JJEE
Volume 11 | Number 1 | March 2025 | Pages 66-88

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Robust Finger Vein Presentation Attack Detection Using XceptionNet-based Modified Depthwise Separable Convolutional Neural Network

Ahmad Atallah Alsawalqah[1*], Bakhtiar Affendi Rosdi[2]

[1, 2] School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, Pulau Penang, Malaysia

E-mail: sawalqah@student.usm.my

Received: Jun 02, 2024 *Revised: Aug 10, 2024* *Accepted: Aug 24, 2024* *Available online: Oct 1, 2024*

***Abstract*—** Finger vein presentation attack detection (FVPAD) biometric systems have seen substantial enhancements through the application of deep learning convolutional neural networks (DCNN). This advancement led to increased complexity, parameters and resource requirements. To address these challenges, a novel modification to the first entry flow of the XceptionNet architecture based on customized depthwise separable convolution (DSC) CNN-based for extracting robust features from FV images to detect spoofing attacks is proposed in this paper. The proposed approach stands out for its simplicity in design, fewer parameters, reduced computational load, minimal resource and equipment needs, and minimum data overflow while maintaining high accuracy in verification and classification tasks. The developed FVPAD system includes FV image data preprocessing and augmentation, a modified XceptionNet architecture based on DSC to deeply extract robust features. Finally, the fully connected (FC) layers exclusively use the SoftMax activation function to normalize, predict and classify output classes. The model was evaluated on cropped FV images from the IDIAP and SCUT-SFVD datasets, achieving high accuracy rates of 100% and 99.499%, respectively. It also has the lowest number of trainable parameters at 131,106 acquired from fifteen convolutional and depth-separable convolution layers.

***Keywords*—** Finger Vein; Presentation attack detection; Deep learning; Depthwise separable convolutional neural network; XceptionNet.

1. INTRODUCTION

In the past, security measures revolved around using faces, eyes, and fingerprints for authentication in various operations like bank transactions, personal access control systems, ATMs, and data centers. The traditional approach involved detecting shapes in an image by identifying its edges and corners and then comparing it to a database. However, this method is vulnerable to attacks, particularly when it comes to fingerprints. For example, an attacker can easily steal someone's identity by using uncooperative methods such as photographs or capturing their fingerprint from surfaces like tape or glass. Biometric technology, such as fingerprint recognition, has gained popularity and is increasingly being integrated into the daily life applications. Unfortunately, the fingerprint-based recognition system is still vulnerable to attack. To fool the system, attackers can steal fingerprints and create molds out of materials such as silicon, gelatin, or latex [1]. The biometric system aims to automatically identify individuals based on biological and behavioral characteristics. for user authentication, it employs a variety of modalities, including fingerprints [2], face recognition [3], alongside iris

* Corresponding author
DOI: https://doi.org/10.5455/jjee.204-1717325207

scanning [4]. The finger vein pattern is particularly valuable due to its resistance to change and stability. Additionally, each finger vein is unique, contactless, difficult to forge, and highly accurate for identification purposes [5, 6].

Researchers are currently exploring innovative methods to enhance the security of liveness operations by utilizing a person's biological characteristics. This is accomplished by combining digital images with biometric data, such as facial or finger vein measurements. These metrics have enabled the development of a robust system that is difficult to exploit. Due to the specialized equipment required for capturing FV features and structures, including an image capture device and near-infrared illumination, the use of finger vein technology has proven effective in preventing theft. In this paper, finger vein biometric characteristics are utilized to reduce spoofing attacks. Unlike previous studies that focused on optimizing the quality of FV images for superior performance [7, 8], this research focuses on improving overall fluency. Unfortunately, these types of detection systems are still vulnerable to spoofing attacks. Attackers have been successful in bypassing the finger vein detection system by developing sophisticated attack methods that mimic the same principles and approaches used by the system [9]. For instance, using ink and sandwich paper, they were able to create fake finger veins that absorb NIR light just like real finger veins [10], due to the properties of the materials used for deception. As a result, there is an ongoing need to develop spoofing detection methods, such as utilizing presentation attack detection (PAD) techniques for finger vein biometrics.

Various finger vein PAD approaches of biometric system have conventional framework and exhibit similar behaviors [11]. However, they vary in terms of the carried data, applied improvement methods, and employed image preprocessing and classification techniques. Consequently, several FVPAD approaches were devised to extract features from FV images. The more traditional PAD approaches rely on distinguishing between fake and real FV images based on differences in feature extraction obtained from these images [12]. Subsequently, researchers employed deep learning and transfer learning with CNN models such as LeNet [13], AlexNet [14], VGG-16 [15], and VGG-19 [16] to extract FV image features for detecting presentation attacks. A comprehensive review of relevant literature on the application of deep learning will be presented in Section 2. Therefore, the application of deep learning techniques with convolutional neural network models like the XceptionNet architecture [17, 18] has resulted in advancements in finger vein PAD models [19]. These enhancements encompass improved efficiency, accuracy, and performance while reducing processing and computation load and time.

The main objective of this research is to propose a modified depthwise separable convolution (DSC) neural network with residual connections and incorporate fully connected (FC) layers including the SoftMax function, for designing a low-complexity FVPAD biometric system. The DSC CNN-based application allows us to achieve minimal parameters by controlling the number of channels/filters, kernel/filter size, input/image size and stride for each channel. Additionally, the structure process is simplified by focusing on modifying only the first flow entry of XceptionNet based on customizing the DSC neural network, while the middle and exit flow entries are eliminated from the proposed model structure. Moreover, the proposed model emphasizes cross-channel correlations and spatial correlations to map and separate all channels based on their respective filter sizes. To further enhance the proposed FVPAD model while maintaining feature richness, changes are introduced in the number of

channels/filters between layers, which speeds up computations and compensates for any loss caused by learning kernel size variations. The training datasets were enlarged using data augmentation techniques to address overfitting issues.

The main contributions of this study are: i) developing a deep learning model architecture by modifying the initial structure of the XceptionNet [19, 20] using a depth-separable convolutional neural network approach. Specifically, the proposed model had two customized modules of DCS layers with residual connections, ii) fine-tunning and setting hyperparameters of the model structure, such as the number of DCNN layers, channels/filters, kernel/filter size, input image resizing, stride, training epochs, and learning rate, iii) applying data augmentation techniques using the ImgAug affine transformation method, including flipping, rotating, shearing, cropping, rescaling, and adjusting brightness, iv) employing the Softmax activation function as a classifier for the prediction and categorization process, which makes decisions based on statistical probabilities.

Accordingly, the proposed model structure was designed to address key challenges faced by researchers, such as achieving high performance while maintaining low complexity, minimal processing time, and low operational losses [21-25], especially in the context of PAD methods. Specifically, the network structure was simplified, and the hyperparameters were adjusted to reduce trainable parameters, minimize computational load, and decrease feature map complexity. Tuning the hyperparameters helped map and separate the network channels, enabling efficient and effective feature extraction while minimizing complexity. This approach aimed to minimize computational complexity without compromising the ability to learn. Additionally, the resizing image method was crucial for quick and efficient training of the FVPAD model, especially for deep learning. Furthermore, the data augmentation approaches enlarged the FV dataset, acted as a regularization tool to reduce overfitting, generated accurate results, and accelerated the training process. Finally, the SoftMax activation function, with its exponential normalization and reliance on uniform input distribution, was used in this study to simplify the design, achieve class normalization, and generate accurate results in multi-class classification tasks.

The remainder of the paper is organized as follows: Section 2 reviews the existing literature on using convolutional neural networks and other deep learning techniques to enhance the performance of the finger vein PAD models. In Section 3, the main research scenarios and methodology have been investigated. Section 4 presents the simulation and experimental results. Finally, Section 5 outlines the key findings from this research and suggests potential directions for future developments.

2. RELATED WORKS

In recent years, finger vein biometric measurement has gained popularity in comparison to other measures such as the iris [26], face [27], palm vein [28], and fingerprint [29]. The unique and permanent nature of each individual's finger veins makes it an efficient method for biometric verification systems [30]. However, there are limitations, including dataset restrictions and the possibility of spoofing or presentation attacks. To address these issues, researchers have been concentrating on using deep learning techniques, such as convolutional neural networks, to improve FVR performance. Various approaches have been employed to tackle the challenges related to a high number of layers and limited datasets in FV identification using subconvolutional neural networks. For instance, Boucherit et al. [31]

developed a deep-fused CNN that merges multiple convolutional neural networks for FV identification. Weng et al. [32] introduced the ZFNet architecture, which utilizes shareable convolution layers. In the study [33], researchers proposed deep learning approaches, such as deep neural networks (DCNN), that have demonstrated the capability of learning robust features from raw pixel images for finger vein image representation and predicting the quality of FV images for biometric verification systems. Moreover, in the field of using deep learning techniques with convolutional neural networks, the study [34] utilized sub-convolutional neural networks to handle challenges posed by numerous layers and data limitations in FV recognition tasks. Although the research focused on finger vein recognition instead of spoofing detection, it employed lightweight CNNs like AlexNet to extract main features deeply in a short time and make classification easier and faster.

