InComTech: Jurnal Telekomunikasi dan Komputer
vol.11, no.3, Desember 2021, 243-253
http://publikasi.mercubuana.ac.id/index.php/Incomtech
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Fingerprint Authenticity Classification Algorithm based-on Distance of Minutiae using Convolutional Neural Network

Hariyanto[1,2]*, Sarifuddin Madenda[1], Sunny Arief Sudiro[2], Tubagus Maulana Kusuma[1]

[1]*Doctoral Program in Information Technology, Gunadarma University*
Jl. Margonda Raya, Depok, Indonesia
[2]*STMIK Jakarta STI&K*
Jl. BRI Radio Dalam, Gandaria Utara, Jakarta 12140, Indonesia
*Corresponding Author's Email: ha07ri@gmail.com

Abstract:
Fingerprint identification systems are vulnerable to attempted authentication fraud by creating fake fingerprints that mimic the live ones. This paper proposes a method to detect whether a fingerprint is a live fingerprint or a fake fingerprint using Convolutional Neural Network (CNN). We construct a features database of distances among minutiaes of fingerprints, where the distance calculation is based-on Euclidean Distance. Furthermore, the distance features database that has been constructed is classified using the CNN. CNN is chosen because of its capability in recognizing and classifying objects in an image. The numerical results have shown that the best accuracy achieves 99.38% when the learning rate is 0.001 with the epoch of 100.

Keywords:
Fingerprint;
Live;
Fake;
CNN;

Article history:
Received November 13th, 2021
Revised November 29th, 2021
Accepted December 1th, 2021
Published December 31th, 2021

DOI:
10.22441/incomtech.v11i3.13770

1. INTRODUCTION

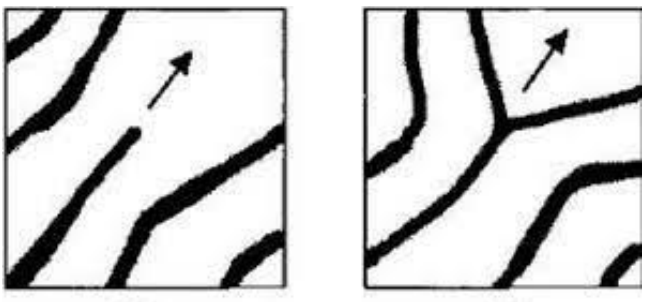
Fingerprint is one of the biometric systems used to identify a person because fingerprint is unique. Everyone has different fingerprint characteristics. With its uniqueness, a fingerprint can be used for authentication in a security system. Fingerprint biometric systems are widely used as well as many attacks they face during the fingerprint authentication process. This makes the fingerprint biometric systems not completely secure. With a fake fingerprint, it will be easy to gain access to the fingerprint scanning system. It is important to develop a way to deal with fake fingerprint attacks. A fingerprint recognition system should have the ability to distinguish real (live) fingerprint images from fake fingerprint images.

Many studies related to the extraction of fingerprint features based on minutiae have been carried out. The use of the Euclidean Distance method is one approach that can be used for fingerprint recognition. The level of accuracy achieved is very good (99%) if it is based on minutiae and classification based on Euclidean distance [1]. Nogueira et al. [2] used CNN to detect fingerprints. The study compared different models to test 50000 datasets sourced from the LivDet 2011 Fingerprint Liveness Detection Competition. CNN is used to train images without changing the original weight values. The results of his research show an accuracy performance of 95.51%. Hindi et al. [3] use the Euclidean distance between each minutiae to replace the coordinates of the minutiae in identifying a fingerprint. AlQadi et al. [4] detect and calculate minutiae using the LBP enhancement method. The result is a vector that can be used as a feature for fingerprint identification. Wang et al. [5] divide the fingerprint image into small plots that do not overlap on the image size of 32 x 32 pixels and then use the four-layer CNN convolution to classify each plot of the image. Jang et al. [6] use the network with four layers of convolution and two layers of fully-connected, inspired by the VGG network architecture, and get a fingerprint spoof detector that has a high accuracy of 99.8% on average.

