Expert Systems With Applications 191 (2022) 116288

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

DS-CNN: A pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition

Kashif Shaheed [a,*], Aihua Mao [a,*], Imran Qureshi [b,c], Munish Kumar [d], Sumaira Hussain [e], Inam Ullah [e], Xingming Zhang [a]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology (MCS-NUST), Islamabad 44000, Pakistan
[c] Key Laboratory of Space Photoelectric Detection and Perception, Ministry of Industry and Information Technology, College of Astronautic, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China
[d] Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India
[e] TIME Lab, School of Software Engineering, Shandong University, Jinan, China

ARTICLE INFO

Keywords:
Biometric
Convolutional neural network
Classification
Deep learning
Finger vein recognition
Transfer learning
Xception

ABSTRACT

Finger vein recognition received special attention among all other biometric traits due to its high security. Adequate recognition and classification accuracy ensure the security of personal authentication. Many convolutional neural networks (CNNs) have been proposed with a promising performance in biometric finger vein recognition. However, their architectures have several problems, such as high complexity, extraction of robust features, degraded performance, etc. Considering the issues of CNNs, the authors present a pre-trained CNN network named Xception model based on depth-wise separable CNNs with residual connection, which is considered to be a more effective, less complex neural network to extract robust features. Our work can be seen as a three-stage process. Initially, the concept of data pre-processing is applied to convert the raw input samples into the standard format. Afterward, data augmentation using different geometrical techniques is incorporated to overcome the lack of training samples required for training the deep learning model. Finally, the feature extraction and classification task is performed through the pre-trained Xception architecture to verify the person's identity. SDUMLA and THU-FVFDT2 datasets are utilized to test and evaluate the proposed multi-layered CNN model performance with existing arts. Our proposed method for the SDUMLA database achieved an accuracy of 99% with an F1-score of 98%. While on THU-FVFDT2, the proposed method obtained an accuracy of 90% with an F1-score of 88%. Experimental results conclude that the proposed work obtained excellent performance compared to existing methods.

1. Introduction

Personal identification using biometric has obtained worldwide attention and growing demand for access control, border control, financial security, and attendance systems (Hou & Yan, 2019). A Biometric system is considered a highly secure and safe method to measure human distinctive physical or behavioral characteristics for a person's fast identification, which is still challenging for scientific and industrial groups. Many types of biometric traits have been proposed, such as fingerprint (Serafim et al., 2019; Win, Li, Chen, Viger, & Li, 2020; Yuan, Xia, Sun, & Wu, 2020), face (Huang, Li, Chen, & Tang, 2019; Ranjan et al., 2019), facial expression (Liu, Zhang, Lin, & Wang, 2020), iris (Liu, Zhou, Shang, & Xu, 2019; Wang, Muhammad, Wang, He, & Sun, 2020), palm vein (Aberni, Boubchir, & Daachi, 2020), ECG (Abdeldayem & Bourlai, 2019), EEG (Wilaiprasitporn et al., 2020), finger vein (Kuzu et al., 2020), retina (Sadikoglu & Uzelaltinbulat, 2016) and so on. Finger vein is an intrinsic biometric method that is harder to steal and forge because it is located inside the hypodermic layer (Qin & El-Yacoubi, 2017). The Finger vein identification method has several advantages compared to other biometric methods: (1) rapid processing with small image data. (2) Non-contact: skin condition does not affect the performance of the system. (3) Biometric: Identification of living body because of the vein pattern of a finger. (4) High Safety: hard to replicate because the vein pattern is inside features (Luo, Yu, Pan, Chu, & Tsai, 2010; Liu

* Corresponding authors.
E-mail addresses: kashifshaheed1@gmail.com (K. Shaheed), ahmao@scut.edu.cn (A. Mao).

https://doi.org/10.1016/j.eswa.2021.116288
Received 3 April 2021; Received in revised form 29 October 2021; Accepted 22 November 2021
Available online 2 December 2021
0957-4174/

et al., 2018; Wu & Ye, 2009). The finger vein recognition method mainly includes four stages, i.e., image acquisition, image pre-processing, feature extraction, and identification. Pre-processing comprises numerous steps: region of interest extraction (ROI), image normalization, and image enhancement. Pre-processing of the finger vein image involves various operations to extract ROI from the original image and enhance the finger vein image (Gupta & Gupta, 2015). Afterward, suitable features are extracted from the finger vein image in the feature extraction step. And finally, data are identified in the identification step.

For image enhancement in the pre-processing several step filter-based and transform-based methods have been used, for instance, Gabor filter (Yang, Shi, & Yang, 2011; Yang & Shi, 2012), Log-Gabor filter (Hajian & Ramli, 2018), Curvelet transform (Zhongbo, Siliang, & Xiao, 2006), and Discrete cosine transforms (Ismail & Mohamed, 2019). Gabor filter was presented to enhance the finger vein image, and then Phase-only-correlation (POC) method was used to identify the finger-vein image (Wu & Ye, 2009). Gabor filter was introduced as a feature extraction method to extract finger vein features in different directions and levels (Yang et al., 2011). These extracted features were used to design a finger vein code to identify finger vein images. Another study (Zhongbo et al., 2006) is based on Curvelet transform, where multi-scale enhancement is employed to improve image quality while finger vein identification would be made with local interconnection neural network. Fusion techniques based on the Discrete Cosine transform scheme were presented to enhance finger vein image and improve image quality (Ismail & Mohamed, 2019). But, these image enhancement approaches are not adjustable with the width of the vein; the noise also enhances if it is closely similar to the vein structure. A repeated line tracking method was proposed to detect valley features by computing the difference between the center pixel value and the pixel value (Miura, Nagasaka, & Miyatake, 2004). Miura, Nagasaka, and Miyatake (2007) have modified the repeated line tracking method and computed the maximum curvature during vein tracking.

Moreover, the sliding window-based ROI extraction-based method was proposed to enhance the finger vein network (Yang, Yang, Yin, & Xiao, 2013). A dual-sliding window model was proposed to further improve the proposed method (Qiu, Liu, Zhou, Huang, & Nie, 2016). However, it is still noticed that the thin vein is not efficiently enhanced because of the small width. After preprocessing, the next step is feature extraction for identification, for instance, local line binary pattern (Rosdi, Shing, & Suandi, 2011), Principal Component Analysis (Wu & Liu, 2011a; Yang, Xi, & Yin, 2012), Random transform (Qin & El-Yacoubi, 2017), pattern map (Beng & Rosdi, 2011) and hybrids methods (Wu & Liu, 2011b) (Yang et al., 2012). E.g., in (Rosdi et al., 2011), a local line binary pattern (LLBP) feature is extracted. Then, the matching accuracy was computed by Hamming distance. A method proposed in (Qin & El-Yacoubi, 2017) used Random transform to obtain the feature, and then artificial neural network classifiers were used for finger vein identification. Pattern maps based on pixel-pattern-based texture features (PPBTF) and PCA were combined for finger vein identification in (Beng & Rosdi, 2011). Feature extraction with popular dimensionality reduction approaches such as PCA (Wu & Liu, 2011a) and $(2D)^2$ PCA (Yang et al., 2012) was implemented as a feature extraction before training the neural network for identification. These methods reported high identification accuracy. Extraction of features in finger vein identification could be carried out in different ways. Wu and Liu (2011b) have proposed a method, which combines PCA and LDA to obtain attributes. And the SVM model was trained for authentication. Also, in (Park, 2011), local and global feature operators were used to extract finger vein features. Carrera, Izurieta, and Carrera (2018) have proposed a biometric finger vein recognition system based on a textural feature without the involvement of segmentation and structural analysis, which is important for the biometric process. The textural features employed in this study were derived from wavelet detail coefficients in finger-vein pictures using grey-level co-occurrence matrices. A machine learning classifier KNN is used for classification task and shows promising accuracy of 91%. However, it is not easy for a finger vein recognition system to design an efficient feature descriptor model because the finger vein shape and structure do not conform to specific patterns. Table 1 introduces a summary of the traditional approaches employed for finger vein identification. It illustrates the used datasets, the achieved results, and finally, the study compares its outcomes to others.

In recent times, a deep learning-based method has been effectively operated in finger vein identification. Deep learning-based approaches have been modeled from a deep neural network, which presents impressive image processing skills. They have a good performance on a noisy dataset with a powerful ability of adaptive feature representation. For example, the study (Huang et al., 2017) designed a finger vein verification method called Deep-Vein, which reported an excellent matching accuracy. He, Li, Chen, and Peng (2017) proposed a segmentation-based way to extract features from finger vein images and improve system recognition performance. To further enhance the finger vein verification performance, various CNN architecture (Ahmad Radzi, Khalil-Hani, & Bakhteri, 2016; Boucherit, Zmirli, Hentabli, & Rosdi, 2020; Das, Piciucco, Maiorana, & Campisi, 2018) and loss function (Bansal & Balas, 2020) have been investigated and researched, which show impressive over other finger vein identification methods. Xie and Kumar (2019) recommended a matching method based on CNN and a supervised discrete hashing method for finger vein authentication. The proposed work significantly improves the finger vein system's matching performance and reduces the template size. Yang, Hui, Chen, Xue, and Liao (2019) suggested a finger vein representation method using generative adversarial networks (FV-GAN) to extract and verify the vein image to address the computational complexity problem. The performance of the system is significantly improved in terms of verification accuracy and error rate. However, the proposed network training was considered unstable and difficult, which might degrade the system's performance. Tang, Zhou, Kang, Wu, and Deng (2019) presented a Siamese CNN with a modified contrastive loss function to extract discriminative features from finger vein images to advance the system performance. Zhao et al. (2019) proposed a CNN model using curvature grey images to improve the finger vein recognition system performance. However, the proposed work involved too much image processing work, making the method not computationally efficient. Table 2 summarizes deep learning-based finger vein recognition arts with the detail of used benchmarks, obtained results, and finally if the study compares its outcomes to others.

Table 1
Traditional finger vein verification methods.

Reference	Method	Database	Number of Samples	Accuracy
(Yang et al., 2011)	Bank of Gabor filter	Own	2100 finger vein images	98%
(Yang et al., 2013)	Sliding Window-Based Finger Vein Images ROI extraction	HKPU	3132 finger vein images	94%
(Qiu et al., 2016)	Dual-sliding window localization and pseudo-elliptical transformer	SDUMLA	636 finger vein images	99 %
(Wu & Liu, 2011a)	PCA and ANFIS	Own	100 finger vein images	99%
(Yang et al., 2012)	$(2D)^2$PCA and Metric Learning	Own	42 finger vein images	99%
(Beng & Rosdi, 2011)	PPBTF and PCA	UTM	2040 finger vein images	98%
(Wu & Liu, 2011b)	PCA and LDA	Own	100 finger vein images	98%
(Carrera et al., 2018)	Textural Feature	SDUMLA	636 finger vein images	91%

Table 2
Deep learning-based finger vein verification arts

Reference	Method	Database	Number of Samples	Accuracy
(He et al., 2017)	Back Propagation Neural Network	Self-Built	600 finger vein images	81%
(Ahmad *et al.*, 2016)	CNN	UTM	500 finger vein images	99%
(Boucherit et al., 2020)	Deeply fused Convolutional Network	THUFVDT2	610 finger vein images	99%
(Das et al., 2018)	CNN with CLAHE	UTFVP	360 finger vein images	98%
(Yang et al., 2019)	FV-GAN	THUFVDT2	540 finger vein images	98%
(Zhao et al., 2019)	CNN using Curvature Gray Image	Self-Built	3000 finger vein images	96%

1.1. Motivation & our contribution

Traditional finger vein recognition schemes involve much processing to remove noise and extract features before classification performance. Therefore, these conventional-based finger vein recognition methods have low robustness and high complexity. CNN has been a potent deep learning tool for various computer vision tasks, for example, object detection (Zhao, Zheng, Xu, & Wu, 2019), recognition, and classification (Al-Waisy, Qahwaji, Ipson, Al-Fahdawi, & Nagem, 2018; Zhang, Wu, You, & Zhang, 2017), medical image analysis (Li et al., 2018). In finger vein recognition, deep learning-based approaches have shown outstanding performance in the recent past. However, the limitation of previously developed deep learning algorithms was that they required a large number of training samples to train the network for model performance, which is unrealistic in finger vein recognition(Ou et al., 2021). Another issue with the existing CNN model is that it requires a large number of parameters and has a high computational cost, making those methods unsuitable for real-time finger vein recognition.