In addition to the previous approaches, a study [35] proposed a lightweight CNN in that combined center loss and dynamic regularization to enhance classification efficiency. Das et al. [36]utilized MatConvNet-1.0-beta24 for identifying larger FV images. In the study[37], researchers implemented FV-Net architecture and focused on Data Augmentation Parameters in Convolutional Neural Networks for finger vein detection and classification. Numerous research studies have also centered around deep CNN techniques for deep finger vein identification. For instance, Huang et al. [38] employed VGG-16 while Lu et al.[39] used pre-trained AlexNet CNN as competitive order CNN-CO for feature extraction and matching purposes.

Different deep convolutional neural network architectures were employed by Jalilian and Uhl [40] to obtain finger vein patterns from images. Noh et al. [41] used DenseNet-16 for texture analysis, while researchers in another study [42] focused on developing a full model using deep CNNs for feature extraction and classification. They utilized ResNet-50 as a pre-trained CNN with lightweight DSC for deeper feature extraction. Nguyen et al. [10] applied transfer learning methods using AlexNet and VGG-16 models. These paired CNN methods were enhanced to overcome overfitting, utilizing transferred parameters. Another research study [43] incorporated transfer learning using AlexNet to improve the reliability of the biometric system, especially in detecting FV presentation attacks with seven additional augmented layers. In 2020, researchers developed the FVRAS-Net [44] real-time model for anti-spoofing detection and FV recognition. This model can perform all essential operations in a biometric system, such as feature extraction, training, classification, and matching. Another study by Hengyi Ren et al. [45] focused on encrypting FV images using the RSA algorithm and using a CNN-based algorithm called ResNet 34 to extract features and recognize the images. In a recent study by Huy H. Nguyen et al. [46], three types of DCNN were utilized: VGG-19 as a pre-trained CNN-based for feature extraction; Capsule NW [47]; and XceptionNet [20] CNN-based specifically designed for classification tasks.

Consequently, deep convolutional neural networks are widely used in a variety of applications to address training time and overfitting issues. For example, reserachers proposed a DSC-LSVM [19] finger-vein PAD model for the IDIAP and SCUT-SFVD datasets. the computational workload was reduced by 10% by using the first entry flow level of DSC DCNN with the linear support vector machine LSVM. Another recent study [48] created a finger-vein PAD model called FV2021, which incorporated deep learning and transfer learning techniques based on DSC CNNs. The FV2021 model also employed spatial convolution by applying depthwise and pointwise convolutions sequentially. This approach proved

beneficial for feature extraction, PRNU correlation analysis, and texture descriptor comparison. In conclusion, previous studies have shown that using deep learning techniques can improve the performance and accuracy of FVPAD models DCNN-based. Incorporating these techniques simplifies model complexity, making it easier to handle large datasets and extract relevant features from FV images. Our proposed approach is to use DSC CNNs in the FVPAD system for identifying the origin of FV images while minimizing model parameters and avoiding unnecessary architectural complexities.

3. METHODOLOGY

3.1. General Deep CNN Structure

A convolutional neural network is a type of deep neural network that utilizes advanced deep learning applications to detect complex patterns. It is particularly effective for tasks such as image processing, verification, object detection, pattern recognition, and classification [49], [50]. CNNs have been widely used in various applications including computer vision due to their ability to automatically learn spatial hierarchies of features from the data. In a subsequent study [51], multiple layers and convolution filters were used to detect local patterns and extract complex features from input FV images. The filters output is then used by one or more fully connected FC layers for feature extraction and prediction. The related CNN architecture [52] diagram is shown in Fig. 1. Therefore, this paper focuses on deep convolutional neural networks applications, specifically the XceptionNet [20] CNN model. It also explores depthwise separable convolution CNNs [20, 53, 54], with a particular focus on the entry flow of XceptionNet as a deeper feature extraction method. The proposed model comprises of convolutional DSC, and FC layers. The first two convolution layers serve as the initial computation, functioning as feature detectors and extracting features from the resized input images. These features are then mapped to the next module, which includes a separable convolution layer. In contrast, the final FC layer followed by a SoftMax function is utilized for prediction and classification purposes. Moreover, the DSC layers with residual connection layers are employed for extracting robust features deeply. Additionally, all convolutional and DSC layers are accompanied by Maxpooling, Batch-normalization (BN), and activation ReLU layers to reduce computational load and feature map complexity.

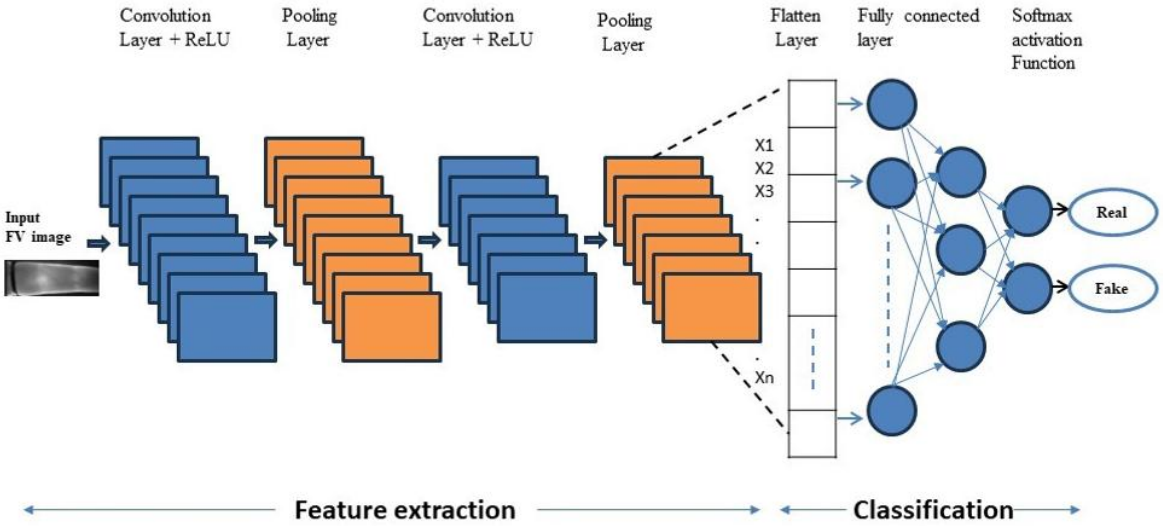


Fig. 1. Basic CNN architecture.

3.2. XceptionNet CNN Model

The XceptionNet model architecture is a linear stack of depthwise separable convolution layers with residual/skip connections. It is also described as a flixable and straightforward approach. The normal XceptionNet architecture consists of three parts of flows: the first entry flow has 21 layers with residual connections, the middle flow has 6 layers with residual connections and is repeated eight times, and the exit flow has 10 levels with residual connections. More details about the XceptionNet architecture can be found in [20].

More precisely, the XceptionNet model incorporates a linear stack of depthwise separable convolution layers with residual connections due to its architecture. This allows the model to learn deeply using fewer parameters and take advantage of DSC's ability to map cross-channel correlations and spatial correlations for each output channel separately. By using depthwise separable convolution instead of traditional convolution, the technique requires fewer parameters while being able to capture more complex representation and acquire complex patterns by applying different filters to each channel. In this research, the first entry flow of XceptionNet and customized the DSC layers have been adapted to extract robust features with the goal of reducing complexity and improving the efficiency of distinguishing between fake and real finger veins. Section 3.3 provides a detailed explanation of the proposed FVPAD model.

3.3. Proposed FVPAD Model (Customized DSC Model)

To further elaborate on the benefits of using depthwise separable convolution DSC CNN-based, four key points can be summarized: First, DSC helps to mitigate overfitting by introducing fewer parameters. Second, it reduces model complexity and computational requirements, making it particularly suitable for computer vision applications, as evidenced by numerous papers [48, 55, 56]. Third, applying different filters to each channel allows for more effective information gathering. Finaly, DSC is a robust DCNN that employs deep learning techniques for automatic feature extraction and classification of FV images from the IDIAP and SCUT-SFVD datasets. Therefore, to reduce the design complexity of the proposed model, the last module of the first entry flow has been excluded, and considering the selected optimal hyperparameters and factors such as input/image size, number of channels/filters, kernel/filter size, padding, and stride for each channel. In addition, determined whether a convolution layer or a depth-separable convolution layer is used for each module, along with residual connection blocks. Thus, the customized DSC model emphasizes the advantages of cross-channel correlations and spatial correlations in mapping and decoupling all network channels. This optimization aims to minimize parameters while improving the proposed model's performance.

Accordingly, the proposed FVPAD model is made up of a linear stack of depth-separable convolution layers with residual connections. The model architecture includes two initial convolutional layers, batch normalization, and the ReLU nonlinearity unit. This is followed by two module blocks, each containing two depth-separable convolution layers, ReLU activation, batch normalization, and Maxpooling. There are also two residual connections in the model that consist of 3x3 skip convolution operator layers for each module. Finally, there is one fully connected layer consisting of flatten and dense operations layer with a SoftMax activation function for prediction and classification. Specifically, the proposed model consists of two

sequential modules; each module block has a residual/skip connection with 3x3 convolution operator layers using a stride of two.

