IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 17, 2021, accepted July 6, 2021, date of publication July 8, 2021, date of current version July 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095666

Deep Learning for Vein Biometric Recognition on a Smartphone

RAUL GARCIA-MARTIN AND RAUL SANCHEZ-REILLO, (Senior Member, IEEE)
Electronic Technology Department, University Carlos III of Madrid, 28911 Leganés, Spain
Corresponding author: Raul Garcia-Martin (raulgarc@ing.uc3m.es)

ABSTRACT The ongoing COVID-19 pandemic has pointed out, even more, the important need for hygiene contactless biometric recognition systems. Vein-based devices are great non-contact options although they have not been entirely well-integrated in daily life. In this work, in an attempt to contribute to the research and development of these devices, a contactless wrist vein recognition system with a real-life application is revealed. A Transfer Learning (TL) method, based on different Deep Convolutional Neural Networks architectures, for Vascular Biometric Recognition (VBR), has been designed and tested, for the first time in a research approach, on a smartphone. TL is a Deep Learning (DL) technique that could be divided into networks as feature extractor, i.e., using a pre-trained (different large-scale dataset) Convolutional Neural Network (CNN) to obtain unique features that then, are classified with a traditional Machine Learning algorithm, and fine-tuning, i.e., training a CNN that has been initialized with weights of a pre-trained (different large-scale dataset) CNN. In this study, a feature extractor base method has been employed. Several architecture networks have been tested on different wrist vein datasets: UC3M-CV1, UC3M-CV2, and PUT. The DL model has been integrated on the Xiaomi© Pocophone F1 and the Xiaomi© Mi 8 smartphones obtaining high biometric performance, up to 98 % of accuracy and less than 0.4 % of EER with a 50-50 % train-test on UC3M-CV2, and fast identification/verification time, less than 300 milliseconds. The results infer, high DL performance and integration reachable in VBR without direct user-device contact, for real-life applications nowadays.

INDEX TERMS Vein biometric recognition, smartphone, deep learning, convolutional neural network (CNN), machine learning, transfer learning, artificial intelligence, contactless wrist vascular database, neural network as feature extractor, biometrics on mobile devices.

I. INTRODUCTION

In the current worldwide pandemic, caused by the SARS-CoV-2 virus, the transmission of COVID-19 disease can even occur through indirect contact between an uninfected person and an object used by an infected person [1]. In this sense, non-contact multi-user systems provide hygienic alternatives, thereby helping to prevent the propagation of this disease.

In the security world, for access control and payments, contactless biometric recognition systems are effective solutions in terms of reliability, robustness, comfort, but even more important, hygiene. There are several non-contact biometric modalities: facial, iris, contactless fingerprint, gait, vascular, etc.

Two main systems define the Vascular or Vein Biometric Recognition (VBR) industry: Fujitsu© PalmSecure, patent US 2005/0148876 A1 [2], and Hitachi© Finger Vein Authentication, patent US 2011/0222740 A1 [3]. The former is a contactless palm vein system and the latter is a contact finger system. In light of these patents, there are two other VBR modalities in the research world: hand dorsal vein and wrist vein. Considering the latter, previous research [4] described a complete contactless wrist VBR system suitable for access control and forensic applications. Additional systems have been reported [5] and [6], however, they require physical contact between the user and the device.

Another example of non-contact wrist VBR, embedded into a smartphone, in this case, was recently stated in [7] for use in future online payments, bank account access, and screen unlocking.

A. MOTIVATION

Vascular or Vein Biometric Recognition (VBR) is one of the most respectful among contactless biometric variants in terms

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

of user privacy (e.g. contrasted with facial or iris recognition) but, unfortunately, has not been well-integrated into daily life yet. It also offers a comfortable user-device interaction as well as a robust recognition technique, useful in preventing spoof attacks. Veins are internal tissues that cannot be captured or stolen in a non-cooperative way.

The above factors, combined with the current increased demand for non-contact systems due to the risk of COVID-19 transmission and the lack of real-life VBR solutions, are the motivation behind this work.

B. CONTRIBUTIONS

In this study, a novel Transfer-Learning-based wrist system for contactless VBR is embedded into smartphones for the first time in a research approach.

The supervised trained Deep Learning (DL) model, designed to verify and identify people based on wrist vascularity, employs a pre-trained Convolutional Neural Network (CNN) to obtain unique features that are then classified with a traditional Machine Learning algorithm. This technique, known as Transfer Learning (TL) using Neural Networks (NN) only as feature extractors, has been revealed instead of following the current biometric DL trend which is to train a CNN from scratch. The latter is usually a more complex and time-consuming research process that obtains similar results to TL in terms of recognition performance. Also, the use of CNNs as a feature extractor is a solution for the reduced number of images per user class in the vascular biometric State-of-the-Art datasets.

To demonstrate this hypothesis, not only a novel algorithm based on networks as arbitrary feature extractors has been revealed, but also this model has been integrated into two smartphone devices to prove the real-life application and evaluate the vascular biometric and processing performance.

The smartphones designed by Xiaomi Inc.©, Xiaomi© Pocophone F1 (Fig. 1 a) and Xiaomi© Mi 8 (Fig. 1 b), are used as whole capture, processing, and storage hardware to complete the VBR system. Both devices mount a near-infrared camera originally used to unlock them with facial recognition.

The results, discussed according to the ISO/IEC 19795-1 standard [8], are obtained on a unique contactless wrist database captured on smartphones, UC3M-CV2 [7]. To compare them with the current State-of-the-Art, other non-contact and contact datasets are employed, UC3M-CV1 [4] and PUT [9], respectively.

C. RELATED WORK

Although this study presents a wrist solution, the current VBR State-of-the-Art that relies on the most recent infrared-images-based studies has been analyzed not only for the wrist vein modality, but also reinforced with palm, finger, and hand dorsal vein variant studies. This section is divided into: acquisition, storage, and processing hardware; recognition algorithms; and existing datasets.

FIGURE 1. Smartphones used for transfer learning VBR integration (image capture, processing, and storage): Xiaomi© devices. (a) Xiaomi© Pocophone F1. (b) Xiaomi© Mi 8.

1) ACQUISITION, STORAGE, AND PROCESSING HARDWARE

In the last year 2020, a reduced number of non-contact acquisition systems were presented: [4] and [10]. The former is a portable wrist-based system composed of a modified near-infrared camera, a near-infrared illumination designed by the authors, and a reduced-size computer. It includes a static-positioning algorithm for user guiding, TGS-CVBR®. Using this system, an image is acquired for each user interaction.

The latter reveals an on-the-fly acquisition method using 4 low-cost near-infrared commercial cameras, a rectangular grid LED light, and a diffusing glass for finger vein recognition. A quick user's hand movement (1-3 seconds) over the system and a video capture ensure the right acquisition.

Jhong *et al.* [11] present a non-contact hand palm vein solution based on a CNN recognition algorithm that is mention in the following section.

The hardware details of these systems are shown in Table 1.

Later in 2020, a capture, storage, and processing system was embedded into smartphones [7] for non-contact wrist VBR. The employed devices, as mentioned previously, mounted a near-infrared and LED illuminator used originally for facial recognition. The same static-positioning algorithm stated in [4] was implemented.

The current work follows this contactless hardware research line that permits the integration of the acquisition, storage, and recognition algorithm (a novel Deep Learning model in this case) into the same device.

Taking into account the motivation behind this work and analyzing these studies and the current industry trends it is probably not wrong to conclude/predict that contactless interaction has come to stay.

TABLE 1. State-of-the-art contactless hardware summary for VBR.

Study	Year	Vascular modality	Camera	Illumination	Processing & storage	Dataset generated
Garcia-Martin et al. [4]	2020	Wrist	USB Logitech® HD Webcam C525 (unknown sensor)	Own PCB with 8 OSRAM© SFH 4715 A (850 nm)	Raspberry® Pi 4 Model B	UC3M-CV1 (own)
Kuzu et al. [10]	2020	Finger	4 PiNoIR-V2 cameras (Sony IMX219 sensor)	20 OSRAM© SFH 4356-UV grid (850 nm) + white diffusion glass	Raspberry® Pi 3 V2	Own (video acquisition)
Jhong et al. in [11]	2020	Hand palm	Low-cost and low-resolution camera (N/A camera and sensor)	LED array (940 nm, N/A)	PC	Own
Proposed	**2021**	**Wrist**	**Xiaomi© Pocophone F1 and Mi 8 facial IR cameras (OmniVision© OV7251 sensor)**	**1 LED (unknown)**	**Xiaomi© Pocophone F1 and Xiaomi© Mi 8**	**UC3M-CV2 (own, smartphones)**

2) RECOGNITION ALGORITHMS

In the last 5-10 years since Deep Learning algorithms for pattern recognition took off, most of the research studies in VBR have relied on DL algorithms, frequently replacing traditional Machine Learning. However, DL has not been applied for the wrist vein recognition variant. Thus, DL technique analysis for the other vascular modalities has been necessary in order to introduce these algorithms for the first time in a wrist research approach.

As the previous work [10] reveals, Deep Convolutional Neural Networks are the main object of study in finger vein recognition. Different CNN architectures have been implemented and trained from scratch with high authentication (Equal Error Rates, EER, under 5 %) and identification (Correct Identification Rates, CIR, higher than 95 %) performance models: VGG16 [12], ResNet50 and ResNet101 [13], DenseNet [14], designed by authors [15], among others.

Huang *et al.* [12] present a model inspired by VGG16 CNN adapting the first convolutional layer to the finger vein image size (Region Of Interest = 128 × 128) because this well-known architecture, based on 3 × 3 Convolutional + Activation (ReLU) layer blocks, has a 224 × 224 input layer.

As Table 2 summarizes, deeper architectures that rely on micro-architectures, such as ResNet50/ResNet101 or DenseNet161, have been applied by Kim *et al.* [13] and Song *et al.* [14], respectively. ResNet uses the residual module which obtains information (reference) from previous convolutional (not consecutive) layers, allowing for deeper training (and thus higher classification accuracy). A similar architecture goal but different reference configuration was introduced by Huang *et al.* [16] with DenseNet.

Moving on to the palm vein modality it is more difficult (in hand dorsal vein even more) to discover DL-based solutions. A reduced number of CNN architectures have been tested. Jhong *et al.* [11] presented a VGG16-inspired solution using contactless images (acquired by the authors) previously enhanced with the CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm [17]. The preprocessing step (apart from ROI extraction) is not very common in CNN solutions due to the high recognition performance achieved with raw images, but it could be optimal if the images have poor quality or the ROI has a reduced size. In the current work, the CLAHE algorithm is also tested and compared with raw images. Obayya *et al.* [18] present a contactless palm solution using the CASIA Multispectral Palmprint Image database [19] for CNN training and testing. A CNN architecture is designed and trained by the authors using a Bayesian optimization to find the optimal network structure and its parameters.

Finally, in wrist VBR, no DL solutions had been revealed prior to the current study. In terms of contributing to the wrist research, as has been previously pointed out in the contributions section (Section I-B), a Deep Learning technique is revealed for the first time. This DL algorithm is not just a novelty for this modality but also for the entire VBR research because, as can be extracted from the current analysis, the CNN architectures are trained from scratch instead of using Transfer Learning. Vein recognition State-of-the-Art networks are trained with infrared images, allowing them to extract unique features and classify them applying the desired user labels. Despite the fact that TL has been previously used in the research world for the development of biometric recognition algorithms, as far as is known, only [20], a hand-dorsal-vein work reveals a Transfer Learning solution (both variants, CNN as feature extractor and CNN fine-tuning) with pre-trained (in ImageNet

TABLE 2. State-of-the-art deep learning systems for VBR.

