Multi-Level Pooling Model for Fingerprint-Based Gender Classification

Sri Suwarno[1], **Erick Kurniawan**[2]
[1]Universitas Kristen Duta Wacana, Yogyakarta, Indonesia
[2]Universiti Teknikal, Melaka, Malaysia

Article Info

Article history:

Received December 27, 2022
Revised February 07, 2023
Accepted February 27, 2023

Keywords:

Features Extraction
Fingerprint
Gender Classification
K-Nearest Neighbor
Pooling Model

ABSTRACT

It has been widely reported that CNN (Convolutional Neural Network) has shown satisfactory results in classifying images. The strength of CNN lies in the type and the number of layers that construct it. However, the most apparent drawbacks of CNN are the requirement for a large labeled dataset and its lengthy training time. Although datasets are available, labeling that data is a significant problem. This work mimics the CNN model but only utilizes its pooling layers. The novelty of this model is removing convolution layers and directly processing fingerprint images using pooling layers. Three pooling layer models, namely maximum pooling, average pooling, and minimum pooling, are used to generate fingerprint features to classify their owner gender. These pooling layers are arranged consecutively up to eight levels. Removing convolution layers makes the process straightforward, and the computation is much faster. This study utilized 200 fingerprint datasets from the NIST (National Institute of Standards and Technology), with male and female fingerprints of 100 samples each. The extracted features were then classified using K-NN (K-Nearest Neighbors) algorithm. The proposed method resulted in an accuracy of 61% to 71.5% or an average of 66.25%.

Corresponding Author:

Sri Suwarno, +628122714719,
Information Technology Faculty,
Universitas Kristen Duta Wacana, Yogyakarta, Indonesia,
Email: srisuwarno@gmail.com

How to Cite:S. Suwarno, ”Multi Level Pooling Model for Fingerprint-Based Gender Classification”, *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 22, no. 2, pp. 195-206, Mar. 2023.

1. INTRODUCTION

One of the problems encountered in fingerprint classification is the selection of features. Fingerprints are composed of dark-colored grooves, called ridges, and these grooves are parallel to light-colored grooves, called valleys. Ridges and valleys form curves that seem to surround the surface of fingerprints. In some fingerprint regions, ridges and valleys form a unique pattern called minutiae. The minutiae feature is relatively easy to be recognized manually, but it is more complicated when processed digitally. Some preprocesses, such as noise removal, background separation, and thinning, are required before the formation of minutiae-based features. The features generated based on minutiae are called spatial-based features [1]. Some studies utilize minutiae to generate features, as conducted by Terhrst et al. [2] and Gnanasivam and Vijayarajan [1].

Many attempts have been made to generate a fingerprint feature based on frequency rather than spatial. Frequency-based methods rely solely on pixel value computation without considering their location. Methods that are widely carried out in the formation of frequency-based features are wavelet transformation and CNN (Convolutional Neural Network). Both methods aim to reduce the image size without losing too much information, but they use different approaches. In wavelet transformation, pixel values are mathematically transformed into components, while CNN utilizes convolve processes using some convolution layers to make feature maps.

Last five years, many papers reported the success of CNN in processing some tasks, especially in classifying images. For example, to classify images related to medical images [3], such as lung cancer [4], colon cancer, Covid-19 [5, 6], and Alzheimers disease [7]. Other uses are classifying fruits [8], plant diseases [9, 10], network attacks [11], gender [12], emotional expressions [13], and for automatic door access [14].

After CNN showed satisfactory performance, many researchers turned to using CNN to classify, identify, verify, and authenticate fingerprints. For example, Miranda et al. [15] classify fingerprints by exploiting Residual Network-50 (ResNet-50) and combining it with Contrast Limited Adaptive Histogram Equalization (CLAHE). They reported that using CLAHE, their proposed model, can perform an accuracy of up to 95,05%.

Aritonang et al. [6] used CNN for fingerprint identification. They grouped fingerprints resulting from RFID attendance machines using CNN to minimize fake data. Based on his model, they reported that it could achieve an accuracy of 95.64%. Meanwhile, Boucherit et al. [16] used Merge CNN to increase the validity of fingerprint recognition. They combine multiple identical CNNs to process a different set of fingerprint images. Using six different quality image sets, the model can achieve an accuracy of up to 99.48%.

