Automated Biometrical Fingerprint Recognition Scheme using Synthesized Images

Erdal Alimovski
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-0909-2047

Jawad Rasheed
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-3761-1641

***Abstract*—The evolution of digitization has engulfed various methods of forensic sciences, such as fingerprint detection, recognition or recovery of partial prints. Prior to computerization, huge fingerprint repositories were manually maintained and involve humans for classification. But the advent of artificial intelligence-based tools performs the fingerprint recognition much faster and easier. Therefore, this study proposed CNN-based deep learning technique to extract effective features from ridges and valleys of skin impression for accurate recognition. The experimental work is based on FVC2020 dataset to train and test the proposed model. Moreover, ResNet50 framework is also tested on this dataset, and results shows that proposed model achieved an accuracy of 81.25%, whereas ResNet50 attained 79.25% accuracy. Furthermore, the incorporation of convolutional auto-encoder (CAE) based model for enhancing the dataset by generating synthetic fingerprint images, improved the recognition accuracy of proposed CNN-based model to 86.0%.**

Keywords*— *Fingerprint, deep learning, convolutional neural network, convolutional auto-encoders

I. INTRODUCTION

With growing demand for security all around the world, biometrics systems become the necessity of institutional life. Biometrics is a science that describes people in terms of physiological and behavioral terms. Physiological features cover characteristics such as palmprint, iris, fingerprint, facial part, while behavioral features include gait, gestures, signature and voice. The advancement in computerization has eased the usage of biometrics technology, thus it is widely adopted in vicinities of airports, banks, schools, or official buildings. Moreover, smart devices have also incorporated biometric features to secure the devices from intruders [1] [2].

Fingerprints are one of the most preferred solutions to authenticate people in biometric technology. A fingerprint is the presentment of the epidermis of a finger: it includes a sample of intermittent edges and valleys [3]. Each person has a matchless fingerprint which is mostly composed of few components such as ridges, grooves, and direction of lines. Ridges consist of three fundamental patterns namely, arch, loop and whorl. In general, patterns of fingerprint are determined by features such as ridge, minutiae and spots. Despite of the fact that fingerprints are distinct and highly distinguishable, but the performance accuracy of fingerprint recognition systems are associated with factors like quality of the image and applied matching algorithm [4].

Machine learning (ML) is a field of computer science that uses classification algorithms to identify patterns in large data for effective possible prediction. Beside ML tools, scientists have widely used Deep learning (DL) techniques, based on complex structures created with hierarchical modules to learn data representations, for tasks like image classification [5], segmentation [6], text detection [7], fingerprint recognition [8], face recognition [9], object detection [10], and fault prediction [11].

DL-based models require huge amount of data for proper training and effective prediction, however, very few and limited datasets related to fingerprint images are publicly available. Thus, this study exploited CAE to generate synthesized fingerprint images that are later used to train and generalize the proposed CNN model. The proposed CNN-based classification model is then evaluated on test set for accurate fingerprint recognition task. Moreover, CNN-based ResNet50 architecture is also implemented for comparative analysis. In addition to this, the effect of data synthesis on classification models is also analyzed.

The organization of this paper is as follows. In Section II, related work on fingerprint recognition task are provided. Section III describes dataset and methodology of the paper, while Section IV analysis the experimental results. Finally, paper ends with Section V by outlining the concluding remarks.

II. RELATED WORK

Finger recognition is key research area since last decade. Computer vision scientists have proposed various artificial intelligence based tools for fast and accurate finger classification, such as Stojanovic et al. [12] proposed DL-based framework for fingerprint recognition using CNN to extract the core of fingerprint Region of Interest (RoI). Experiments were performed in two variations with and without Gaussian noise. The proposed model obtained promising results while tested on FVC2002 dataset. Similarly, Shrein [13] suggested CNN-based Lenet-5 model for fingerprint image classification. Moreover, some image pre-processing techniques were applied to boost the performance while sufficiently reducing the training time. Authors evaluated the model using NIST-DB4 dataset to get an accuracy of 95.9%.

Darlow et al. [14] proposed deep CNN-based minutia extraction network (MENet), that consists of five convolutional layers, followed by 2 fully connected layers with softmax activation function. Later, some post-processing is performed to identify minutiae locations from the output of proposed MENet model. It is noted from experimental section that MENet outperformed previous minutiae extractor.