Study	Year	Vascular modality	Dataset	DL technique	Train-test (%)	Biometric performance
Huang et al. [12]	2017	Finger	Own (train) FVRC2016 (test)	VGG16	99-1	EER = 0.39 %
Kim et al. [13]	2018	Finger + Finger-shape	SDUMLA-HMT PolyU (version 1)	ResNet50 and ResNet101	50-50	EER = 2.34 % EER = 0.79 %
Song et al. [14]	2019	Finger	SDUMLA-HMT PolyU (version 1)	DenseNet161	50-50	EER = 2.35 % EER = 0.33 %
Jhong et al. [11]	2020	Hand palm	Own (private)	VGG16	90-10	TPIR = 96.54 %
Obayya et al. [18]	2020	Hand palm	CASIA Multispectral Palmprint	Own CNN architecture	80-20	EER = 0.07 % TPIR = 99.40 %
Al-Johania et al. [20]	2019	Hand dorsal	Bosphorus and Dr. Badawi + ImageNet	Transfer Learning (feature extractor and fine-tuning): AlexNet, VGG16,and VGG19	80-20	TPIR = 99.25 % TPIR = 100.00 %
Proposed	**2021**	**Wrist**	**UC3M-CV1, UC3M-CV2, and PUT + ImageNet**	**Transfer Learning (feature extractor): VGG16, VGG19, ResNet50, and ResNet152**	**50-50**	**EER = 0.38 % TPIR = 98.67 % EER = 0.78 % TPIR = 97.67 % EER = 0.78 % TPIR = 97.67 %**

TABLE 3. Wrist VBR datasets.

Dataset	Subjects	Wrists	Samples	Sessions	Images	Year	Contactless
Singapore (NIR) [25]	150	2	3	N/A	900	2007	NO
UC3M [26]	121	1 (right)	5	1	605	2011	NO
PUT [9]	50	2	4	3	1200	2011	NO
Raghavendra et al. [5]	50	2	5	2	1000	2016	NO
FYOWV [24]	160	2	1	2	640	2020	NO
UC3M-CV1 [4]	50	2	6	2	1200	2020	YES
UC3M-CV2 [7]	50	2	12	2	2400	2020	YES (smartphones)

dataset [21]) AlexNet, VGG16, and VGG19 architectures. Kuzu *et al.* [22] present something similar to the fine-tuning (TL) technique, clearly defined by Rosebrock in [23], with a pre-trained (ImageNet dataset [21]) DenseNet161 architecture.

The TL concept "*proposes a different training paradigm*" [23] and is exhaustively explained and discussed in detail in this work, in particular in Section II.

3) DATASETS

Finally, in order to conclude the State-of-the-Art analysis, Table 3 indicates the existing wrist vein databases and their main features. To compare the biometric performance, only the existing wrist datasets are shown and analyzed not considering the rest of the VBR modalities.

It is worth noting that PUT [9] and FYO [24] (FYOWV part), as far as is known, are the only existing public

datasets: Singapore [25] and UC3M [26], collected respectively in 2007 and 2011, are privately-distributed and [5], UC3M-CV1 [4] and UC3M-CV2 [7] are, at this time, private datasets.

Table 3 infers that only the 2020 recent databases, UC3M-CV1 [4] and UC3M-CV2 [7], contain contactless images.

Since the PUT dataset was collected in 2011, the number of images per class (unique wrist user) is acceptable considering the previous datasets, but quite small applying the general DL rule of thumb of 1000-5000 examples per class described in [23] by A. Rosebrock. UC3M-CV2 contains the highest number of unique wrist user images, 24 samples (6 samples $\times$ 2 sessions $\times$ 2 devices $=$ 24 images), with a total of 2400 images (50 subjects $\times$ 2 wrists $\times$ 6 samples $\times$ 2 sessions $\times$ 2 devices $=$ 2400 images). The limited training data (samples per class) in DL solutions supposes the most challenging and persistent problem in all VBR variants because the interclass features are very similar and therefore difficult to distinguish. In this sense, this issue is analyzed in the next sections and the proposed recognition Transfer Learning CNN algorithm is put through its paces not only over the UC3M-CV2 datasets but also over UC3M-CV1, and PUT.

II. TRANSFER LEARNING

TL is a Deep Learning technique that takes a pre-trained Neural Network model as a starting point. This pre-trained NN is then modified and trained with the data of interest which may be completely different from that used in the pre-training.

For CNNs, there are two ways of applying TL:

1) CNN as a feature extractor.
2) Fine-tuning.

A. CNN AS FEATURE EXTRACTOR

This TL variant relies on using the pre-trained weights from the shallowest layers of the CNN architecture to acquire unique features from the images. Then, a traditional Machine Learning algorithm is applied to these features in order to precisely classify/recognize the images. The pre-training using some massive datasets like ImageNet, which consists of 1000 different object classes with over 1.2 million images, have demonstrated the viability of this TL technique with excellent results in the computer vision world.

In this way, the traditional biometric scheme of ''feature extraction + feature comparison'' stays the same as before the DL irruption using the CNNs as end-to-end image classifiers.

Fig. 2 shows the comparison between a CNN as a feature extractor and a CNN as an end-to-end classifier. The CNN architecture shown is VGG16, as it is one of the structures studied in this work.

The VGG16 original architecture (Fig. 2 a) consists of:

- 2 blocks of 2 convolutional layers (3×3) + 1 max pooling layer ($C_1 + C_2 + MP_1$ and $C_3 + C_4 + MP_2$).

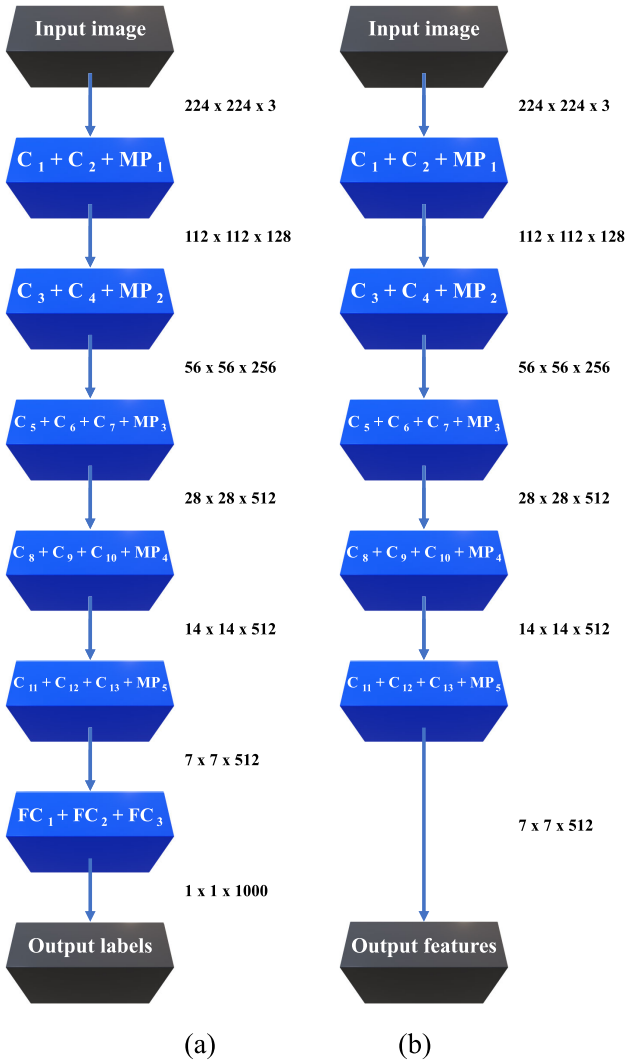

FIGURE 2. Transfer learning using the VGG16 network architecture as feature extractor: (a) Original VGG16 end-to-end image classifier with the 1000 ImageNet output class labels. (b) VGG16 as feature extractor ($7 \times 7 \times 512 = 25088$ unique features).

- 3 blocks of 3 convolutional layers (3×3) + 1 max pooling layer ($C_5 + C_6 + C_7 + MP_3$, $C_8 + C_9 + C_{10} + MP_4$ and $C_{11} + C_{12} + C_{13} + MP_5$).
- 3 fully-connected layers ($FC_1 + FC_2 + FC_3$).

The VGG16 network as a CNN feature extractor (Fig. 2 b) follows the same structure but the propagation is stopped before the final 3 fully-connected layers that are obviated, obtaining $7 \times 7 \times 512 = 25088$ unique features instead of the probabilities for each class label. Directly using the weights previously obtained when pre-training this CNN architecture in a massive dataset, like ImageNet, the image features of the desired dataset are extracted. Then, they are classified with a traditional Machine Learning algorithm completing the recognition model.

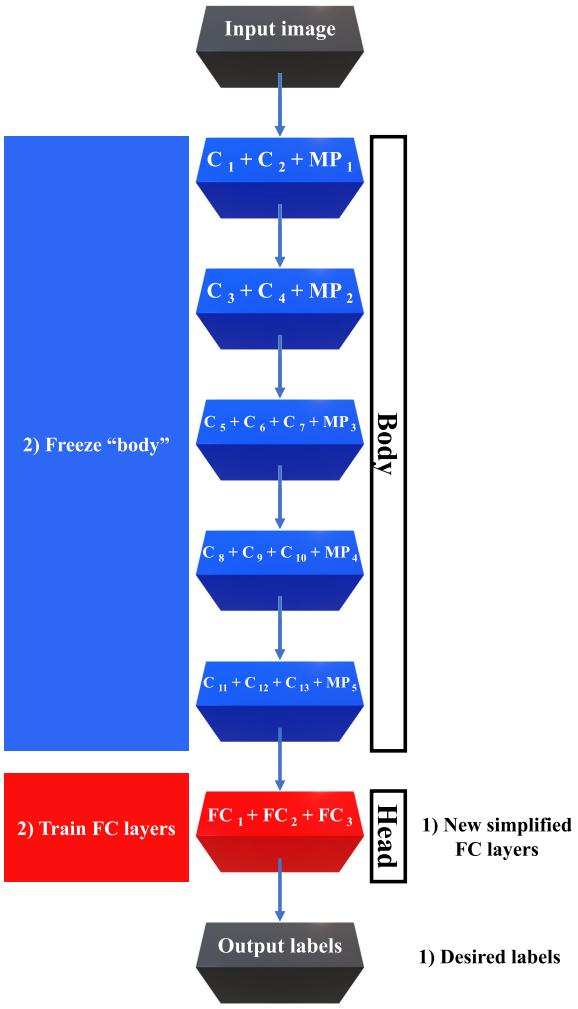

FIGURE 3. Transfer learning using the fine-tuning process for the VGG16 architecture, Step 1 and 2: add the new simplified FC layers and freeze the CNN "body" training only the new FC layers.

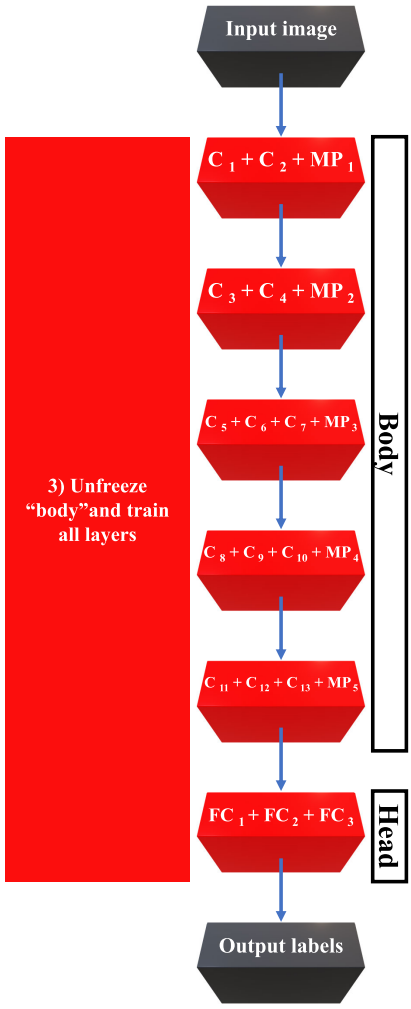

FIGURE 4. Transfer learning using the fine-tuning process for the VGG16 architecture, Step 3: unfreeze the CNN "body" training the entire architecture.

B. FINE-TUNING

Instead of using a CNN as a feature extractor, fine-tuning as a TL method proposes maintaining the CNN as an end-to-end classifier. For this purpose, a pre-trained CNN is modified (deeper layers or "head" of the network) and trained by parts again with the desired images.