Based on the observations, a modified DSC CNN structure is designed and inspired by the first entry flow of the XceptionNet network. The kernel size was adjusted to 3x3 across all layers, including fully connected, skip connection, and depth-separable convolution layers. The padding is set to "same" for fully connected and modules layers, while stride is consistently set at two for all layers.

To optimize performance, various hyperparameters and factors have been considered. This includes the number of DSC modules used, filter sizes employed, data generation methods applied, including data augmentation techniques. Furthermore, the number channels/filters have been adjusted to 32 for the first convolution layer and 64 for the second convolution layer. Additionally, the number of channels/filters has been optimized and fixed at 64 for two DSC modules. Finally, the fully connected layer is also set up with a similar number of channels/filters of 64. The complete architecture can be seen in Fig. 2. More details regarding its structure are provided in Table 1.

Table 1. Architecture of the customized DSC CNN-based system for the proposed FVPAD model.

Layer name	Output size	No. of parameters
Input layer	96, 96, 3	0
conv2d_8 (Conv2D)	48, 48, 32	864
batch_normalization_16 (BatchNormalization)	48, 48, 32	128
re_lu_10 (ReLU))	48, 48, 32	0
conv2d_9 (Conv2D)	48, 48, 64	18432
batch_normalization_17 (BatchNormalization)	48, 48, 64	256
re_lu_11 (ReLU)	48, 48, 64	0
separable_conv2d_8 (SeparableConv2D)	48, 48, 64	4672
batch_normalization_18 (BatchNormalization)	48, 48, 64	256
re_lu_12 (ReLU)	48, 48, 64	0
separable_conv2d_9 (SeparableConv2D)	48, 48, 64	4672
conv2d_10 (Conv2D)	24, 24, 64	36864
batch_normalization_19 (BatchNormalization)	48, 48, 64	256
batch_normalization_20 (BatchNormalization)	24, 24, 64	256
max_pooling2d_4 (MaxPooling2D)	24, 24, 64	0
add_4 (Add)	24, 24, 64	0
re_lu_13 (ReLU)	24, 24, 64	0
separable_conv2d_10 (SeparableConv2D)	24, 24, 64	4672
batch_normalization_21 (BatchNormalization)	24, 24, 64	256
re_lu_14 (ReLU)	24, 24, 64	0
separable_conv2d_11 (SeparableConv2D)	24, 24, 64	4672
conv2d_11 (Conv2D)	12, 12, 64	36864
batch_normalization_22 (BatchNormalization)	24, 24, 64	256
batch_normalization_23 (BatchNormalization)	12, 12, 64	256
max_pooling2d_5 (MaxPooling2D)	12, 12, 64	0
add_5 (Add)	12, 12, 64	0
flatten_2 (Flatten)	9216	0
dense_2 (Dense)	2	18434
Total parameters		132,066

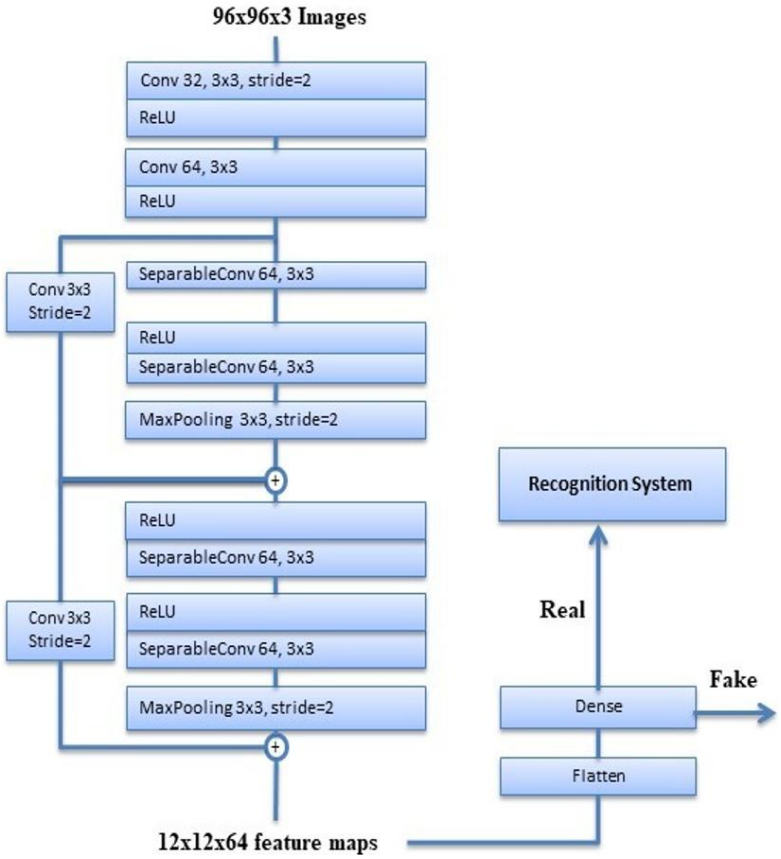


Fig. 2. DSC CNN-based proposed model.

3.4. FVPAD system

In fact, the finger vein PAD structure is consistent in architecture and behavior across all solutions in practice. However, there are variations in dataset types, data input size, prediction and mapping features, and performance improvement methods. Additionally, there are differences in data generation approaches including data preprocessing and augmentation techniques as well as feature extraction strategies and implemented classification methods. As a result of these differences, the FVPAD has the potential for further improvements to enhance its effectiveness. Furthermore, adjusting hyperparameters and attributes of model structure, as well as the criteria for data augmentation could lead to significant enhancements. This optimization not only affects the number of parameters but also impacts the choice of classification technique. In addition, it is crucial to consider factors like processing time, duration of training, the platform used for training and finding suitable datasets for FVPAD tasks.

In this context, Fig. 3 illustrates the complete architecture of the proposed FVPAD model, incorporating the DSC CNN-based approach. The proposed FVPAD model comprises three blocks. The first block involves data generating for FV images, utilizing preprocessing and augmentation techniques to enhance the reliability and performance of feature extraction by the CNN. Data generation significantly contributes to expanding FV data size, mitigating overfitting concerns, and improving overall CNN performance. Additionally, the second block of the FVPAD system includes a newly developed DSC CNN-based system for training, feature extraction, and mapping as illustrated in Fig. 2. Finally, it incorporates a fully connected layer comprising of a flattened layer and dense layer with SoftMax function. This

aids in determining whether an FV image is real or forged by adjusting hyperparameters such as number of channels/filters of 64, kernel size of 3x3, and stride of two. The system handles fake FV images differently - either ignoring them or repeating them - while real images undergo further processing in the final block known as the FVR system block for evaluation.

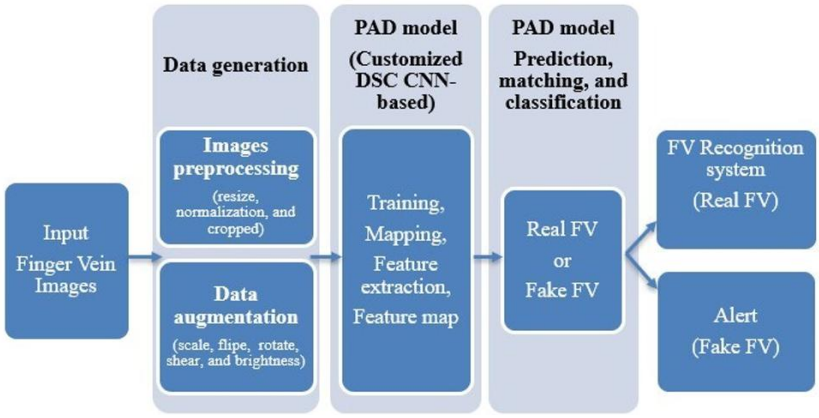


Fig. 3. Overall structure of the proposed FVPAD biometric system.

Furthermore, advanced techniques, such as deep learning CNN-based methods enhance the FVPAD model and create tailored PAD systems for finger vein images. The customized DSC uses the convolution layer to establish the developed model, which is considered the core of building the CNN block. It is responsible for major computations and incorporates fewer components like stride, number of channels, and kernel size. These are used to perform initial functions like feature detection, extraction, and mapping to the next module resulting in simplified model processing leading to improved performance.

4. SAMPLE DATA AND RESULTS

4.1. Sample Data

The proposed FVPAD model is evaluated using two publicly available datasets, namely IDIAP VERA [24] and SCUT-SFVD [25]. These datasets consist of cropped FV images, as demonstrated below:

- The IDIAP VERA is a finger vein database that includes 440 index finger images from 110 clients. These images were captured by a sensor to evaluate the performance of FVPAD systems in detecting finger vein spoofing or presentation attacks. Additionally, there are 440 forged finger images in the database, making it a total of 880 real and fake finger vein images in the dataset. Likewise, the IDIAP database contains two types of finger vein sizes – original size and cropped FV images for both fake and real finger veins. The original images have dimensions of 650x250 pixels while the cropped ones are sized at 550x150 pixels. This study utilizes the cropped FV images to exclude the surrounding environment and solely concentrate on the area around the actual finger veins.
- The SCUT-SFVD database, developed by South China University of Technology [25], is specifically designed for evaluating the performance of FVPAD systems in detecting finger vein spoofing or presentation attacks. The dataset consists of 3600 images, including both fake and real FV images. Each image has been cropped from its original

size of 158x467 pixels to a size of 150x450 pixels, focusing solely on the finger vein area. This study utilized the cropped FV images for analysis and evaluation purposes.

