Left or Right Hand Classification from Fingerprint Images Using a Deep Neural Network

Junseob Kim[1], Beanbonyka Rim[1], Nak-Jun Sung[1] and Min Hong[2, *]

Abstract: Fingerprint security technology has attracted a great deal of attention in recent years because of its unique biometric information that does not change over an individual's lifetime and is a highly reliable and secure way to identify a certain individuals. AFIS (Automated Fingerprint Identification System) is a system used by Korean police for identifying a specific person by fingerprint. The AFIS system, however, only selects a list of possible candidates through fingerprints, the exact individual must be found by fingerprint experts. In this paper, we designed a deep learning system using deep convolution network to categorize fingerprints as coming from either the left or right hand. In this paper, we applied the Classic CNN (Convolutional Neural Network), AlexNet, Resnet50 (Residual Network), VGG-16, and YOLO (You Only Look Once) networks to this problem, these are deep learning architectures that have been widely used in image analysis research. We used total 9,080 fingerprint images for training and 1,000 fingerprint to test the performance of the proposed model. As a result of our tests, we found the ResNet50 network performed the best at determining if an input fingerprint image came from the left or right hand with an accuracy of 96.80%.

Keywords: Deep Learning, convolution neural network, fingerprint classification.

1 Introduction

Biometric information encompasses an individual's measurable, unique characteristics including physical characteristics such as fingerprint, iris, face, and vein, and behavioral characteristics such as voice, signature, and handwriting. Biometric information is attracting great attention mainly as a security technology because of its high reliability and security due to it being unique for each individual. In particular, the fingerprint has a characteristic that everyone's fingerprint is different in shape, and this shape does not change over a lifetime. Therefore, it is possible to efficiently identify and authenticate a person with a fingerprint, this has recently been widely used as a simple authentication means for a mobile device such as a smartphone [Maltoni, Maio, Jain et al. (2009)].

In general, fingerprint recognition is performed based on image processing technology

[1] Department of Computer Science, Soonchunhyang University, 31538, Asan, Korea.

[2] Department of Computer Software Engineering, Soonchunhyang University, 31538, Asan, Korea.

[*] Corresponding Author: Min Hong. Email: mhong@sch.ac.kr.

Received: 07 November 2019; Accepted: 18 December 2019

that, finds feature points (cores and deltas) of a fingerprint and analyzes correlations between these feature points. Fingerprint recognition systems work to narrow classification using such characteristics as arch, whorl, loop, etc., in order to perform recognition quickly and accurately. Recently, research for narrowing the fingerprint search through various features has been conducted. This research looks not only into feature extraction of fingerprints through image processing technology, but also investigates detecting spoofing of fingerprints by applying deep learning technology [Darlow and Rosman (2017); Menotti, Chiachia, Falcao et al. (2015)].

Most of the existing research has only studied how to secure fingerprints and improve the hit rate of fingerprint recognition. In the Republic of Korea, the fingerprint recognition technology employed at actual crime scenes is called AFIS (Automated Fingerprint Identification System), which is a system that takes as inputs the pattern of a fingerprint, attempts to find a fingerprint similar to that pattern and selects a candidate group. AFIS matches one input to multipl -fingerprints in database of more than 1 million fingerprints, this pre-processing stages reduces the number of suspects to about 30-50 people. These candidates fingerprints are then visually inspected by fingerprint experts for final judgment. In such a system, even after the candidate group is selected, the fingerprint experts have to examine a 40 additional fingerprints. However, only 11,494 fingerprints out of 19,632 cases were successfully identified using by this system, asuccess rate of 58.5%. Therefore, better results can be expected by improving the traditional AFIS's ability to narrow down the search before experts appraise the candidates.