To implement this technique the next steps are usually followed (Fig. 3 and Fig. 4):

1) Replace the final fully-connected layers ("head" of the network) with other simplified fully-connected layers. These new layers are randomly initialized (just like any other layer in a new network). The previous layer, the max polling MP5 in the case of the VGG16 network shown in Fig. 3, is treated as the output of a feature extractor.
2) Train the new architecture by "freezing" the parameters of the previous-to-the-head layers (the "body"), i.e., not back propagating through these layers in order to avoid destroying the rich feature filters previously learned. The new fully-connected layers are randomly initialized, so that this part of the network is "foolish" at this stage.
3) After the network "head" has started to learn patterns in the dataset of interest, the training is paused, the "body" is unfrozen allowing the entire network back propagation, and the training is resumed with a reduced learning rate (it is not usually desirable to dramatically modify the pre-trained convolutional filters) in order to obtain, if it is necessary, higher accuracy.

III. VEIN BIOMETRIC RECOGNITION SYSTEM

Fig. 5 shows the implemented DL system with the Transfer Learning model completely embedded into a smartphone and following the ISO/IEC 19795-1 standard [8]. It is divided into 5 subsystems:

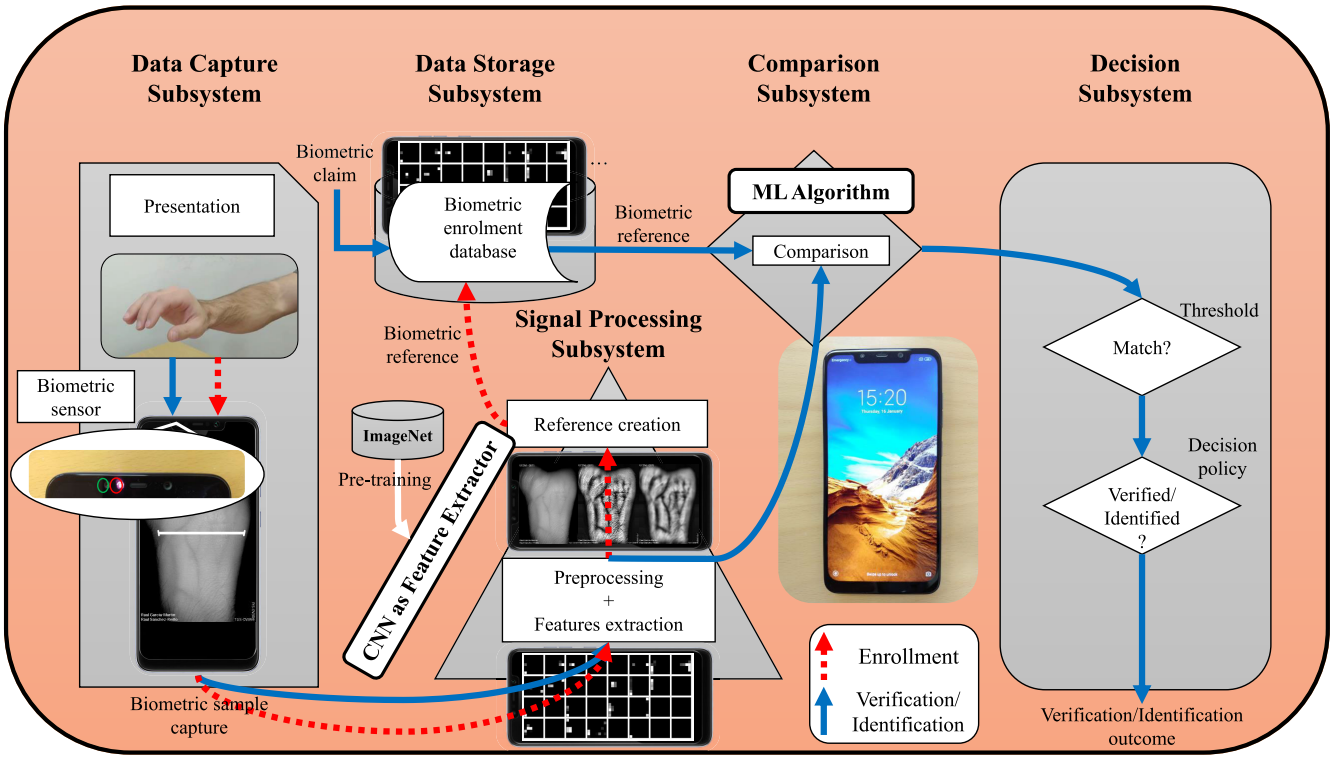

FIGURE 5. Components of the designed transfer learning VBR system embedded into a smartphone.

1) Data Capture Subsystem: The first step in the verification/identification task is to present and capture the biometric sample. The wrist vascular presentation is acquired using the biometric sensor: the near-infrared camera, and the near-infrared light, both originally integrated into the smartphone for facial recognition. The capture and guiding algorithm software, TGS-CVBR®, presented in [7] is in charge of this task.
2) Signal Processing Subsystem: The captured images are preprocessed or not (both cases have been tested) and the designed TL model obtains the unique discriminating features using 4 different CNN structures. Fig. 5 shows the CNN as a feature extractor pre-trained with the ImageNet dataset.
3) Data Storage Subsystem: If the system is working in enrollment mode the features are stored in the data storage subsystem: the internal memory of the smartphone.
4) Comparison Subsystem: As has been discussed, after applying this TL solution a Machine Learning algorithm is in charge of the feature comparison. For the CNNs trained from scratch this block is part of the network (fully-connected layers).
5) Decision Subsystem: According to the values obtained in the previous subsystem, the verification/identification is taken.

A. INTEGRATED BIOMETRIC HARDWARE

The only devices used in the designed system as image capture, processing, and storage hardware, are two smartphones designed by Xiaomi Inc.© : Xiaomi© Pocophone F1 (Fig. 1 a) and Xiaomi© Mi 8 (Fig.1 b). Both of them were used in [7] to collect the UC3M-CV2 dataset. All their main features are detailed in [7] and summarized in the following points.

1) CAPTURE: BIOMETRIC SENSOR

The unknown front near-infrared camera and a near-infrared LED (Light Emitting Diode) light integrated into the screen-notch of the smartphone is the acquisition hardware of the system. This ToF (Time-of-Flight) technology was originally used for facial recognition in these devices. It is worth pointing out that the infrared LED light that emits a wide spectrum of infrared light (around 960 nm) is still unknown, however, it has been discovered that the camera mounts an OmniVision© OV7251 sensor [28]. The light spectrum response of this sensor, shown in Fig. 6, infers that it is probably not wrong to think that its lens mounts a visible-light-block filter.

2) PROCESSING

The hardware in charge of the real-time-image processing is the computing unit (CPU) Qualcomm® Snapdragon SDM845 (Octa-Core, 2.8 GHz) and the graphical unit (GPU) Qualcomm® Adreno 630 (710 MHz) with Android 9 Pie OS.

3) STORAGE

The 64 GB internal memory of the smartphone represents the storage data subsystem.

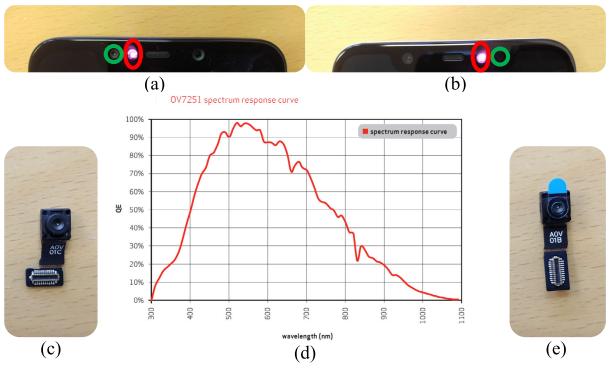

FIGURE 6. Image capture hardware used for the VBR integration: (a) Xiaomi© Pocophone F1 near-infrared camera (surrounded in green) and near-infrared LED (surrounded in red) embedded into the left side of the notch. (b) Xiaomi© Mi 8 near-infrared camera (surrounded in green) and near-infrared LED (surrounded in red) on the right side of the notch. (c) Xiaomi Pocophone© F1 near-infrared camera (non-mounted replacement). (d) OmniVision© OV7251 light spectrum response [28]. Quantum Efficiency (QE) vs. wavelength (nm). The OV7251 camera sensor is mounted on both devices. (e) Xiaomi© Mi 8 near-infrared camera (non-mounted replacement).

B. DATABASES

This study bases its experiments and results on the contactless UC3M-CV2 dataset collected in [7]. Thus, following this smartphone solution line research, the Deep Learning model revealed and detailed in the next sections has been integrated into the same UC3M-CV2 smartphones. However, in order to compare the robustness of the proposed model, another non-contact dataset, UC3M-CV1 [4], and another contact dataset, PUT [9], have been tested. Also, as could not be otherwise, the dataset for the pre-training of the Transfer Learning model, ImageNet [21], is detailed.

1) UC3M-CV2

The UC3M-CV2 (UC3M-Contactless Version 2) consists of 2400 wrist vein infrared images captured in the last year 2020 using the Xiaomi© Pocophone F1 and Xiaomi© Mi 8 devices in a non-contact user-device interaction. The register software algorithm, TGS-CVBR®, was in charge of leading the user through wrist positioning and capturing the samples.

The greyscale images (8 bit/pixel monochromatic images with values from 0, black, to 255, white) with 640 × 480 resolution were collected in JPEG (Joint Photographic Experts Group)/JFIF (JPEG File Interchange Format) compress format.

Table 3 and the following equation (1) summarize the values of this dataset:

$$50\ subjects \times 2\ wrists \times 6\ samples \times 2\ sessions = 1200\ images \quad (1)$$

2) UC3M-CV1

The UC3M-CV1 (UC3M-Contactless Version 1) was previously collected in 2020 using a modified webcam, USB Logitech® HD Webcam C525 (unknown sensor), as a near-infrared camera and a near-infrared illumination designed by the authors, PCB (Printed Circuit Board) with 8 OSRAM© SFH 4715 A (850 nm) LEDs. This contactless system [29] captured the greyscale (8 bit/pixel) infrared images with 640 × 480 resolution and JPEG/JFIF compress format.

This database contains 1200 images collected in two different sessions. Table 3 and the following equation (2) summarize the values of this database:

$$50\ subjects \times 2\ wrists \times 6\ samples \times 2\ sessions \times 2\ devices = 2400\ images \quad (2)$$

3) PUT

This single existing public wrist VBR database was published also in 2011. The contact system proposed by the authors (Kabacinski and Kowalski [9]), consisting of a USB camera and 850 nm LED light (unknown devices), captured 1200 infrared images from 50 users in 3 sessions. The greyscale (24 bit/pixel) 1024 × 768 images were stored in BMP format.

Table 3 and the following equation (3) summarize the values of this database:

$$50\ subjects \times 2\ wrists \times 4\ samples \times 3\ sessions = 1200\ images \quad (3)$$

4) ImageNet

Finally, the ImageNet dataset, revealed in 2009 by Deng *et al.* [21], is the most popular set of data in the computer vision world, due to its huge size and the well-known recognition challenge, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [30]. At the time of the current work, this massive dataset consists of over 14 million RGB images with more than 1000 possible object categories. The images go from any kind of animal to all types of vehicles.

It is not surprising that this gigantic amount of data is an excellent starting ground to pre-train the designed and revealed TL model in order to extract unique image features.

C. PREPROCESSING

Usually, since the DL irruption, no preprocessing method has been considered necessary to preprocess vein images and isolate the vascular patterns due to the excellent performance guaranteed by the CNNs in this mission: to extract discriminating features. Only ROI extraction is usually applied. For DL biometric solutions, this process resizes (reduces) the image to fit it into the input layer of the CNN architecture. All the tested datasets in this work consist of 640 × 480 (VGA resolution) images, except the PUT dataset that presents 1024 × 768 (XGA resolution) images. All images have been resized as described in the feature extraction section.