An end-to-end fingerprint recognition framework is proposed by Minaee et al. [15]. Before passing the PolyU dataset to proposed CNN-based network, affine transformation is employed to increase the data for each class. Thus, the data augmentation empowered the model to fit better that accomplished an accuracy of 95.7%. Likewise, Fanfeng et al. [16] proposed DL-based recognition framework to recognize partial fingerprint images. To improve the recognition of partial fingerprints, authors used two loss functions, one for training while another for feature extraction. The model is trained and evaluated on NIST-DB4 and self-built NCUT-FR datasets that performed better than many existing techniques developed for partial fingerprint classification.

Authors in [17] blended a novel discriminative restricted Boltzmann machines (DRBM) and deep Boltzman machines (DBM) model to examine fingerprints. DBM is used to extract features from grayscale images, which are then feed to k-nearest-neighbors (kNN) classifier to analyze spoof forgeries. A novel method inspired from VGG-16 is proposed in [18] for aligned fingerprints matching. The model is tested using two publicly available datasets; NIST SD04, and NIST SD14. Besides various developed frameworks, there is still room to suggest more methods with higher performance accuracy.

III. Materials and Methods

The publicly available dataset is downloaded and later data augmentation technique is employed using CAE to produce synthesized images of fingerprints. Both datasets (original and augmented) are then used to train the proposed method. This section outlines the dataset and proposed methods in details.

A. Dataset

In this study FVC2002 [19] dataset is used as original dataset. FVC2020 dataset contains four classes and each class consist of 80 images. Hence the total number of images is 320. Except images in one class all the images were obtained from different sensors. Some of sample images of FVC2002 dataset are demonstrated in Fig. 1.

Fig. 1. Fingerprint samples of FVC2002 dataset.

B. Data Augmentation using Convolutional Auto-encoders

A data augmentation technique is employed to generate synthetic images using CAE. Auto-encoders are unsupervised machine learning algorithms that aims to rebuild the input data back using lower-dimensional representation [20]. Usually, auto-encoders consist of two parts: encoder, and decoder. The encoder has the ability to transform the input x into a representation h, also called as code, by using deterministic function as below

$$h = f_{\theta}(x) = \sigma(Wx + b) \quad (1)$$

with parameters $\theta = \{W, b\}$, where W is matrix's weight, b represents the bias of vector, while σ refers to the activation function. Similarly, reverse mapping of f is done by:

$$r = g_{\theta'}(h) = \sigma'(W'h + b') \quad (2)$$

While dealing with images, CAE can be more effective [20]. Generally, the structure of CAE is very similar to any other auto-encoder. Compared with normal auto-encoders, CAE uses convolutional and pooling layers to extract the features and minimizes the size of the input image. Therefore, a CAE architecture is proposed, considering the suggestion described in [21] and [22]. A detailed network topology of proposed CAE is outlined in Table I.

TABLE I. Network Topology of Proposed Convolutional Auto-Encoders (CAE)

CAE Architectural Details		
Layer type	***Output shape***	***No. of Prams***
input_1(InputLayer)	None, 224, 224, 1	0
conv2d_1 (Conv2D)	None, 224, 224, 32	320
max_pooling2d_1	None, 112, 112, 32	0
conv2d_2 (Conv2D)	None, 112, 112, 64	18496
max_pooling2d_2	None, 56, 56, 64	0
conv2d_3 (Conv2D)	None, 56, 56, 128	73856
conv2d_4 (Conv2D)	None, 56, 56, 128	147584
up_sampling2d_1	None, 112, 112, 128	0
conv2d_5 (Conv2D)	None, 112, 112, 128	73792
up_sampling2d_2	None, 224, 224, 64	0
conv2d_6 (Conv2D)	None, 224, 224, 1	577

FVC2002 dataset is used to train the proposed model to generate synthetic images of fingerprints. The training is performed to reduce the differences between input and its reconstructed images. Fig. 2 shows few synthetic images of fingerprints generated by proposed CAE after 300 epochs. The augmented dataset constitutes of combination of original images and synthetic images, thus raised the dataset tally to 430 samples.

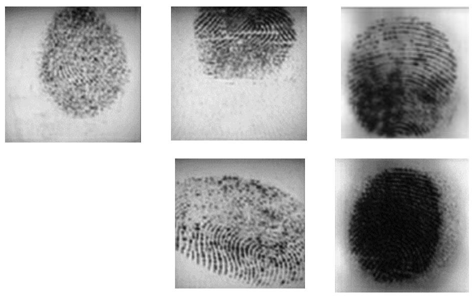


Fig. 2. Samples of generated sythetic fingerprint images.