This paper proposes a method to further classify fingerprints using deep learning technology to increase the efficiency of fingerprint retrieval systems such as AFIS. Deep learning is a machine learning technique that creates virtual neurons and connects them in a structure similar to the human brain. This technology allows a large number of virtual neurons to discover various features from the data and to classify and analyze more complex data better than humans. In this paper, Classic CNNs (Convolutional Neural Network), a deep learning network widely used for data classification: AlexNet [Krizhevsky, Hinton and Sutskever (2012)], ResNet (Residual Network) [He, Ren, Sun et al. (2016)], VGG-16 [Simonyan and Zisserman (2014)], and YOLO (You Only Look Once) [Redmon and Farhadi (2018)] are used to train the system to improve fingerprint recognition.

This paper consists of five sections. Section 2 introduces research related to fingerprint classification. The preprocessing method of fingerprint images and the architecture of the deep learning networks used are described in Section 3. Section 4 describes the fingerprint classification results derived from deep learning networks. Finally, Section 5 describes the results of deep learning in the fingerprint domain and future research directions.

2 Related works

Previous research related to the classification of fingerprints has been conducted on the methods of classifying fingerprints into arch, tented arch, left loop, right loop, and whorl characterisitcs. For example, Michelsanti et al. [Michelsanti, Ene, Guichi et al. (2017)] conducted research on how to accelerate fingerprint classification using a CNN architecture based on deep learning technology. Their method has 94.4-95.05% accuracy using pre-trained CNN, VGG-F and VGG-S, also it is fast to classify the above types without the need for a time-consuming pretreatment step. The fingerprint classification of this method was mostly carried out with the types mentioned above and focuses on getting results faster and more accurately.

Gnanasivam et al. [Gnanasivam and Vijayarajan (2019)] conducted a study to infer gender from fingerprints using image processing techniques based on the number of fingerprints and the size of the fingertips. They proposed an OSA (Optimal Score Assignment) method to calculate optimal scores for men and women from an internally collected fingerprint database. Using the fingerprints of four age groups collected by the scanner, the number of ridges from the core to the delta were counted to determine gender. The success rate of their method was between 84.1% and 90.11%.

Nogueira et al. [Nogueira, de Alencar Lotufo and Machado (2016)] distinguished real fingerprint images mixed with spoofing images based on pre-trained CNNs. They trained fingerprint data using CNN-VGG, CNN-AlexNet, and CNN-Random with an accuracy of 97.1% and 95.5% accuracy in LivDet 2015 (The Fingerprint Liveness Detection Competition). In this paper, they confirmed that their pre-trained CNN outperforms the classic LBP (Local Binary Pattern) pipeline. Tab. 1 gives a summary of fingerprint related researches.

In this paper, we are not focused on how to classify fingerprints in to specific forms such as arch, tented arch, left loop, right loop, and whorl, but how to extract new features from fingerprints using deep learning technology. The proposed method classifies whether the input fingerprint is from the left-hand or right-hand using a learning model with deep learning technology.

Table 1: Summary of fingerprint related research

	Applying Field	**Feature**
Daniel Michelsanti	Fingerprint Type Classification	95% Acc, Deep Learning, Fiinger Type Classification
P.Gnanasivam	Gender Classification using Fingerprint	90% acc, Image Processing, professional knowledge of fingerprint
Rodrigo Frasetto Nogueira	Fingerprint Liveness Detection	95% Acc, Deep Learning, Livness Detection

Deep learning is a set of algorithms that provides a high level of abstraction through a combination of several nonlinear transformations. Deep learning techniques are a way of representing data in a computer-readable form. Among them, CNN can be used for voice and one-dimensional data, but mainly for image recognition, such as Tyan et al., Shen et al. [Tyan and Kim (2018); Shen, Tao, Li et al. (2019)]. CNN has shown good performance with various image datasets such as MNIST, CIFAR-10, CIFAR-100, and ImageNet.