This research aims to obtain a high level of accuracy performance from the feature of the distance between minutiae dataset trained through the CNN method, so that it can determine the accuracy of the difference between 'live' and 'fake' fingerprints. Based on this model of CNN, the fingerprint recognition system can be developed with higher reliability.

2. METHOD

There are several methods to characterize a fingerprint based on its features [7], which are global features and local features. The first is from the global features. Global feature look at fingerprints from patterns formed such as pattern area, core area, delta, basic ridge pattern (loop, arch, whorl). The second, the local features, is called the minutiae point, where this minutiae has several ridge characteristics including ridge ending, ridge bifurcation, ridge divergence, dot, and enclosure [8]. Minutiae characteristics are most widely used for fingerprint detection, ridge ending and ridge bifurcation features (shown in Figure 1) are the most widely used for research purposes, such as minutiae detection and fingerprint recognition.



(a) (b)

Figure 1. (a) Minutiae Ridge Ending (b) Minutiae Bifurcation [9]

The first step is the formation of fingerprint features that will be used during classification. The feature that will be used is the fingerprint feature based on the distance pattern between the minutes. After the fingerprint feature is formed, the second step is to get the accuracy from the results of fingerprint feature classification.

Human fingerprints contain several unique and varied objects called minutiae, as shown in Figure 2. The number, type, and location of these minutiae differ from one person to another.

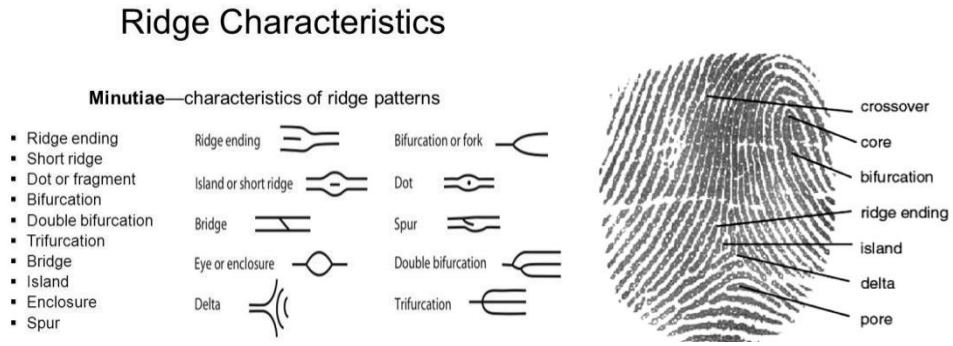


Figure 2. Type of Minutiae [8]

Minutiae are strokes found on the skin of human fingers that form fingerprint patterns. Minutiae have different types, where the shape of the minutiae becomes an important feature in fingerprint feature extraction. This study only took minutiae with ridge ending and ridge bifurcation types as extracted features.

The purpose of this paper is to obtain a high level of accuracy from the classification results in order to determine whether the fingerprint is genuine or fake. The fingerprint feature that will be used as the basis for classification is in the form of distance data between minutiae. The research stages of this fingerprint feature dataset were obtained through a series of processes starting from preprocessing the input image and producing minutiae points through the minutiae extraction process. The preprocessing stage for minutiae extraction is taken from the research stages that have been carried out previously [10]. The stages of the research can be seen in Figure 3.

2.1. Preprocessing and Minutiae Extraction stages

The processing stages in this research are conducted based on Figure 3 and developed using Matlab programming. All stages are described as follows:

1) Data Preparation: The first thing to do is to prepare fingerprint data as an input. The data is sourced from the Joint Multi-modal Biometric Dataset Release Agreement (Liveness Detection (Livdet) Data Set-Fingerprint 2015) Clarkson University- University of Cagliari [11]. For this paper, we used 800 datasets consisting of 400 live fingerprints and 400 fake fingerprints.