Hariyanto et al. [17] used CNN to detect fingerprint authenticity, whether it is a live or fake fingerprint. They calculated the distance of the fingerprint ridge using the Euclidean distance formula and saved it as a reference. They reported that the method could reach up to 99.34% accuracy.

Likewise, Lin and Kumar [18] used CNN to compare the matching accuracy between fingerprints taken on a contactless basis with those contact-based. Firstly, they trained multi-Siamese CNN using a ridge map and a specific ridge map region. Then, fingerprint representation is generated based on the distance-aware loss function. They reported that the model performed a better result compared to other models.

All of these papers reported that CNN shows promising performance in processing fingerprints. However, most proposed methods involved convolution layers that need much computation. This study aims to bypass convolution layers and directly implement polling layers to fingerprint images.

2. RESEARCH METHOD

2.1. Typical CNN Architecture

Figure 1 illustrates a typical architecture of CNN. Input data is convolved using a convolution layer consisting of some filters or kernels to extract the features of a small part of the input data. The output of the convolution layer is then filtered by ReLu (Rectified Linear Unit) activation function. The ReLu function passes all positive values but converts to zeros all negative values. This function is practically not required for image data because all pixel values are positive. The bigger the input data dimension and the more kernels are implemented, the longer the convolve process.

After being filtered by the ReLu function, the output is forwarded to the pooling layer. These three consecutive steps can be repeated many times depending on the purpose of the design. At the end of the layers, the fully connected layer is applied. This layer(s) aims to classify or determine the input data based on the trained data. This layer can be composed of more than one.

This study mimics one of the steps in CNN architecture, namely pooling, as shown in a blue colored box in Figure 1. This research uses a quantitative approach. The fingerprint features are generated based on polling layers outputs. To increase the uniqueness of features, three polling layers, namely max-pooling, average-pooling, and min-pooling, are arranged sequentially. The method is applied to the NIST (National Institute of Standards and Technology) dataset. The computation would be simpler and quicker than CNN because the pooling process only finds the biggest, the smallest, and the average values of the matrix elements.

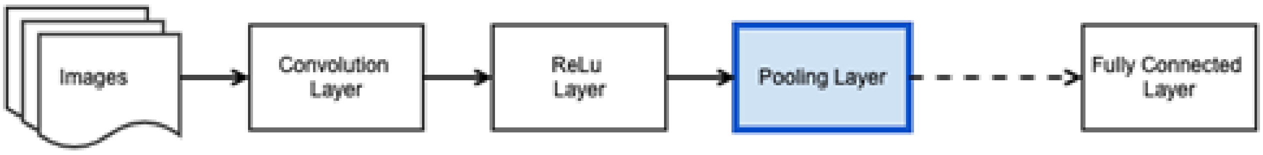

Figure 1. Typical CNN Architecture

2.2. Pooling design

In CNN architecture, the primary purpose of the pooling layer is to reduce the matrix size produced by the convolutional layer in front of it. The pooling layer works the same as convolutional layers, which run a filter on matrix elements in the horizontal and vertical directions with selectable steps. However, unlike CNN, the pooling layer implements a function to the matrices, while the convolution layer implements a convolutional process.

In general, pooling layers use a 22 filter and apply a function to the filter-sized matrix. Figure 2 shows a typical pooling process. The left side of the figure shows the original matrix, and the right side part shows the result matrix. Element P1 is resulted from matrix [x1, x2, x5, x6], P2 is resulted from matrix [x3, x4, x7, x8], P3 is resulted from matrix [x9, x10, x13, x14], and P4 is resulted from matrix [x11, x12, x15, x16].

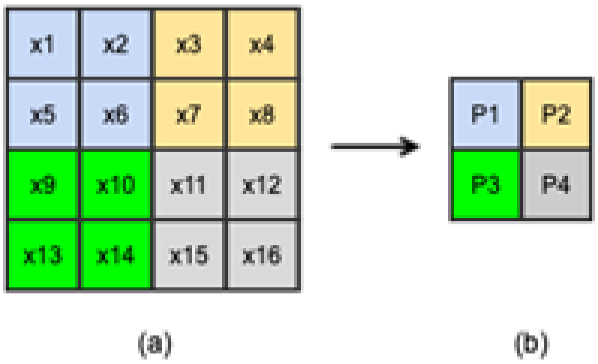

Figure 2. Typical Pooling Process, (a) Original Matrix, (b) Result Matrix

There are three function options in pooling layers: max-pooling, average-pooling, and min-pooling. Mathematically, max-pooling looks for the largest element value on the matrix, average pooling calculates the average of the matrix elements, and min-pooling looks for the smallest element value on the matrix.