As a result, this study aimed to primarily enhance system performance by reducing the input size of the FV images to 96x96 pixels. Fig. 4 provides visual examples of data preprocessing approaches using resizing method for the original cropped real and fake FV images from the IDIAP and SCUT-SFVD datasets. Additionally, Table 2 presents detailed information about the distribution of these datasets, including training, validation, and testing data subsets.

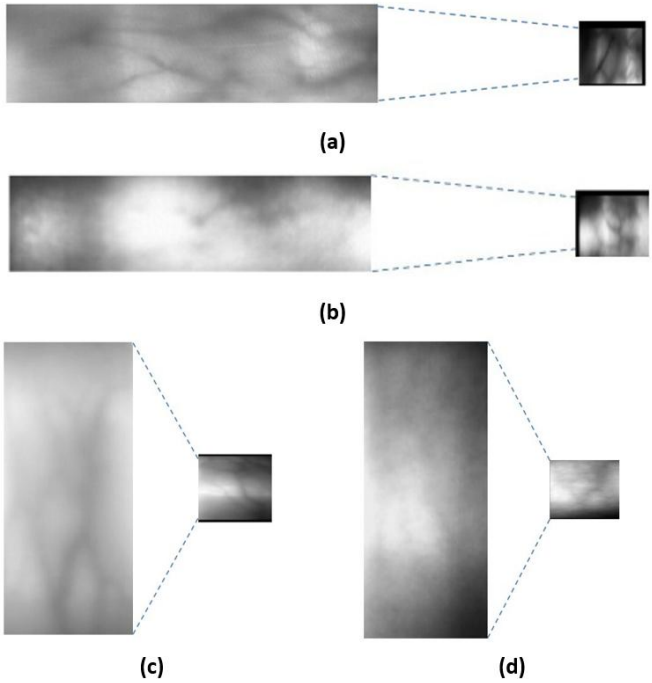


Fig. 4. Preprocessing method - FV images resized to 96x96 pixels: a) IDIAP (real FV); b) IDIAP (fake FV); c) SCUT-SFVD (real FV); d) SCUT-SFVD (fake FV).

Table 2. IDIAP and SCUT-SFVD datasets distribution.

Database	Real FV image			Fake FV image		
	Training	Testing	Validation	Training	Testing	Validation
IDIAP FV [24]	440	200	120	440	200	120
SCUT-SFVD [25]	720	2160	180	720	2160	180

Accordingly, the first block of the proposed FVPAD structure was used to preprocess and augment the collected samples sourced from the IDIAP and SCUT-SFVD datasets.

4.1.1. Data Preprocessing

In order to optimize the FVPAD model, it is necessary to preprocess the finger vein images before using them in the training and detection blocks, as previously illustrated in Fig. 4. This involves regenerating the FV image datasets by applying various transformations such as normalization, rescaling, and resizing. For consistency across datasets like IDIAP and SCUT-SFVD, all resized FV images are standardized to 96x96 pixels. Fig. 5 displays the set of preprocessed and FV images after resizing and normalization. Data preprocessing enables the proposed model to efficiently process data without causing delays or compromising accuracy

during detection, especially in problematic areas of the images. Therefore, preprocessing plays a crucial role in identifying optimal features for normalizing FV images. For example, resizing the images to smaller dimensions such as 96x96 pixels, enables deep learning models to train more quickly and efficiently.

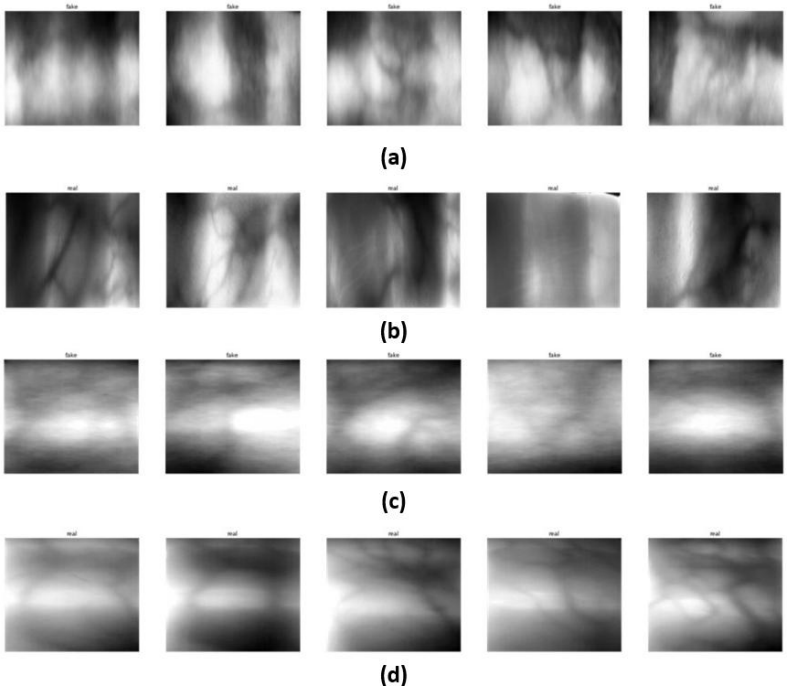


Fig. 5. FV images resized after preprocessing operation to 96x96 pixels: a) IDIAP datasets (real); b) IDIAP Vera datasets (fake); c) SCUT-SFVD dataset (real); d) SCUT-SFVD dataset (fake).

To ensure compatibility for matching and classification, the FV images are normalized using techniques such as numerical stability methods and the min-max function to transform them within a range of 0 to 1, as outlined in Eqs. 1 and 2, respectively.

$$Rescale = 1/255 \tag{1}$$

$$(x)' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

4.1.2. Data Augmentation

Data augmentation is a crucial technique in deep learning to enhance the diversity of a training dataset. This technique involves creating copies of existing data and making small adjustments to the images. It helps enlarge the dataset's feature vectors, extract more robust features, achieve reliable predictions, and reduce overfitting in deep-learning models. Additionally, it plays a role in regularizing and normalizing the FV datasets. In particular, the proposed FVPAD model applies affine transformations such as ImgAug augmenters techniques to FV datasets using approaches like shearing, flipping, rotating, cropping, zooming in and out as well as changing brightness and contrast. Those augmented methods are applied to both the IDIAP and SCUT-SFVD databases, except for the flipping method. Specifically, the images in the FV dataset from both databases differ in size; therefore, the IDIAP dataset was horizontally flipped while the SCUT-SFVD dataset vertically flipped. As a result, Table 3 details the techniques used to augment the FV training datasets. Regarding the results of augmentation techniques, Table 4 describes the distribution of augmented real and fake FV images for both IDIAP and SCUT-SFVD databases, which consist of approximately 4,400 and 7,200 cropped real and fake FV images respectively.

Table 3. Data augmentation techniques used to expand the FV training datasets.

IDIAP datasets		SUCT-SFVD datasets	
ImgAug augmenters	Affine transformations	ImgAug augmenters	Affine transformations
rotation	15	rotation	15
rescale	1./255	rescale	1./255
shear	0.2	shear	0.2
Horizontal flip	1.0	Vertical flip	1.0
Multiply (random brightness)	1.2, 1.5	Multiply (random brightness)	0.8, 1.2

Table 4. The augmented real and fake FV images distribution for both IDIAP and SCUT-SFVD databases.

Dataset	Real FV image training	Fake FV image training	FV image datasets	FV image augmentation	Total FV images
IDIAP FV [24]	440	440	880	5	4400
SCUT-SFVD [25]	720	720	1440	5	7200

As a result, Fig. 6 shows the FV sample results of five different augmentation methods applied to the FV training datasets. This highlights how data generation can be valuable in reducing overfitting, expediting the training process, and extracting robust features to achieve high performance, especially when applying deep learning in the proposed FVPAD model.