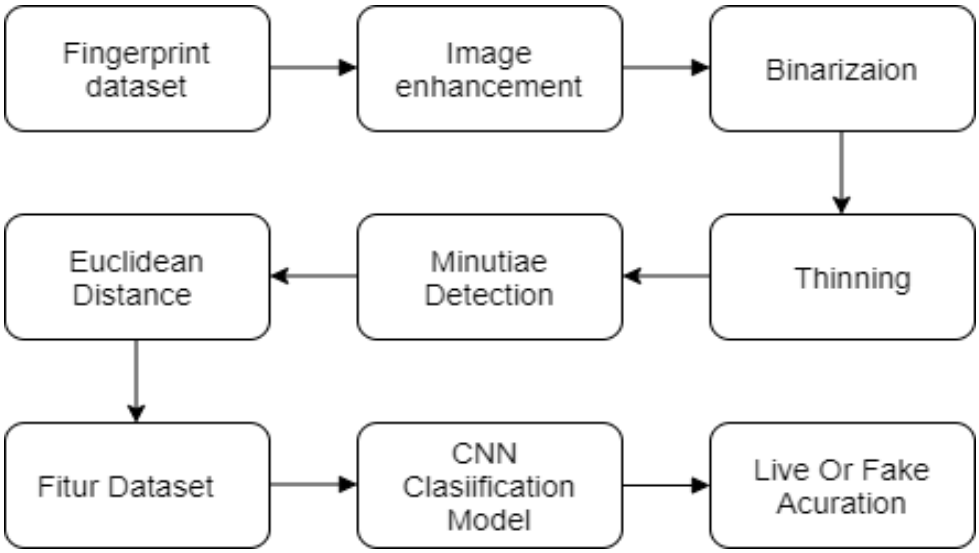


Figure 3. Fingerprint Classification Stages

2) Image Enhancement: The preprocessing stage is carried out to obtain good fingerprint image quality by eliminating unwanted distortion. The technique used to improve image quality in this study is the Gabor filter as a band pass filter [12] to remove noise and show the actual structure of ridges and valleys.
3) Binarization is the process of converting a fingerprint image from a gray scale to a black and white color image according to a specified threshold value [1]. Threshold value > pixel value = Black-0, threshold value < pixel value = White-1. In general, the process of binary gray scale imagery to produce a binary image is as follows.

$$G(x,y) = \begin{cases} 1 \; if \; f(x,y) \geq T \\ 0 \; if \; (x,y) < T \end{cases} \quad (1)$$

Where g(x,y) is the binary image of the gray image f(x,y) and T represents the threshold value.

4) Thinning: This thinning process aims to obtain a one-pixel wide representation of the fingerprint image and does not change the original fingerprint structure. Zhang and Suen's algorithm described in [13, 14] was changed or modified by considering the thinning process to be carried out on the image with a pixel value of '0' (valley structure). The thinning process is based on evaluating the pixel values in 8-neighbor pixels with reference to the pixel P1 and the transition from pixel value '0' to '1' for the surrounding pixels, as shown in Figure 4. This modification has been presented in [14].

0 →	1	0
1	P1(i,j)	0
1	1 ←	0

Figure 4. Pixel Transition

5) Minutiae Extraction: To get the presence of minutiae point on the fingerprint and the type of minutiae obtained, feature extraction is carried out. Figure 4 shows the pixel transition in 8 neighborhood pixel values. The transition is 0 to 1 or 1 to 0 and then calculated based on the Crossing Number equation (eq. 2) to determine the type of minutiae in the fingerprint image. Feature extraction in this paper uses the algorithm proposed by [15], where they use the Crossing Number (CN) method at point P in determining the minutiae obtained. In general, the formula for the Crossing Number is:

$$CN = 0.5 \sum_{i=1}^{8} P_i - P_{i+1} with\ P_9 = P_1 \quad (2)$$

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

This process detects the presence of minutiae small dots and defines the type of small dots based on the Crossing Number (CN) method. For example, Pi is the pixel of the binary element belonging to the 3x3 window P. If CN = 1, the end point is obtained, and if CN = 3, the bifurcation point is obtained. Other CN values are inapplicable. This algorithm is quite fast as it only works for specific foreground blocks.