The variant of deep learning, transfer learning, is something that most researchers and data scientists believe can further advance computational intelligence tasks. Andrew Ng, a renowned professor, and data scientist from Stanford University, USA, already mentioned that Transfer learning would be the next diver of machine learning commercial success. Transfer learning is broadly categorized under three settings: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning (Ribani & Marengoni, 2019). Transfer learning using deep neural networks has already been used in various computer vision and recognition problems, one of the common types of inductive transfer learning (Lu et al., 2015). In the biometric domain, transfer learning using a deep neural network model has already been introduced and showed remarkable performance in many biometric traits such as fingerprint (Liu et al., 2018), face (Uchoa, Aires, Veras, Paiva, & Britto, 2020), iris (Gautam & Mukhopadhyay, 2018), ECG (Zhang, Zhao, Guo, Huang, & Xu, 2019), etc. Few studies have also been reported, which show the implementation of a pre-trained model in the finger vein recognition system (Fairuz, Habaebi, & Elsheikh, 2019; Fairuz, Habaebi, & Elsheikh, 2018; Hong, Lee, & Park, 2017). However, the model used in these methods was also computationally complex. Few studies have been found about the pre-trained CNN Xception model in the literature. Recently, the Xception model used in various image classification (Ayan & Ünver, 2019; Kassani et al., 2019; Lo, Yang, & Wang, 2019) and computer vision tasks showed remarkable performance compared to other pre-trained CNN models (Chen, Yang, & Zhang, 2020). It is considered a more powerful CNN model with fewer overfitting issues than other pre-trained models (Lo et al., 2019). This special CNN architecture performs well because of its inception module, residual block, and depth-wise separable convolution. Kang *et al.* (2020) developed a 3D finger vein feature extraction and matching strategies based on CNN depth-wise separable convolution. The study demonstrates excellent performance on a full view of 3D finger vein images. Inspired by these works, we introduced a pre-trained Xception model based on a depth-wise separable convolution neural network to overcome the existing finger vein system deficits and improve the finger vein system performance. To the best of my knowledge, this paper is the first attempt to use them in single mode finger vein image authentication and address the complexity problem in existing network methods. We used the pre-trained Xception architecture, which was already trained on a large ImageNet dataset. The pre-trained model performs excellently on a small dataset. We used the Xception model to extract the feature from finger vein images, and a fully connected layer was used to verify the person's identity.

The main contribution of our paper is defined below:

- A pre-trained model called Xception is based on a depth-wise separable convolutional neural network (DS-CNN) to recognize and classify finger vein images. The depth-wise separable convolution layers and residual connection make the Xception model learn the feature efficiently from finger vein images.
- The significant merit is designing a cost-efficient, automated, and lighter deep neural network for finger vein recognition which advanced the system performance.
- A comparative assessment of the learning model is presented regarding recognition accuracy and computational complexity to investigate other CNN model performances.
- Experimental result shows that the proposed method shows promising results and achieved an acceptable level of recognition accuracy on small datasets

The manuscript is systematized as follows: Section 2 gives theoretical background about transfer learning and CNN architecture with common layers, Section 3 presents the method and materials in detail, followed by experiment result and analysis in Section 4, disscussion of this research work is presented in Section 5, and finally, the paper concluded in Section 6 with future remarks.

2. Background knowledge

2.1. Transfer learning

Since 1995, transfer learning-based research has gained growing interest with different titles: life-long learning, learning to learn, inductive transfer, knowledge transfer, knowledge consolidation, multitask learning, context-sensitive learning, *meta*-learning, and incremental learning (Pan & Yang, 2010). The multitask learning method is a very much related learning scheme to transfer knowledge because it learns multiple tasks simultaneously (Ribani & Marengoni, 2019). Fig. 1 represents the difference between the traditional machine learning process of two different functions and the transfer learning approach for the same tasks. In transfer learning, task 1 is trained on large datasets, and task 2 uses the knowledge acquired from task 1 to learn faster and enhance accuracy.

There are various strategies of transfer learning methods, such as inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. These TL strategies are categorized based on the availability of the labelled data. Our work has adopted a pre-trained deep learning model, one of the common types of inductive TL method using source domain and task, and fine-tunning different layers for the target task. We adopted the pre-trained TL method because there is insufficient finger vein training data to train the model from scratch. Also, the computation cost is the other element to use a pre-trained network.

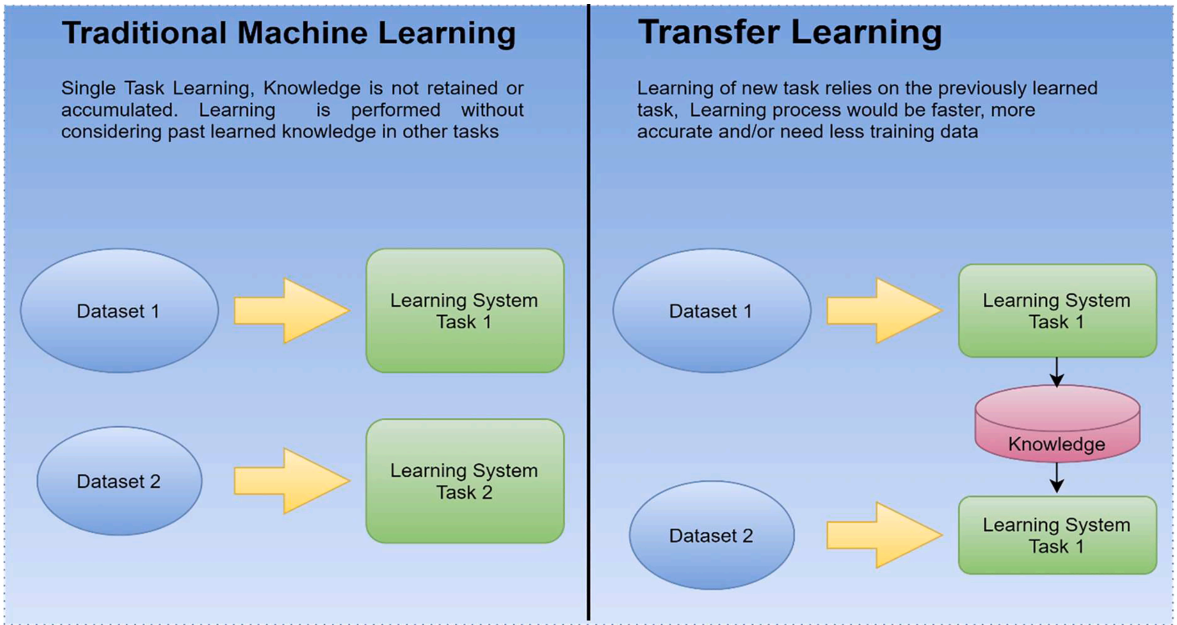


Fig. 1. Difference between Traditional Machine Learning and Transfer Learning (Dipanjan Sarkar, 2018).

2.2. Transfer learning concept in CNN

Practically, a vast amount of data is needed to train the CNN model. It is complicated to organize a large amount of data for important problems in some scenarios. In most cases, for real-world applications, it is a difficult task to obtain matching training and testing data. Hence, a transfer learning concept has been introduced. Transfer learning is one of the most renowned machine learning concepts that learned the background information used to solve one problem and reused it on the other related issue. First, the base network is trained for a particular problem on their respective database and then moved to the target dataset target problem (Khan, Islam, Jan, Ud Din, & Rodrigues, 2019).

2.3. Convolutional neural network

In 1998, CNN was first introduced by Fukushima; now, CNN has wide object detection applications, image classification, facial expression recognition, etc. (Dhillon & Verma, 2020). The deep learning scheme CNN is composed of convolutional, pooling, and fully connected layers. These layers play a vital role in training efficiently. Every layer of CNN has neurons with weight and biases that can be learned and processed the input. Each layer has the aim to extract patterns in the local region of the input image.

Generally, the convolution layer consists of a learnable filter, F with kernel size l, against the small set of $\times$ images of size $A \times B$, with weight matrices w. The convolution operator, V for n local connection and one dimensional can be computed by equation (1);

$$V_j = \sum_{i=1}^{l} w_i + x_{i+j-1} j = n \quad (1)$$

Fig. 2 represents the basic structure of the CNN model.

Because of its enormous potential, the CNN model is accepted by the research community. The convolution layer in CNN operates a mutual weighting mechanism to locate every image pixel using a single filter. There are numerous benefits of using the CNN model. A limited number of pre-processing steps are needed; human interaction and past knowledge are not necessary to train the CNN model (Abbas, Ibrahim, & Jaffar, 2018). Moreover, CNN is considered more computationally effective than other networks due to the fewer parameters used in hidden layers to run multiple experiments (Qureshi, Ma, & Abbas, 2021).

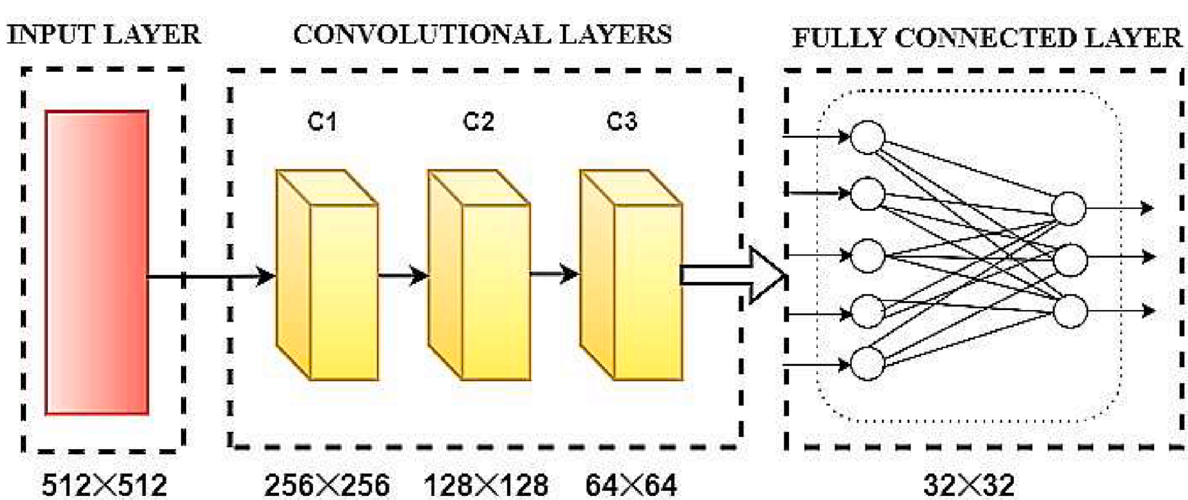


Fig. 2. Primary Taxonomy of CNN Model.

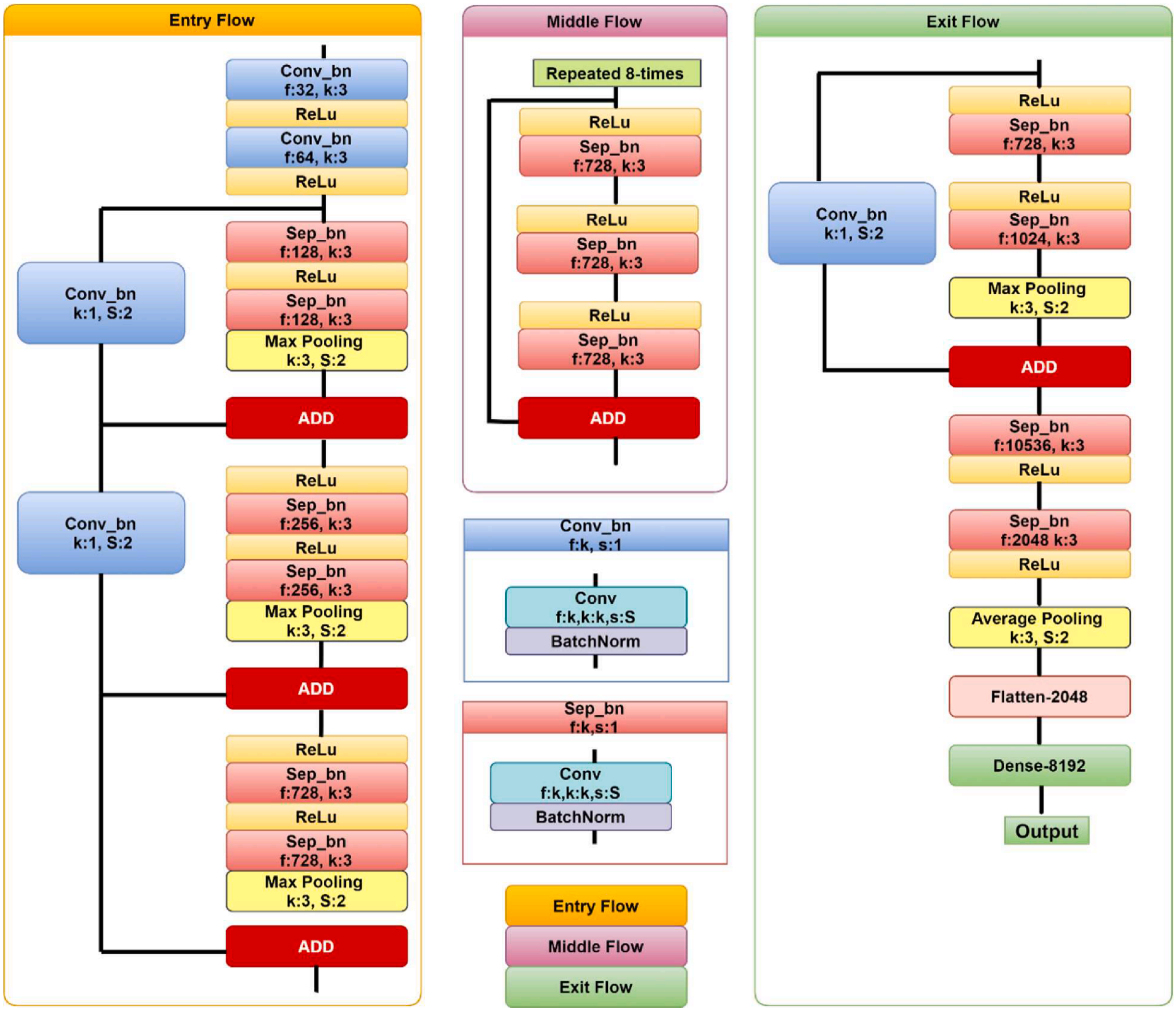


Fig. 3. The architecture of the Proposed Pre-trained Network Model.