C. Classification using Convolutional Neural Network

CNN is deep learning based powerful tool for recognizing local patterns in data instances [23]. As images are interrelated data, CNN based architectures are widely adopted for image classification task. CNNs detect local patterns by creating feature maps through conducting element wise multiplication using kernel. As data at hand can't be described with linear functions thus a nonlinear function, such as rectified linear unit (ReLU), is usually used to normalize the extracted feature map values. A pooling layer, which comes in variations of maximum, average and sum pooling, is also sometimes applied for dimensionality reduction that results in less network parameters. Later, the resultant is flattened, converted to one long vector. Afterwards, a regular feed forward backpropagation neural network methodology is applied. Usually, at end of architecture, a fully connected layer is placed to calculate the probabilities for different classes using the features detected from prior steps. Finally, the network is back propagated based on the selected optimizer function for adjusting the weights.

The proposed CNN model consist of four convolutional layers, followed by flattening layer and fully connected layer. For this study, max pooling is utilized as it has been found more effective in previous studies [24]. It reduces the dimensions of the feature map while maintaining the most important identity values [25]. Between each convolutional layers, max pooling layers are placed. In order to eliminate overfitting among third convolutional layer and fully connected layers dropout layers are used. ReLU is used as activation function in each convolutional layer except last that uses Softmax. Adam is used as optimizer while batch size is set to 10.

In order to compare proposed CNN model, as a next phase we apply ResNet50 model. ResNet50 model is also trained and tested in both original and augmented datasets. ResNet50 architecture were performed with distinct hyper-parameters in order to reach the best accuracy. So, with following hyper-parameters achieved the best performance: optimizer Adam, learning rate 0.002, number of epoch 9 and batch size 10.

IV. EXPERIMENTAL RESULTS

In this study, all the experiments were performed using Jupyter Notebook with Python (3.7) programming language. In addition, we used Tensorflow, Keras, Sklearn and Seaborn libraries.

As described earlier, original dataset and augmented datasets are used separately for experimental work. Note that, for each experiment 90% of both datasets were used for training and remaining 10% for testing. In order to analyze the effect of data augmentation and the performance of proposed CNN model, various performance metrics have been measured, such as accuracy, confusion matrix, precision, recall and f1-score, using Equations (3)-(6)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

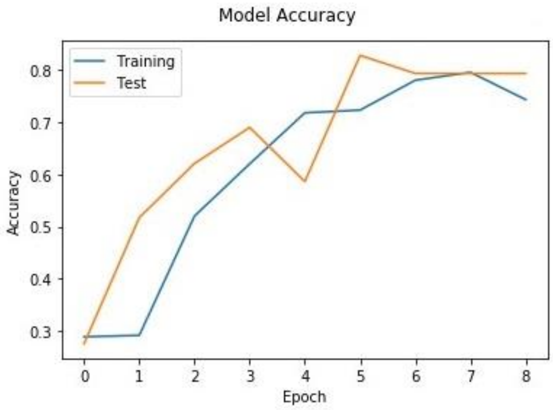
$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

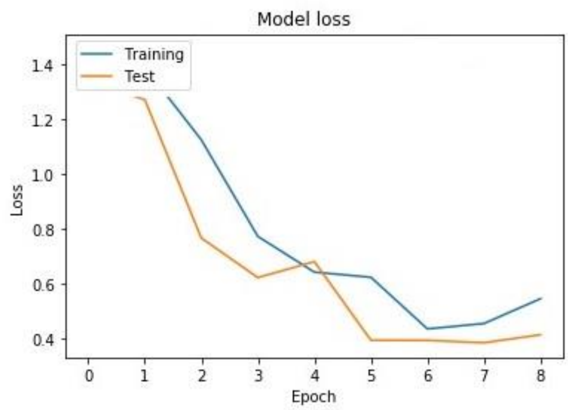
$$F1 - score = \frac{2 * (Recall + Precision)}{(Recall + Precision)} \tag{6}$$

where TP refers to true positive, TN refers to true negative, FP refers to false positive and FN is false negative.