6) Ecludience Distance Calculation: The resulting minutiae point, in addition to providing minutiae type (ridge end, bifurcation), also provides minutiae point location, which is marked as coordinates (x,y). The first minutiae point (T1) on the fingerprint is connected to the second minutiae point (T2), and then the distance is calculated. Calculation of the distance between two points uses the Euclidean Distance method as shown in equation (3).

$$J = \sqrt{(\Delta x)^2 + (\Delta y)^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

In this paper, not all the distance between the coordinate points is calculated. From a fingerprint image, only 36 coordinate points will be taken. These 36 coordinate points are sufficient to represent the characteristics of a fingerprint [16]. The calculation of the distance from 36 minutiae points connected to each other will produce 630 distance segments for each fingerprint. Figure 5 shows an illustration of the distance network between minutiae points, namely from T1 to T36.

First minutiae point : T1 = (x1,y1)
Second minutiae point : T2 = (x2,y2)

The first distance (J1) is formed from the distance between T1 and T2, then calculated by the Euclidean Distance formula. The distance calculation is

carried out until all 36 minutiae points are connected to each other. The last distance segment that is calculated is between points T35 and T36 as well as the 630th distance. See Table 1, example of a distance point. A minutiae distance feature dataset is formed, which will be used for the classification of real or fake fingerprints using the CNN method.

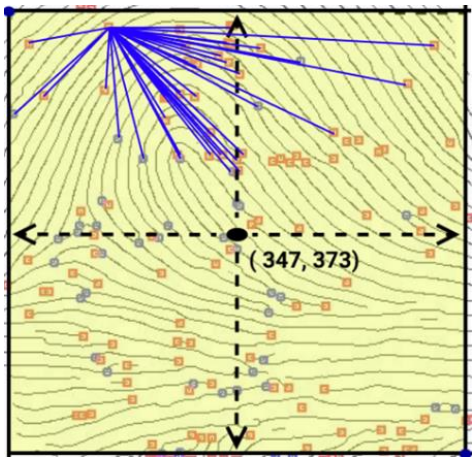


Figure 5. Illustration of the Distance Network from One point to Another Minutiae Point

Tabel 1. Distance Between Point Minutiae

Distance Code	Minutiae Point
J1	T1 – T2
J2	T1 – T3
J3	T1 – T4
…..	…
…..	…
…..	…
…..	…
…..	…
J628	T34 – T35
J629	T34 – T36
J630	T35 – T36

2.2. CNN Classification Stages

The CNN architecture used to classify fingerprints as real or fake consists of 2 convolution layers and 2 pooling layers. The first convolution uses a 1x2 kernel with 8 filters with a stride value of 1, after which the ReLU (Retrified Linear Unit) activation function is added. The next layer is maxpooling. The maxpooling filter used is 1x2 with a stride value of 1. The second convolution is carried out on a 1x2 kernel with 16 filters and a stride value of 1. The second pooling process is the same as the first pooling process, namely using maxpooling with a 1x2 filter and shifting the stride 1. The last is the fully connected process. At the fully connected layer, first the flattening process is carried out, namely the process of converting the feature map into a one-dimensional vector [17]. This vector will be used as input to the fully connected layer with softmax as the activation function. In this fully connected layer, the data is trained to recognize the characteristics of each

fingerprint. Furthermore, the softmax activation function will determine which feature classification is most correlated with real or fake fingerprints. Figure 6 shows the CNN architecture proposed in this study.