Since CNNs can learn without losing spatial information, these artificial neural networks can extract features and learn efficiently. CNNs effectively recognizes the characteristics of adjacent images while maintaining the shape of the input/output data at each layer as well as maintaining spatial information. CNNs consists of a convolution layer, which abstracts and compresses a specific part of the image and expresses it in that specific layer, and a pooling layer that collects and enhances the features of the extracted image. In addition, since a filter is used as a shared parameter, the learning parameter is very small compared to general artificial neural networks.

The performance of CNNs in the image classification field is so high that many CNN-based networks have produced excellent performances in the ILSVRC (ImageNet Large Scale Recognition Competition), which is an image recognition competition as shown by LeCun et al., Shrein. [LeCun, Bengio and Hinton (2015); Shrein (2017)]. Due to these characteristics, Li et al. [Li, Cai, Zhou et al. (2014)] and Rezende et al. [Rezende, Carvalho, De Geus et al. (2017)] showed that learning fingerprints with CNN-based architectures helped them achieve their research goals.

Li et al. [Li, Cai, Wang et al. (2014)] classified lung images into five types (normal, emphysema, ground glass, fibrosis, and micronodules) using a CNN. They created the best CNN framework for classification purposes and trained it to derive the results. Their work solved the problem of ambiguous visual structures and limited training data size in adapting to IDL (Interstitial Lung Disease) classification.

Rezende et al. [Rezende, Carvalho, De Geus et al. (2017)] proposed a malware classification approach based on the ResNet50 architecture using a deep neural network. This network trained the last softmax with a number of malware classification classes using ResNet50 pre-trained on the ImageNet dataset. Their method was experimented with a data set of 9,339 grayscale image samples and showed that ResNet50 can be effectively used to classify malware with 98.62% accuracy.

Kim et al. [Kim and Yu (2016)] used YOLO to detect the position of objects in sonic images, which are sound wave images of moving materials obtained with a forward sonar. 1,607 images were used for training, and 1,000 images were designated as test data for the CNN to detect objects in the images. Through their research, it was possible to navigate underwater with a robot without an arm, in contrast to an existing robot that needed an arm, in doing so they created a robot that shows good performance in object detection underwater.

In this paper, we applied classic CNN and Alexnet among the CNN architectures of deep learning technology. In addition, we also applied Resnet50 and VGG-16, which solved the problem of deepening the network,a problem inherent in deep networks with more layers than the CNN had already trained. Finally, we used YOLO to derive the results. The traditional CNN architecture uses three channels of image data, but since the fingerprint image has only one channel of grayscale image data, we modified the CNN architecture to one channel to optimize the computation. In addition, segmentation was performed on the original high-resolution images to extract only the fingerprint area in a rectangular shape, and then this was resized to adjust it to the input size of each architecture, which is described in detail in the methodology of Section 3.

3 Methodology

The proposed method aims to create a model that can readily classify images from the left-hand or right-hand. There are two things that are essential to using deep learning technology. The first is the computing power of the computer. To speed up learning, a GPGPU environment that can perform computations in parallel with multiple GPUs is essential. The typical GPGPU configuration used is Nvidia's Tesla and CUDA architectures. The second is the quality and quantity of the dataset for learning. When using deep learning techniques, although most of the learning model's structure is fine, there is a lot of data that causes errors, so we need a good dataset with the right amount and quality of data. This section describes the pre-processing process for fingerprint image data used in deep learning, the process of collecting data, and how to apply deep learning technology.

3.1 System overview

Fig. 1 shows the contents of the proposed system that distinguishes fingerprints from the left orright hand through the deep learning model proposed in this paper. The proposed system is divided into a data preprocessing section, input data section, DL model building section, and identification section. In the data preprocessing section, fingerprint images are collected and processed using image processing algorithms, they are then divided into training data sets for learning and validation data sets for validation. The input data section consists of a section that performs the process of putting data into the system. In the DL model building section, the classification model is trained and verified by selecting a network whose performance is already proven among CNN architectures that are strong in classification. Finally, the identification section outputs the results of the images from the input data section to the classification model created using deep learning techniques.