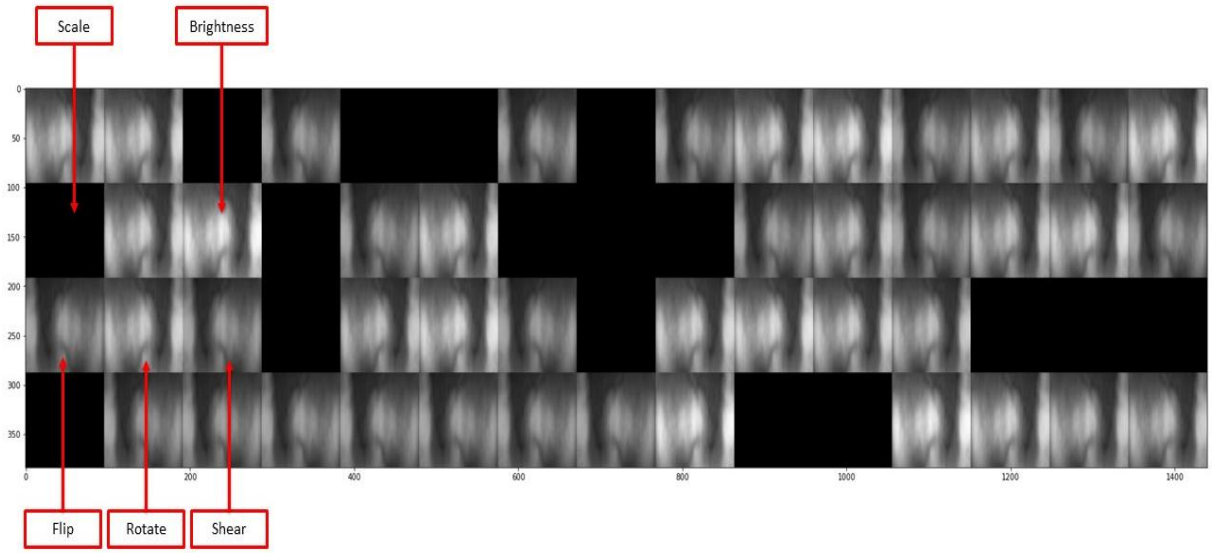


Fig. 6. Sample of the five data augmentation approaches applied on real and fake cropped FV images from the IDIAP training datasets.

4.2. Training Infrastructure

For the implementation phase, the proposed model employes TensorFlow platform [56-58]. TensorFlow offers reliability, user-friendliness, support for deep learning, and can be easily installed on machines with minimal specifications. The proposed model does not require high-end specifications but relies on certain components and hyperparameters to function properly. Thes include Jupiter Notebook version v2.4 and Python version 3.8 which installed on a laptop equipped with an Intel processor (core i7 2.7GHz, 4 CPU cores), and Intel

HD 620 graphics display device, along with NVIDIA GeForce 930MX graphics display with a share of 4GB memory capacity. For training purposes, fixed and optimized hyperparameters are utilized including a learning rate of 0.0001, batch size set at 32, and maximum epochs set at 40. Additionally, it incorporates categorical cross-entropy, Adam optimizer, ReLU activation function, early stopping technique, and maxpooling. Consequently, Table 5, outlines the specific training infrastructure needs specifications and the fixed hyperparameters used in the proposed FVPAD model for FV images generated from IDIAP and SCUT-SFVD datasets.

Table 5. Proposed FVPAD model training infrastructure and fixed hyperparameters.

Basic datasets	CPU	GPU	Tensor flow version	Python version	Optimizer	Learning rate	Batch	Epoch	Kernel
IDIAP and SCUT-SFVD	Intel core i7-2.7GHz 4CPU	NVIDIA GeForce 930MX 4GB	2.8.0	3.8	Adam	0.0001	32	40	3x3

4.3. Experimental Evaluations, Results, and Comparisons

4.3.1. Performance Evaluation Criteria

This section outlines the evaluation process and results for the proposed FVPAD model. In this experiment, the performance is evaluated using global standard evaluation metrics benchmarks based on ISO/IEC 30107-3 [59]. The following metrics and expressions provide a detailed explanation of how the FVPAD methods are evaluated:

- The attack presentation classification error rate (APCER): is a measurement used by biometric systems to assess their ability to detect attacks or presentation attacks. It represents the proportion of attack presentations using the same presentation attack instrument species PAIs that are incorrectly classified as bona fide presentations in a specific scenario. A lower APCER indicates that the system is more effective at detecting spoofing attempts.
- The bona fide presentation classification error rate (BPCER): is a measure of how well a biometric system can detect attempts to bypass it. It represents the proportion of bona fide presentations that are incorrectly classified as presentation attacks in a specific scenario. In this context, a presentation attack occurs when an attacker presents a legitimate biometric sample that has been registered in the system but does not correspond to the person who is present. A lower BPCER indicates higher effectiveness of the system in detecting such attempts to bypass its security measures.

• The average classification error rate (ACER): is the average of the APCER and BPCER. ACER is used to evaluate the overall performance of a biometric system. A lower ACER indicates better overall performance. The metrics APCER, BPCER, and ACER are calculated according to Eqs. 3, 4, and 5 respectively.

$$APCER = 1 - \left(\frac{1}{NPA}\right)\sum_{i=1}^{NPA}(Ri) \quad (3)$$

$$BPCER = \frac{\sum_{i=1}^{NPA}(Ri)}{NBF} \quad (4)$$

$$ACER = \frac{APCER+BPCER}{2} \quad (5)$$

The *NPA* variable represents the number of presentation attacks for a specific instrument species, and *NBF* represents the number of bona fide presentations. *Ri* is equal to 1 when the i_{th} presentation is classified as an attack presentation, and 0 if it is classified as a bona fide presentation. The evaluation metric for assessing the performance of the PAD system includes global standard metrics like precision, recall, accuracy, and ACER. These metrics are represented by Eqs. (6) to (9) respectively. They are also utilized in evaluating the proposed FVPAD model. It should be noted that the values of ACER in Eqs. (9) and (5) are identical.

$$Percision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$Accuarcy = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

$$ACER = \frac{2TP}{2TP+FP+FN} \tag{9}$$

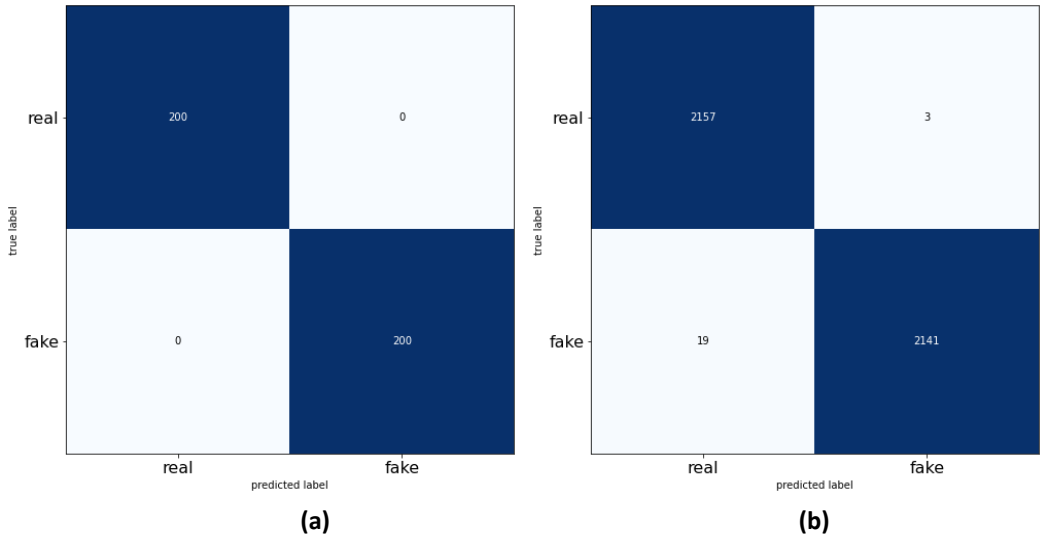
$$F1score = 2 \times \frac{Percision \times Recall}{Percision + Recall} \tag{10}$$

The metrics of precision, recall, and accuracy are used to evaluate the performance of biometric systems. *Precision* measures how many positive patterns were correctly predicted, while *Recall* calculates the fraction of positive patterns that were accurately classified. *Accuracy* (ACC) is the ratio of correct predictions across all systems. The *F1score* assesses both recall rates and accuracy for consistency. Furthermore, true positives, false positives, true negatives, and false negatives are values that are used to determine the tested datasets whether datasets are real or fake. As shown in Table 6, *TP, FP, TN,* and *FN* are used as inputs for confusion matrices that demonstrate classification performance.

Table 6. Binary confusion matrix.

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Consequently, the confusion matrix provides valuable insights into the performance of classification models by identifying areas where the system is not performing well and detecting challenging patterns for the model. Specifically, the confusion matrix metrics, such as accuracy, precision, recall, and F-score, are employed to evaluate the performance of the proposed model, as detailed in Section 4.3.2 (see Fig. 7).



(a) **(b)**

Fig. 7. Confusion matrix outputs: a) IDIAP Vera; b) SCUT-SFVD.

4.3.2. Experiment Results

The proposed FVPAD model was evaluated using global metrics described in the previous section. It was applied to cropped FV images from the IDIAP and SCUT-SFVD datasets. Subsequently, this section presents the initial results and the generated outputs of proposed FVPAD model. It includes fixed hyperparameters and attributes such as a total of four depth-separable convolution layers, two traditional convolution layers, and two residual/skip convolution layers. The overall structure consists of fifteen layers excluding the classification stage layers. Therefore, the IDIAP dataset has a validation step number of 240 and a training step number of 27, while the SCUT-SFVD dataset has recorded 360 validation steps and 45 training steps. Other fixed hyperparameters such as cross-entropy, pooling, activation function, and model normalization are set to default values including maxpooling, ReLU, categorical cross-entropy, and BN were utilized in testing the proposed FVPAD model.

Table 7. The proposed FVPAD model CNN-based hyperparameters, attributes and factors.