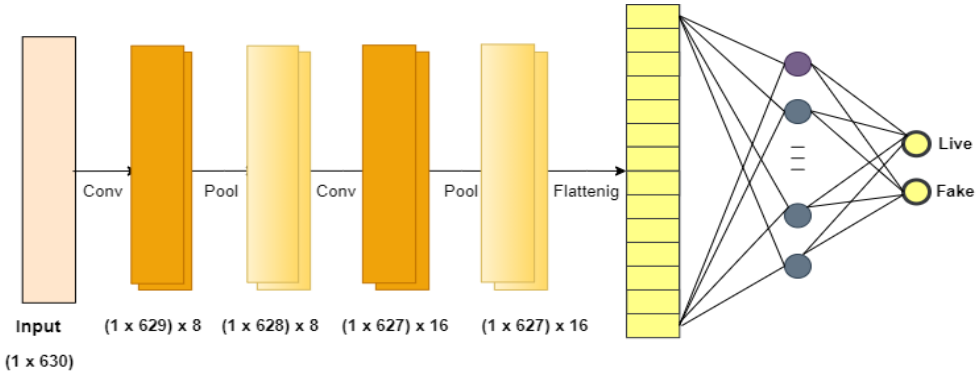


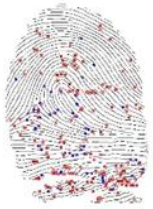
Figure 6. CNN architecture used in this research

3. RESULT AND DISCUSSION

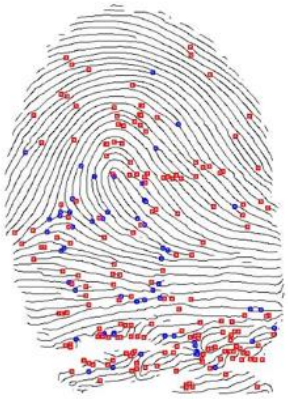
The preprocessing and minutiae extraction steps have been carried out. The result is a fingerprint feature data-set based on the distance between the minutiae. Table 2 presents the results of preprocessing and minutiae extraction.

From Table 2, the first column contains input images and the second column contains images of the image enhancement, binarization, and thinning processes. The third column is the result image of the feature extraction process. The fingerprint feature extraction process shows clearly the minutiae points, both ridge ending and bifurcation points. The location of each minutiae point is marked with its coordinates as the horizontal x axis and the vertical y axis, written as (x, y), so that from the extracted image a number of coordinate points are obtained. Table 3 shows one of the minutiae points vector data from one of the datasets. These features are obtained based on the fingerprint feature extraction algorithm using the Crossing Number method.

Table 2. Results of Preprocessing and Extraction Images

Input Image	**Preprocessing Image**	Enhancement	Binarization	Thinning	**Minutiae Extraction Image**
					

Tabel 3. Minutiae Point Coordinate

Fingerprints Image	Minutiae Point Coordinate					
		x	y		x	y
	T1	330	185	T19	349	251
	T2	229	198	T20	476	258
	T3	237	201	T21	247	263
	T4	342	214	T22	304	266
	T5	344	214	T23	201	267
	T6	250	215	T24	315	268
	T7	304	217	T25	292	269
	T8	340	227	T26	294	269
	T9	190	228	T27	363	275
	T10	342	228	T28	178	281
	T11	308	229	T29	293	286
	T12	496	229	T30	327	295
	T13	365	231	T31	420	295
	T14	333	237	T32	257	300
	T15	310	241	T33	439	301
	T16	395	241	T34	430	302
	T17	341	242	T35	508	303
	T18	317	243	T36	458	308

Coordinate points obtained are calculated by the distance between coordinate points. Some of the results are presented in Table 4.

Table 4. Example of Result Calculation the Distance Between Minutiae Points in a Fingerprint (Pixel Unit)

Minutiae Point	Distance Code	Distance
T1 – T2	J1	101.83
T1 – T3	J2	94.366
T1 – T4	J3	31.385
T1 – T5	J4	32.202
T1 – T6	J5	85.44
…….	…	……
T35 – T36	J630	50.249

3.1. Minutiae Distance Feature Classification with CNN

This dataset, based on the minutiae distance feature, will then be classified using the CNN method. All 800 datasets consist of 400 live fingerprints and 400 fake fingerprints. Then the data is divided into two groups: 80% of the data is used for training and 20% is used for validation.