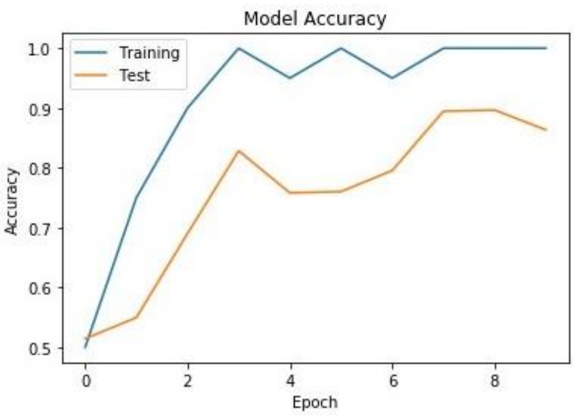
The proposed CNN model is first trained and evaluated on original dataset, which secured an accuracy of 81.25%. The accuracy and loss curves of model on original dataset is depicted in Fig. 3(a)-(b), respectively. During experiments, the model is trained with 5, 9, 10 and 15 number of epochs, respectively, thus it accomplished the highest accuracy epochs were set to 9. Moreover, experiments were repeated with



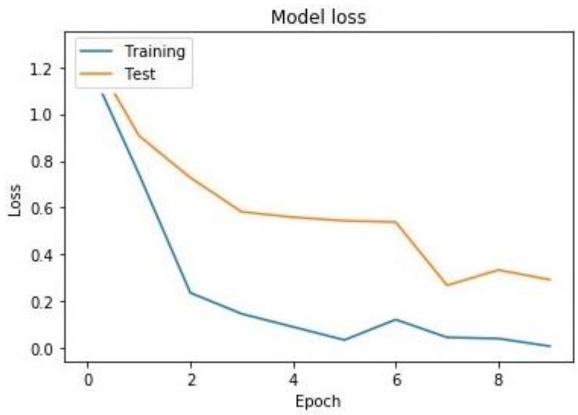
(a)



(b)

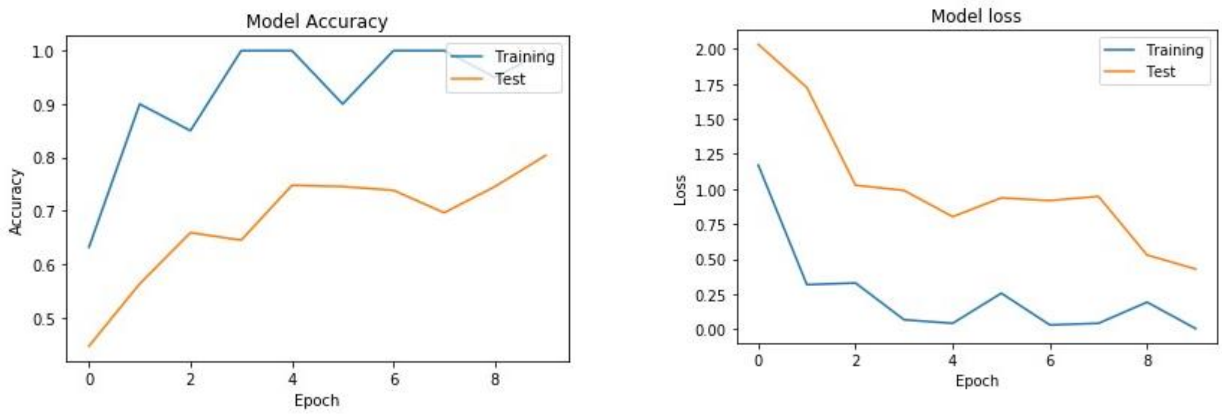


(c)



(d)

Fig. 3. Training and loss curves of proposed model, (a) accuracy on original dataset, (b) loss on original dataset, (c) accuracy on augmented dataset, and (d) loss on augmented dataset.

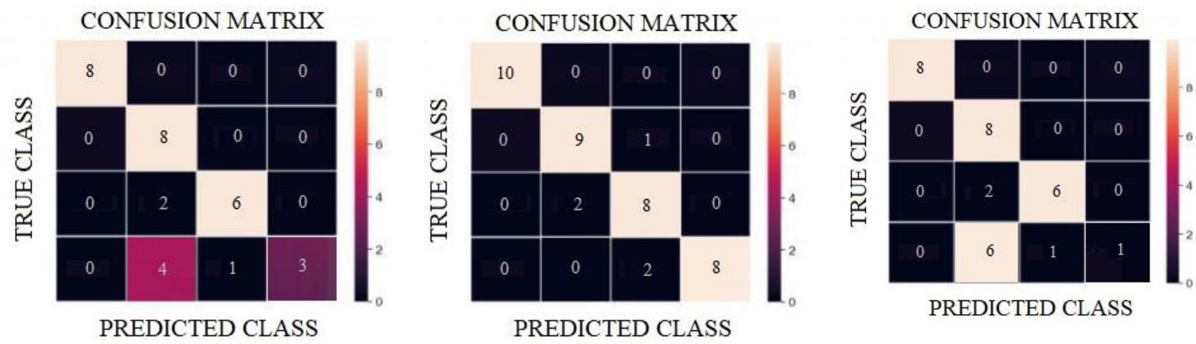


(a) (b)