In an attempt to compare both solutions, preprocessing the vein images or not, before introducing them into the CNN, a contrast increase between the veins and the surrounding living tissue has been performed. The software

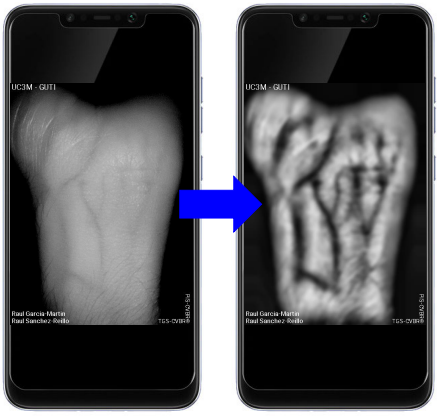

FIGURE 7. Raw vs. Preprocessed (PIS-CVBR®, [7]) sample (User 0 of the UC3M-CV2 dataset) as CNN input image in order to compare the TL model performance.

algorithm sequence followed was presented in [7], PIS-CVBR®. As Fig. 7 shows, this technique combines the CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm, to increase the contrast and a sequence of noise-removal filters.

It is worth pointing out that no image enhancement method has been applied to the datasets.

D. FEATURE EXTRACTION: DEEP CNN TRANSFER LEARNING

Four CNN architectures have been followed independently to extract unique features and create the Transfer Learning models: VGG16, VGG19, ResNet50, and ResNet152. All their weight values have been tangled from the pre-training networks in the ImagenNet dataset. As it has been explained in Section II-A, the "head" of the networks, the 3 final fully-connected layers of VGG16 and VGG19, and the final fully-connected layer of ResNet50, have been removed.

1) VGG16

Fig. 8 a and Table 4 show the network architecture and Fig. 8 b illustrates its output feature maps for each layer. As has been previously mentioned and shown, the VGG family architecture is based on consecutive 3 × 3 Convolutional + Activation (ReLU) layer blocks that end with a max pooling layer to reduce the volume size.

The VGG16 structure proposed to extract unique image features (and also the original) consists of 7 layer blocks:

1) Block 1 (I layer): This is the input layer where the square greyscale (only one channel) images are introduced. The size of the input image is 224 × 224 thus the images from the tested vein datasets have been reduced to the required square size maintaining the original aspect ratio using the nearest-neighbor interpolation.
2) Block 2 ($C_1 + C_2 + MP_1$): As all the convolutional layers learned in this architecture, C_1 and C_2 layers, are 3 × 3 filters followed by a ReLU activation layer. This well-known non-linear function is described by equation (4).

$$y(x) = max\{0, x\} \tag{4}$$

Each of these layers, C_1 and C_2, is composed of 64 filters (depth) with a stride (pixel step between each movement of the convolutional matrix) of 1 and a zero-padding of 1 (filling 1 extra matrix border with zeros) to retain the original input volume size. The max pooling layer, MP_1, reduces the spatial size of the input volume (i.e., width and height) with a 2 × 2 kernel and a stride of 2 obtaining the following output feature map volume size

$$\begin{aligned} W_O &= \frac{W_I - MP}{S} + 1 = \frac{224 - 2}{2} + 1 = 112 \\ H_O &= \frac{H_I - MP}{S} + 1 = \frac{224 - 2}{2} + 1 = 112 \\ D_O &= D_I = 64 \end{aligned} \tag{5}$$

where:

- W_I and W_O: weight of the input and output volume.
- H_I and H_O: height of the input and output volume.
- D_I and D_O: depth of the input and output volume.
- MP: size of the max polling filter because it has a square shape.
- S: stride

3) Block 3 ($C_3 + C_4$ z MP_2): This block is equal to the previous one but in this case, the depth of the convolutional layers is 128. The max pooling layer, MP_2, reduces the output volume, in the same way, obtaining a 56 × 56 × 128 output feature map.
4) Block 4 ($C_5 + C_6 + C_7 + MP_3$): In this block, an extra convolutional layer is added. The depth of the 3 convolutional layers is 256. The max pooling layer, MP_3, reduces the output volume in the same way, obtaining a 28 × 28 × 256 output feature map.
5) Block 5 ($C_8 + C_9 + C_{10} + MP_4$): This block is equal to the previous one but in this case, the depth of the convolutional layers is 512. The max pooling layer, MP_4, reduces the output volume in the same way, obtaining a 14 × 14 × 512 output feature map.
6) Block 6 ($C_{11} + C_{12} + C_{13} + MP_5$): In this block, an extra convolutional layer is added. The depth of the 3 convolutional layers is 256. The max pooling layer, MP_5, reduces the output volume in the same way, obtaining the final 7 × 7 × 512 output features.
7) Block 7 (OF layer): This is the output features layer. As has been mentioned in order to use this pre-trained (with the ImageNet dataset) CNN as a feature extractor, the fully-connected layers of this architecture are omitted. The CNN produces 7 × 7 × 512 = 25.088 unique features for each dataset image that passes through the pre-trained network. For example, for the UC3M-CV2 dataset, the obtained features

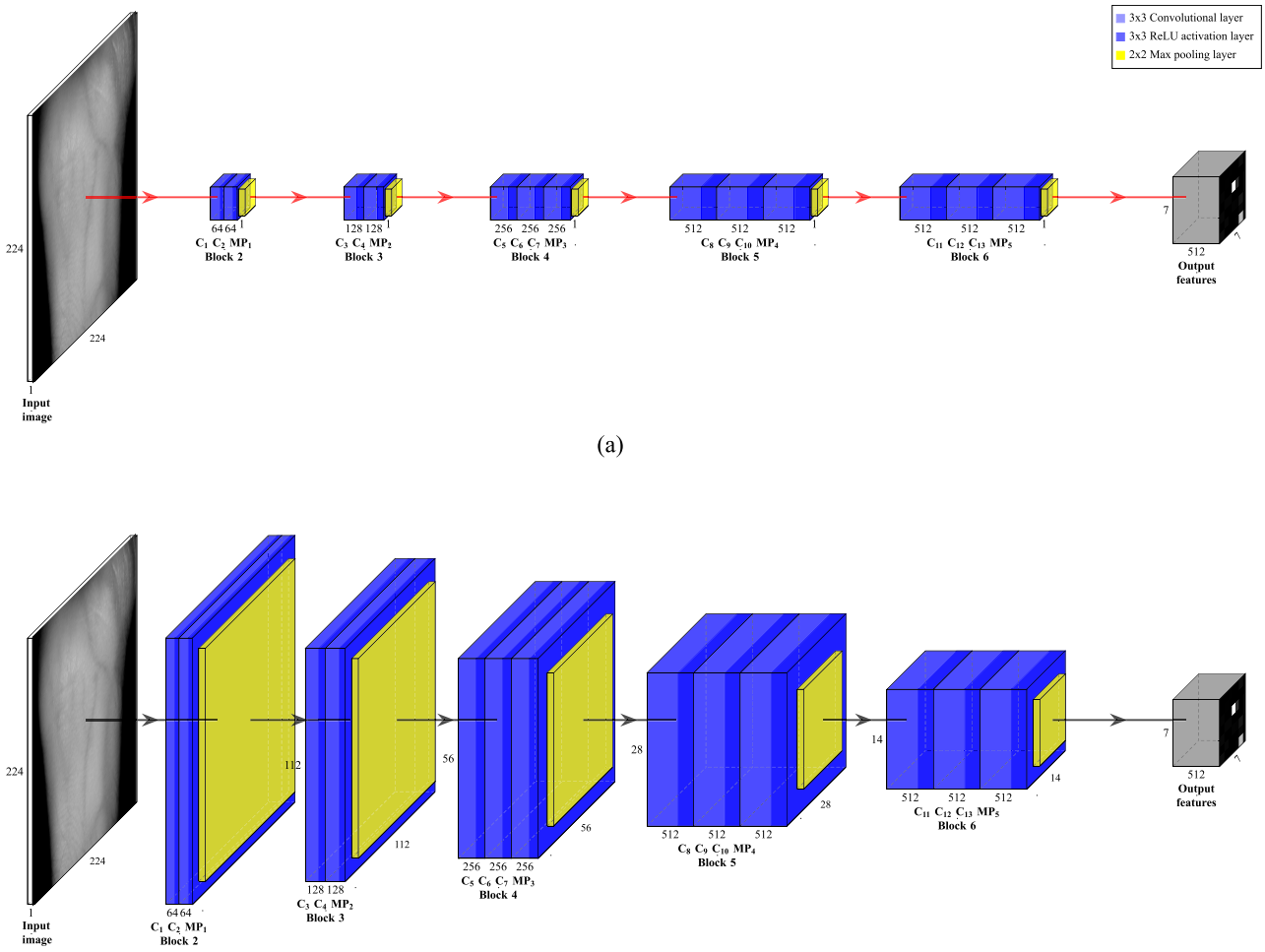

FIGURE 8. Original VGG16 architecture applied for TL. (a) VGG16 architecture as a feature extractor. (b) Output feature map for each layer of VGG16 as a feature extractor.

follow the equation (6).

$$7\ pixels \times\ 7\ pixels\ \times\ 512\ filters \times 2400\ images\ = 60.211.200\ unique\ features\ for\ UC3M-CV2 \quad (6)$$

The huge amount of output features are stored in binary format in an HDF5® file (Hierarchical Data Format version 5) [31] in order to access them easily. The file is dived into 3 sub-datasets:

1) Label names: existing image class label names for the studied datasets. In UC3M-CV2: U000 (User 0), U001 (User 1), U002 (User 2),..., U099 (User 99).
2) Labels: The integer associated with each class and corresponding to every dataset image. In UC3M-CV2, there are 24 zeros (User 0, 24 samples), 24 ones (User 1, 24 samples), 24 twos (User 2, 24 samples),..., 24 ninety-nines (User 99, 24 samples).
3) Features: The extracted features for each image.

Table 5 summarizes the HDF5 file structure for UC3M-CV2.

This pre-trained architecture as feature extractor has been implemented using Python 3.7.9 programming language, TensorFlow© 2.0 (2.3.0 version) open-source library, and its Keras© (1.1.2 version) API.

The processing has been performed using the NVIDIA® GeForce® RTX 2080 Ti (11 GB GDDR6 memory) GPU and the 9th Generation Intel® Core i9k (64-bit, 16 GB of RAM, 3.6 GHz) CPU of the Dell© Alienware Aurora R8 computer with Windows 10 Home OS.

The Cuda® parallel computing platform (version 11.1) has been in charge of performing, broadly speaking, the CPU-GPU communication in order to parallelize the computations.

It is worth noting that the entire TL model, as it is going to be explained in the next session (Section III-D), has been integrated into the smartphones for real-time verification/identification using TensorFlow© Lite library and Android programming language.

TABLE 4. VGG16 architecture for TL.

Block	Layer reference	Layer	Depth	Output feature map size	Kernel size	Stride step size	Number of zero padding
Block 1	I	Input	-	224 x 224 x 1	-	-	-
Block 2	C_1	Convolutional 1	64	224 x 224 x 64	3 x 3	1 x 1	1
	C_2	Convolutional 2	64	224 x 224 x 64	3 x 3	1 x 1	1
	MP_1	Max Pooling 1	1	112 x 112 x 64	2 x 2	2 x 2	0
Block 3	C_3	Convolutional 3	128	112 x 112 x 128	3 x 3	1 x 1	1
	C_4	Convolutional 4	128	112 x 112 x 128	3 x 3	1 x 1	1
	MP_2	Max Pooling 2	1	56 x 56 x 128	2 x 2	2 x 2	0
Block 4	C_5	Convolutional 5	256	56 x 56 x 256	3 x 3	1 x 1	1
	C_6	Convolutional 6	256	56 x 56 x 256	3 x 3	1 x 1	1
	C_7	Convolutional 7	256	56 x 56 x 256	3 x 3	1 x 1	1
	MP_3	Max Pooling 3	1	28 x 28 x 256	2 x 2	2 x 2	0
Block 5	C_8	Convolutional 8	512	28 x 28 x 512	3 x 3	1 x 1	1
	C_9	Convolutional 9	512	28 x 28 x 512	3 x 3	1 x 1	1
	C_{10}	Convolutional 10	512	28 x 28 x 512	3 x 3	1 x 1	1
	MP_4	Max Pooling 4	1	14 x 14 x 512	2 x 2	2 x 2	0
Block 6	C_{11}	Convolutional 11	512	14 x 14 x 512	3 x 3	1 x 1	1
	C_{12}	Convolutional 12	512	14 x 14 x 512	3 x 3	1 x 1	1
	C_{13}	Convolutional 13	512	14 x 14 x 512	3 x 3	1 x 1	1
	MP_5	Max Pooling 5	1	7 x 7 x 512	2 x 2	2 x 2	0
Block 7	**OF**	**Output features**	**-**	**7 x 7 x 512**	**-**	**-**	**-**

2) VGG19

This deeper well-known VGG architecture is similar to the previous one but in this case, the 3 final convolutional blocks consist of 4 convolutional layers instead of 3. Also, the 3 final fully-connected layers are removed extracting $7 \times 7 \times 512 = 25.088$ unique features.