In this paper, 9 scenarios of the CNN model are carried out based on differences in the learning rate and epoch parameters. The determination of the learning rate and epoch values in this paper is based on trials, because there is no appropriate learning rate and epoch value for each model we made. The purpose of determining the two parameters of the learning rate difference and the epoch parameter is to compare the model that has the best accuracy. The results of the classification can be seen in Table 5.

Based on Table 5, it can be seen that the results of the training carried out with a learning rate of 0.01 and epoch 150 achieved the highest accuracy of 98.75% in 12 minutes and 11 seconds. Next, training with a learning rate of 0.001 and epoch 100 obtained the highest accuracy of 99.38% in 10 minutes 40 seconds. The results of this study show that the greater the number of epochs, the higher the level of training accuracy is obtained and vice versa with the learning rate parameter. The learning rate with a large value (0.01) produces an accuracy level that is not as good as the learning rate that has a smaller value (0.001). These two parameters, epoch and learning rate, greatly affect the accuracy of the CNN network architecture.

Table 5. CNN Model Training Results based on Minutiae Distance Feature

Learning Rate	Epoch	Accuracy (%)	Time
0.01	30	93,75	2 m 52 dt
0.01	50	95,63	5 m 8 dt
0.01	80	96,88	8 m 40 dt
0.01	100	97.50	10 m 35 dt
0.01	150	98.75	12 m 11 dt
0.001	30	98,13	3 m 11 dt
0.001	50	98,75	5 m 16 dt
0.001	80	99,38	12 m 9 dt
0.001	100	99,38	10 m 40 dt

The graphs of training results that illustrate the highest accuracy values for each learning rate and epoch are presented in Figure 7 and Figure 8. Figure 7 shows a graph of the training progress with LR 0.01 and Epoch 150. It can be seen that the accuracy graph shows an accuracy value that continues to increase slowly until the last epoch with an accuracy value of 98.75. While the loss chart shows the beginning of the epoch rising, then drastically decreasing until the end of the epoch.