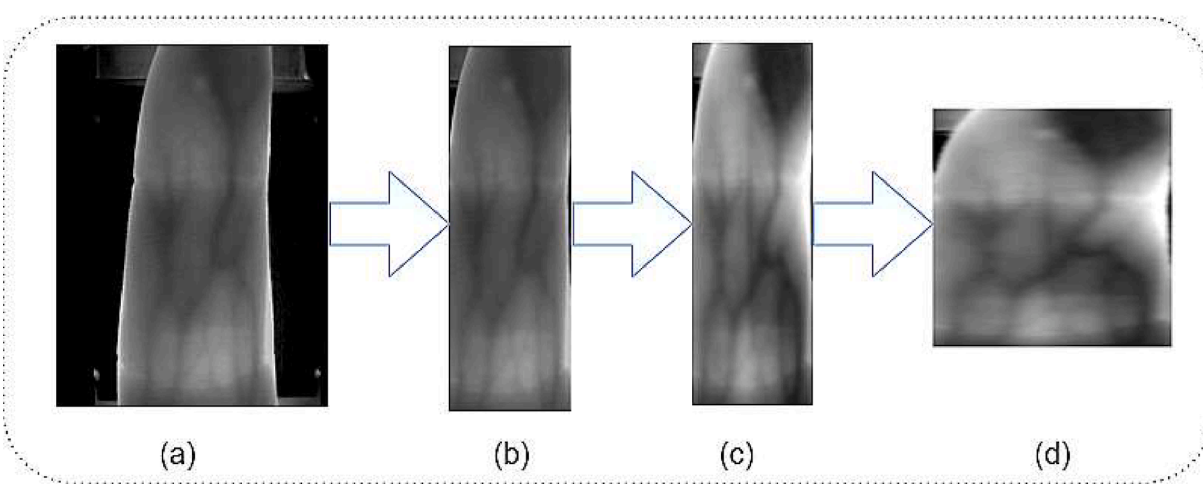


Fig. 4. Shows the pre-processing step of our work. (a) Raw finger vein image (b) ROI Extraction (c) Enhanced Image (d) Reshape image.

Table 3
Data Augmentation with parameters.

Method	Parameters
Rotation_range	15
Width_shift_range	0.2
Shear_range	0.2
Zoom_range	0.2
Horizontal_flip	True
Vertical_flip	False

We have used the pre-trained CNN model Xception model, which we have discussed in the next section.

3. Methods and materials

This section provides a detailed introduction of each part of the proposed CNN architecture capable of identifying and classifying finger vein images. The pre-processing step is first performed on both datasets images; then, low-level features from finger vein images are extracted using a pre-trained model called Xception. Finally, the output feature is fed into two fully connected layers for classification tasks shown in Fig. 3.

3.1. Data Pre-processing and data augmentation

The data pre-processing step is an important step in finger vein recognition that removes the image noise. In the proposed work, the finger vein image is enhanced by the method used in (Shaheed et al., 2018). All the image of both datasets used in our work is resized to 199 × 199, focusing the center of the finger vein image on ensuring each venous image line is at the center of the image. For our model, all the datasets images are normalized from [0, 255] to [-1, 1].

Steps involved in the proposed finger vein image pre-processing are as follows.

1. ROI Extraction from raw finger vein image by cropping to 240 × 240 pixels. Images are separated into two halves and apply a filter to detect upper and lower boundaries. We keep the region between upper and lower limits.
2. Enhance the ROI image with the method proposed to remove non-uniform illumination and improve the contrast level (Shaheed, Yang, Yang, Qureshi, & Yin, 2018).
3. Resize the image to achieve a smaller edge of 199 pixels long.
4. Extract the center region 199 × 199 for clearer pattern detail for our proposed model.

Below, Fig. 4 shows the steps involved in our pre-processing work.

CNN model required a large dataset to achieve high-performance accuracy. Moreover, CNN architecture performance declined with a small dataset due to an over-fitting problem, which means that the model performs well on training data but underperforms test data. The data augmentation method is used to increase the data set and degrade the over-fitting problem in our work. Therefore, the dataset size can expand using a simple image processing technique with the data augmentation method. Table 3 illustrate the data augmentation method with the parameter used in this proposed work. Fig. 5 shows a single image and augmented images of that particular image on two different datasets.

3.2. Pre-trained CNN model

The Xception network model was proposed by Francois Chollet (Chollet, 2017), an extension of inception architecture entirely based on depth-wise distinguished convolutions.

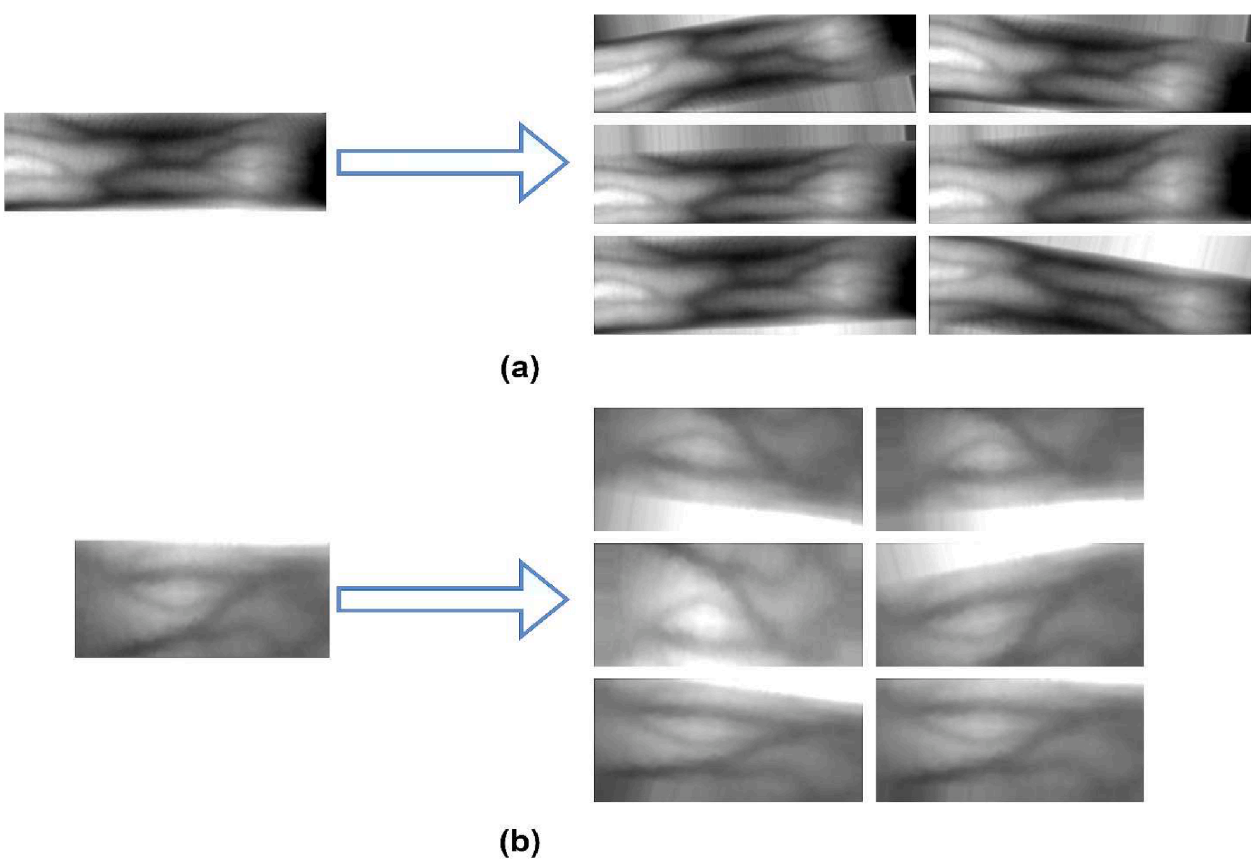


Fig. 5. Images after data augmentation in both datasets (a) SDUMLA (b) THU-FVFDT2.

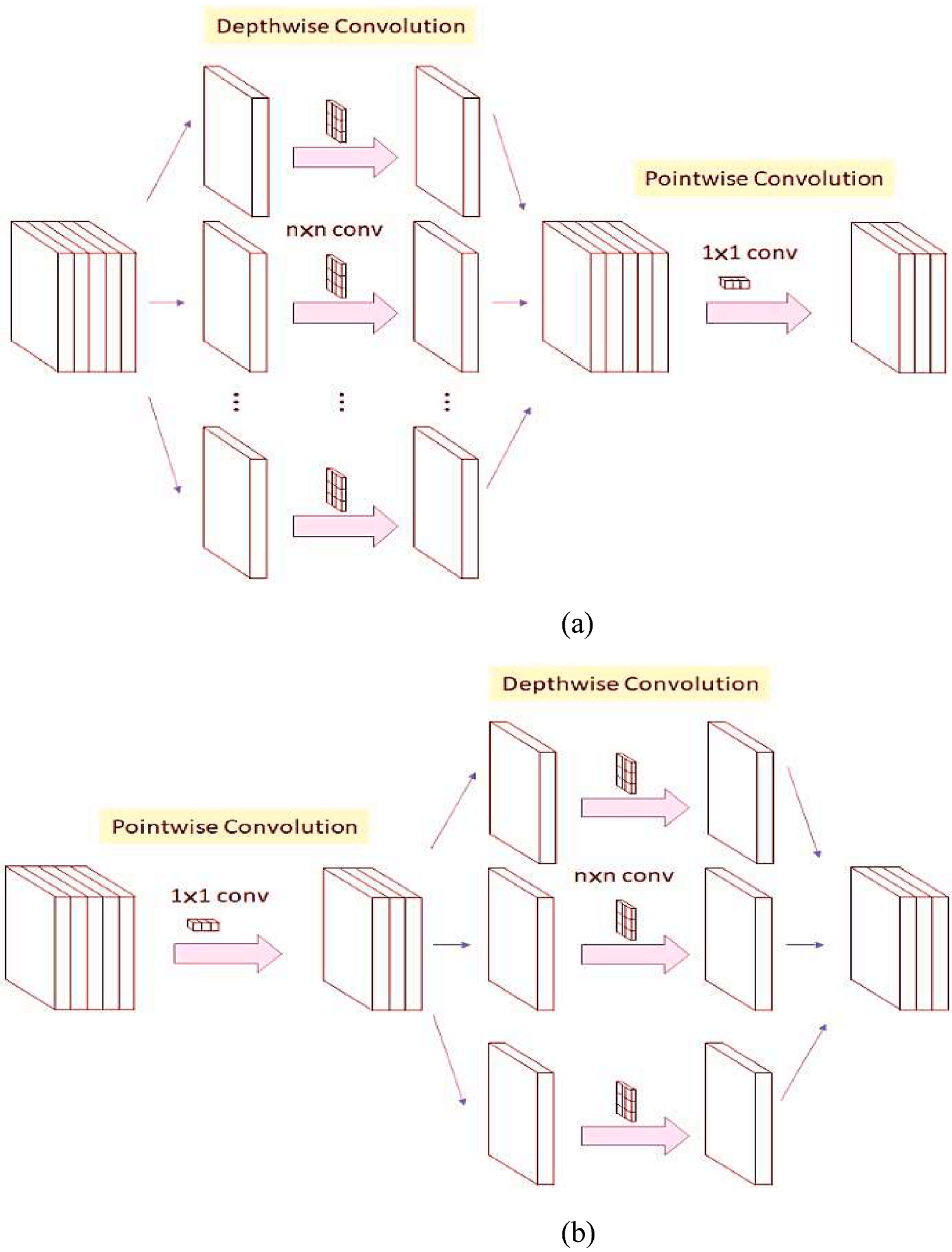


Fig. 6. (a) Original depth-wise separable convolution (b) modified depth-wise separable convolution (Tsang, 2018).