3) ResNet50

This architecture introduced by He *et al.* in their 2015 work, Deep Residual Learning for Image Recognition [32], is considerably deeper than the VGG family architectures, allows the training of networks with depths greater than 50-100 layers. The ResNet CNNs rely on the residual module micro-structure. Fig. 9 shows the most common residual module: the bottleneck.

As is shown, the bottleneck residual model consists of 3 convolutional layers followed by a batch normalization layer and a ReLU activation layer. The output is added to the input of the block (identity or ''residual input'') in an addition node via ''shortcut''. If a traditional layer is defined with the

TABLE 5. Dataset of the VGG16 extracted features. HDF5 file.

	Index	Values
Label names	0	U000
	1	U000
	2	U000
	...	...
	99	U099
Labels	0	0
	1	0
	...	...
	23	1
	24	1
	...	...
	2400	99
Features	0	0., 0., ...
	1	0., 0., ...
	...	0., 0., ...
	2400	0., 0., ...

equation (7),

$$y(x) = f(x) \tag{7}$$

a residual module could be represented as (8)

$$y(x) = f(x) + x, \tag{8}$$

where x is the (''residual'') input.

The size of the 3 convolutional layer is respectively 1×1, 3×3 and 1×1, and their depth is $D = DC_1 = DC_2 = DC_3 / 4$. Due to this volume depth increment, this module is called a ''bottleneck''. The ResNet50 architecture, Fig. 10, consists of 4 different stages or residual modules (Block 3, 4, 5, and 6). The output of each residual stage, the input or identity of the following one, passes through another 1×1 convolutional layer ($+ 1 \times 1$ batch normalization layer) of $DC_3 = DC_4$ and stride 2, as Fig. 10 shows, reducing the volume size. This non-identity shortcut is only applied between stages. The ResNet50 version 1 is the implemented architecture that presents post-activation instead of pre-activation

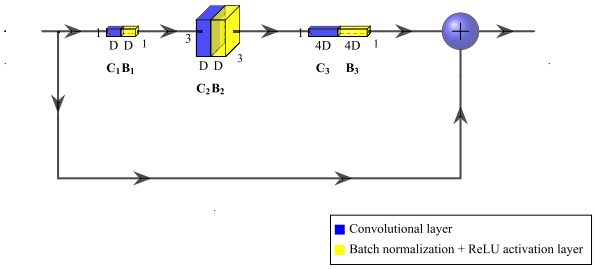

FIGURE 9. Residual ResNet module, the bottleneck. Residual micro-structure applied in the proposed TL configuration and most common residual module. The last ReLU activation layer is performed after the add operation.

(version 2), i.e. the batch normalization and ReLU activation layers are applied after the convolutional layers of the bottleneck.

It is very important to indicate that this architecture, in contrast with the VGG family, doesn't use pooling layers to reduce the weight and height of the feature maps but rather its own convolutional layers applying strides > 2 (except on the layers of the first stage).That is why the ResNet network only has two pooling layers: MP (max pooling) and AP (average pooling). The average pooling layer is not shown in Fig. 10 because the CNN top or head (fully-connected + average pooling layers) has been removed in order to applied TL and obtain the $7 \times 7 \times 2048$ feature output volume.

As Table 6 shows this architecture consists of 7 blocks. The first one is the 224×224 input layer (I) where the vein dataset images, previously resized (640×480 to 224×224 or 1024×768 to 224×224), enter. Block 2 is composed of one convolutional layer (C_1) and one max pooling (MP). Both of them, with a stride of 2, reduce the feature map output to 112×112. Block 3, 4, 5, and 6 (the 4 stages) consist of, respectively, 3, 4, 6, and 3 residual blocks ($C_2 + C_3$ z C_4 to $C_{11} + C_{12} + C_{13}$) with depth 64, 64, and 4 * 64 to 512, 512, and 4 * 512.

Finally, Block 7 is the output layer where the $7 \times 7 \times 2048 = 100.352$ unique features are obtained from this proposed ImageNet pre-trained ResNet50 (version 1) network due to the final fully-connected layer omissions. For example, for the UC3M-CV2 dataset, the obtained features follow the equation (9).

$$\begin{aligned} &7\ pixels \times 7\ pixels \times 2048\ filters \\ &\quad \times 2400\ images = 240.844.800\ unique\ features \\ &\quad for\ UC3M - CV2 \end{aligned} \tag{9}$$

4) ResNet152

ResNet152 (version 1), architecture that belongs to the ResNet family, presents a similar structure to ResNet50 (version 1) although deeper. But in this case, the same size bottleneck residual modules are repeated with a depth of 3, 8, 36, and 3, respectively reaching the 152 layers. From this revealed ImageNet pre-trained ResNet152 network as feature

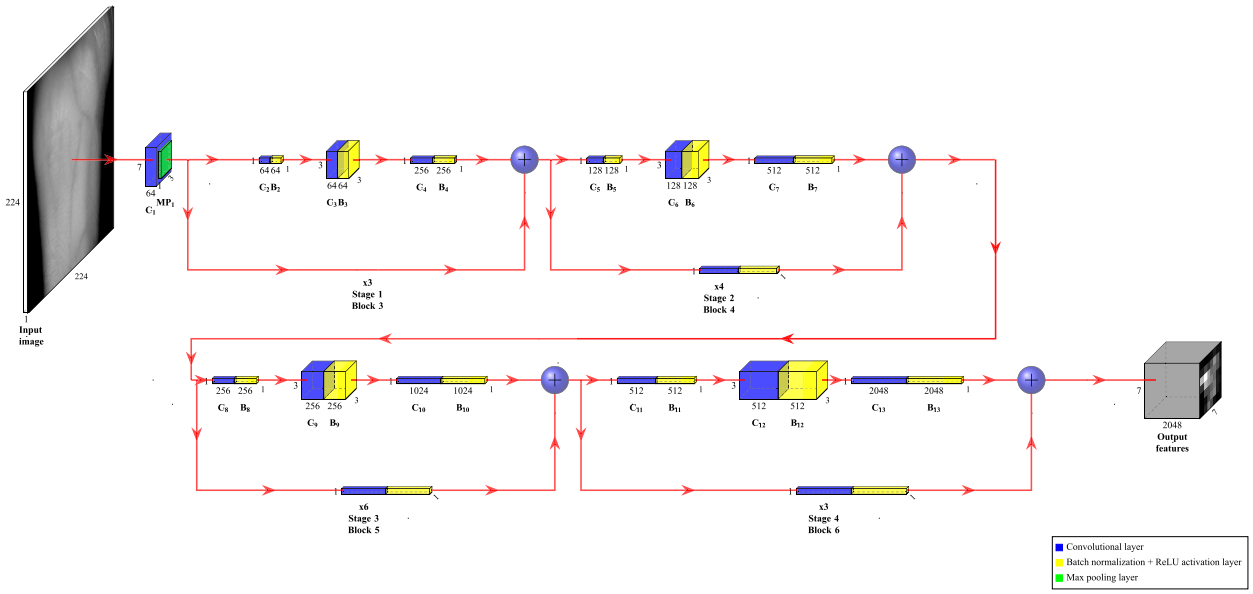

FIGURE 10. Original simplified ResNet50 architecture applied for TL.

extractor, $7 \times 7 \times 2048 = 100.352$ unique features are obtained.

E. FEATURE COMPARISON: A MACHINE LEARNING METHOD

After extracting unique features for each vascular image with the CNN pre-trained architecture, an ML algorithm is applied in order to verify/identify as is shown in the scheme in Fig. 11. This figure summarizes the completed software TL algorithm model designed. First, the model begins with two paths (blue block): one in which input images are preprocessed with PIS-CVBR® and resized to 224×244 (input square size for the preprocessed CNNs) and the second path in which raw images are only resized to 224×244. Both solutions are compared in the experiments and results section.

Then, these resized images pass through one of the pre-trained CNN architectures, and their unique features are extracted. Finally, these singular points are compared with a traditional ML algorithm.

The supervised traditional Machine Learning algorithm applied is Logistic Regression which relies on the Sigmoid function (10):

$$P_\theta = \frac{1}{1 + e^x} \tag{10}$$

where:

- P_θ: Prediction function, is the probability estimated (between 0 and 1).
- x: is the input function learned by the model.

The solver or optimization algorithm used to find the decision boundary to separate the unique vein features of each user is Software for Large-scale Bound-constrained Optimization (L-BFGS-B) [33]. The maximum number of iterations taken to converge is 1000 and the cost function applied in order to evaluate the errors is the cross-entropy loss.

The different image vein feature vectors have been split into the following train-test percentages: 75-25 % and 50-50 %, e.g., in the UC3M-CV2 dataset these percentages represent 1800 images for training and 600 images for testing (18-6 images for each user class), and 1200 training images and 1200 testing images (12-12 images for each user class). As it is also noted in the results section, the 50-50 % rate is considered in this work to be the minimum restrictive value that should be applied for biometric solutions.

In an attempt to obtain a proof of concept of a real-application system, the entire model runs on the smartphones through the TensorFlow© Lite (0.0.0-nightly version) library framework and the developed Android application. The model has been trained and generated using TensorFlow© 2.0 (2.3.0 version) open-source library, its Keras© (1.1.2 version) API, and the previously described processing computer equipment (CPU + GPU). After extracting the unique features and training the Logistic Regression classifier, the model has been stored as a binary HDF5 file and then converted to a binary TFLITE file using TensorFlow© Lite. This framework allows the interpretation of DL models in real-time on embedded devices, as the proposed smartphones.

In Section IV-B the computational time of these real-time processing video tests is collected.

IV. EXPERIMENTS AND RESULTS

In order to evaluate (offline) the proposed model and system, experiments and results have been divided into two possible analyzable features: biometric and computational time performance. Within each of the two characteristics, the section is divided into the verification and the identification tasks.

TABLE 6. ResNet50 (version 1) architecture applied for TL.

Block	Layer reference	Layer	Depth	Output feature map size	Kernel size	Stride step size	Number of zero padding
Block 1	I	Input	-	224 x 224 x 1	-	-	-
Block 2	C_1	Convolutional 1	64	112 x 112 x 64	7 x 7	2 x 2	0
	MP_2	Max Pooling 2	1	56 x 56 x 64	3 x 3	2 x 2	0
Block 3	$C_{2\text{-}4}$	Convolutional 2-4	3	56 x 56 x 256	1 x 1, 64 3 x 3, 64 1 x 1, 256	1 x 1	1
Block 4	$C_{5\text{-}7}$	Convolutional 5-7	4	28 x 28 x 512	1 x 1, 128 3 x 3, 128 1 x 1, 512	2 x 2	1
Block 5	$C_{8\text{-}10}$	Convolutional 8-10	6	14 x 14 x 1024	1 x 1, 256 3 x 3, 256 1 x 1, 1024	2 x 2	1
Block 6	$C_{11\text{-}13}$	Convolutional 11-13	3	7 x 7 x 2048	1 x 1, 512 3 x 3, 512 1 x 1, 2048	2 x 2	1
Block 7	**OF**	**Output features**	**-**	**7 x 7 x 2048**	**-**	**-**	**-**

A. BIOMETRIC PERFORMANCE

1) VERIFICATION

According to ISO/IEC 19795-1, the False Match Rate (FMR) and False Non-Match Rate (FNMR) are reported (mandatory in verification systems) in different Detection Error Trade-Off (DET, recommended) plots for each experiment performed. Although the EER is deprecated according to the standard, this well-known working point has been reported in each experiment. The Failure-To-Enrol Rate (FTER) and the Failure-To-Acquire Rate (FTAR) are unknown for all the datasets. Table 7 summarizes 3 experiments that have been carried out changing 3 different influence variables: preprocessing (raw images vs. images preprocessed with PIS-CVBR®), CNN architecture for feature extraction (VGG16, VGG19, ResNet50, and ResNet152, trained with ImageNet dataset), and train-test dataset split (50-50 % vs. 75-25 %). As was previously mentioned, the train-test dataset split means that for UC3M-CV2, for example, a 50-50 % represents that 1200 images (12 images per user) are applied for training the Logistic Regression classifier and 1200 (12 images per user) for testing it.