Datasets	Images size	Overall layers	Separable layers	Conv layers	Skipped ConV layer	Training steps no.	Validation steps no.
IDIAP	96x96	15	4	2	2	27	240
SCUT-SFVD	96x96	15	4	2	2	45	360

The proposed FVPAD model, built using the DSC deep learning application, demonstrated impressive efficiency with a total of 132,066 parameters. Of these, 131,106 were trainable parameters, and 960 were non-trainable parameters. This streamlined parameter count reflects the model's commitment to achieving low complexity while ensuring effectiveness. Subsequently, the proposed FVPAD model was applied to train FV datasets from the IDIAP and SCUT-SFVD databases, yielding impressive results for both tested datasets. It achieved an overall accuracy of 1.0 (100%) for the IDIAP dataset and 0.99499 (99.499%) for the SCUT-SFVD dataset. Table 8 provides a detailed breakdown of parameters and the model's accuracy; it offers a comprehensive overview of its performance and parameter distribution while highlighting its robustness in accurately detecting presentation attacks in finger vein biometric systems.

Table 8. DSC CNN-based accuracy and performance measurement.

Datasets	ACC Test datasets	Loss Test datasets	Total parameters	Trainable parameters
IDIAP	100%	0.000	132,066	131,106
SCUT-SFVD	99.499%	0.014	132,066	131,106

Furthermore, the outputs of the training data generator are utilized with the proposed FVPAD model for the training process in both IDIAP and SCUT-SFVD datasets. New fit parameters, such as training accuracy, training loss, validation accuracy, and validation loss, are evaluated as metrics. These parameters act as validation data for fitting deep learning models and indicate whether the proposed model is overfitting, underfitting, or a good fit.

As a result, Fig. 8a, the IDIAP dataset shows that training accuracy ranged from 91% to 100% across all epochs, while validation accuracy started at 79% and reached a perfect score of 100% by the final epoch. The loss values also exhibited notable improvement during training, with training loss ranging from 0.12% to 0.03%, reaching its lowest value in the last epoch,.

Similarly, validation loss decreased progressively from an initial value of 65% to reach 0.06% as shown in Fig. 8b. The test datasets showed a minimal loss of only 0.05% in overall performance. Similarly, Fig. 8c demonstrate significant improvements in both training and validation accuracy for the SCUT-SFVD dataset, increasing from 93.5% to 100% and from 50% to 100%, respectively. However, there was a wide range of training losses, which decreased from 16% to 0.03%. The validation loss also varied but consistently declined during training, starting at 70% and reaching 0.27% at the last epoch, as illustrated in Fig. 8d. Additionally, the overall loss on test datasets was measured at 1.3%.

Upon completing the initial prediction and assessment phases, as well as obtaining the initial outputs, data, and results using the proposed FVPAD model based on DSC CNN for the FV training and validation datasets, the classification stage was conducted as the third step in the biometric system. This stage involves utilizing the FV testing datasets to enhance the accuracy and efficiency of the FVPAD model. The achievement of improved classification performance is determined through a performance evaluation method using a confusion matrix applied to our proposed model, especially on the testing datasets. The main objective at this stage is to differentiate between genuine and forged FV images in testing sets comprising 400 FV images from the IDIAP database and 4320 from the SCUT-SFVD database.

The confusion matrix represents the distribution of the testing datasets. In the IDIAP dataset, the confusion matrix values for TP, FP, TN, and FN were 200, 0, 200, and 0 respectively. Similarly in the SCUT-SFVD dataset: TP:2141, FP:3, TN:2157, and FN :19. These values obtained from the Confusion Matrix are used to compute performance measurements such as precision, recall, F1 score, and ACER for both databases

Table 9 shows the performance measurements of the confusion matrix. The precision value is around 100% for both databases, while the recall of 100% for IDIAP database and 99.12% for SCUT-SFVD database. In addition, in addition to the F1 score, the IDIAP database achieved a success rate of 100%, while SCUT-SFVD database achieved a success rate of 99.499%. The evaluation using confusion matrix method showcased high classification performance of the proposed FVPAD model.

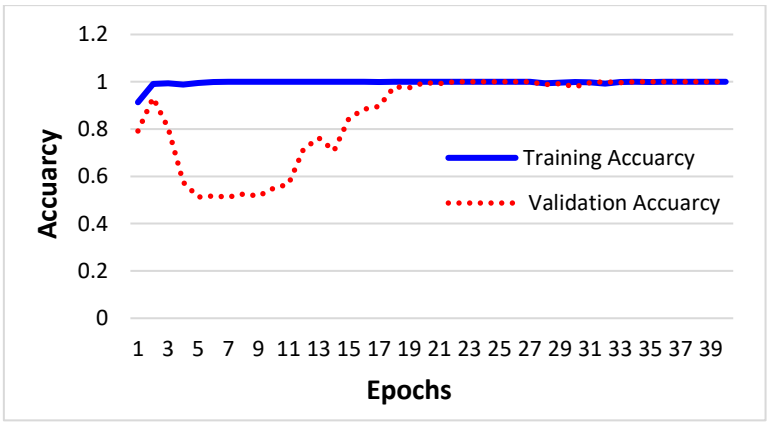
Table 9. Accuracy measurements of the proposed FVPAD model using DSC CNN-based approach.

Datasets	Accuracy	Precision	Recall	F1score
IDIAP	100%	100%	100%	100%
SCUT-SFVD	99.499%	100%	99.12%	99.49%

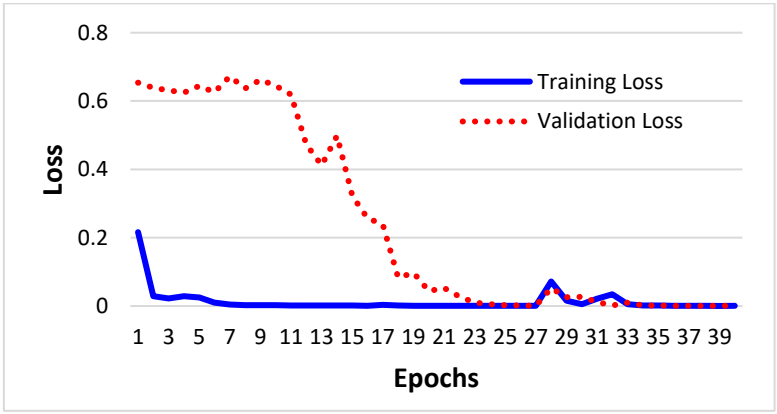
Consequently, Table 10 presents the performance results of the proposed FVPAD model for the FV datasets in both IDIAP and SCUT-SFVD databases. The evaluation is based on standardized ISO/IEC 30107-3 [59] metrics: APCER, BPCER, and ACER. For the IDIAP datasets, both APCER and BPCER achieved a low error rate of 0.0%, indicating strong performance by the proposed model for detecting fake and real-live FV images. The obtained results demonstrate that the proposed FVPAD model performs exceptionally well with an overall average ACER metric of 0.0%.

Table 10. Performance of the proposed FVPAD model using DSC CNN-based approach.

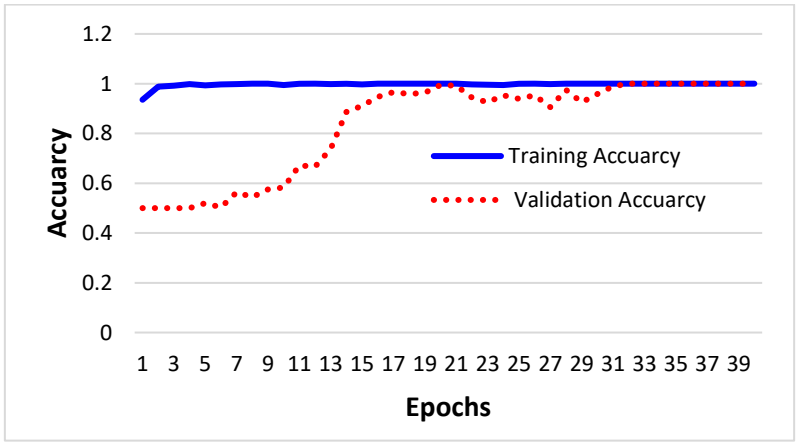
Datasets	ACER	APCER	BPCER
Proposed FVPAD (IDIAP)	0.0%	0.0%	0.0%
Proposed FVPAD (SCUT-SFVD)	0.005%	0.0013%	0.0087%



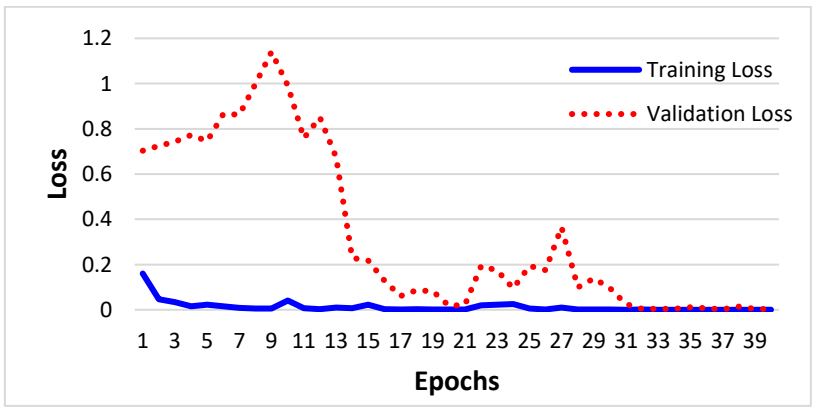
(a)



(b)



(c)



(d)

Fig. 8. The overall training and validation accuracy and losses: a) IDIAP dataset, training accuracy; b) IDIAP dataset, training loss; c) SCUT-SFVD dataset, training accuracy; d) SCUT-SFVD dataset, training loss.