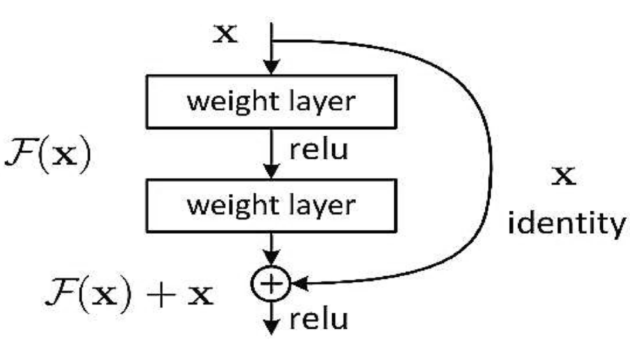


Fig. 7. Single residual block (Sahoo, 2018).

3.2.1. Depth-wise separable convolutional layer

The Xception network has a foundation based on earlier model inception and inception-V3 (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, & Rabinovich, 2015; Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016). The Xception model used a depth-wise separable convolution neural network instead of standard convolution. A depth-wise separable convolution is commonly called “Separable Convolutional”. This layer is an alternative to the classical convolutional layer that is supposed to

Table 4
Proposed Network Hyperparameter.

Loss Function	Learning Rate	Optimizer	No. of epoch	Batch size
Binary Cross Entropy	1×10^{-3}	Adam	20	32

Table 5
Configuration of Proposed CNN architecture.

	Layer	Number of Filters	Feature Map Size	Kernel Size	Stride
	Input Layer		199 × 199 × 3		
Block 1	block1_conv1(Conv2D)	32	149 × 149 × 32	3 × 3	1 × 1
	block1_conv1_bn	64	149 × 149 × 32	3 × 3	1 × 1
	block1_conv1_act (Activation)		149 × 149 × 32		
	block1_conv2 (Conv2D)	64	147 × 147 × 64	3 × 3	1 × 1
	block1_conv2_bn	64	147 × 147 × 64	3 × 3	1 × 1
	block1_conv2_act (Activation)		147 × 147 × 64		
Block 2	block2_sepconv1(SeparableConv2D)	128	147 × 147 × 128	3 × 3	1 × 1
	block2_sepconv1_bn	128	147 × 147 × 128	3 × 3	1 × 1
	block2_sepconv2_act (Activation)		147 × 147 × 128		
	block2_sepconv2 (SeparableConv2D)	128	147 × 147 × 128	3 × 3	1 × 1
	block2_sepconv2_bn	128	147 × 147 × 128	3 × 3	1 × 1
	conv2d (Conv2D)	128	74 × 74 × 128		
	block2_pool (MaxPooling2D)	1	74 × 74 × 128	3 × 3	2 × 2
	batch_normalization	128	74 × 74 × 128		
	add (Add)		74 × 74 × 128		
Block 3	block3_sepconv1_act (Activation)		74 × 74 × 128		
	block3_sepconv1 (SeparableConv2D)	256	74 × 74 × 256	3 × 3	1 × 1
	block3_sepconv1_bn	256	74 × 74 × 256	3 × 3	1 × 1
	block3_sepconv2_act (Activation)		74 × 74 × 256		
	block3_sepconv2 (SeparableConv2D)	256	74 × 74 × 256	3 × 3	1 × 1
	block3_sepconv2_bn	256	74 × 74 × 256	3 × 3	1 × 1
	conv2d_1 (Conv2D)	256	37 × 37 × 256		
	block3_pool (MaxPooling2D)	1	37 × 37 × 256	3 × 3	2 × 2
	batch_normalization_1	256	37 × 37 × 256		
	add_1 (Add)		37 × 37 × 256		
Block 4	block4_sepconv1_act (Activation)		37 × 37 × 256		
	block4_sepconv1 (SeparableConv2D)	728	37 × 37 × 728	3 × 3	1 × 1
	block4_sepconv1_bn	728	37 × 37 × 728	3 × 3	1 × 1
	block4_sepconv2_act (Activation)		37 × 37 × 728		
	block4_sepconv2 (SeparableConv2D)	728	37 × 37 × 728	3 × 3	1 × 1
	block4_sepconv2_bn	728	37 × 37 × 728	3 × 3	1 × 1
	conv2d_2 (Conv2D)	728	19 × 19 × 728		
	block4_pool (MaxPooling2D)	1	19 × 19 × 728	3 × 3	2 × 2
	batch_normalization_2	728	19 × 19 × 728		
	add_2 (Add)		19 × 19 × 728		
Block 5 Repeated 8 times	block5_sepconv1_act (Activation)		19 × 19 × 728		
	block5_sepconv1 (SeparableConv2D)	728	19 × 19 × 728	3 × 3	1 × 1
	block5_sepconv1_bn	728	19 × 19 × 728	3 × 3	1 × 1
	block5_sepconv2_act (Activation)		19 × 19 × 728		
	block5_sepconv2 (SeparableConv2D)	728	19 × 19 × 728	3 × 3	1 × 1
	block5_sepconv2_bn	728	19 × 19 × 728	3 × 3	1 × 1
	block5_sepconv3_act (Activation)		19 × 19 × 728		
	block5_sepconv3 (SeparableConv2D)	728	19 × 19 × 728	3 × 3	1 × 1
	block5_sepconv3_bn	728	19 × 19 × 728	3 × 3	1 × 1
	add_3 (Add)		19 × 19 × 728		
Block 6	block6_sepconv1_act (Activation)		19 × 19 × 728		
	block6_sepconv1 (SeparableConv2D)	728	19 × 19 × 728	3 × 3	1 × 1
	block6_sepconv1_bn	728	19 × 19 × 728	3 × 3	1 × 1
	block6_sepconv2_act (Activation)		19 × 19 × 728		
	block6_sepconv2 (SeparableConv2D)	1024	19 × 19 × 1024	3 × 3	1 × 1
	block6_sepconv2_bn	1024	19 × 19 × 1024	3 × 3	1 × 1
	conv2d_3 (Conv2D)	1024	10 × 10 × 1024		
	block6_pool (MaxPooling2D)	1	10 × 10 × 1024	3 × 3	2 × 2
	batch_normalization_3	1024	10 × 10 × 1024		
	add_4 (Add)		10 × 10 × 1024		
	block7_sepconv1 (SeparableConv2D)	1536	10 × 10 × 1536	3 × 3	1 × 1
	block7_sepconv1_bn	1536	10 × 10 × 1536	3 × 3	1 × 1
	block7_sepconv1_act (Activation)		10 × 10 × 1536		
	block7_sepconv2 (SeparableConv2D)	2048	10 × 10 × 2048	3 × 3	1 × 1

(*continued on next page*)

Table 5 (*continued*)

Layer	Number of Filters	Feature Map Size	Kernel Size	Stride
block7_sepconv2_bn	2048	10 × 10 × 2048	3 × 3	1 × 1
block7_sepconv2_act (Activation)		10 × 10 × 2048		
average_pooling2d (Average Pooling)		2 × 2 × 2048	3 × 3	2 × 2
flatten (Flatten)		8192 × 1		
dense (Dense)		64 × 1		
dense_1 (Dense)		1 × 1		

be more efficient in computation costs. The original depth-wise separable convolutional layer is the depth-wise separable where point-wise convolution is operated after depth-wise separable convolution. In the Xception model, a modified depth-wise detachable convolution layer is used, which means that point-wise convolution, i.e., a 1 × 1 convolution, is performed on each channel before using a depth-wise convolution operator, as shown in Fig. 6 below.

3.2.2. Residual connection

The residual connection, also called skipped connection, bypass the two or three layers of the network. Below, Fig. 7 shows the single residual connection block in the deep learning network. There are three residual blocks in our proposed CNN model, as shown in Fig. 3. The benefit of using the residual connection in the deep learning model is that it adds the feature from the preceding layer to the next layer in the network model, which empowers the feature that increases the deep network performance. The residual connection can be mathematically defined as

$$H(x)=F(x)+x \tag{2}$$

H(x) represent the true output value and $F(x)$ shows residual learning of layers with the network input.

Using depth-wise separable convolutional layer and residual connection in the CNN network makes the network computation easy, helping to learn the efficient feature from the images and improve the model performance. Therefore, we have adopted this model for finger vein recognition.

3.2.3. CNN architecture

We design a framework for finger vein verification by a modified Xception model. We utilized transfer learning and fine-tuning on the Xception model, hence employed ImageNet weights before starting to train the model; the last layer was frozen. After average pooling, we added two fully connected and one output layer with a sigmoid activation function. Xception model used as the base network for deep feature extraction, separated by three flows: entry flow, Middle flow, and Exit flow. The input data first go through the entry flow, then through the middle, repeated eight times, and finally through the exit flow. Our proposed network includes 4 Convolution batch normalization blocks, 10 Separable batch normalization blocks, four max-pooling layers, one average pooling layer, 14 ReLu activation functions, and two fully connected layers, as shown in Fig. 3. We remove the top layer from the base network to transfer pre-trained knowledge from the ImageNet dataset to intermediate layers. And include two fully connected layers and an output layer that classified the finger vein image. This is the key difference between our proposed network and the original Xception architecture.

The proposed network is computationally efficient because of the depth-wise separable convolution layer. It reduces the amount of arithmetic operation while persevering the generalization ability of the model. Depth-wise separable convolution divided the convolution into two stages of calculation: 1) Each channel was convoluted for filtering using depth-wise convolution; 2) Pointwise convolution was used with a 1 × 1 convolution kernel for combining. The depth-wise convolution outputs were then attached using a pointwise convolution. In one phase, criterion convolution was used to filter and combine the input, resulting in a new output set.

This distinct approach was applied to reduce computation and parameter quantity effectively. We demonstrated how to calculate the total size (in bytes) of the weights in a single original convolutional layer in Equation (3) (Lu et al., 2021).

$$Weights = ch \times n_{filter} \times h_{filter} \times w_{filter} \tag{3}$$

where ch was the number of channels, n_{filter} was the number of filters, h_{filter} was the height of the filter, and w_{filter} was the width of the filter.

The parameter computation in separable convolution layer as shown in equation 4. In depth-wise separable convolution, the number of parameters is reduced compared to the original convolution layer.

$$Weights_{Depthwise\_separable\_convolution} = h_{filter} \times w_{filter} \times ch_{L-1} \times ch_L \times 1 \times 1 \times n_{filter} \tag{4}$$

where L represented the convolution layer, ch was the channel , n was the number of filters, h_{filter} represent the height and w_{filter} denote the width of the filter.

The depth-wise separable in the Xception network make the computation easy. The transformation strategy of Xception makes learning more efficient and improves performance. Table 5 shows the proposed network configuration. Model hyperparameter tunning plays a vital role in training a network model. The best model for the identification/Classification of finger vein was achieved after several configurations. The hyperparameters used for our proposed method are shown in Table 4.

The loss function indicated in the table is Binary Cross-Entropy, which is also known as the cost function. Mathematically defined as following

$$Loss = -\frac{1}{outputsize}\sum_{i=1}^{outputsize} y_i.log\,\widehat{y}_i + (1-y_i).log(1-\widehat{y}_i) \tag{5}$$

where y_i is the i - th scalar value in the model output, y_i is the corresponding target value, and the output size is the number of scalar values in the model output.

As mentioned in Table 4, Adam optimizer was used in our proposed network, with a learning rate of 1×10^{-3} and batch size 32. Adam can be mathematically defined as:

$$w_{t+1} = w_t - \eta\frac{v_t}{\sqrt{s_t+\varepsilon}} \times g_t \tag{6}$$

w_{t+1} and w_t denote the new and old weight, η represent the initial learning rate, vt is the exponential average of the gradient, st is the exponential average of a square and g_t denote gradient at time t .

The proposed network model is compared with the Visual Geometry Group (VGG) Net model (VGG16 and VGG19), Inception V3, and Dense Net the network of a selected network as follows:

- VGG Net Model

It's a CNN model introduced by VGG from the University of Oxford (Dhillon & Verma, 2020). VGG16 consists of 13 convolutional layers and three fully connected layers, while VGG19 comprises 16 convolutional layers and three fully connected layers. They utilize channels with little receptive field: 3 × 3. And each layer is got up with a rectifier linear unit.

- Inception V3

The development of the Inception CNN model was considered an important milestone. Christian Szegedy proposed this Inception module for computer vision tasks with a named Google Net/Inception-v1. The inception V3 is the third generation of inception modules developed by Google brain, comprising 159 layers. Such architecture's main idea is to reduce computation cost and learn multi-scale representation using large kernel sizes with fewer parameters.