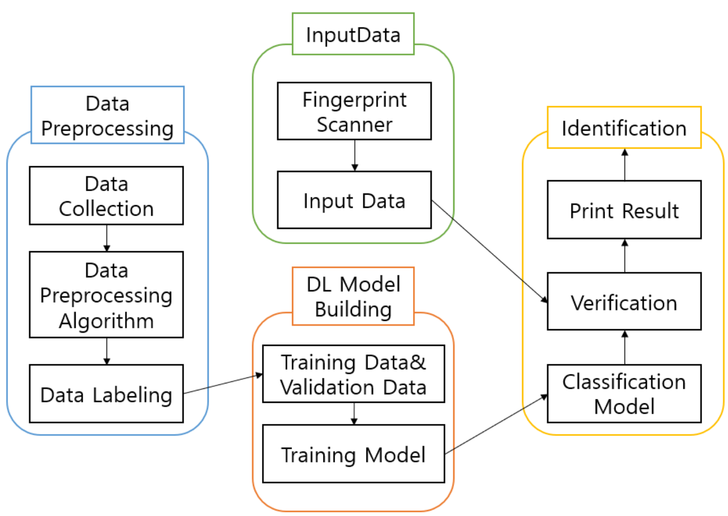

Figure 1: Architecture of proposed fingerprint classification system

3.2 Data pre-processing

We collected and scanned fingerprints directly using Integrated Biometrics' Sherlock scanner model, considering the characteristics of data it is difficult to disclose our dataset due to the personal information contained within. The collected fingerprint images were scanned at 800 (W)×750 (H) pixel size. A fingerprint data set was constructed by scanning 10 fingerprints for 1,008 people using the scanner, then data labeling appended left-hand, right-hand, finger information, gender, age and occupation information.

Then, in order to achieve high accuracy, pre-processing of the collected images was performed. The collected images contained some meaningless white backgrounds in addition to the fingerprints. Therefore, we extracted only the area of the fingerprint by applying rotation and segmentation appropriately to use only the necessary part of the source image. We also resized the segmented fingerprint images to a size of 224×224 to match the input size of the deep learning model being used.