This study uses fingerprint images in grayscale format. It means that the method works only on one channel. For grayscale images, max-pooling results in an image that accentuates bright-intensity pixels, while min-pooling is the opposite. In average pooling, sharp corners or edges will be smoothed. In this study, all three functions were applied sequentially, as presented in Figure 3.

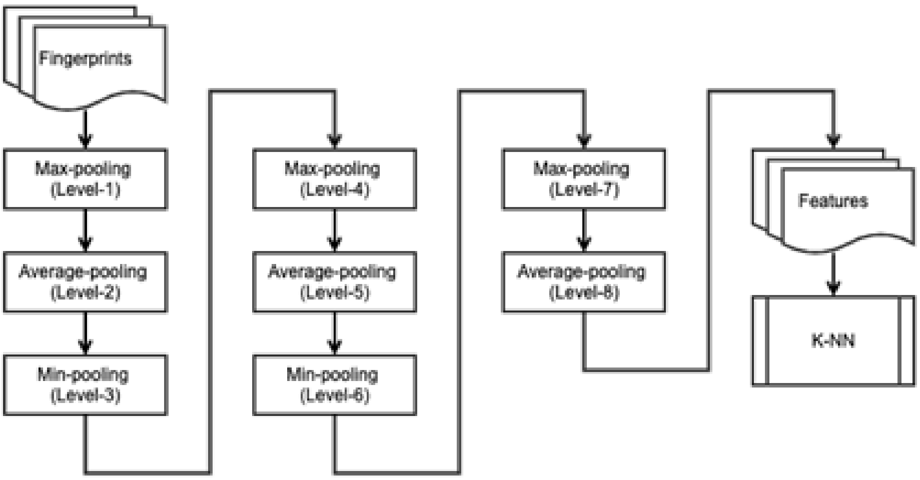

Figure 3. The Proposed Pooling Model

Figure 3 shows that each fingerprint is processed with pooling filters up to eight levels, consisting of max-pooling, average-pooling, and min-pooling. These consecutive processes are repeated to produce the proper size of the feature. The feature size should be small enough but still, preserve essential information. The larger the feature size would increase the complexities of classification, while the smaller feature size would overfit the classification. The primary purpose of this repeated pooling process was to find the appropriate matrix size suitable for the K-NN process.

This study used a filter size of 22. If the size of the fingerprint image is nn, then the image size of level t can be calculated by Equation 1. The image size specifies the matrix size to be processed in the subsequent step. The pooling process decreases the matrix size gradually.

$$A_t = \frac{n}{2^t} \times \frac{n}{2^t} \tag{1}$$

Where A is the image size and t is the pooling level.

2.3. Experiment Design

Some algorithms can be used to classify objects; however, this study used K-NN (K-Nearest Neighbors). This algorithm is chosen because it is simple and does not require training. Moreover, the K-NN algorithm does not depend on big sample data.

Theoretically, the K-NN algorithm computes the distance between two values, in the context of this study, two feature vectors. A new value is classified based on its distance (k) relative to predefined values. The parameter k was chosen at 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. This work used the Euclidean distance formula, as shown in Equation (1), to compute the distance between two vectors.

$$dist(p,q) = \sqrt{\Sigma_{i=1}^{n}(q_i - p_i)^2} \tag{2}$$

Where p and q are points in Euclidean space, and n is the number of points.

Because in this work, the points to be calculated are feature vector elements, the size of the vector would affect the number of distance calculations. The original images' size is 512512, and when converted into a vector, the size is too big to be classified by K-NN. So the size is reduced by pooling them eight times. The output size of each pooling level is shown in Table 1. It shows that the output of pooling level 1, level 2, and level 3 is 65,536, 16,384, and 4,096, respectively. These output sizes are still too large. Meanwhile, the vector size of the output of pooling levels 4, 5, 6, 7, and 8 is moderate enough to be input to the K-NN algorithm. However, theoretically, when the dimension of features is too small, much important information is lost.