- Dense Net

Dense Net architecture was developed by (Huang, Liu, & Laurens van der Maaten, 1978), where each layer is linked in feed-forward mode. This model has $L(L+1)/2$ direction connection with the L layer. This model collective knowledge from the preceding layer is required to concatenate the output feature map with the input feature map. The advantage of using this model is that it reduces the vanishing gradient problem, requires minimum parameters, and can reuse features. The architecture shows a promising result on CIFAR100 and ImageNet databases. The only disadvantage is such a network requires extra memory to concatenate tensors.

3.3. Material

We applied two publicly available databases in practical work, i.e., SDUMLA (Yin, Liu, & Sun, 2011) and THU-FVFDT2 (Yang, 2014) datasets, to evaluate the proposed method performance. The SDUMLA dataset collected from 106 subjects has six images of 6 different fingers, a total of 3816 images, while the THU-FVFDT2 dataset consists of 610 images where each image belongs to a single subject. The images were obtained in two sessions, and the subject was volunteers and students of Graduate School, Tsinghua University, at Shenzhen.

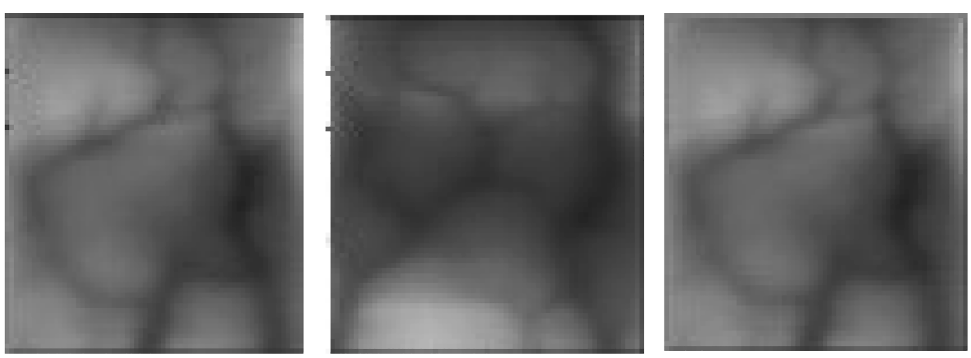
In our work, from the SDUMLA dataset, we consider 80 subjects with five images from 2 fingers to evaluate our model performance. Among 800 finger vein images, 640 finger vein images were used for model training and 160 images for verification. While in THU-FVFDT2 total of 600 images were used from each subject. For model training, 480 images were used and 120 images for model verification. However, both datasets were small. Therefore, to remove the overfitting problem from the network model, we apply the data augmentation technique on both datasets by rotating, width shift, zoom range, horizontal flip, and shear to produce a total of 4000 and 3000 images for both datasets, respectively. Below, Fig. 8 shows both the dataset input sample to proposed CNN architecture.

3.4. Evaluation metric

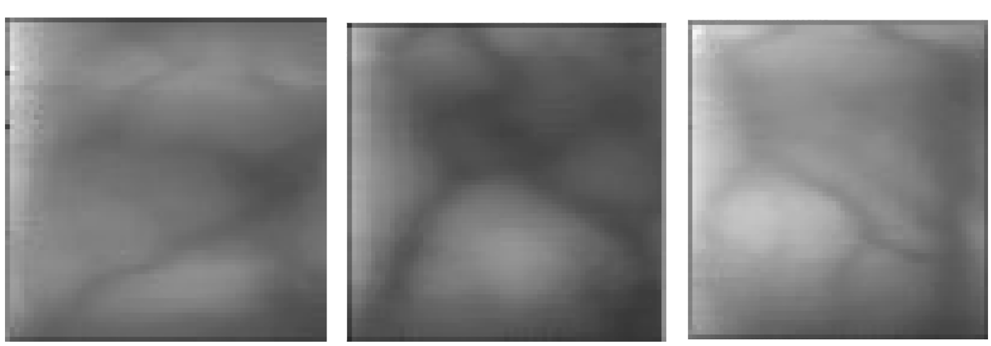
To obtain good accuracy, it is vital to propose and select a suitable parameter for the proposed network architecture. A wide range of experiments is performed for finger vein recognition performance to test the network structure. The proposed model performance was assessed using standard assessment metrics such as accuracy (Acc), precision (Precision), recall (Recall), f-score (F1), Sensitivity (Sen), Specificity (Spe), and receiver operating curve (ROC). These metric is defined as follows

$$Precision = \frac{TP}{(TP+FP)} \tag{7}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{8}$$



(a)



(b)

Fig. 8. (a) SDUMLA datasets three Input images to the CNN model (b) THU-FVFDT2 datasets three Input images to the CNN model.

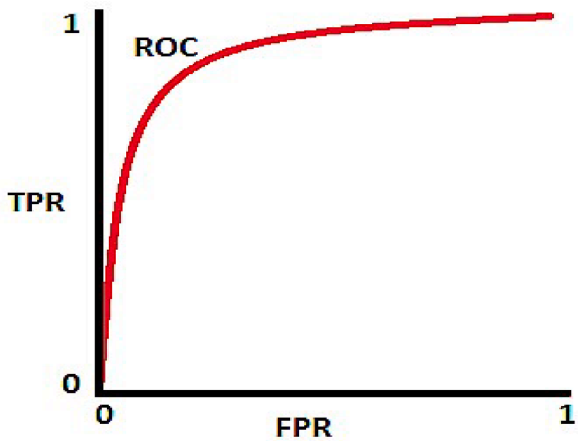


Fig. 9. Receiver Operating Curve.

$$Sensitivity = \frac{TP}{TP + FN} \quad (9)$$

$$Specificity = \frac{TN}{(TN + FP)} \quad (10)$$

$$F1 - Score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (11)$$

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \quad (12)$$

where TP denoted the number of accurate positive sample predictions, TN represented the number of correct negative sample predictions, FN indicated the number of wrong positive sample predictions, and FP referred to the number of incorrect negative sample predictions.

ROC is a graphical representation of model performance at various threshold settings. The curve plot on two parameters: True Positive Rate and False Positive Rate, as shown in Fig. 9.

4. Results and analysis

This section will explain the data we used, the parameter we employ to evaluate our proposed model, and the experimental result. The proposed method was assessed on two available public benchmark databases. We conduct both quantitative and qualitative evaluation comparisons with other contemporary methods.

4.1. Computational cost

We measure the processing time of different transfer learning methods for two different databases, as shown in Table 6 and Table 7. The experiment was performed in Python (3.6) using the Google collab environment with Tesla T4 GPU, having system configuration Intel(R) Core (TM) i5 CPU and windows 10 professional operating system. We used MATLAB 2019a for pre-processing. Feature extraction time refers to the time taken by the model to extract features. Training time refers to the time to train the network model. Prediction time refers to predicting a finger vein image label. Table 6 and Table 7 show that our proposed method overall time is less than the overall time taken by other transfer learning models such as VGG16, VGG19, InceptionV3, and Dense Net121 on both databases. Thus, our proposed method outperforms other network models in terms of processing time and is considered less computationally expensive. It is because of image pre-processing and also the use of depth-wise-separable convolution layer in our proposed model. Two main advantages of using depth-wise-separable convolution layer in deep learning model (1) reduce the parameter quantity, which made the proposed model computationally efficient as shown in Table 6 and Table 7 and (2) also it improves the generalization ability (Lu et al., 2021). Therefore, the proposed model is helpful to use in real-time applications.

Table 6
Processing time using SDUMLA.

Method	Feature Extraction	Training	Prediction	Overall
VGG16	1.4 s	294.0 s	1.9 s	297.1 s
VGG19	2.8 s	221.3 s	2.7 s	226.8 s
InceptionV3	4.3 s	353.9 s	1.6 s	359.8 s
Dense Net121	175.4 s	142.1 s	1.3 s	318.8 s
Proposed Method	**8.2 s**	**140.9 s**	**1.8 s**	**150.9 s**

Table 7
Processing time using THU-FVFDT2.

Method	Feature Extraction	Training	Prediction	Overall
VGG16	2.0 s	136.6 s	1.8 s	140.4 s
VGG19	1.6 s	242.6 s	2.3 s	246.5 s
InceptionV3	5.7 s	355.1 s	1.3 s	362.1 s
Dense Net121	133.0 s	121.3 s	1.5 s	255.8 s
Proposed Method	**8.4 s**	**120.9 s**	**1.7 s**	**131.0 s**

4.2. Performance of the proposed method on SDUMLA

First, we evaluate the training accuracy of our network model. For our network model, the size of the input image is normalized to 199 × 199 pixels. The batch size for dataset 1 is 32, and the learning rate is 0.003. Our proposed method performs better than Dense Net 121, Inception V3, VGG19 and VGG16 by achieving 98% accuracy, as shown in Fig. 10. These training data results illustrate that our scheme achieves high performance in terms of accuracy in identifying finger vein images.

After our proposed model's excellent performance on training data, we consider our proposed network model to evaluate further and test using the validation dataset. During the proposed scheme experiments, the dataset is split into train and test datasets. Three different procedure is used to split the datasets: 90%-10%, 80%-20%, and 70%-30%. The 70%-30% means that 70 percent of data is employed for training, while

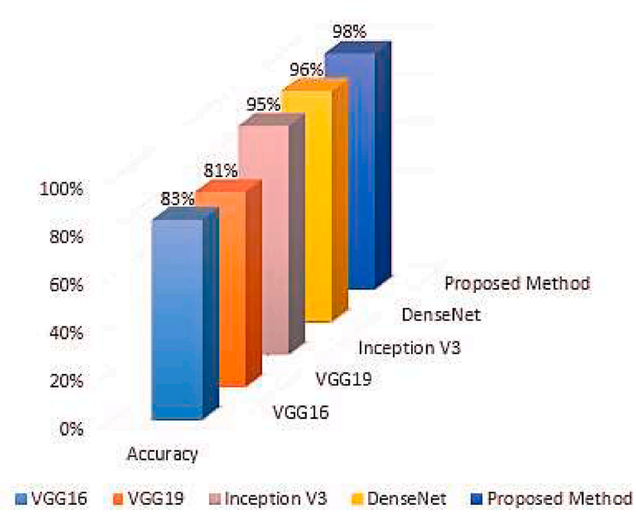


Fig. 10. Training Accuracy on SDUMLA datasets.

Table 8
Comparative Analysis of Proposed method with other models using SDUMLA.

Classifier	Training-testing data splitting	Precision	Recall	F1	Sen	Spe	Acc	Average Acc
VGG16	90%-10%	92%	90%	91%	90%	91%	91.2%	89.4%
	80%-20%	95%	77%	86%	77%	90%	86.8%	
	70%-30%	98%	75%	90%	75%	89%	90.4%	
VGG19	90%-10%	97%	75%	85%	75%	97%	86.2%	87.3%
	80%-20%	95%	63%	85%	63%	95%	86.2%	
	70%-30%	87%	98%	90%	98%	47%	89.5%	
InceptionV3	90%-10%	89%	92%	94%	92%	90%	93.7%	94%
	80%-20%	93%	97%	96%	97%	88%	95.6%	
	70%-30%	87%	87%	93%	87%	94%	92.5%	
DenseNet121	90%-10%	97%	94%	96%	94%	93%	96.2%	96.5%
	80%-20%	95%	87%	98%	87%	96%	97.5%	
	70%-30%	93%	95%	96%	95%	94%	95.8%	
Proposed Method	**90%-10%**	**97%**	**97%**	**97%**	**97%**	**97%**	**98.8%**	**98.5%**
	80%-20%	**95%**	**100%**	**98%**	**100%**	**95%**	**99.0%**	
	70%-30%	**95%**	**99%**	**97%**	**99%**	**97%**	**98.0%**	

the rest 30 percent is used to test the proposed architecture. A comparative evaluation of the proposed scheme with other network architecture based on data splitting is given in Table 8, which provides precision, recall, F1 score, sensitivity, specificity, and accuracy. Average accuracy is also provided for each architecture based on splatting procedures. It is observed from Table 10 that our proposed method gives high average accuracy compared to VGG16, VGG19, Inception V3, DenseNet 121. Also, with other metrics, the performance of our proposed method outperforms other network models. As shown in Fig. 11, the ROC curve predicts positive classes 98.5% of the time.

4.3. Performance of the proposed method on THU-FVFDT2

For dataset 2, we also first evaluate the training accuracy. As we see in Fig. 12, our presented network model performs better than the rest of the network model. The proposed network achieved an accuracy of 89% on training data.

For dataset 2, after a good performance of the proposed method on training data, we evaluate the proposed network with the validation dataset by splitting data into training and testing: 90%-10%, 80%-20%, and 70%-30%. Table 9 clearly shows that the proposed method outperforms other network models in precision, recall, F1score, sensitivity, specificity, accuracy, and average accuracy. However, the InceptionV3 and DenseNet121 model's performance increase if we increase the testing data compared to our proposed method. ROC curve in Fig. 13 also shows that the overall performance of the proposed method on dataset 2 is reasonably good with different data splitting.