Fig. 4. Training and loss curves of ResNet50 on original dataset.

TABLE II. COMPARATIVE ANALYSIS OF PROPOSED MODEL WITH RESNET50

Performance Analysis					
Method	***Dataset***	***Accuracy***	***Precision***	***Recall***	***F1-score***
Proposed	Original	0.81	0.72	0.78	0.75
	Augmented	0.86	0.78	0.83	0.80
ResNet50	Original	0.79	0.70	0.75	0.72



(a) (b) (c)

Fig. 5. Confusion matrix of (a) proposed model on original dataset, (b) proposed model on augmented dataset, and (c) ResNet50 on original dataset.

different combinations of convolutional layers without pooling layers, but the network's performance degraded. In addition to this, experiments were repeated for augmented dataset. The proposed classification CNN-based model is trained on augmented dataset that achieved accuracy of 86% on respective test set. The accuracy and loss curves for augmented dataset is shown in Figure 3(c)-(d), respectively. Evidently, the experimental results shows that the incorporation of convolutional auto-encoders for data augmentation has significantly increased the performance accuracy of proposed recognition network.

Besides the proposed classification model, a state-of-the-art DL based image classification framework, known as ResNet50 is also trained and evaluated for comparative analysis. For the problem at hand, fingerprint recognition task, ResNet50 performed less accurately when compared with proposed model of this study. ResNet50 model hardly attained an accuracy of 79.25% when trained and evaluated on original dataset, as shown in Figure 4(a), while Figure 4(b) depicts its loss curve. A detailed comparative analysis is presented in Table II.

For better visualization, confusion matrix for each experiment is also composed. It can be observed from Figure 5(a), which depicts the confusion matrix of proposed model when trained and test on original dataset that first two classes are correctly distinguished but 2 fingerprint images of Class-3 are misclassified. Similarly, half of the images in Class-4 are also misclassified by proposed model. On the other hand, with the addition of data augmentation technique, the model performed better while distinguishing Class-4 samples as shown in Figure 5(b). The ResNet50 performed worse than others while recognizing Class-3 and Class-4 instances as illustrated in Figure 5(c). It wrongly classified seven fingerprint images of Class-4, whereas only one image is recognized correctly. Thus, the experimental results shows that the proposed model outperformed other models when trained on augmented dataset.

V. Conclusion

In this paper we proposed a CNN based model to perform fingerprint recognition task. At first, the proposed model is trained on publicly available FVC2002 dataset, which is then evaluated on remaining 10% set that achieved an accuracy of 81.25%. As DL models are data hunger, thus this study also proposed a model to increase the fingerprint images samples. Therefore, the dataset is enhanced using convolutional auto-encoder that generated synthetic fingerprint images. The proposed model is then trained on augmented dataset, which contains original images as well synthetized images. Evidently, the synthesized images technique increased the overall performance of classification network by 4.75%. Moreover, ResNet50 framework is also implemented and trained on downloaded dataset, which attained an accuracy of 79.25%, thus performed worse than the proposed model. Even though, the proposed model with data augmentation approach attained promising results, but still there is room for further enhancement which can be done in future works by training on other datasets.

References

[1] D. Bouchaffra and A. Amira, "Structural hidden Markov models for biometrics: Fusion of face and fingerprint," *Pattern Recognition*, vol. 41, no. 3, pp. 852–867, 2008, doi: 10.1016/j.patcog.2007.06.033.

[2] M. D. Security, "A Mobile Biometric System-," no. April, pp. 13–19, 2010.

[3] E.- Book, S. Vishwas, and L. D. Resolution, "E-Handbook Sabka Vishwas (Legacy Dispute Resolution) Scheme , 2019," vol. 2019, 2019.