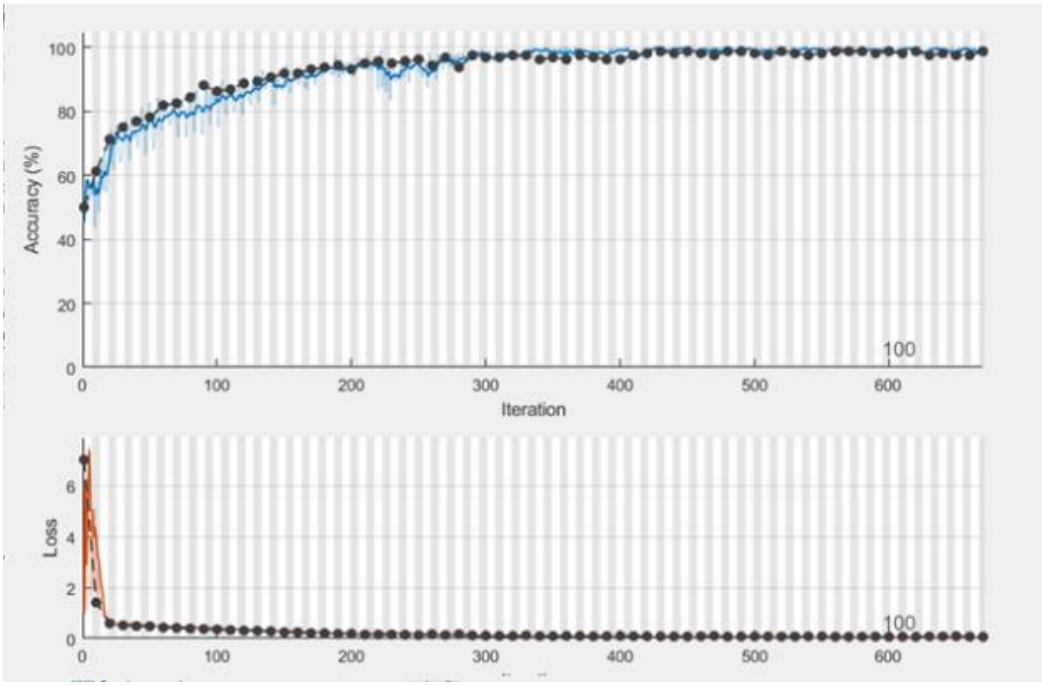


Figure 7. Graph of Training Progress with LR 0.01 and Epoch 150

Figure 8 shows a graph of the progress of the Learning Rate training 0.001 and Epoch 100. At the beginning of the process, it appears that the loss value decreases, then increases at epoch 10, and then drops back down at epoch 15. Until the end of the epoch, the loss value tends to decrease slowly, which means the loss value is getting smaller. The best accuracy performance result for this paper is 99.38%. Compared to existing research, it shows that the results of the proposed research are slightly better. Table 6 shows the comparison with other researchers.

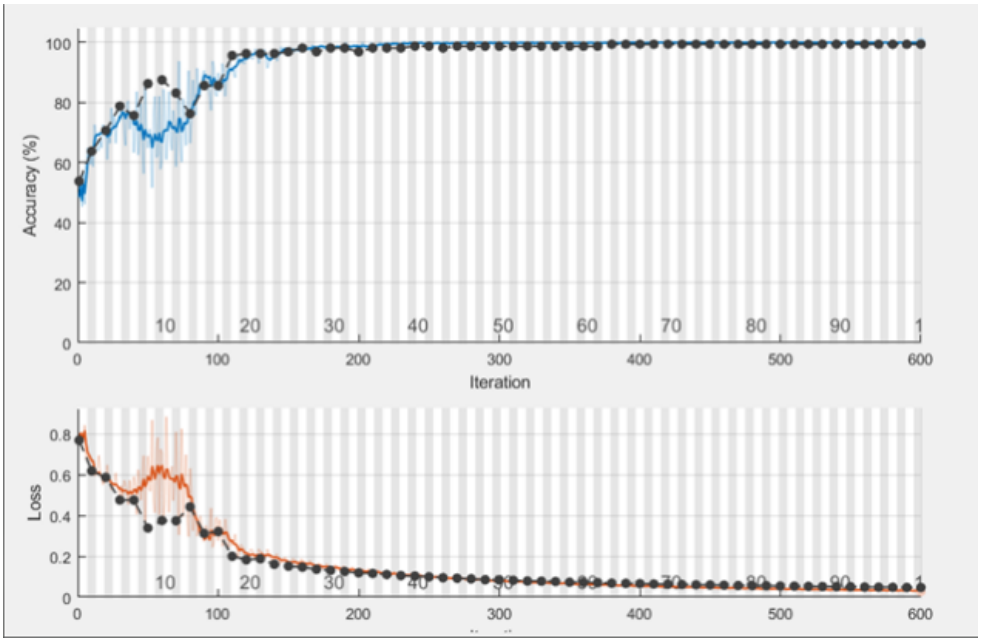


Figure 8. Graph of training progress for LR 0.001 and Epoch 100

Table 6. Comparation Result feature

	Research by Martin et al. [4]	Research by Jang et al. [9	This Article
Accuracy (%)	99	93,75	99,38
Layers	NA	6	5

4. CONCLUSION

This research generated fingerprint features in the form of distance between minutiae points. The distance segment is obtained from the calculation of the distance between two minutiae points using the Euclidean Distance method. This feature is then used as a data-set for classification in determining the authenticity of fingerprints through the CNN method. From a number of classification trials performed in the experiments, two results with the best accuracy were obtained, the first for a learning rate of 0.01 and an epoch of 150 with an accuracy of 98.75% and the second for a learning rate of 0.001 and an epoch of 100 with an accuracy of 99.38%. This shows that the lower the learning rate value, the better the obtained accuracy, while on the contrary, the higher the epoch value, the higher the achieved accuracy. The results showed that the CNN training model that was formed had a fairly high level of accuracy.