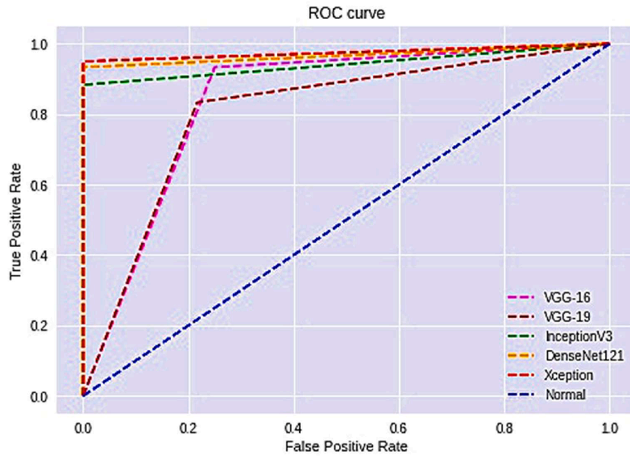
4.4. Comparative analysis of accuracy with the existing state of the art algorithm

To evaluate the proposed method strength and comparatively analyze the result, the proposed framework is compared with other deep-learning-based finger vein verification systems given in Table 10. The existing deep learning-based finger vein recognition method performance and proposed is evaluated on the SDUMLA dataset for comparative analysis. It can be observed from Table 10 that the method LeNet, Alex Net, VGG-16, CNN, deeply fused CNN, Dense Net give an accuracy of 96%, 91%, 89%, 98%, 98%, 87%, respectively. The accuracy we achieved with the proposed method is 99%, which is better than all other methods. We achieved an accuracy of 99% at just 20 epoch and required a significantly less average training time of 152.7 s. On the contrary, LeNet obtained an accuracy of 96% at 23 epochs with an average training time of 190.s, AlexNet achieved an accuracy of 91% at 40 epochs with an average training of 754.9 s for model training, VGG-16 required 25 epochs and took 294.s average training time to achieve accuracy of 89%, CNN needs 25 epochs and training time of 240.s for model training to obtain a result of 98%. Deeply fused CNN also needs 25 and 230.s average time taken to train the model and obtain an accuracy of 98%. CAE and FV-GAN methods achieved 98% and 97% recognition accuracy using an average training time of 180.0 s and 250.0 s. These results represent the proposed technique strength compared to different related network approaches.

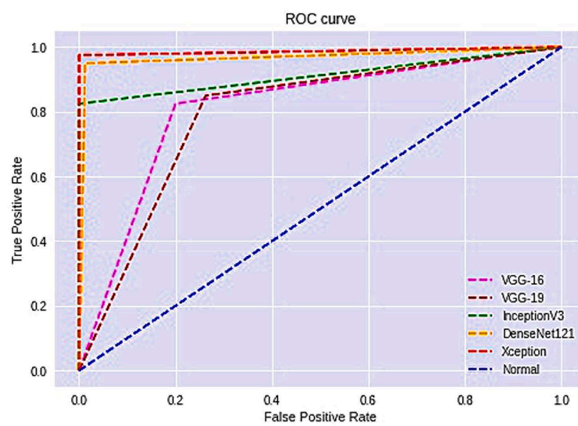
5. Discussion

This section will analyze the proposed method compared to other models based on the result shown in section 4. We will argue the advantage of our approach for the recognizing and classification of finger vein images. Here are some of the facts that inspired us to employ a pre-trained CNN model called the Xception model with data augmentation methods are: (1) Inspired by developing deep transfer neural network approaches. (2) Computational complexity problem in existing finger vein deep learning model. (3) Lack of finger vein training data. (4) to achieve an acceptable level of authentication accuracy on small datasets. In the proposed work, we first apply the data pre-processing method to enhance the data and normalize it into a standard format. Afterward, the data augmentation method is used to increase the finger vein image sample for model training. Finally, the DS-CNN pre-trained model was employed to extract finger vein image features and classify/identify finger vein images.

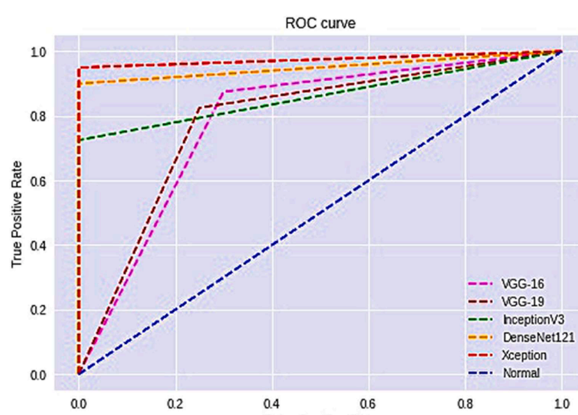
The results are shown in Table 8, Table 9, Table 10, Fig. 10, and Fig. 12 demonstrates that the proposed method DS-CNN (Xception model) attained remarkable performance among all the other schemes studied in our work using two different datasets. Our proposed method has shown impressive results on all the statistical parameters used in our paper. For example, it is worth noting that the F1 score and accuracy are not less than 97% and 86% on three different data split using two benchmark datasets, i.e., the SDUMLA dataset THUFVFDT2 dataset, respectively. We also elaborated and showed the ROC curve recommended for model performance; we compute the ROC curve, which shows that our proposed model outperforms all the other models, as shown in Fig. 11 and Fig. 13. Moreover, compared with the current work, the best CNN model identification accuracy in existing schemes is improved by 1% using the proposed DS-CNN model with a data augmentation approach. A deep learning algorithm



(a)



(b)



(c)

Fig. 11. (a) ROC curve with data splitting 90%-10% (b) ROC curve with data splitting 80%-20% (c) ROC curve with data splitting 70%-30%.

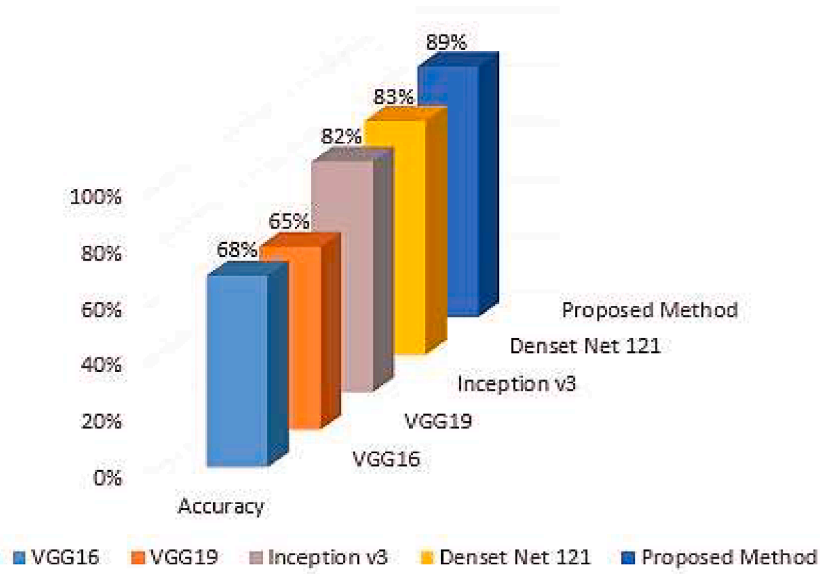


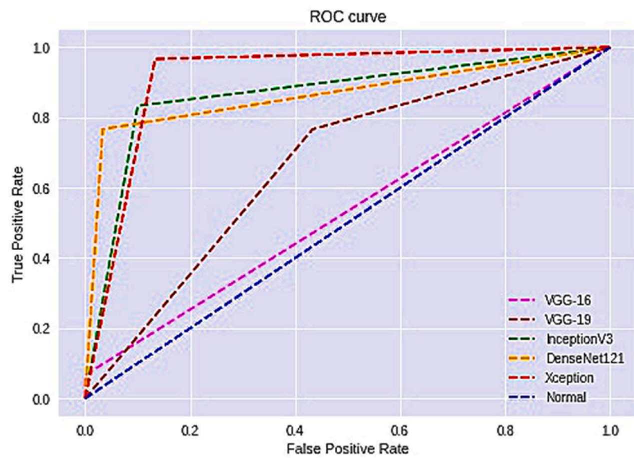
Fig. 12. Training accuracy on THU-FVFDT2 datasets.

training time is important in selecting a specified model for a specific task. As mentioned in Table 7 and Table 8, our proposed model training time is 140.9 s and 120.0 s for two different datasets, which shows that our model takes less time to train than VGG-16, VGG-19, inception-V3, and DenseNet121. We can also notice from Table 6 and Table 7 that the overall time taken by our proposed method to achieve accuracy of 99% and 90% on two benchmarks datasets takes much less time compared to other baseline models. Thus, it shows that our proposal can converge a better solution and suggest that our proposed approach is not computationally complex. One reason is the use of residual connection and depth-wise separable convolution layer, which significantly improves the training speed (Chollet, 2017). Also, it makes the model capable of learning richer representation from a finger vein image using fewer parameters. Implementing the depth-wise separable convolution layer and residual connection in Xception architecture makes our model easy to define and modify, unlike VGG16, VGG19, and inception V3, which are far more complex not easy to define (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017). Also, the absence of non-linearity between depth-wise separable convolution layers leads to high identification performance. Therefore, our proposed method is highly efficient in all measures, for instance, accuracy, recall, precision, F1 score, specificity, sensitivity, ROC curve compared to other deep learning CNN networks. It must be noted that the scores we attained with a transfer learning approach from ImageNet are of a different context. We used the weights of the pre-trained Xception model to initialize the proposed model. This has enabled us to train a proposed model with the finger vein data sets quicker.

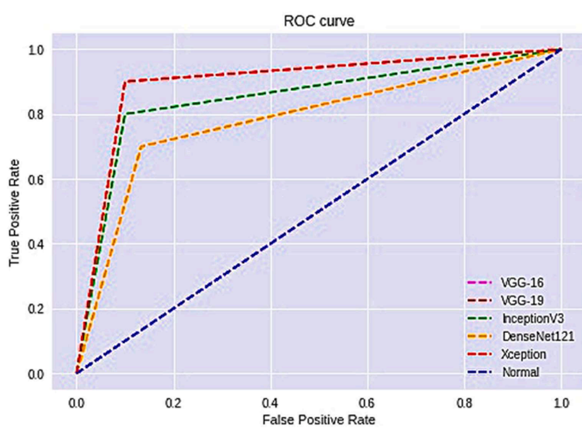
There are two main concerns with the existing deep learning-based finger vein system: the high computation cost, complexity, and low convergence rate. The conventional convolution operation involves many multiplication operations, which increases the inference time and results in a low convergence rate. Therefore, the result achieved with the existing approaches is computationally complex and obtained low accuracy. Our proposed work used a depth-wise separable convolution layer to make it a computationally efficient deep learning-based finger vein system and

Table 9
Comparative Analysis of Proposed method with other models using THU-FVFDT2.

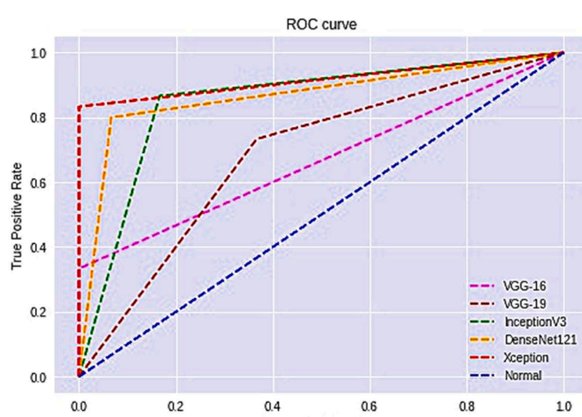
Classifier	Training-Testing data splitting	Precision	Recall	F1	Sen	Spe	Acc	Average Acc
VGG16	90%-10%	70%	75%	72%	75%	40%	65.6%	57%
	80%-20%	60%	70%	62%	70%	10%	50.0%	
	70%-30%	82%	60%	69%	60%	95%	55.8%	
VGG19	90%-10%	60%	77%	67%	77%	20%	53.3%	60%
	80%-20%	56%	60%	57%	60%	85%	50.0%	
	70%-30%	80%	70%	74%	70%	24%	59.4%	
InceptionV3	90%-10%	92%	80%	85%	80%	90%	82.3%	84%
	80%-20%	82%	78%	79%	78%	90%	84.0%	
	70%-30%	88%	88%	86%	88%	82%	86.7%	
DenseNet121	90%-10%	85%	90%	87%	90%	70%	81.6%	85%
	80%-20%	81%	80%	80%	80%	93%	85.7%	
	70%-30%	88%	83%	85%	83%	87%z	88.5%	
Proposed Method	**90%-10%**	**92%**	**95%**	**93%**	**95%**	**92%**	**93.5%**	**90%**
	80%-20%	**90%**	**88%**	**88%**	**88%**	**85%**	**89.0%**	
	70%-30%	**87%**	**90%**	**88%**	**90%**	**80%**	**86.6%**	



(a)



(b)



(c)

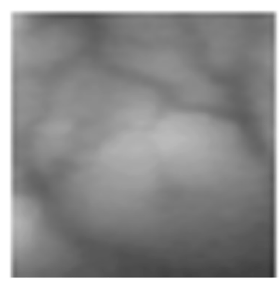
Fig. 13. (a) ROC curve with data splitting 90%-10% (b) ROC curve with data splitting 80%-20% (c) ROC curve with data splitting 70%-30%.