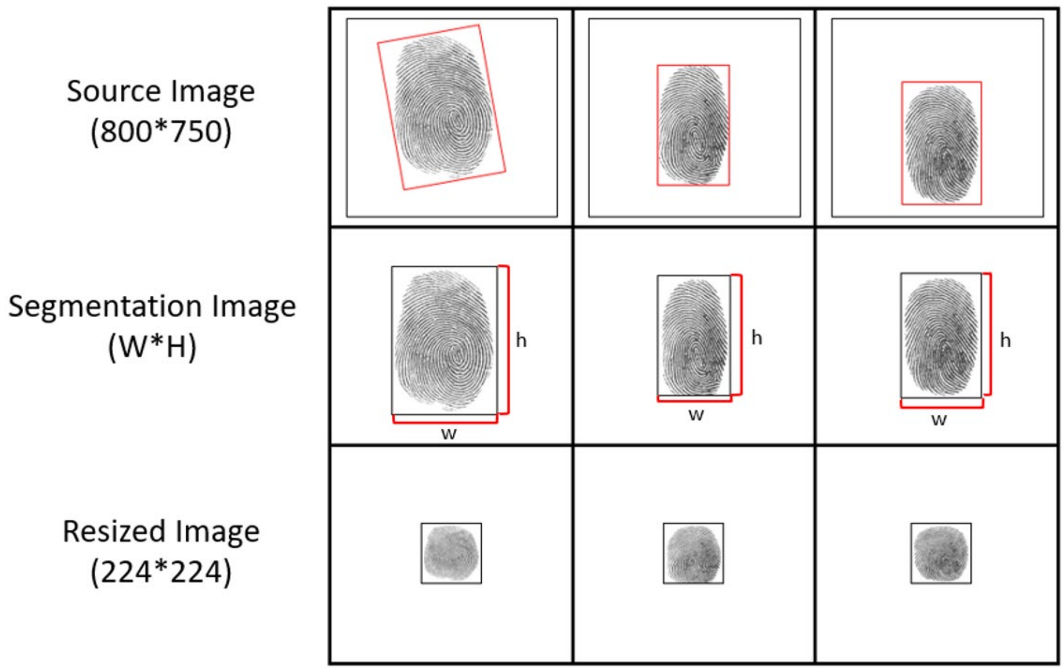

Figure 2: Fingerprint image resizing on pre-processing state

3.3 Deep learning models

Tab. 2 shows a summary of the models used in this paper. The deep learning models used in this paper were chosen for their excellent performance in image classification, they were Classic CNN, AlexNet, Resnet50, VGG-16, and YOLO. Since this paper is not intended to classify more than 3-class, some of the softmax functions were modified and a layer was changed from 1,000 to 2 in the fully connected layer, the rest of deep learning pipelines are the same as in the reference papers.

Table 2: Summary of network models

Model Name	Pipeline
Classic CNN	4 Convolutional Layers + 2 Pooling Layers + 2 Fully Connected Layers
AlexNet	5 Convolutional Layers + 2 Pooling Layers + 3 Fully Connected Layers
Resnet50	49 Convolutional Layers + 2 Pooling Layers + 1 Fully Connected Layers + softmax
VGG-16	13 Convolutional Layers + 5 Pooling Layers + 3 Fully Connected Layers
Yolo	52 Convolutional Layers + 1 Pooling Layers + 23 Residual + 1 Fully Connected Layers + softmax

AlexNet is a deep convolutional neural network which can classify high-resolution images. It consists of 5 hidden convolutional layers, and 3 fully connected layers [Krizhevsky, Hinton and Sutskever (2012)]. Its main contribution is to reduce the overfitting problem by employing a dropout method that strives for high accuracy. The modified architecture of AlexNet is shown in Tab. 3.

Table 3: The modified architecture of AlexNet training on 1 GPU

Layer	Conv. filter	Kernel size	Stride
Convolution	96	11×11	4×4
Max pooling		2×2	2×2
Convolution	256	11×11	1×1
Max pooling		2×2	2×2
Convolution	384	3×3	1×1
Convolution	384	3×3	1×1
Convolution	256	3×3	1×1
Max pooling		2×2	2×2
Fully-connected	4,096		
Dropout	0.4		
Fully-connected	4,096		
Dropout	0.4		
Fully-connected	1,000		
Dropout	0.4		
Fully-connected	2		

ResNet50 has 50 layers for the convolutional layer and the fully connected layer [He, Ren, Sun et al. (2016)]. ResNet is a network designed to solve the problem of degradation, where the depth of the model becomes worse than a shallow model at some point. The ResNet model has 3.8 billion FLOPs and consists of residual blocks and identity blocks. Fig. 3 and Tab. 4 described the architecture of ResNet.

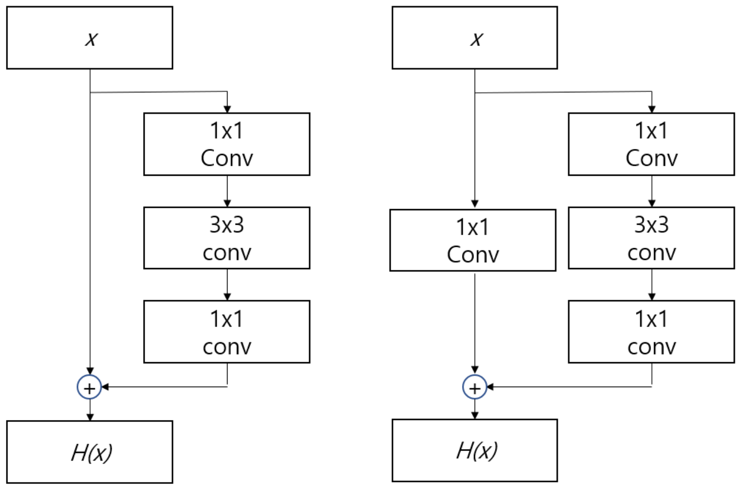

Figure 3: Residual Block (Left) & Identity Block architecture (Right)