In contrast, the SCUT-SFVD datasets achieved a low error measurement for the APCER metric of only 0.0013% and a BPCER metric of only 0.0087% due to using large FV datasets, resulting in an overall ACER average metrics of only 0.005%. Additionally, this value is identical to the ACER metric value achieved based on confusion matrix performance evaluation method. These values further highlight the effectiveness, reliability, and strong performance of the proposed model.

Finally, Table 11 compares the proposed FVPAD model with existing CNN-based models such as XceptionNet [19], FV2021 [48], VGG-16 [10, 19], and AlexNet [10, 19].

Table 11. Performance and hyperparameters comparisons of the Xception, FV2021, AlexNet, VGG-16 CNN-based models, and the proposed FVPAD model on the cropped FV datasets.

PAD method	No. of datasets	Resized image pixels	No. of epochs	Learning rate	No. of layers	CNN network based	No. of parameters	ACC
Xception CNN-based [48] (IDIAP)	120	96x96	NA	0.001	36 Conv + 1 FC	Original Xception CNN-based	20,822,768	(AUR-ROC) ROI = 100% Precision= 100%
FV2021 [48] (IDIAP)	120	96x96	NA	0.001	6 Conv + 1 FC	modified the First entry flow XceptionNet	314,632	(AUR-ROC) ROI= 99.97% Precision= 100%
PAD Alex-Net [10], [19] (IDIAP-PVD)	500	87×151	10	0.001	5 ConV + 3 FC	Customized AlexNet	1,191,168	ACC= 100%
PAD VGG-16 [10], [19] (IDIAP PVD)	500	128×256	10	0.001	2x8 ConV + 3 FC	Customized VGG 16	3,532,576	ACC= 100%
Proposed FVPAD (SCUT-SFVD)	1440	96x96	40	0.0001	15 ConV +1 FC	Customized the First entry flow XceptionNet	132,066	ACC= 99.499% Precision= 100%
Proposed FVPAD (IDIAP)	880	96x96	40	0.0001	15 ConV +1 FC	Customized the First entry flow XceptionNet	132,066	ACC= 100%, Precision= 100%

The evaluation criteria for this comparison include accuracy, hyperparameters, CNN-based approach, and FV datasets. Table 11 presents the performance and hyperparameters of a set of existing models based on deep convolutional neural networks, as well as the proposed FVPAD model. While all existing CNN models achieved excellent performance on the cropped FV datasets, ranging between 99-100%, they exhibited complexity in their structure with a large number of parameters and high computational load. For instance, deeper models such as XceptionNet, VGG-16, and AlexNet showed high performance but required a significant

increase in parameters and computational requirements during implementation. On the other hand, FV2021 achieved high performance with few parameters by using a 7x7 kernel size, which allowed for greater coverage while potentially sacrificing precise details. Additionally, the FV2021 dataset was sourced from eight databases, including the IDIAP database, which contained 120 FV images as presented in Table 11.

Among these results, the proposed FVPAD model achieved impressive scores of 100% for the IDIAP Database and 99.499% for SCUT-SFVD Database. Additionally, this model demonstrated remarkable outcomes with a low number of trainable parameters of 131.106, resulting in less complexity, simplicity, accelerated speed, and minimizing computational loads all without requiring special equipment's or specific environment for implementation.

Finally, the models were compared based on their CNN-based architecture, hyperparameters, and the number of parameters used. The proposed FVPAD model demonstrated high accuracy and efficiency while outperforming existing models in similar areas with comparable parameter usage. This led to the development of a less complex model that achieves high performance in detecting finger vein spoofing attacks using DSC CNN-based applications.

5. CONCLUSIONS AND FUTURE WORK

In this study, a FVPAD model was developed using deep learning techniques based on CNN. The proposed model was tested on the IDIAP and SCUT-SFVD databases, effectively distinguishing between fake and real (live) FV images. Additionally, a DSC CNN-based model was created and validated. Experimental results demonstrate that the proposed FVPAD achieved superior performance with low complexity, resulting in 131.106 trainable parameters using 15 convolution layers. Furthermore, it is characterized by efficient resource utilization, depth training, robust feature extraction capability as well as speed and high accuracy in detection and classification with rates of 100% for IDIAP datasets and 99.5% for SCUT-SFVD datasets. Therefore, when applying the FVPAD model, using a large number of parameters is unnecessary as they do not affect the computation process. As a result, it is concluded that the proposed model is lightweight and does not demand specific resources or an operating environment with strict specifications.

In future work, the proposed finger vein presentation attack detection model will be targeted in order to establish a biometric FV detection system that is easy to implement and highly efficient. Specifically, this will involve developing a minimally complex CNN-based deep learning model with fewer parameters, making it adaptable to new datasets growth or capable of adjusting when presented with unfamiliar data, while effectively detecting spoofing attacks.

REFERENCES

[1] J. Hashimoto, "Finger Vein authentication technology and its future," Symposium on VLSI Circuits, 2006, pp. 5–8. doi: 10.1109/VLSIC.2006.1705285.

[2] K. Win, K. Li, J. Chen, P. Viger, K. Li, "Fingerprint classification and identification algorithms for criminal investigation: a survey," *Future Generation Computer Systems*, vol. 110, pp. 758–771, 2020, doi: 10.1016/j.future.2019.10.019.

[3] R. Ramachandra, C. Busch, "Presentation attack detection methods for face recognition systems: a comprehensive survey," ACM Computing Surveys, vol. 50, no. 1, pp. 1-37, 2017, doi: 10.1145/3038924.

[4] A. Boyd, Z. Fang, A. Czajka, K. Bowyer, "Iris presentation attack detection: where are we now?," *Pattern Recognition Letter*, vol. 138, pp. 483–489, 2020, doi: 10.1016/j.patrec.2020.08.018.

[5] K. Wang, H. Ma, O. Popoola, J. Liu, "Finger vein recognition," *Biometrics*, 2011, doi: 10.5772/18025.

[6] P. Taylor, T. Dargahi, A. Dehghantanha, R. Parizi, K. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147–156, 2020, doi: 10.1016/j.dcan.2019.01.005.

[7] R. Krishnan., "Image quality assessment for fake biometric detection: application to finger-vein images," *International Journal of Advanced Research*, vol. 4, no. 8, pp. 2015–2021, 2016, doi: 10.21474/IJAR01/1418.

[8] L. Chen, T. Guo, L. Li, H. Jiang, W. Luo, Z. Li, "A finger vein liveness detection system based on multi-scale spatial-temporal map and light-vit model," *Sensors*, vol. 23, no. 24, p. 9637, 2023, doi: 10.3390/s23249637.

[9] J. Kolberg, M. Gomez-Barrero, S. Venkatesh, R. Ramachandra, C. Busch, *Presentation Attack Detection For Finger Recognition*, *Handbook of Vascular Biometrics*, Cham: Springer International Publishing, 2020.

[10] D. Nguyen, H. Yoon, T. Pham, K. Park, "Spoof detection for finger-vein recognition system using NIR camera," *Sensors*, vol. 17, p. 2261, 2017, doi: 10.3390/s17102261.

[11] S. Dargan, M. Kumar, "A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities," *Expert Systems with Applications*, vol. 143, p. 113114, 2020, doi: 10.1016/j.eswa.2019.113114.

[12] R. Ramachandra, C. Busch, ''Presentation attack detection algorithms for finger vein biometrics: a comprehensive study,'' International Conference on Signal-Image Technology & Internet-Based Systems, 2015. doi: 10.1109/SITIS.2015.74.

[13] C. Szegedy, W. Liu, Y, Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi: 10.1109/CVPR.2015.7298594.

[14] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 2012, doi: 10.1145/3065386.

[15] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint,* vol. 1 2014, doi: 10.48550/arXiv.1409.1556

[16] Y. Wang, D. Shi, W. Zhou, "Convolutional neural network approach based on multimodal biometric system with fusion of face and finger vein features," *Sensors*, vol. 22, no. 16, 2022, doi: 10.3390/s22166039.

[17] S. Kassani, P. Kassani, R. Khazaeinezhad, M. Wesolowski, K. Schneider, R. Deters, "Diabetic retinopathy classification using a modified xception architecture," IEEE international symposium on signal processing and information technology, 2019, doi: 10.1109/ISSPIT47144.2019.9001846.

[18] N. Kumar, M. Gupta, D. Gupta, S. Tiwari, "Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 469–478, 2023, doi: 10.1007/s12652-021-03306-6.