REFERENCES

[1] K. M. Sagayam, D. N. Ponraj, J. Winston, Yaspy J C, E. Jeba D, A. Clara, "Authentication of Biometric System using Fingerprint Recognition with Euclidean Distance and Neural Network Classifier", *Int. J. Innov. Technol. Explor. Eng*, vol. 8, no.4, pp.766-771, 2019.

[2] R. F. Nogueira, R. de Alencar Lotufo and R. Campos Machado, "Fingerprint Liveness Detection Using Convolutional Neural Networks," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206-1213, June 2016, doi: 10.1109/TIFS.2016.2520880.

[3] A. Hindi, M. O. Dwairi, Z. Alqadi, "Analysis of Fingerprint Minutiae to form Finger Identifier," *International Journal of Computer Science and Mobile Computing*, vol.9, no. 2, pp. 38-48, 2020.

[4] Z.A. AlQadi, Y. Eltous, M. Abuzalata, G.M. Qaryouti, "Detecting and Counting Minutiae in Human Fingerprint," *Open Science Journal*, vol. 5, no. 1, 2020.

[5] C. Wang, K. Li, Z. Wu and Q. Zhao, "A DCNN Based Fingerprint Liveness Detection Algorithm with Voting Strategy". *In: Yang J., Yang J., Sun Z., Shan S., Zheng W., Feng J. (eds) Biometric Recognition. CCBR 2015. Lecture Notes in Computer Science*, vol 9428. Springer, Cham, 2015. doi: 10.1007/978-3-319-25417-3_29.

[6] H.U. Jang, H.Y. Choi, D. Kim, J. Son, H.K. Lee, "Fingerprint Spoof Detection Using Contrast Enhancement and Convolutional Neural Networks," *In Information Science and Applications; Kim, K., Joukov*, N., Eds. Springer: Singapore, pp. 331-338, 2017.

[7] D. Maltoni, Handbook of Fingerprint Recognitioni, 2nd Edition, Springer: London, 2009.

[8] Digital Persona inc., "Guide to Fingerprint Recognition", *White Paper* [Online] http://dl.fecpos.com/CustomerService/Peripheral/Finger_Print/Manual/Guide to Fingerprint Recognition.pdf [Accessed 07 Mei 2021].

[9] S. Bhattacharya, K Mali., "Fingerprint Recognition Using Minutiae Extraction Method", *Proc. of International Conferance on Emerging Technologies (ICET-2011)*, 2011, pp. 0975-4830.

[10] S.A, Sudiro, "Fingerprint Recognition using FPGA Devices", *Disertation*, Doctoral Program Gunadarma University, 2009.

[11] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay and S. A. Schuckers, "LivDet 2015 fingerprint liveness detection competition 2015," *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015, pp. 1-6, doi: 10.1109/BTAS.2015.7358776.

[12] Lin Hong, Yifei Wan and A.K Jain., "Fingerprint image enhancement: algorithm and performance evaluation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777-789, doi: 10.1109/34.709565 , 1998.

[13] T. Zhang and C. Sue, "A Fast Parallel Algorithm For Thinning Digital Patterns," *Communications of the ACM*, vol. 27, no. 3, pp 236–239, 1984, doi: 10.1145/357994.358023.

[14] S.A. Sudiro,"Thinning Algorithm for Image Converted in Fingerprint Recognition System," *National Seminar Soft-Computing Inteligent Systems and Information Technology 2005*, Universitas Kristen Petra, Surabaya, 2005.

[15] S. Kasaei, M. Deriche and B. Boashash, "Fingerprint feature extraction using block-direction on reconstructed images," TENCON '97 Brisbane - Australia. *Proceedings of IEEE TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications* (Cat. No.97CH36162), 1997, pp. 303-306 vol.1, doi: 10.1109/TENCON.1997.647317.

[16] S. Z. Li and A.K Jain, *Encyclopedia of Biometrics*, Springer Science & Business Media, 2009, doi: 10.1007/978-0-387-73003-5.

[17] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.