Table 4: ResNet50 architecture

Type	Output size	Kernel size	Count
Convolution	112×112	7×7, 64, stride 2	
		3×3 max pool, stride 2	
Conv2_x	56×56	1×1, 64 3×3, 64 1×1, 256	×3
Conv3_x	28×28	1×1, 128 3×3, 128 1×1, 512	×4
Conv4_x	14×14	1×1, 256 3×3, 256 1×1, 1024	×6
Conv5_x	7×7	1×1, 512 3×3, 512 1×1, 2048	×3
Average pool, Fully Connected, softmax	1×1		

VGG-16 [Simonvan and Zisserman (2014)] is a large-scale image recognition and classification convolutional network and its main contribution is to increase the depth to 16 and 19 weight layers with a very small convolution filter (3×3). The VGG-16, which consists of 16 convolutional hidden layers, requires less memory compared to VGG-19. The VGG-16 network consists around 134 million trainable parameters. Since we have only 2 mode classes of classification, we modified the fully connected layer 1,000 to 2. The architecture of VGG-16 is shown in Tab. 5.

Table 5: VGG-16 architecture

Layer	Conv. filter	Kernel size	Stride	Padding
2× Convolution	64	3×3		1×1
Max pooling		2×2	2×2	
2× Convolution	128	3×3		1×1
Max pooling		2×2	2×2	
3× Convolution	256	3×3		1×1
Max pooling		2×2	2×2	
3× Convolution	512	3×3		1×1
Max pooling		2×2	2×2	
3× Convolution	512	3×3		1×1
Max pooling		2×2	2×2	
Fully-connected	4,096			
Dropout	0.5			
Fully-connected	4,096			
Dropout	0.5			
Fully-connected	2			

YOLO (You Only Look Once) v3 [Redmon and Farhadi (2018)] is used to classify fingerprint image into left-hand or right-hand categories. The neural network architecture of YOLO v3 has 53 convolutional layers which is a hybrid architecture combining from YOLO v2, Darknet-19, and the residual network. The YOLO v3 network is described in Tab. 6.

Table 6: YOLO v3 architecture

Layer	Conv. filter	Kernel size	Output
Convolution	32	3×3	256×256
Convolution	64	3×3/2	128×128
Convolution	32	1×1	
Convolution	64	3×3	
Residual			128×128

Convolution	128	3×3/2	64×64
2× Convolution	64	1×1	
2× Convolution	128	3×3	
2× Residual			64×64
Convolution	256	3×3/2	32×32
8× Convolution	128	1×1	
8× Convolution	256	3×3	
8× Residual			32×32
Convolution	512	3×3/2	16×16
8× Convolution	256	1×1	
8× Convolution	512	3×3	
8× Residual			16×16
Convolution	1024	3×3/2	8×8
4× Convolution	512	1×1	
4× Convolution	1024	3×3	
4× Residual			8×8
Avgpool	Global		
Connected	1000		
Softmax			

The dataset, with 224×224 resolution images, is fed into that neural network. We used 9,080 images for the training, 4,540 left-hand images and 4,540 right-hand images. Then we used 1,000 images for validation, 500 in each category. Since all fingerprint patterns are modified to fit a resolution of 224×224, it was necessary to generate the the image labels in the way shown in Tab. 7.