1) Preprocessing experiment: First, in order to compare the initial two paths of the model (PIS-CVBR® preprocessing/not preprocessing, Fig. 11, blue block) Fig. 11 shows the DET curve for both solutions and each dataset applying the first architecture of the TL model, VGG16 (Fig. 11, green block) trained with ImageNet. The implemented Logistic Regression classifier process (Fig. 11, yellow block) has been trained and tested in the 3 previously describe wrist vein datasets:

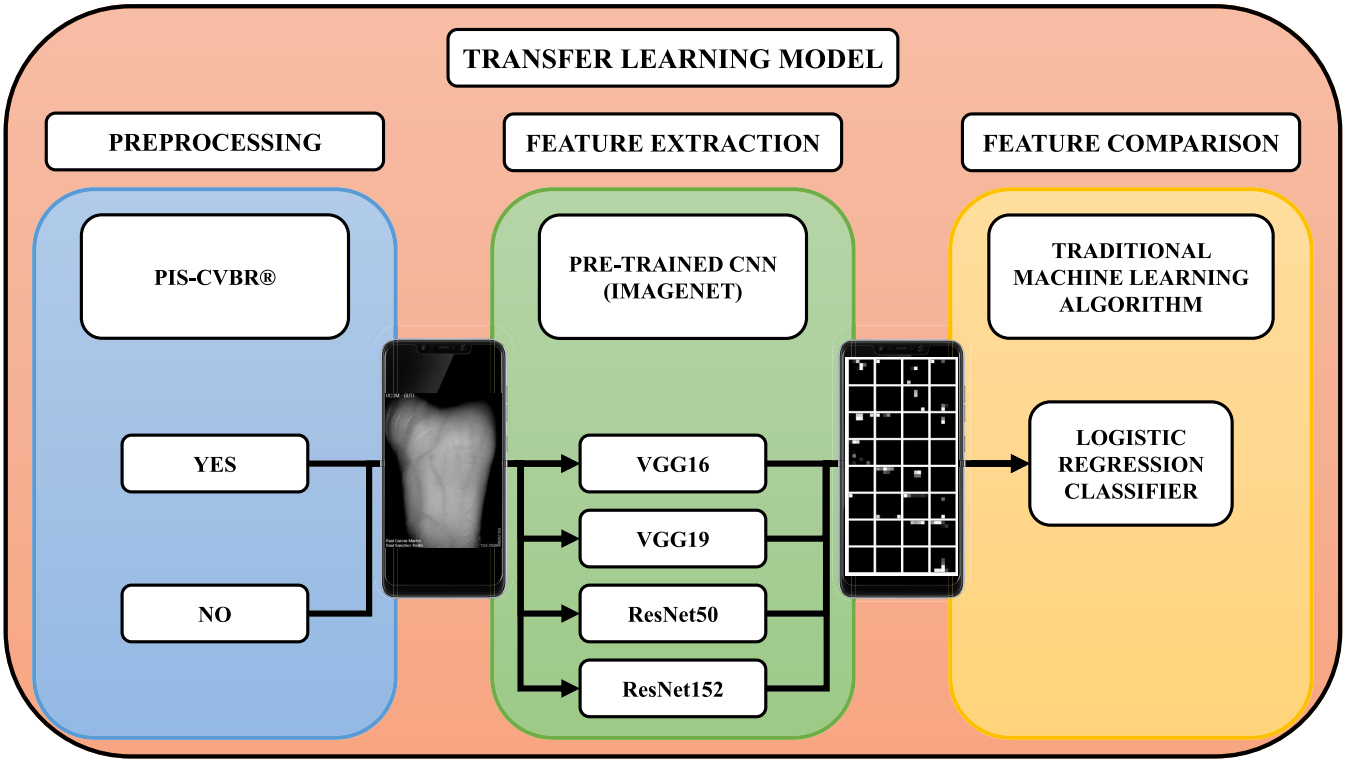

FIGURE 11. Transfer learning proposed model. ImageNet Pre-trained CNN (VGG16, VGG19, ResNet50, or ResNet152) as feature extractor and Logistic Regression classifier as Machine Learning feature comparator. The smartphone screenshots show the intermediate steps for one raw image passed through the VGG16 ImageNet network. The first 32 feature maps (7 × 7) from the TL feature extraction process are shown.

UC3M-CV2, UC3M-CV1, and PUT. For a more robust and demanding test, UC3M-CV1 has been combined with UC3M-CV2 in a single dataset, UC3M-CV1+CV2, due to images belong the same users but with a different capture device. The train-test percentage is 50-50 % in this experiment. The continuous, dot-dot, and line-line curves belong, respectively, the UC3M-CV2, UC3M-CV1+CV2, and PUT dataset. As Fig. 12 infers the best results are obtained respectively in the following order: UC3M-CV2, UC3M-CV1+CV2, and PUT. This fact could be caused due to a higher image illumination quality of the UC3M-CV2 against UC3M-CV1 and a better user wrist position during the capture (TGS-CVBR®) and higher illumination quality of the UC3M-CV2 and UC3M-CV1+CV2 against PUT. All this, despite the fact that PUT is a contact database.

The performance of the model with preprocessed images is only clearly slightly higher for the PUT database than with raw images. It could be hypnotized that this fact is a direct consequence of the remarkable image quality enhancement provided by PIS-CVBR® to the low image quality (improvable illumination and user wrist position) of the PUT dataset (compared with UC3M-CV1 and UC3M-CV2). However, as it is shown in the train-test experiment the CNN architecture as a feature extractor also is a critical factor related to the preprocessing, e.g. ResNet152 obtains better performance with raw PUT images than preprocessed PIS-CVBR® PUT images.

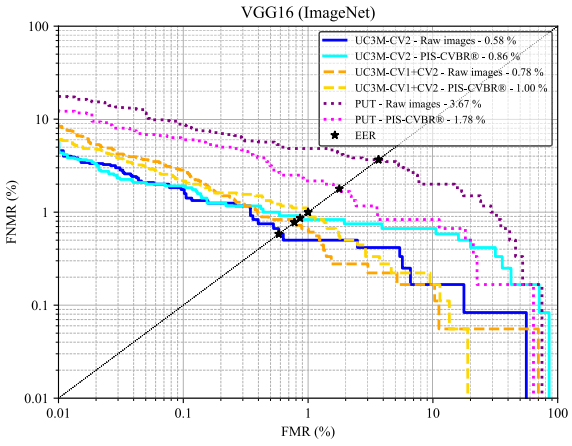

FIGURE 12. Biometric performance: Verification. DET curves for the VGG16 Transfer Learning CNN architecture as a feature extractor using ImageNet dataset and wrist vein raw and preprocessed (PIS-CVBR®) dataset images. Continuous blue, line-line orange, and dot-dot purple curves show the performance, respectively, of the UC3M-CV2, UC3M-CV1+CV2, and PUT dataset. The EER for each curve is provided in the legend.

2) Architecture experiment:To evaluate the different architectures applied in the TL process (Fig. 11, green block), VGG16, VGG19, ResNet50, and ResNet152,

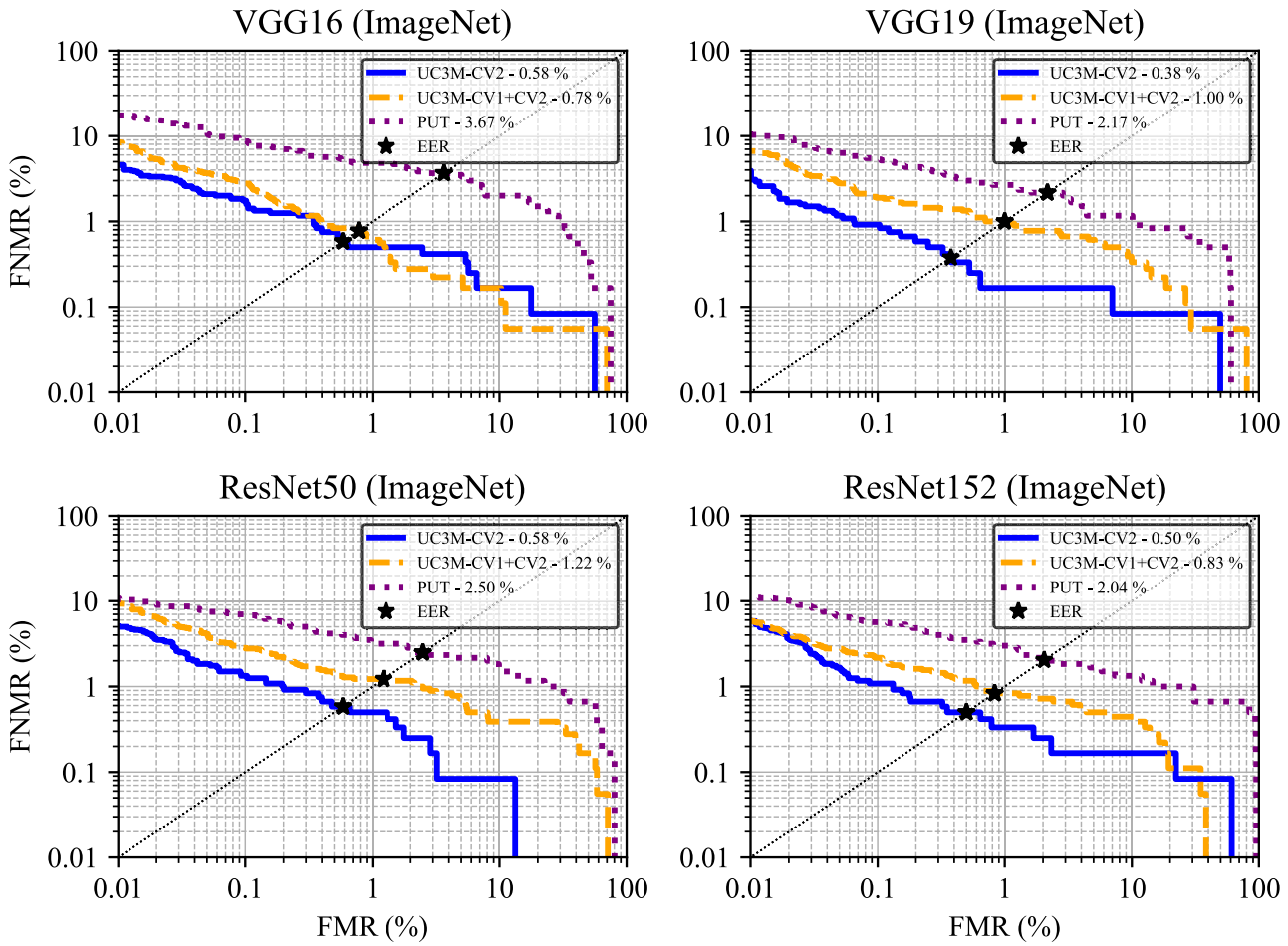

FIGURE 13. Biometric performance: Verification. DET curves for the 4 Transfer learning CNN architectures as a feature extractor using ImageNet dataset. Continuous blue, line-line orange, and dot-dot purple curves show the performance, respectively, of the UC3M-CV2, UC3M-CV1+CV2, and PUT dataset. The EER for each curve is provided in the legend.

all datasets have been used in their raw state (no preprocessing) and the train-test percentage is again 50-50 % in this experiment. Fig. 13 shows the DET curve for each CNN architecture and the 3 databases.