Table 10
Comparison of the proposed method with existing deep learning-based finger vein identification framework.

Method	Best Epoch	Average Training time	RecognitionAcc
LeNet (Syafeeza et al., 2016)	23	190.0 s	96%
Alex Net (Fairuz et al., 2018)	40	754.9 s	91%
VGG-16 (Hong et al., 2017)	25	294.0 s	89%
CNN (Das et al., 2018)	25	240.0 s	98%
Deeply fused CNN (Boucherit et al., 2020)	28	230.0 s	98%
CAE (Hou & Yan, 2020)	25	180.0 s	98%
FV-GAN (Yang et al., 2019)	22	250.0 s	97%
Proposed Method	**20**	**152.7 s**	**98.5%**

improve the converging rate without affecting the model accuracy.

As we can observe from (a) and (b) in Fig. 14, the model correctly identifies the person with a finger vein image where the vein region and non-venous region are differentiated clearly. Our model misclassified the images with low quality where the vein region is not very clear, as seen from (c) and (d) in Fig. 14. Therefore, our proposed method performance on dataset 2 is slightly below par, where most of the images' quality was low, even after pre-processing. Another reason was maybe a small data size compared to dataset 1. Therefore, a better pre-processing method and introducing a huge amount of training data would be prospects to evaluate the model potential further.



(a)

(b)

(c)

(d)

Fig. 14. (a) and (c) shows finger vein images where vein and non-vein region are clear, (b) and (d) shows finger vein images where vein and non-vein region are not obvious and clear.

6. Conclusion with future remarks

This paper proposes a new deep learning method for finger vein authentication and classification, employing transfer learning. This method extracts features from finger vein images using the CNN architecture Xception model using transfer learning ideas to improve accuracy. Also, we employ and propose a data augmentation concept to enlarge the dataset size to enhance the efficacy of the proposed pre-trained Xception model. Lastly, the performance of our presented scheme is competed with other CNN architecture using two datasets separately. Also, the proposed work is compared with the existing deep learning-based finger vein identification system. It has been noted that the proposed method provides perfect results regarding accuracy without training from scratch, which shows the superiority of our approach compared to other deep learning schemes. Moreover, the proposed model is also cost-efficient in terms of network parameter complexity compared to other transfer learning models.

The following steps can be taken in future work: First, both traditional and in-depth features will be employed to further boost the identification system accuracy. Second, a huge number of training data could be provided to assess the model performance further. Lastly, other commonly used pre-processing methods, such as Histogram equalization, Contrast limited adaptive histogram equalization, can be employed for a comprehensive comparison.

CRediT authorship contribution statement

Kashif Shaheed: Conceptualization, Methodology, Data curation, Formal Analysis, Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Aihua Mao:** Supervision, Project Administration. **Imran Qureshi:** Writing – review & editing. **Munish Kumar:** Writing – review & editing. **Sumaira Hussain:** Resources. **Inam Ullah:** Resources. **Xingming Zhang:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

I would like to thank my supervisors, Professor Aihua Mao and Professor Xingming Zhang, who supervises my Ph.D. project. And Thanks to all anonymous reviewers for their valuable comments. This work is financially supported by the NSF of Guangdong Province (No. 2019A1515010833) and the Fundamental Research Funds for Central Universities (No. 2020ZYGXR089). It is a part of a Ph.D. research project conducted in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

References

Abbas, Q., Ibrahim, M. E. A., & Jaffar, M. A. (2018). Video scene analysis: An overview and challenges on deep learning algorithms. *Multimedia Tools and Applications, 77*(16), 20415–20453. https://doi.org/10.1007/s11042-017-5438-7

Abdeldayem, S. S., & Bourlai, T. (2019). A Novel Approach for ECG-Based Human Identification Using Spectral Correlation and Deep Learning. *IEEE Transactions on Biometrics, Behavior, and Identity Science, 2*(1), 1–14. https://doi.org/10.1109/TBIOM.2019.2947434

Aberni, Y., Boubchir, L., & Daachi, B. (2020). Palm vein recognition based on competitive coding scheme using multi-scale local binary pattern with ant colony optimization. *Pattern Recognition Letters, 136*, 101–110. https://doi.org/10.1016/j.patrec.2020.05.030

Ahmad Radzi, S., Khalil-Hani, M., & Bakhteri, R. (2016). Finger-vein biometric identification using convolutional neural network. *Turkish Journal of Electrical Engineering and Computer Sciences, 24*(3), 1863–1878. https://doi.org/10.3906/elk-1311-43

Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S., & Nagem, T. A. M. (2018). A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications, 21*(3), 783–802. https://doi.org/10.1007/s10044-017-0656-1

Ayan, E., & Ünver, H. M. (2019). Diagnosis of pneumonia from chest X-ray images using deep learning. *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science. EBBT, 2019*. https://doi.org/10.1109/EBBT.2019.8741582

Bansal, P., & Balas, V. E. (2020). *Finger Vein Recognition Using Deep Learning.*

Beng, T. S., & Rosdi, B. A. (2011). Finger-vein identification using pattern map and principal component analysis. In *2011 IEEE International Conference on Signal and Image Processing Applications*. https://doi.org/10.1109/ICSIPA.2011.6144093

Boucherit, I., Zmirli, M. O., Hentabli, H., & Rosdi, B. A. (2020). Finger vein identification using deeply-fused Convolutional Neural Network. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.04.002

Carrera, E. V., Izurieta, S., & Carrera, R. (2018). A finger-vein biometric system based on textural features. *Advances in Intelligent Systems and Computing, 721*(Icits), 367–375. https://doi.org/10.1007/978-3-319-73450-7_35

Chen, H., Yang, Y., & Zhang, S. (2020). Learning Robust Scene Classification Model with Data Augmentation Based on Xception. *Journal of Physics: Conference Series, 1575*(1). https://doi.org/10.1088/1742-6596/1575/1/012009

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2017.195

Das, R., Piciucco, E., Maiorana, E., & Campisi, P. (2018). Convolutional Neural Network for Finger-Vein-Based Biometric Identification. *IEEE Transactions on Information Forensics and Security, 14*(2), 360–373. https://doi.org/10.1109/TIFS.2018.2850320

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: A review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence, 9*(2), 85–112. https://doi.org/10.1007/s13748-019-00203-0

Dipanjan Sarkar. (2018). A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning | by Dipanjan (DJ) Sarkar | Towards Data Science. https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

Fairuz, M. S., Habaebi, M. H., & Elsheikh, E. M. A. (2019). Pre-trained based CNN model to identify finger vein. *Bulletin of Electrical Engineering and Informatics, 8*(3), 855–862. https://doi.org/10.11591/eei.v8i3.1505

Fairuz, S., Habaebi, M. H., & Elsheikh, E. M. A. (2018). Finger Vein Identification Based on Transfer Learning of AlexNet. In *Proceedings of the 2018 7th International Conference on Computer and Communication Engineering*. https://doi.org/10.1109/ICCCE.2018.8539256

Gautam, G., & Mukhopadhyay, S. (2018). Contact Lens Detection using Transfer Learning with Deep Representations. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2018.8489590

Gupta, P., & Gupta, P. (2015). An accurate finger vein based verification system. *Digital Signal Processing: A Review Journal, 38*(i), 43–52. https://doi.org/10.1016/j.dsp.2014.12.003

Hajian, A., & Ramli, D. A. (2018). Sharpness Enhancement of Finger-Vein Image Based on Modified Un-sharp Mask with Log-Gabor Filter. *Procedia Computer Science, 126*, 431–440. https://doi.org/10.1016/j.procS.2018.07.277

He, C., Li, Z., Chen, L., & Peng, J. (2017). Identification of finger vein using neural network recognition research based on PCA. In *Proceedings of 2017 IEEE 16th International Conference on Cognitive Informatics and Cognitive Computing*. https://doi.org/10.1109/ICCI-CC.2017.8109788

Hong, H., Lee, M., & Park, K. (2017). Convolutional neural network-based finger-vein recognition using NIR image sensors. *Sensors (Switzerland), 17*(6), 1297. https://doi.org/10.3390/s17061297

Hou, B., & Yan, R. (2019). Convolutional Auto-Encoder Model for Finger-Vein Verification. *IEEE Transactions on Instrumentation and Measurement, PP(c)*, 1. https://doi.org/10.1109/tim.2019.2921135

Hou, B., & Yan, R. (2020). Convolutional Autoencoder Model for Finger-Vein Verification. *IEEE Transactions on Instrumentation and Measurement, 69*(5), 2067–2074. https://doi.org/10.1109/TIM.1910.1109/TIM.2019.2921135

Huang, G., Liu, Z., & Laurens van der Maaten, K. Q. W. (1978). Densely Connected Convolutional Networks. *Computer Vision Foundation, 39*(9), 1442–1446.

Huang, C., Li, Y., Chen, C. L., & Tang, X. (2019). Deep Imbalanced Learning for Face Recognition and Attribute Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(c)*, 1. https://doi.org/10.1109/tpami.2019.2914680

Huang, H., Liu, S., Zheng, H., Ni, L., Zhang, Y., & Li, W. (2017). DeepVein: Novel finger vein verification methods based on Deep Convolutional Neural Networks. In *2017 IEEE International Conference on Identity, Security and Behavior Analysis*. https://doi.org/10.1109/ISBA.2017.7947683

Ismail, B., & Mohamed, O. Z. (2019). In *Finger Vein Image Enhancement Technique based on Gabor filter and Discrete Cosine Transform* (pp. 43–48).

Kassani, S. H., Kassani, P. H., Khazaeinezhad, R., Wesolowski, M. J., Schneider, K. A., & Deters, R. (2019). Diabetic Retinopathy Classification Using a Modified Xception Architecture. *2019 IEEE 19th International Symposium on Signal Processing and Information Technology, ISSPIT 2019*, 0–5. 10.1109/ISSPIT47144.2019.9001846.

Khan, S. U., Islam, N., Jan, Z., Ud Din, I., & Rodrigues, J. J. P. C. (2019). A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters, 125*, 1–6. https://doi.org/10.1016/j.patrec.2019.03.022

Kuzu, R. S., Member, S., Piciucco, E., Member, S., Maiorana, E., Member, S., … Member, S. (2020). On-the-Fly Finger-Vein-Based Biometric Recognition Using Deep Neural Networks. *IEEE Transactions on Information Forensics and Security, 15*, 2641–2654.

Li, X., Pang, T., Xiong, B., Liu, W., Liang, P., & Wang, T. (2018). Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In *Proceedings - 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017, 2018-Janua (978),* (pp. 1–11). https://doi.org/10.1109/CISP-BMEI.2017.8301998

Liu, K., Zhang, H., Ng, J.-Y., Xia, Y., Feng, L., Lee, V. C. S., & Son, S. H. (2018). Toward Low-Overhead Fingerprint-Based Indoor Localization via Transfer Learning: Design, Implementation, and Evaluation. *IEEE Transactions on Industrial Informatics, 14*(3), 898–908. https://doi.org/10.1109/TII.942410.1109/TII.2017.2750240

Liu, M., Zhou, Z., Shang, P., & Xu, D. (2019). Fuzzified Image Enhancement for Deep Learning in Iris Recognition. *IEEE Transactions on Fuzzy Systems, PP(c)*, 1. https://doi.org/10.1109/tfuzz.2019.2912576

Liu, Y., Zhang, X., Lin, Y., & Wang, H. (2020). Facial expression recognition via deep action units graph network based on psychological mechanism. *IEEE Transactions on Cognitive and Developmental Systems, 12*(2), 311–322. https://doi.org/10.1109/TCDS10.1109/TCDS.2019.2917711

Lo, W. W., Yang, X., & Wang, Y. (2019). An xception convolutional neural network for malware classification with transfer learning. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019 - Proceedings and Workshop* (pp. 1–5). https://doi.org/10.1109/NTMS.2019.8763852

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems, 80*, 14–23. https://doi.org/10.1016/j.knosys.2015.01.010

Lu, Y., Jiang, M., Wei, L., Zhang, J., Wang, Z., Wei, B., & Xia, L. (2021). Automated arrhythmia classification using depthwise separable convolutional neural network with focal loss. *Biomedical Signal Processing and Control, 69*, 102843. https://doi.org/10.1016/j.bspc.2021.102843

Luo, H., Yu, F. X., Pan, J. S., Chu, S. C., & Tsai, P. W. (2010). A survey of vein recognition techniques. *Information Technology Journal, 9*(6), 1142–1149. https://doi.org/10.3923/itj.2010.1142.1149

Miura, N., Nagasaka, A., & Miyatake, T. (2004). Feature extraction of finger vein patterns based on iterative line tracking and its application to personal identification. *Systems and Computers in Japan, 35*(7), 61–71. https://doi.org/10.1002/(ISSN)1520-684X10.1002/scj.v35:710.1002/scj.10596

Miura, N., Nagasaka, A., & Miyatake, T. (2007). Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE Transactions on Information and Systems, E90-D*(8), 1185–1194. https://doi.org/10.1093/ietisy/e90-d.8.1185

Ou, W. F., Po, L. M., Zhou, C., Rehman, Y. A. U., Xian, P. F., & Zhang, Y. J. (2021). Fusion loss and inter-class data augmentation for deep finger vein feature learning. *Expert Systems with Applications, 171*, 114584. https://doi.org/10.1016/j.eswa.2021.114584

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Park, K. R. (2011). Finger vein recognition by combining global and local features based on SVM. *Computing and Informatics, 30*(2), 295–309.