Table 7: Generating image label for training in YOLO v3

Input image	Image bounding box	YOLO Label
224×224	<object-class> <left><top> < width >< height >	<object-class> <x_center> <y_center> <width><height>
224×224	left 0 0 224 224	0 0.5 0.5 1.0 1.0
224×224	right 0 0 224 224	1 0.5 0.5 1.0 1.0

We assigned <object-class>=0 for the left-hand class and <object-class>=1 for the right-hand class. Since we used whole image as a bounding box, the top, left position is designated as the 0th pixel. The width and height of the images are 224. The <x_center>, <y_center>, <width> and <height> are the normalized values of the center of x, y, width and height of the images, respectively. Thus, those normalized values are 0.5, 0.5, 1.0, 1.0, respectively.

4 Discussion

We trained our network using GPU parallelism. The learning environment ran on an Intel (R) Xeon (R) W-2155 CPU @ 3.30Ghz, 8GB RAM, NVIDIA Quadro P4000, and was coded using the Python language. We also used keras to write the main framework, tensorflow to implement the architecture and evaluation was performed using Su et al., Hong et al., Li et al. [Su, Chen, Lai et al. (2017); Hong, Jain and Wan (1998); Li, Chen, Feng et al. (2018)]. Since the input image size required by each network is different, the resizing of the images was done during the pre-processing process, Tab. 8 summarizes the image sizes used by each network.

Table 8: Size of image for each network

Network Name	Image Size	Type
Classic CNN	224×224	Segmentation+resized Image
AlexNet	224×224	Segmentation+resized Image
ResNet50	224×224	Segmentation+resized Image
VGG-16	224×224	Segmentation+resized Image
YOLO	224×224	Segmentation+resized Image

In order to use the same dataset to accurately and find the best performing network, we set constant hyperparameters for training. The learning rate was set to 0.001, the batch size and the number of training iterations were the same, and the differences in validation accuracy of the various networks at the same level are compared in Tab. 9.

Table 9: Network performance comparison

Model	Epoch (Iteration)	Time/Sample	Train acc.	Valid acc.
Classic CNN	300 Epoch	8s 1ms	95%	84.18%
	500 Epoch	8s 1ms	94%	**84.50%**
	1,000 Epoch	8s 1ms	92%	81.85%
AlexNet	300 Epoch	18s 2ms	99.82%	94.50%
	500 Epoch	18s 2ms	99.91%	**95.10%**
	1,000 Epoch	18s 2ms	99.98%	91.20%
VGG-16	300 Epoch	180s 2ms	100%	95.80%
	500 Epoch	180s 2ms	100%	96.00%
	1,000 Epoch	180s 2ms	100%	**96.20%**
ResNet50	300 Epoch	25s 3ms	99.98%	**96.80%**
	500 Epoch	25s 3ms	99.97%	95.56%
	1,000 Epoch	25s 3ms	99.98%	96.45%
YOLO v3	9,000 Iteration	113s	86.63%	81.94%
	10,000 Iteration	113s	88.41%	**86.25%**
	11,000 Iteration	113s	88.15%	85.41%

In this paper, we experimented with three different iterations for all networks, all networks showed a more than 80% accuracy rate. Therefore, the applied network models were able to distinguish whether the input fingerprint was from a left hand or right hand with a significant success rate. As we can see from the results above, when the Epoch of ResNet50 was 300, the prediction rate was 96.80%, which was the highest prediction success rate. The Classic CNN model with the lowest hidden layer recorded 81.85%, the lowest prediction rate.

5 Conclusions

In this paper, we used various CNNs for image classification to determine whether the input fingerprint image was from a left hand or right. We found various features that can be extracted from a fingerprint image in addition to the shape of the fingerprint, we then implemented a deep learning model that can compensate for the shortcomings of the AFIS system.

The proposed method uses a pre-processing stage to remove unnecessary parts of the fingerprint image and to make the images uniform before the deep learning technique was applied. After that, we trained image classification networks such as Classic CNN, ResNet50, AlexNet, VGG-16, and YOLO v3 using a uniform dataset and measured the training accuracy and validation accuracy. The highest validation accuracy of 96.80% was obtained by ResNet50. In general, if fingerprints are not presented to AFIS it is very long and arduous task to narrow down uspects from a large database of fingerprint. TIf the classification model presented in this paper is added to the AFIS system, the search results are faster than with conventional AFIS. It is expected that this system will help by significantly reducing the time needed by fingerprint identification experts to identify the matching fingerprints. In the future, we plan to create a system for identifying finger type, fingerprint shape, fingerprint scratch, etc. based on fingerprint images.