Table 1 reveals a gradual decrease in the matrix and vector sizes resulting from each pooling level. It can be seen from the table that the output size decreases a quarter from the previous level in every pooling step.

Table 1. The Matrix Size at each level

Level	Pooling type	Matrix size	Vector size
1	Max-pooling	256 × 256	65,536
2	Average-poolong	128 × 128	16,384
3	Min-pooling	64 × 64	4,096
4	Max-pooling	32 × 32	1,024
5	Average-pooling	16 × 16	256
6	Min-pooling	8 × 8	64
7	Max-pooling	4 × 4	16
8	Average-pooling	2 × 2	4

The main goal of this work is to find a more reasonable method to generate reliable features rather than to find the finest classifier. The K-NN algorithm is employed only to compare the performance of the generated features.

In this study, the dataset used was from the NIST (National Institute of Standart and Technology), as many as 200, consisting of 100 male and 100 female fingerprints. In this dataset, the image size is 512512 in grayscale format. The grayscale images are composed of pixel values ranging from 0 to 255. Values close to 0 have darker intensity compared to values close to 255.

Before being classified by the K-NN algorithm, the fingerprint dataset was processed by pooling layers. By considering the size of feature vectors, only the output of five layers was used: max-pooling at level 4, average-pooling at level 5, min-pooling at level 6, max-pooling at level 7, and average-pooling at level 8. The result of these selected layers can determine the most reasonable number of levels.

Based on Equation 1 and summarized in Table 1, at the 4th level of pooling, the resulting matrix size is 3232, the 5th level is 1616, the 6th level is 88, the 7th level is 44, and the 8th level is 22. The matrices were then reshaped into a vector of size 1024, 256, 64, 16, and 4, respectively. The reshaped vector then functioned as a feature to be processed with the K-NN algorithm. The proposed method was processed using Image Processing and Deep Learning Toolboxes of MATLAB Version R2022a.

3. RESULT AND ANALYSIS

The research started by selecting fingerprint data samples from the NIST (National Institute of Standards and Technology) dataset. These data samples have been labeled Male and Female and converted to 1 and 0, respectively. After that, pooling layers directly process the dataset without any preprocessing. The output of pooling layers is only one dimension, whereas CNN results in more than one dimension output, depending on the number of kernels used.

3.1. The Pooling layers output

This research used 200 labeled fingerprints, composed of 100 male and 100 female fingerprints in grayscale format. The largest pixel value of this format is 255, and the smallest is 0. The max-pooling layer of this level processes original fingerprint images in grayscale format, so the output values are around 255. It means that the max-pooling layer at this level detects light-colored pixels. The output of this layer is a vector of size 65,536. It is impossible to present the vector visually.

The outputs of the max-pooling layer of this level are then forwarded to the average-pooling layer, which calculates the average values of these vector elements. Starting from this level, elements of vectors are floating point numbers. Next, the vectors resulting from the average pooling are inputted to min-pooling, which determines the smallest value of the vector. These consecutive processes are repeated three times and produce vectors that are getting smaller in size. The size of the vectors of each level is presented in Table 1.

3.2. The KNN process

Prior to classifying the features by K-NN, they are required to be labeled. These features utilize the label from the image from which they were generated. This research is binary classification, so the label is 1 for males and 0 for females. After that, male features and female features are combined to make data input to the K-NN algorithm.

Due to the feature dimensionality, this research only chooses features of sizes 1024, 256, 64, 16, and 4. The K-NN algorithm does not need training. It requires only calculating the distance between the features of new and existing data. The result of the experiments is presented in Table 2, Table 3, Table 4, Table 5, and Table 6.

3.3. The performance

The accuracy of the proposed method was calculated based on Equation 3, and the confusion matrix is shown in Figure 4. In binary classification, the confusion matrix consists of four values, namely TP (True Positive), FP (False Positive), FN (False Negative), and TN (True Negative). These values are calculated based on the percentage of prediction to the real fingerprint class.

In this confusion matrix, TP is the number of male fingerprints perfectly predicted as male, and TN is the number of female fingerprints indicated as female. Meanwhile, FP is the number of male fingerprints wrongly predicted as female, and FN is the number of female fingerprints indicated as male.