The results could reveal that a deeper architecture as ResNet provides slightly more easy-discriminant features using the ImageNet TL dataset according to the implemented Logistic Regression classifier. Nevertheless, this hypothesis is not conclusive because the VGG architecture as a feature extractor for some cases in the identification performance section presents a higher performance with the same classification technique, Logistic Regression.

3) Train-test experiment: As has been previously mentioned in this work, the dataset percentage split for testing the DL and traditional ML classifiers should be at least 50 % in biometrics, even more, when the dataset size is very far from reaching the general DL classification rule of thumb of 1000-5000 examples per class. Nonetheless, Fig. 14 shows the best model performance, obtained with ResNet152 as a feature extractor (pre-trained with ImageNet), over the 3 raw wrist vein datasets using 50-50 % train-test sets but also 75-25 %.

Best results, in terms of EER, 0.38, 0.78, and 2.04 % have been reached with a 50-50 % train-test split, respectively, for the raw UC3M-CV2, UC3M-CV1, and PUT datasets and pre-trained CNN architectures as feature extractors: VGG19, VGG16, and ResNet152.

2) IDENTIFICATION

Following ISO/IEC 19795-1, the identification performance of the proposed system has been obtained in a closed-set test (all the test subjects utilized in the evaluation are known) and has been shown as Cumulative Match Characteristic (CMC) plots. This type of curve shows the True-Positive Identification Rate (TPIR) over the returned rank (R). The Failure-To-Enrol Rate (FTER) and the Failure-To-Acquire Rate (FTAR) are unknown for all the datasets.

Fig. 15 shows the CMC curve for the 3 raw state datasets (split in 50-50 % and 75-25 % train-test sets) for every 100 users (N = 100) and each CNN architecture. The TPIR

TABLE 7. Biometric performance: Verification experiments.

#	Experiment	Dataset	Architecture	Preprocessing	Train-test (%)
1	Preprocessing	All	VGG16	**NO** **YES**	50-50
2	Architecture	All	**VGG16** **VGG19** **ResNet50** **ResNet152**	NO	50-50
3	Train-Test	All	VGG16	YES (PUT)	**50-50** **75-25**

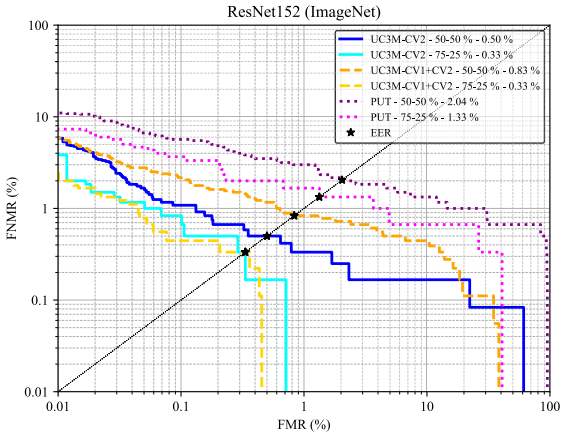

FIGURE 14. Biometric performance: Verification. DET curves for the ResNet152 Transfer learning CNN architecture as a feature extractor using ImageNet over wrist vein datasets split in 50-50 % and 75-25 % train-test sets. Continuous blue, line-line orange, and dot-dot purple curves show the performance, respectively, of the UC3M-CV2, UC3M-CV1+CV2, and PUT dataset. The EER for each curve is provided in the legend.

is provided over the rank R (1, 5), and with the threshold T set to 0 (minimum possible similarity score) [8]: TPIR (N = 100, R, T = 0).

Analyzing the 50-50 % train-test and rank 1 results, VGG19 presents a slightly higher identification performance than VGG16, where the TPIR for PUT is completely out of range with 91.00 %. This wide difference in VGG16 is also notable comparing the complete curve against PUT 75-25 % that is 6 % more accurate in the rank 1 starting point (probably due to the reduced number of testing images: 3). The same case could be noted by comparing ResNet50 against ResNet152, with the former architecture presenting a reduced TPIR (N = 100, R = 1, T = 0) of 93.83 %.

In general terms, VGG16 as a feature extractor provides the best vascular biometric identification performance in the proposed model over the tested vein datasets. However, it is worth pointing out and has been demonstrated that all the applied CNN architectures, pre-trained over the huge ImageNet dataset, are completely effective in the Transfer Learning process with reduced differences among them.

Best results, in terms of TPIR at rank 1, 98.67, 97.67, and 95.00 % have been reached with a 50-50 % train-test split, respectively, for the raw UC3M-CV2, UC3M-CV1, and PUT datasets and pre-trained CNN architectures as feature extractors: VGG16, VGG19, and ResNet152.

B. COMPUTATIONAL TIME PERFORMANCE

The computational time performance or computational workload has been stated providing the accept/reject transaction time for the verification task and the direct (rank 1) identification transaction time for the identification task. For this purpose, two different experiments have been carried out for both missions. On the one hand, as an offline test (computer equipment, CPU), the total time cost of every verification/identification transaction is measured and divided by the total number of tested transactions to obtain the unit average computational cost. On the other hand, as an online test, the computing verification/identification time of the smartphone real-time video capture for a user sample presentation is quantified.

1) VERIFICATION

Table 8 (experiment 1, offline) shows the workload time, total and unit average, in the verification and identification transaction for each dataset user (train and test: UC3M-CV2, UC3M-CV1+CV2, and PUT) and the VGG16 architecture. The total transaction time represents the time-lapse between the starting features extraction and the model predictions for the 50 % (train set) of the entire databases. The unit transaction time is the arithmetic average for each image.

As Table 8 infers, the workload increases according to the number of processed images. However, PUT, the smaller dataset (1200 images), presents the maximum unit transaction time (70 ms) due to the fact that its images have a 1024 × 768 resolution (XGA) instead of the 640 × 480 resolution (VGA) of the UC3M-CV2 and UC3M-CV1-CV2. This means more information is processed by the model for each transaction.

2) IDENTIFICATION

Comparing the verification and identification, it should be noted that the second task is faster. This counterintuitive

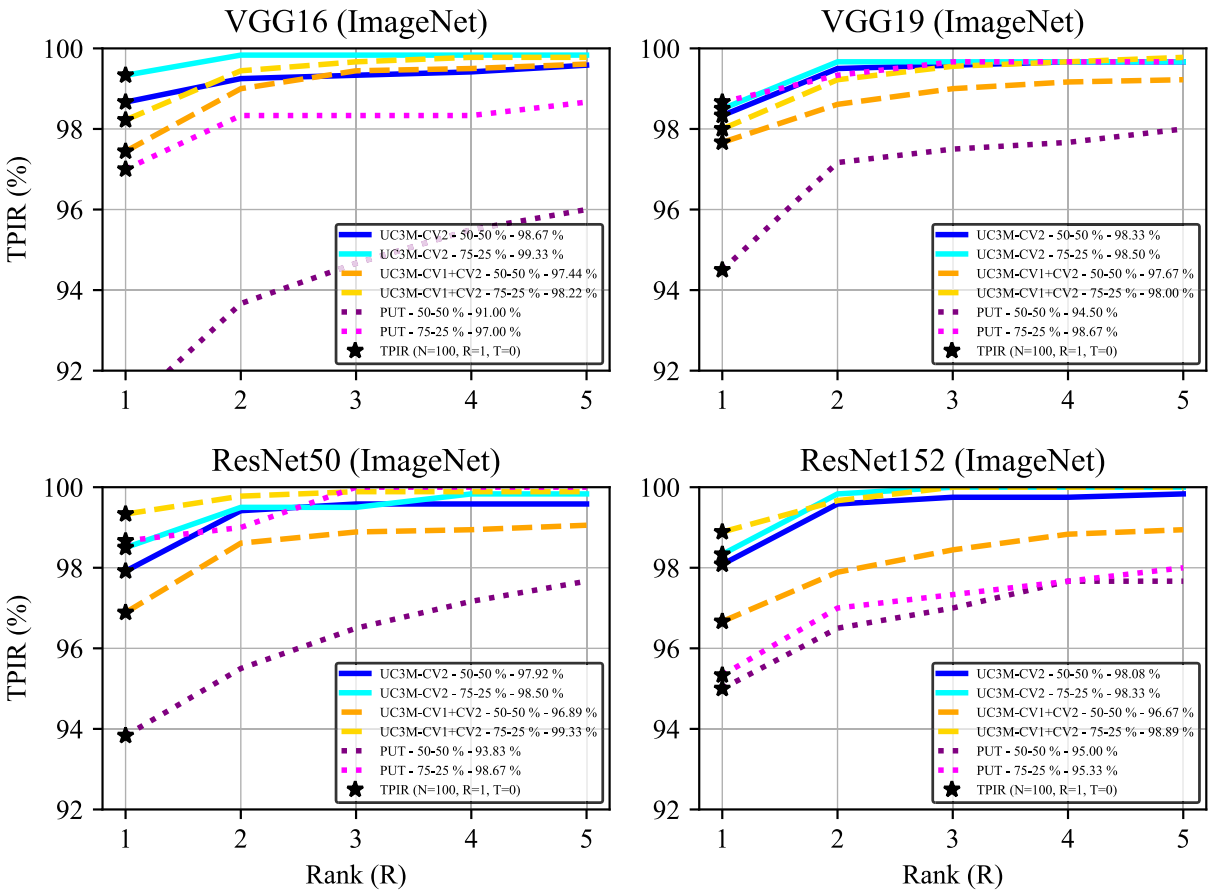

FIGURE 15. Biometric performance: Identification. CMC curves for the ResNet152 transfer learning CNN architecture as a feature extractor using ImageNet over the wrist vein datasets split in 50-50 % and 75-25 % train-test sets. Continuous blue, line-line orange, and dot-dot purple curves show the performance, respectively, of the UC3M-CV2, UC3M-CV1+CV2, and PUT dataset. The TPIR at rank 1 TPIR (N = 100, R = 1, T = 0) for each curve is provided in the legend.

result occurs because the verification model follows a one-vs-all/one-vs-the-rest (Logistic Regression) training strategy, instead of the multinomial (Logistic Regression) implemented in the identification model. The one-vs-the-rest strategy generates and fits one classifier per class. For each classifier (Logistic Regression classifier) the user or class is trained against all the other users to reduce the problem to a binary solution. That is the reason why the inferences of the proposed verification model are slower than those of the identification model.

Table 8 also shows the computational unit workload (experiment 2, online), in the verification and identification transactions for the input real-time video sample captured by the smartphones (Xiaomi© Pocophone F1 and Xiaomi© Mi 8) and the model train with the raw UC3M-CV2 and the VGG16 architecture. This value is obtained by averaging 10 correct attempts. In order to run TensorFlow© model inferences in embedded devices, the TensorFlow© Lite framework allows reducing and optimizing models. In this sense, the proposed Transfer Learning model has been adapted to fit the TensorFlow© Lite constraints: the Logistic Regression algorithm has been implemented adding the following trained layers: flatten (25088 × 1 output feature), fully-connected (512 × 1 output features), sigmoid activation, fully-connected (100 × 1 output labels) and softmax activation. The body of the CNN has been "frozen" with the pre-trained ImageNet weights. The obtained TensorFlow© Lite model (tflite format) infers over the input real-time video capture for both missions (verification and identification) using different decision policies (Fig. 5, Decision subsystem).

The tflite model has been run over the CPU (Qualcomm® Snapdragon SDM845 Octa-Core, 2.8 GHz) and the GPU (Qualcomm® Adreno 630) of both smartphones. Table 8 provides the time-lapses for both processing units. As could be expected, the transaction time is clearly short for the GPUs (unit transaction average time of 331.9 ms vs. 870.9 ms on the Xiaomi Pocophone F1) and there is no noticeable difference between the smartphones as could be anticipated given that they present the same processing units.