Acknowledgement: This work was supported by the Technology development Program (S2688148) funded by the Ministry of SMEs and Startups (MSS, Korea) and was supported by the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Darlow, L, N.; Rosman, B. (2017): Fingerprint minutiae extraction using deep learning. *2017 IEEE International Joint Conference on Biometrics*, pp. 22-30.

Gnanasivam, P.; Vijayarajan, R. (2019): Gender classification from fingerprint ridge count and fingertip size using optimal score assignment. *Complex & Intelligent Systems*, vol. 5, no. 3, pp.1-10.

He, K.; Ren, S.; Sun, J.; Zhang, X. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Hong, L.; Jain, A.; Wan, Y. (1998): Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777-789.

Kim, J.; Yu, S. C. (2016): Convolutional neural network-based real-time ROV detection using forward-looking sonar image. *2016 IEEE/OES Autonomous Underwater Vehicles*, pp. 396-400.

Krizhevsky, A.; Hinton, G, E.; Sutskever, I. (2012): Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097-1105.

LeCun, Y.; Bengio, Y.; Hinton, G. (2015): Deep Learning. *Nature*, vol. 521, no. 7553, pp. 436-444.

Li, Q.; Cai, W.; Chen, M.; Feng, D.; Wang, X. et al. (2014): Medical image classification with convolutional neural network. *2014 13th International Conference on Control Automation Robotics & Vision*, pp. 844-848.

Li, W.; Chen, Y.; Feng, C.; Xiao, B. (2018): Binary hashing CNN features for action recognition. *KSII Transactions on Internet & Information Systems*, vol. 12, no. 9, pp. 4412-4428.

Maltoni, D.; Jain, A. K.; Maio, D.; Prabhakar, S. (2009): Handbook of fingerprint recognition. Springer Science & Business Media.

Menotti, D.; Chiachia, G.; Falcao, A, X.; Pedrini, H.; Pinto, A. et al. (2015): Deep representations for Iris, Face, and Fingerprint Spoofing Detection. *IEEE*, vol. 10, no. 4, pp. 864-879.

Michelsanti, D.; Ene, A, D.; Guichi, Y.; Moeslund, T, B.; Nasrollahi, K. et al. (2017): Fast fingerprint classification with deep neural networks. *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 202-209.

Nogueira, R, F.; de Alencar Lotufo, R.; Machado, R. C. (2016): Fingerprint liveness detection using convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206-1213.

Redmon, J.; Farhadi, A. (2018): Yolov3: An incremental improvement. *CoRR*, vol. abs/1804.02767.

Rezende, E.; Carvalho, T.; De Geus, P.; Ramos, F.; Ruppert, G. (2017): Malicious software classification using transfer learning of ResNet-50 deep neural network. *16th IEEE International Conference on Machine Learning and Applications*, pp. 1011-1014.

Shen, J.; Tao, X.; Li, Q.; Liu, N.; Sun, H. (2019): Vehicle detection in Aerial images based on hyper feature map in deep convolutional network. *KSII Transactions on Internet & Information Systems*, vol. 13, no. 4, pp. 1989-2011.

Shrein, J. M. (2017): Fingerprint classification using convolutional neural networks and ridge orientation images. *IEEE Symposium Series on Computational Intelligence*, pp. 1-8.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations.*

Su, H, R.; Chen, K, Y.; Lai, S, H.; Wong, W. J. (2017): A deep learning approach towards pore extraction for high-resolution fingerprint recognition. *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 2057-2061.

Tyan, V.; Kim, D. (2018): Convolutional neural network with particle filter approach for visual tracking. *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 693-709.