification [represention read representation]," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 279–284.
[101] X. Wu, K. Wang, and D. Zhang, "Palmprint texture analysis using derivative of Gaussian filters," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 1, Nov. 2006, pp. 751–754.
[102] Z. Guo, D. Zhang, L. Zhang, and W. Zuo, "Palmprint verification using binary orientation co-occurrence vector," *Pattern Recognit. Lett.*, vol. 30, no. 13, pp. 1219–1227, Oct. 2009.
[103] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An online system of multispectral palmprint verification," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 480–490, Oct. 2009.
[104] L. Zhang, H. Li, and J. Niu, "Fragile bits in palmprint recognition," *IEEE Signal Process. Lett.*, vol. 19, no. 10, pp. 663–666, Oct. 2012.
[105] L. Leng and J. Zhang, "PalmHash code vs. PalmPhasor code," *Neurocomputing*, vol. 108, pp. 1–12, May 2013.
[106] L. Fei, Y. Xu, W. Tang, and D. Zhang, "Double-orientation code and nonlinear matching scheme for palmprint recognition," *Pattern Recognit.*, vol. 49, pp. 89–101, Jan. 2016.
[107] Q. Zheng, A. Kumar, and G. Pan, "Suspecting less and doing better: New insights on palmprint identification for faster and more accurate matching," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 3, pp. 633–641, Mar. 2016.
[108] L. Fei, Y. Xu, and D. Zhang, "Half-orientation extraction of palmprint features," *Pattern Recognit. Lett.*, vol. 69, pp. 35–41, Jan. 2016.
[109] M. Tabejamaat and A. Mousavi, "Concavity-orientation coding for palmprint recognition," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 9387–9403, Apr. 2017.
[110] L. Fei, B. Zhang, Y. Xu, and L. Yan, "Palmprint recognition using neighboring direction indicator," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 6, pp. 787–798, Dec. 2016.
[111] L. Fei, J. Wen, Z. Zhang, K. Yan, and Z. Zhong, "Local multiple directional pattern of palmprint image," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3013–3018.
[112] Y. Xu, L. Fei, J. Wen, and D. Zhang, "Discriminative and robust competitive code for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 232–241, Feb. 2018.
[113] L. Fei, B. Zhang, Y. Xu, Z. Guo, J. Wen, and W. Jia, "Learning discriminant direction binary palmprint descriptor," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3808–3820, Aug. 2019.
[114] L. Fei, B. Zhang, Y. Xu, D. Huang, W. Jia, and J. Wen, "Local discriminant direction binary pattern for palmprint representation and recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 468–481, Feb. 2020.
[115] J. Chen and Y.-S. Moon, "Using SIFT features in palmprint authentication," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[116] A. Morales, M. A. Ferrer, and A. Kumar, "Towards contactless palmprint authentication," *IET Comput. Vis.*, vol. 5, no. 6, p. 407, 2011. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-cvi.2010%.0191

[117] Q. Zhao, X. Wu, and W. Bu, "Contactless palmprint verification based on SIFT and iterative RANSAC," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 4186–4189. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6738862

[118] W. Kang, Y. Liu, Q. Wu, and X. Yue, "Contact-free palm-vein recognition based on local invariant features," *PLoS ONE*, vol. 9, no. 5, 2014, Art. no. e97548.

[119] K. P. Hollingsworth, K. W. Bowyer, and P. J. Flynn, "The best bits in an iris code," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 964–973, Jun. 2009.

[120] G. Peters, N. Krüger, and C. Von Der Malsburg, "Learning object representations by clustering banana wavelet responses," in *Proc. 1st STIPR*, 1997, pp. 113–118.

[121] F. Zhong and J. Zhang, "Face recognition with enhanced local directional patterns," *Neurocomputing*, vol. 119, pp. 375–384, Nov. 2013.

[122] A. R. Rivera, J. R. Castillo, and O. O. Chae, "Local directional number pattern for face analysis: Face and expression recognition," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1740–1752, May 2013.

[123] L. Fei, G. Lu, W. Jia, J. Wen, and D. Zhang, "Complete binary representation for 3-D palmprint recognition," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 12, pp. 2761–2771, Dec. 2018.

[124] S. Iitsuka, K. Ito, and T. Aoki, "A practical palmprint recognition algorithm using phase information," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[125] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94

[126] N. Charfi, H. Trichili, A. M. Alimi, and B. Solaiman, "Novel hand biometric system using invariant descriptors," in *Proc. 6th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Aug. 2014, pp. 261–266.

[127] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. ICCV*, 2011, vol. 11, no. 1, p. 2.

[128] B. G. Srinivas and P. Gupta, "Palmprint based verification system using surf features," in *Proc. Int. Conf. Contemp. Comput.* Berlin, Germany: Springer, 2009, pp. 250–262.

[129] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," in *Computer Vision and Image Understanding*, vol. 110, no. 3. Berlin, Germany: Springer, Jun. 2008, pp. 346–359.

[130] L. Dian and S. Dongmei, "Contactless palmprint recognition based on convolutional neural network," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 1363–1367.