[4] W. J. Wong and S. H. Lai, "Multi-task CNN for restoring corrupted fingerprint images," *Pattern Recognition*, vol. 101, p. 107203, 2020, doi: 10.1016/j.patcog.2020.107203.

[5] J. Rasheed, H. B. Dogru, and A. Jamil, "Turkish Text Detection System from Videos Using Machine Learning and Deep Learning Techniques," in *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Aug. 2020, pp. 116–120, doi: 10.1109/DSMP47368.2020.9204036.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "A deep convolutional encoder-decoder architecture for image segmentation}," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] J. Rasheed, A. Jamil, H. B. Dogru, S. Tilki, and M. Yesiltepe, "A Deep Learning-based Method for Turkish Text Detection from Videos," in *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, Nov. 2019, pp. 935–939, doi: 10.23919/ELECO47770.2019.8990633.

[8] H.-R. Su, K.-Y. Chen, W. J. Wong, and S.-H. Lai, "High-Resolution Fingerprint Recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2017*, pp. 2057–2061, 2017, [Online]. Available: https://ieeexplore.ieee.org/document/7952518/authors#authors.

[9] J. Rasheed, E. Alimovski, A. Rasheed, Y. Sirin, A. Jamil, and M. Yesiltepe, "Effects of Glow Data Augmentation on Face Recognition System based on Deep Learning," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2020, pp. 1–5, doi: 10.1109/HORA49412.2020.9152900.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] H. Rashid, E. Khalaji, J. Rasheed, and C. Batunlu, "Fault Prediction of Wind Turbine Gearbox Based on SCADA Data and Machine Learning," in *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*, Sep. 2020, pp. 391–395, doi: 10.1109/ACIT49673.2020.9208884.

[12] B. Stojanovic, O. Marques, A. Neskovic, and S. Puzovic, "Fingerprint ROI segmentation based on deep learning," *24th Telecommunications Forum, TELFOR 2016*, pp. 5–8, 2017, doi: 10.1109/TELFOR.2016.7818799.

[13] J. M. Shrein, "Fingerprint classification using convolutional neural networks and ridge orientation images," *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, vol. 2018-Janua, pp. 1–8, 2018, doi: 10.1109/SSCI.2017.8285375.

[14] L. N. Darlow and B. Rosman, "Fingerprint minutiae extraction using deep learning," *IEEE International Joint Conference on Biometrics, IJCB 2017*, vol. 2018-Janua, pp. 22–30, 2018, doi: 10.1109/BTAS.2017.8272678.

[15] S. Minaee, E. Azimi, and A. Abdolrashidi, "FingerNet: Pushing the limits of fingerprint recognition using convolutional neural network," *arXiv*, 2019.

[16] F. Zeng, S. Hu, and K. Xiao, "Research on partial fingerprint recognition algorithm based on deep learning," *Neural Computing and Applications*, vol. 31, no. 9, pp. 4789–4798, 2019, doi: 10.1007/s00521-018-3609-8.

[17] D. M. Uliyan, S. Sadeghi, and H. A. Jalab, "Anti-spoofing method for fingerprint recognition using patch based deep learning machine," *Engineering Science and Technology, an International Journal*, vol. 23, no. 2, pp. 264–273, 2020, doi: 10.1016/j.jestch.2019.06.005.

[18] Y. Liu, B. Zhou, C. Han, T. Guo, and J. Qin, "A novel method based on deep learning for aligned fingerprints matching," *Applied Intelligence*, vol. 50, no. 2, pp. 397–416, 2020, doi: 10.1007/s10489-019-01530-4.

[19] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," *Proceedings - International Conference on Pattern Recognition*, vol. 16, no. 3, pp. 811–814, 2002, doi: 10.1109/icpr.2002.1048144.

[20] K. G. Kim, "Deep learning book review," *Nature*, vol. 29, no. 7553, pp. 1–73, 2019.

[21] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 52–59, 2011, doi: 10.1007/978-3-642-21735-7_7.

[22] F. Chollet, "Building Autoencoders in Keras," 2016. https://blog.keras.io/building-autoencoders-in-keras.html (accessed Dec. 15, 2020).

[23] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Cnn实际训练的," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2012.

[24] D. C. Cires, U. Meier, J. Masci, and L. M. Gambardella, "IJCAI11-210.pdf," *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence Flexible*, pp. 1237–1242, 2003, [Online]. Available: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/viewFile/3098/3425.

[25] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *ArXiv e-prints*, no. November, 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.