[19] K. Shaheed, A. Mao, I. Qureshi, Q. Abbas, M. Kumar, X. Zhang, "Finger-vein presentation attack detection using depthwise separable convolution neural network," *Expert Systems with* Applications, vol. 198, p. 116786, 2022, doi: 10.1016/j.eswa.2022.116786.

[20] F. Chollet, "Xception: deep learning with depthwise separable convolutions," IEEE Conference on Computer Vision and Pattern Recognition, 2017, doi: 10.1109/CVPR.2017.195.

[21] Z. Zhang, M. Wang, "Finger vein recognition based on lightweight convolutional attention model," *IET Image Process*, vol. 17, 2023, doi: 10.1049/ipr2.12761.

[22] S. Li, B. Zhang, S. Zhao, J. Yang, "Local discriminant coding based convolutional feature representation for multimodal finger recognition," *Information Sciences*, vol. 547, pp. 1170–1181, 2021, doi: https://doi.org/10.1016/j.ins.2020.09.045.

[23] F. Juefei-Xu, V. N. Boddeti, M. Savvides, "Local binary convolutional neural networks," IEEE Conference on Computer Vision and Pattern Recognition, 2016, doi: 10.1109/CVPR.2017.456.

[24] J. Schuiki, B. Prommegger, A. Uhl, "Confronting a variety of finger vein recognition algorithms with wax presentation attack artefacts," IEEE International Workshop on Biometrics and Forensics, 2021, doi: 10.1109/IWBF50991.2021.9465091.

[25] P. Tome, M. Vanoni, S. Marcel, "On the vulnerability of finger vein recognition to spoofing," International Conference of the Biometrics Special Interest Group, 2014, https://api.semanticscholar.org/CorpusID:4883590.

[26] D. Gragnaniello, C. Sansone, L. Verdoliva, "Iris liveness detection for mobile devices based on local descriptors," *Pattern Recognition Letters*, vol. 57, pp. 81–87, 2015, doi: 10.1016/j.patrec.2014.10.018.

[27] I. Chingovska, A. Anjos, S. Marcel, ''On the Effectiveness of local binary patterns in face anti-spoofing,'' Proceedings of the International Conference of Biometrics Special Interest Group , 2012.

[28] X. Qiu, W. Kang, S. Tian, W. Jia, Z. Huang, "Finger vein presentation attack detection using total variation decomposition," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 465–477, 2018, doi: 10.1109/TIFS.2017.2756598.

[29] A. Toosi, S. Cumani, A. Bottino, *On Multiview Analysis For Fingerprint Liveness Detection*, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Cham: Springer International Publishing, 2015.

[30] H. Zayed, H. Hamid, Y. Kamal, A. Zekry, "A comprehensive survey on finger vein biometric," *Journal of Advances in Information Technology*, vol. 14, no. 2, 2023, doi: 10.12720/jait.14.2.212-223.

[31] I. Boucherit, M. Zmirli, H. Hamza, B. Rosdi, "Finger vein identification using deeply-fused convolutional neural network," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, 2020, doi: 10.1016/j.jksuci.2020.04.002.

[32] L. Weng, X. Li, W. Wang, ''Finger vein recognition based on deep convolutional neural networks,'' International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, 2020, doi: 10.1109/CISP-BMEI51763.2020.9263601.

[33] H. Qin, M. El-Yacoubi, "Deep representation for finger-vein image-quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1677–1693, 2018, doi: 10.1109/TCSVT.2017.2684826.

[34] Y. Zhang, Z. Liu, "Research on finger vein recognition based on sub-convolutional neural network," International Conference on Computer Network, Electronic and Automation, 2020, doi: 10.1109/ICCNEA50255.2020.00051.

[35] D. Zhao, H. Ma, Z. Yang, J. Li, W. Tian, "Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization," *Infrared Physics & Technology*, vol. 105, p. 103221, 2020, doi: https://doi.org/10.1016/j.infrared.2020.103221.

[36] R. Das, E. Piciucco, E. Maiorana, P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2019, doi: 10.1109/TIFS.2018.2850320.

[37] H. Hu, W. Kang, Y. Lu, Y. Fang, H. Liu, J. Zhao, F. Deng, ''FV-Net: learning a finger-vein feature representation based on a CNN,'' International Conference on Pattern Recognition, 2018, doi: 10.1109/ICPR.2018.8546007.

[38] H. Huang, S. Liu, H. Zheng, L. Ni, Y. Zhang, W. Li, "DeepVein: novel finger vein verification methods based on Deep Convolutional Neural Networks," International Conference on Identity, Security and Behavior Analysis, 2017, doi: 10.1109/ISBA.2017.7947683.

[39] Y. Lu, S. Xie, S. Wu, "Exploring competitive features using deep convolutional neural network for finger vein recognition," *IEEE Access*, vol. 7, pp. 35113–35123, 2019, doi: 10.1109/ACCESS.2019.2902429.

[40] E. Jalilian, A. Uhl, "Finger-vein recognition using deep fully convolutional neural semantic segmentation networks: the impact of training data," IEEE International Workshop on Information Forensics and Security, 2018, doi: 10.1109/WIFS.2018.8630794.

[41] K. Noh, J. Choi, J. Hong, K. Park, "Finger-vein recognition based on densely connected convolutional network using score-level fusion with shape and texture images," *IEEE Access*, vol. 8, pp. 96748 - 96766, 2020, doi: 10.1109/ACCESS.2020.2996646.

[42] S. Tang, S. Zhou, W. Kang, Q. Wu, F. Deng, "Finger vein verification using a Siamese CNN," *IET Biomatrices*, vol. 8, no. 5, pp. 306–315, 2019, doi: 10.1049/iet-bmt.2018.5245.

[43] R. Ramachandra, S. Venkatesh, K. Raja, C. Busch, ''Transferable deep convolutional neural network features for fingervein presentation attack detection,'' 2017, International Workshop on Biometrics and Forensics, doi: 10.1109/IWBF.2017.7935108.

[44] W. Yang, W. Luo, W. Kang, Z. Huang, Q. Wu, "FVRAS-Net: an embedded finger-vein recognition and antispoofing system using a unified CNN," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8690–8701, 2020, doi: 10.1109/TIM.2020.3001410.

[45] H. Ren, L. Sun, J. Guo, C. Han, F. Wu, "Finger vein recognition system with template protection based on convolutional neural network," *Knowledge-Based Systems*, vol. 227, p. 107159, 2021, doi: 10.1016/j.knosys.2021.107159.

[46] H. Nguyen, J. Yamagishi, I. Echizen, "Use of a capsule network to detect fake images and videos," IEEE International Conference on Acoustics, Speech and Signal Processing, doi: 10.1109/ICASSP.2019.8682602

[47] S. Sabour, N. Frosst, G. Hinton, "Dynamic routing between capsules," Conference on Neural Information Processing Systems, 2017, doi: arxiv.org/abs/1710.09829

[48] B. Maser, A. Uhl, "Using CNNs to identify the origin of finger vein sample images," IEEE International Workshop on Biometrics and Forensics, 2021, doi: 10.1109/IWBF50991.2021.9465077.

[49] M. Taye, "Theoretical Understanding of convolutional neural network: concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, 2023, doi: 10.3390/computation11030052.

[50] R. Yamashita, M. Nishio, R. Do, K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.

[51] L. Alzubaidi, J. Zhang, A. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. Fadhel, M. Al-Amidie, L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[52] M. Hussain, J. Bird, D. Faria, ''A study on CNN transfer learning for image classification,'' Advances in Computational Intelligence Systems, 2018, doi: 10.1007/978-3-319-97982-3_16.

[53] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: efficient convolutional neural networks for mobile vision applications," *Computer Vision and Pattern Recognition*, vol. 1, 2017, doi: 10.48550/arXiv.1704.04861.

[54] H. Farag, L. Said, M. Rizk, M. Ahmed, ''Hyperparameters optimization for ResNet and Xception in the purpose of diagnosing COVID-19,'' *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 2, pp. 3555-3571, 2021. doi: 10.3233/JIFS-210925.

[55] Y. Nan, J. Ju, Q. Hua, H. Zhang, B. Wang, "A-MobileNet: an approach of facial expression recognition," *Alexandria Engineering Journal*, vol. 61, no. 6, pp. 4435–4444, 2022, doi: https://doi.org/10.1016/j.aej.2021.09.066.

[56] J. Shang, N. Phipps, I. Wey, T. Teo, "A-DSCNN: depthwise separable convolutional neural network inference chip design using an approximate multiplier," *Chips*, vol. 2, no. 3, pp. 159–172, 2023, doi: 10.3390/chips2030010.

[57] J. Brownlee, *Deep Learning with Python: Develop Deep Learning Models on Theano and Tensorflow Using Keras,* Machine Learning Mastery, 2016.

[58] J. Brownlee, *Develop Deep Learning Models on Theano and Tensorflow Using Keras*, Machine Learning Mastery, 2019.

[59] C. Busch, H. Darmstadt, "The ISO/IEC standards for testing of presentation attack detection," *International Organization for Standardization*, 2017, https://www.nist.gov/system/files/documents/2020/09/15/12_buschthieme-ibpc-pad-160504.pdf.