[131] A. S. Tarawneh, D. Chetverikov, and A. B. Hassanat, "Pilot comparative study of different deep features for palmprint identification in low-quality images," 2018, *arXiv:1804.04602*. [Online]. Available: http://arxiv.org/abs/1804.04602

[132] A. Hassanat, M. Al-Awadi, E. Btoush, A. Al-Btoush, E. Alhasanat, and G. Altarawneh, "New mobile phone and webcam hand images databases for personal authentication and identification," *Procedia Manuf.*, vol. 3, pp. 4060–4067, Mar. 2015.

[133] R. Ramachandra, K. B. Raja, S. Venkatesh, S. Hegde, S. D. Dandappanavar, and C. Busch, "Verifying the newborns without infection risks using contactless palmprints," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 209–216.

[134] A. Genovese, V. Piuri, K. N. Plataniotis, and F. Scotti, "PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3160–3174, Dec. 2019.

[135] L. Zhang, Y. Shen, H. Li, and J. Lu, "3D palmprint identification using block-wise features and collaborative representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1730–1736, Aug. 2015.

[136] L. Dian and S. Dongmei, "Contactless palmprint recognition based on convolutional neural network," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 209–212.

[137] L. Zhang, Z. Cheng, Y. Shen, and D. Wang, "Palmprint and palmvein recognition based on DCNN and a new large-scale contactless palmvein dataset," *Symmetry*, vol. 10, no. 4, p. 78, 2018.

[138] S. Zhao, B. Zhang, and C. L. Philip Chen, "Joint deep convolutional feature representation for hyperspectral palmprint recognition," *Inf. Sci.*, vol. 489, pp. 167–181, Jul. 2019.

[139] P. Geladi and B. R. Kowalski, "Partial least-squares regression: A tutorial," *Anal. Chim. Acta*, vol. 185, pp. 1–17, Jan. 1986.

[140] J. Svoboda, J. Masci, and M. M. Bronstein, "Palmprint recognition via discriminative index learning," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 4232–4237.

[141] D. Zhong, Y. Yang, and X. Du, "Palmprint recognition using Siamese network," in *Proc. Chin. Conf. Biometric Recognit.* Cham, Switzerland: Springer, 2018, pp. 48–55.

[142] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[143] H. Shao, D. Zhong, and Y. Li, "PalmGAN for cross-domain palmprint recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1390–1395.

[144] X. Du, D. Zhong, and P. Li, "Low-shot palmprint recognition based on meta-Siamese network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 79–84.

[145] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[146] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[147] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[148] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 1735–1742.

[149] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 499–515.

[150] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[151] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[152] H. Shao and D. Zhong, "Few-shot palmprint recognition via graph neural networks," *Electron. Lett.*, vol. 55, no. 16, pp. 890–892, Aug. 2019.

[153] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[154] S. S. L. Oskouei, H. Golestani, M. Kachuee, M. Hashemi, H. Mohammadzade, and S. Ghiasi, "GPU-based acceleration of deep convolutional neural networks on mobile platforms," in *Proc. Distrib. Parallel Clust. Comput*, 2015, pp. 1–7.

[155] Y. Deng, "Deep learning on mobile devices: A review," in *Proc. Mobile Multimedia/Image Process., Secur., Appl.*, May 2019, Art. no. 109930.

[156] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum, M. Wu, L. Xu, and L. Van Gool, "AI benchmark: All about deep learning on smartphones in 2019," 2019, *arXiv:1910.06663*. [Online]. Available: http://arxiv.org/abs/1910.06663

[157] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on Android smartphones," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[158] V. Kanhangad, A. Kumar, and D. Zhang, "Contactless and pose invariant biometric identification using hand surface," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1415–1424, May 2011.

[159] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "GANerated hands for real-time 3D hand tracking from monocular RGB," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 49–59. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/

[160] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," 2016, *arXiv:1605.04711*. [Online]. Available: http://arxiv.org/abs/1605.04711

ADRIAN-STEFAN UNGUREANU received the B.S. and M.S. degrees in electrical engineering from the Politechnica University of Bucharest, in 2012 and 2014, respectively, and the Ph.D. degree in electrical and electronics from the National University of Ireland, Galway, in 2020. His field of work is image processing techniques and biometric technologies used in the context of consumer devices.

SAQIB SALAHUDDIN (Associate Member, IEEE) received the B.Sc. degree (Hons.) in computer engineering from the COMSATS Institute of Information Technology, Abbottabad, Pakistan, in 2007, the M.Sc. degree in microelectronics systems design from the University of Southampton, U.K., in 2009, and the Ph.D. degree in electrical and electronics from the National University of Ireland, Galway, Ireland, in 2019.

PETER CORCORAN (Fellow, IEEE) is a Personal Chair at the College of Engineering and Informatics, NUI Galway. In addition to his academic career, he is also an occasional entrepreneur, an industry consultant, and a compulsive inventor. His research interests include biometrics and consumer electronics. He is the coauthor of 250+ technical publications and a co-inventor on c.250 granted U.S. patents. He is a Board Member of the IEEE Biometrics Council. He is the Editor-in-Chief of the *IEEE Consumer Electronics Magazine*.