Qin, H., & El-Yacoubi, M. A. (2017). Deep Representation-Based Feature Extraction and Recovering for Finger-Vein Verification. *IEEE Transactions on Information Forensics and Security, 12*(8), 1816–1829. https://doi.org/10.1109/TIFS.2017.2689724

Qiu, S., Liu, Y., Zhou, Y., Huang, J., & Nie, Y. (2016). Finger-vein recognition based on dual-sliding window localization and pseudo-elliptical transformer. *Expert Systems with Applications, 64*, 618–632. https://doi.org/10.1016/j.eswa.2016.08.031

Qureshi, I., Ma, J., & Abbas, Q. (2021). Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. *Multimedia Tools and, 80*(8), 11691–11721.

Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., … Chellappa, R. (2019). A Fast and Accurate System for Face Detection, Identification, and Verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science, 1*(2), 82–96. https://doi.org/10.1109/TBIOM.2019.2908436

Ribani, R., & Marengoni, M. (2019). A Survey of Transfer Learning for Convolutional Neural Networks. In *Proceedings - 32nd Conference on Graphics, Patterns and Images Tutorials*. https://doi.org/10.1109/SIBGRAPI-T.2019.00010

Rosdi, B. A., Shing, C. W., & Suandi, S. A. (2011). Finger vein recognition using local line binary pattern. *Sensors, 11*(12), 11357–11371. https://doi.org/10.3390/s111211357

Sadikoglu, F., & Uzelaltinbulat, S. (2016). Biometric Retina Identification Based on Neural Network. *Procedia Computer Science, 102*(August), 26–33. https://doi.org/10.1016/j.procs.2016.09.365

Sahoo S. (2018). *Residual blocks — Building blocks of ResNet*. https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec.

Serafim, P. B. S., Medeiros, A. G., Rego, P. A. L., Maia, Gilvan R. J., Trinta, F. A. M., Maia, M. E. F., De MacEdo, J. A. F., & Neto, A. V. L. (2019). A Method based on Convolutional Neural Networks for Fingerprint Segmentation. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–8). https://doi.org/10.1109/IJCNN.2019.8852236

Shaheed, K., Liu, H., Yang, G., Qureshi, I., Gou, J., & Yin, Y. (2018). A systematic review of finger vein recognition techniques. *Information (Switzerland), 9*(9), 213. https://doi.org/10.3390/info9090213

Shaheed, K., Yang, L., Yang, G., Qureshi, I., & Yin, Y. (2018). Novel Image Quality Assessment and Enhancement Techniques for Finger Vein Recognition. In *International Conference on Security, Pattern Analysis, and Cybernetics, SPAC 2018* (pp. 223–231). https://doi.org/10.1109/SPAC46244.2018.8965537

Sik-Ho Tsang. (2018). Review: Xception — *With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification) | by Sik-Ho Tsang | Towards Data Science*. https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568#5cb1.

Syafeeza, A. R., Itqan, K. S., Gong, F. G., Mustafa, N., Wong, Y. C., & Ibrahim, M. M. (2016). User identification system based on finger-vein patterns using Convolutional Neural Network. *ARPN Journal of Engineering and Applied Sciences, 11*(5), 3316–3319.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June*, 1–9. 10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.308

Tang, S., Zhou, S., Kang, W., Wu, Q., & Deng, F. (2019). Finger vein verification using a Siamese CNN. *IET Biometrics, 8*(5), 306–315. https://doi.org/10.1049/bme2.v8.510.1049/iet-bmt.2018.5245

Uchoa, V., Aires, K., Veras, R., Paiva, A., & Britto, L. (2020). Data Augmentation for Face Recognition with CNN Transfer Learning. *International Conference on Systems, Signals, and Image Processing, 2020-July*, 143–148. 10.1109/IWSSIP48289.2020.9145453.

Wang, C., Muhammad, J., Wang, Y., He, Z., & Sun, Z. (2020). Towards Complete and Accurate Iris Segmentation Using Deep Multi-Task Attention Network for Non-Cooperative Iris Recognition. *IEEE Transactions on Information Forensics and Security, 15*, 2944–2959. https://doi.org/10.1109/TIFS.1020610.1109/TIFS.2020.2980791

Wilaiprasitporn, T., Ditthapron, A., Matchaparn, K., Tongbuasirilai, T., Banluesombatkul, N., & Chuangsuwanich, E. (2020). Affective EEG-Based Person Identification Using the Deep Learning Approach. *IEEE Transactions on Cognitive and Developmental Systems, 12*(3), 486–496. https://doi.org/10.1109/TCDS10.1109/TCDS.2019.2924648

Win, K. N., Li, K., Chen, J., Viger, P. F., & Li, K. (2020). Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Generation Computer Systems, 110*, 758–771. https://doi.org/10.1016/j.future.2019.10.019

Wu, J. D., & Liu, C. T. (2011a). Finger-vein pattern identification using principal component analysis and the neural network technique. *Expert Systems with Applications, 38*(5), 5423–5427. https://doi.org/10.1016/j.eswa.2010.10.013

Wu, J. D., & Liu, C. T. (2011b). Finger-vein pattern identification using SVM and neural network technique. *Expert Systems with Applications, 38*(11), 14284–14289. https://doi.org/10.1016/j.eswa.2011.05.086

Wu, J. D., & Ye, S. H. (2009). Driver identification using finger-vein patterns with Radon transform and neural network. *Expert Systems with Applications, 36*(3 PART 2), 5793–5799. https://doi.org/10.1016/j.eswa.2008.07.042

Xie, C., & Kumar, A. (2019). Finger vein identification using Convolutional Neural Network and supervised discrete hashing. *Pattern Recognition Letters, 119*, 148–156. https://doi.org/10.1016/j.patrec.2017.12.001

Yang, G., Xi, X., & Yin, Y. (2012). Finger vein recognition based on (2D) 2 PCA and metric learning. *Journal of Biomedicine and Biotechnology, 2012*, 1–9. https://doi.org/10.1155/2012/324249

Yang, J., & Shi, Y. (2012). Finger-vein ROI localization and vein ridge enhancement. *Pattern Recognition Letters, 33*(12), 1569–1579. https://doi.org/10.1016/j.patrec.2012.04.018

Yang, J., Shi, Y., & Yang, J. (2011). Personal identification based on finger-vein features. *Computers in Human Behavior, 27*(5), 1565–1570. https://doi.org/10.1016/j.chb.2010.10.029

Yang, L., Yang, G., Yin, Y., & Xiao, R. (2013). Sliding window-based region of interest extraction for finger vein images. *Sensors (Switzerland), 13*(3), 3799–3815. https://doi.org/10.3390/s130303799

Yang, Wengming. (2014). *Tsinghua University Finger Vein and Finger Dorsal Texture Database (THU-FVFDT)*. https://www.sigs.tsinghua.edu.cn/labs/vipl/thu-fvfdt.html.

Yang, W., Hui, C., Chen, Z., Xue, J. H., & Liao, Q. (2019). FV-GAN: Finger Vein Representation Using Generative Adversarial Networks. *IEEE Transactions on Information Forensics and Security, 14*(9), 2512–2524. https://doi.org/10.1109/TIFS.2019.2902819

Yin, Y., Liu, L., & Sun, X. (2011). SDUMLA-HMT: A multimodal biometric database. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7098 LNCS*, 260–268. 10.1007/978-3-642-25449-9_33.

Yuan, C., Xia, Z., Sun, X., & Wu, Q. M. J. (2020). Deep Residual Network with Adaptive Learning Framework for Fingerprint Liveness Detection. *IEEE Transactions on Cognitive and Developmental Systems, 12*(3), 461–473. https://doi.org/10.1109/TCDS.2019.2920364

Zhang, S., Wu, X., You, Z., & Zhang, L. (2017). Leaf image based cucumber disease recognition using sparse representation classification. *Computers and Electronics in Agriculture, 134*, 135–141. https://doi.org/10.1016/j.compag.2017.01.014

Zhang, Y., Zhao, Z., Guo, C., Huang, J., & Xu, K. (2019). ECG Biometrics Method Based on Convolutional Neural Network and Transfer Learning. *Proceedings - International Conference on Machine Learning and Cybernetics, 2019-July*, 1–7. 10.1109/ICMLC48188.2019.8949218.

Zhao, J.-Y., Gong, J., Ma, S. T., Lu, Z. M., Chu, S. C., & Roddick, J. F. (2019). Finger vein recognition scheme based on convolutional neural network using curvature gray image. *Journal of Network Intelligence, 4*(3), 114–123.

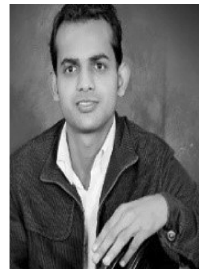
Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems, 30*(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

Zhongbo, Z., Siliang, M., & Xiao, H. (2006). Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network. *Proceedings - International Conference on Pattern Recognition, 4*, 145–148.

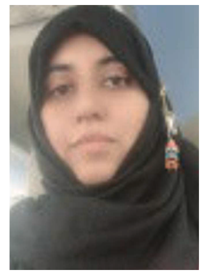
Kashif Shaheed is a Ph.D. Candidate at School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received his Master Degree from Shandong University, Shandong, China in 2019. He has published more than 10 research papers in both reputable journals and conferences. He is a reviewer of IEEE Access, IEEE Transaction on instrumentation and measurement, Concurrency and Computation: Practices and several other reputed journals. His area of research interests are Biometric finger vein recognition, Deep learning, Computer Vision and Pattern recognition.

Mao Aihua is a associate professor with the School of Computer Science and Engineering, South China University of Technology (SCUT), China. He received the PhD degree from Hong Kong Polytechnic University in 2009, the M.Sc. degree from Sun Yat Sen University in 2005 and B.Eng. degree from Hunan University in 2002. His research interests include physical-based simulation, computer aided design and computer graphic.

Dr. Imran Qureshi received his BCS (Hons), MS(CS) and Ph.D. degrees from the University of Peshawar in Pakistan, COMSATS University in Pakistan and Shandong University in China, respectively. He is currently working as an Assistant Professor in the Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan. He is also engaged as a Postdoctoral Researcher with the College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, P.R. China. He has published more than 14 research papers in both reputable journals and conferences. His research interests include image processing, pattern recognition and computer vision.



Munish Kumar received his Master's degree in Computer Science Engineering from Thapar University, Patiala, India in 2008. He received his Ph.D. degree from Thapar University, Patiala, India in 2015. He started his career as an Assistant Professor in computer application at Jaito Centre of Punjabi University, Patiala. Presently, he is working as Assistant Professor in the Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India. He has published more than 100 articles in reputed international journals and conferences. He has 1300+ citations for his articles at Google scholar. His research interests include Character Recognition, Computer Vision and Pattern Recognition. He is a Professional Member of ACM and IEEE.



Sumaira Hussain received the M.S. degree in computer science from Virtual University, Pakistan, in 2014. She is currently pursuing the Ph.D. degree in computer science with Shandong University, Jinan, Shandong, China. Her research interests include machine learning, medical image analysis, and computer vision.

Inam Ullah is currently pursuing a Ph.D. degree from the School of Software Engineering, Shandong University, Jinan, China. He received his bachelor's degree from the University of Peshawar, Pakistan, and his Masters degree from the International Islamic University of Islamabad, Pakistan. His current research interests include image processing, Computer vision, Machine Learning, deep learning, Salient Object Detection, and biometric recognition.

Xingming Zhang received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1996. He is currently a Professor, Doctoral Supervisor, and Vice-dean with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is a member of the Standing Committee of the Technical Committee of Education, China Computer Federation, and an Executive Director of the Computer Federation of Guangdong Province. His research focuses on image processing, video coding and surveillance.