Finally, it is worth pointing out that the smartphone infrared cameras provide real-time video captures with a frame rate of 30 FPS. During inferences, this frame rate drops to approximately 8 FPS in the GPUs whereas it decreases to

TABLE 8. Computational time performance: Verification and identification.

Experiment	Hardware	Task	Dataset (train-test)	Total transaction (ms)	Unit transaction (ms)
Offline	9th Generation Intel® Core™ i9k (64-bit, 16 GB of RAM, 3.6 GHz) CPU	Verification	UC3M-CV2	19278	16
			UC3M-CV1+CV2	69552	39
			PUT	39750	70
		Identification	UC3M-CV2	14391	12
			UC3M-CV1+CV2	21070	12
			PUT	9707	16
Online	Pocophone F1 (GPU)	Verif./Ident.	UC3M-CV2 - real-time video capture	3319*	331.9
	Pocophone F1 (CPU)			8709*	870.9
	Mi 8 (GPU)			2662*	266.2
	Mi 8 (CPU)			9225*	922.5

(*) For 10 correct attempts.

6 FPS in the CPUs. This difference in the frame rate could be noticeable to the user.

V. CONCLUSION

In this work, a novel Convolutional Neural Network model based on Transfer Learning for contactless vein biometric recognition has been designed, implemented, integrated, and tested on smartphones for the first time using a research approach.

The proposed investigation represents the first Deep Learning solution for wrist vascularity and shows both the necessary and the real-life viability of these types of advanced biometric systems. Furthermore, TL has only been clearly addressed by a few other researchers in the Vascular Biometric Recognition field. This technique exploits and takes advantage of the unique image features "knowledge" acquired by a Convolution Neural Network pre-trained over more than 1 million images (ImageNet dataset) instead of training a CNN from scratch (with a reduced number of vein images), with the consequent investment of training time and difficulty that this entails.

The proposed TL model has been configured in 4 different well-known CNN shapes or architectures: VGG16, VGG19, ResNet50, and ResNet152. Each of these CNNs, previously pre-trained on the ImageNet dataset, as a feature extractor, is in charge of obtaining unique arbitrary features that are then independently learned/classified using the Logistic Regression algorithm (traditional Machine Learning algorithm).

The wrist vein near-infrared images from the public PUT dataset and the contactless private UC3M-CV2 (smartphones) and UC3M-CV1 datasets, split into 50-50 % and 75-25 % train-test sets, have been passed through the 4 CNN architectures. For each network, the unique feature vectors have been stored and used to train 4 different Logistic Regression classifiers. The classifiers have been adapted in order to verify or identify each user according to the following training strategies, respectively, one-vs-the-rest or multinomial.

The results, divided into biometric and computational time performance, and provided according to the ISO/IEC 19795-1 standard shows the real-life viability of the proposed system.

On the one hand, the Detection Error Trade-Off (DET) curves provide the verification performance (FNMR vs. FMR) according to each CNN (as a feature extractor) architecture, the preprocessing (raw images or PIS-CVBR® preprocessing), and the train-test split (50-50 % or 75-25 %). The results infer the high importance and influence of the dataset image quality and size (PUT is the smaller dataset consisting of the poorest image quality/wrist positioning and as a result the worst biometric performance) and the train-test split (as could be expected, 75-25 % provides a better performance), above the CNN architecture. Best results, in terms of EER, 0.38, 0.78, and 2.04 % have been obtained with a 50-50 % train-test split, respectively, for the raw UC3M-CV2, UC3M-CV1, and PUT datasets.

On the other hand, the Cumulative Match Characteristic (CMC) plots provide the identification performance of the proposed model showing the True-Positive Identification Rate (TPIR) over the returned rank (R, between 1 and 5 despite the fact that only rank 1 should be considered as acceptable for biometric solutions with the reduced size of the database used). The same conclusion is inferred in this case. Best results, in terms of TPIR at rank 1, 98.67, 97.67, and 95.00 % have been obtained with a 50-50 % train-test split, respectively, for the raw UC3M-CV2, UC3M-CV1, and PUT datasets and different pre-trained CNN architectures as feature extractors.

In order to obtain a real-life system application, the computational/workload time performance has been studied and reported.

As an offline test, the time consumption for each transaction or inference made by the model on the computer CPU (9th Generation Intel® Core i9k) has been estimated as an average. The results show that the identification is faster than the verification due to the Logistic Regression model strategy. Also, as could be expected, the higher the image resolution (PUT dataset, 1024×768) the slower the transaction because the model has to process more information.

Finally, the proposed online test demonstrates the real reachable integration of advanced DL biometric models on smartphones and other embedded systems without compromising the recognition performance. For this purpose, the adapted TL TensorFlow© model has been converted to a TensorFlow© Lite model which can be executed on a smartphone. The model has been run over the CPU and the GPU of both smartphones. As could be expected, the performance is significantly better in the GPU. But both solutions could fit into a real-life application. As input, the near-infrared camera captures a 30 FPS video on which inferences are made. The adapted model is the same for verification and identification but with a different decision policy. That is why there is no notable variance between their inferences.

For future works, all efforts will be focused on increasing the recognition performance without negatively affecting the time workload performance. This will be achieved by applying advanced DL techniques and leveraging previous knowledge, such as Transfer Learning, for a better and safer user experience in terms of comfort, hygiene, and security.

ACKNOWLEDGMENT

R.G.M. truly appreciates the grammar review and moral support provided by Maribeth Hoath-Perez. This work is dedicated to Laura Martinez-Pastor and her father who is very proud of her.

REFERENCES

[1] *Modes of Transmission of Virus Causing COVID-19: Implications for IPC Precaution Recommendations*, World Healtth Org., Geneva, Switzerland, Mar. 2020.

[2] T. Endoh, T. Aoki, M. Goto, and M. Watanabe, "Individual identification device," U.S. 2005 0 148 876 A1, Jul. 7, 2005.

[3] K. Kitane, "Fingervein authentication unit," US 2011 0 222 740 A1, Sep. 15, 2011.

[4] R. Garcia-Martin and R. Sanchez-Reillo, "Wrist vascular biometric recognition using a portable contactless system," *Sensors*, vol. 20, no. 5, p. 1469, Mar. 2020.

[5] R. Raghavendra and C. Busch, "A low cost wrist vein sensor for biometric authentication," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Chania, Greece, Oct. 2016, pp. 201–205.

[6] J. E. S. Pascual, J. Uriarte-Antonio, R. Sanchez-Reillo, and M. G. Lorenz, "Capturing hand or wrist vein images for biometric authentication using low-cost devices," in *Proc. 6th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Darmstadt, Germany, 2010, pp. 318–322.

[7] R. Garcia-Martin and R. Sanchez-Reillo, "Vein biometric recognition on a smartphone," *IEEE Access*, vol. 8, pp. 104801–104813, 2020, doi: 10.1109/ACCESS.2020.3000044.

[8] *Information Technology Biometric Performance Testing and Reporting Part 1: Principles and Framework*, Standard ISO/IEC 19795-1, American National Standards Institute, 2021.

[9] R. Kabacinski and K. Kowalski, "Vein pattern database and benchmark results," *Electron. Lett.*, vol. 47, no. 20, pp. 1127–1128, Sep. 2011.

[10] R. S. Kuzu, E. Piciucco, E. Maiorana, and P. Campisi, "On-the-fly finger-vein-based biometric recognition using deep neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2641–2654, 2020.

[11] S.-Y. Jhong, P.-Y. Tseng, N. Siriphockpirom, C.-H. Hsia, M.-S. Huang, K.-L. Hua, and Y.-Y. Chen, "An automated biometric identification system using CNN-based palm vein recognition," in *Proc. Int. Conf. Adv. Robot. Intell. Syst. (ARIS)*, Taipei, Taiwan, 2020, pp. 1–6.

[12] H. Houjun, S. Liu, H. Zheng, L. Ni, Z. Yi, and W. Li, "DeepVein: Novel finger vein verification methods based on deep convolutional neural networks," in *Proc. IEEE Int. Conf. Identity, Secur. Behav. Anal. (ISBA)*, Feb. 2017, pp. 1–8.

[13] W. Kim, J. M. Song, and K. R. Park, "Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (NIR) camera sensor," *Sensors*, vol. 18, no. 7, p. 2296, 2018.

[14] J. M. Song, W. Kim, and K. R. Park, "Finger-vein recognition based on deep DenseNet using composite image," *IEEE Access*, vol. 7, pp. 66845–66863, 2019.

[15] H. Qin and M. A. El-Yacoubi, "Deep representation for finger-vein image-quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1677–1693, Aug. 2018.

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

[17] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Comput. Vis., Graph., Image Process.*, vol. 39, no. 3, pp. 355–368, Sep. 1987.

[18] M. I. Obayya, M. El-Ghandour, and F. Alrowais, "Contactless palm vein authentication using deep learning with Bayesian optimization," *IEEE Access*, vol. 9, pp. 1940–1957, 2021.

[19] *CASIA Multispectral Palmprint Database*. Accessed: Jan. 20, 2021. [Online]. Available: http://biometrics.idealtest.org/

[20] N. A. Al-johania and L. A. Elrefaei, "Dorsal hand vein recognition by convolutional neural networks: Feature learning and transfer learning approaches," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 3, pp. 91–178, 2019.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

[22] R. S. Kuzu, E. Maiorana, and P. Campisi, "Vein-based biometric verification using transfer learning," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Milan, Italy, 2020, pp. 403–409.

[23] A. Rosebrock. *Deep Learning For Computer Vision*. PyImageSearch. Accessed: Jan. 20, 2021. [Online]. Available: https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/

[24] O. Toygar, F. O. Babalola, and Y. Bitirim, "FYO: A novel multimodal vein database with palmar, dorsal and wrist biometrics," *IEEE Access*, vol. 8, pp. 82461–82470, 2020.

[25] L. Wang, G. Leedham, and S. Gho, "Infrared imaging of hand vein patterns for biometric purposes," *IET Comput. Vis.*, vol. 1, no. 3, pp. 113–122, Dec. 2007.

[26] J. Uriarte-Antonio, D. Hartung, J. E. S. Pascual, and R. Sanchez-Reillo, "Vascular biometrics based on a minutiae extraction approach," in *Proc. Carnahan Conf. Secur. Technol.*, Barcelona, Spain, Oct. 2011, pp. 1–7.

[27] *OmniVision OV7750/OV7251 Datasheet Product Specification*, OmniVision, Santa Clara, CA, USA, 2020.

[28] R. Garcia-Martin, R. Sanchez-Reillo, and J. E. Suarez-Pascual, "Wrist vascular biometric capture using a portable contactless system," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Chennai, India, Oct. 2019, pp. 1–6.

[29] *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*. Accessed: Jan. 20, 2021. [Online]. Available: https://www.hdfgroup.org/
[30] The HDF Group. *Hierarchical Data Format Version 5*. Accessed: Jan. 20, 2021. [Online]. Available: https://www.hdfgroup.org/
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778. [Online]. Available: https://ieeexplore.ieee.org/document/7780459, doi: 10.1109/CVPR.2016.90.

RAUL GARCIA-MARTIN received the bachelor's degree in industrial electronics and automation engineering and the master's degree in electronics systems and applications engineering from the University Carlos III of Madrid (UC3M), in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the University Group for Identification Technologies (GUTI), with a focus on vascular biometric recognition. At the same time, he acquired experience in industrial applications as a software and hardware developer in several companies.

RAUL SANCHEZ-REILLO (Senior Member, IEEE) received the Ph.D. degree in telecommunication engineering from the Universidad Politecnica de Madrid, in 2000. He is currently a Full Professor with the University Carlos III of Madrid (UC3M). He is also the Head of the University Group for Identification Technologies (GUTI). His research and development group is involved in project development related to identification technologies, either by the user of secure elements (such as smartcards) and by using biometrics. In addition to research and development activities, he has also managed projects concerning a broad range of applications from Social Security Services till financial payment methods. He has taken part in European projects, being WP leader in most of them. He is member of SC17, SC27, and SC37 Standardization Committees, holding the international secretary of SC17 WG11 and SC37 WG2. In 2009, he founded the IDTestingLab and Evaluation Laboratory for identification products.