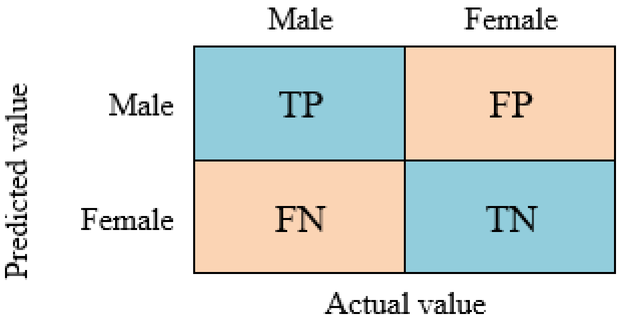

Figure 4. Confusion Matrix

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

All the computation results are presented as tables. Table 2, Table 3, Table 4, Table 5, and Table 6 shows the confusion matrix elements with features dimension of 32 × 32, 16 × 16, 8 × 8, 4 × 4, and 2 × 2, respectively. In these tables, the k value is varied from 5 to 50 with a step of 5 to compare the result of K-NN classification using the same feature size. These tables are presented as charts in Figure 5, Figure 6, Figure 7, Figure 8, and Figure 9, which emphasize the result comparison.

Figure 5 presents the confusion matrix values when the feature size is 1024, converted from a 1616 matrix. What can be clearly seen in Figure 6 is the variability of the prediction for male or female fingerprints. For a k value of more than 20, the prediction for female fingerprints is higher than for male fingerprints. On the contrary, for a k-size less than 20, the prediction for male and female fingerprints is unstable. Overall, the correct prediction for male fingerprints (TP) tends to be lower than for female fingerprints (TN).

Table 2. Confusion Matrix Elements of K-NN with input dimension of 32 × 32

k	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)
5	64	36	32	68	65
10	76	24	38	62	69
15	60	40	32	68	64
20	63	37	37	63	63
25	60	40	25	75	67.5
30	60	40	29	71	65.5
35	62	38	21	79	70.5
40	64	36	27	73	68.5
45	61	39	24	76	68.5
50	65	35	22	78	71.5

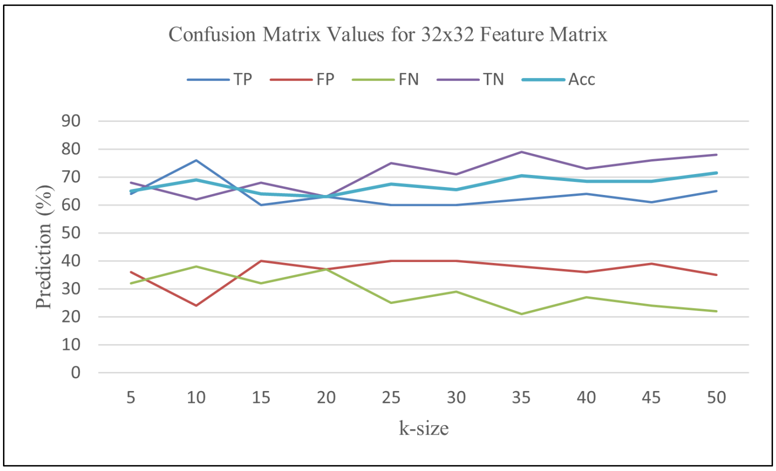

Figure 5. Chart of Confusion Matrix Values for 32 × 32 Feature Matrix

Table 3. Confusion Matrix Elements of KNN with input dimension of 16 × 16

k	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)
5	64	36	27	33	68.5
10	66	34	35	65	65.5
15	62	38	23	77	69.5
20	65	35	27	73	69
25	60	40	18	82	71
30	61	39	25	75	68
35	59	41	21	79	69
40	58	42	21	79	68.5
45	50	50	20	80	65
50	56	44	23	77	66.5

The confusion matrix values for a feature size of 256 are presented in Figure 6. This chart shows the steady trend of the correct predictions for a k-size greater than 10. Moreover, correct predictions for female fingerprints are higher than for male fingerprints for a k value higher than 10. As an effect, the false prediction for the female fingerprint is low.

It is interesting to look closely at Table 4 and Figure 7 that female fingerprint prediction (TN) is higher than male fingerprint prediction (TP) in all k values. The male fingerprint and female fingerprints are separated linearly without overlapping. As an effect, the accuracy also has consistent trends. Besides that, it exhibits that the greater the k-size, the better the female fingerprint prediction. However, the correct prediction of male fingerprints needs to be higher. The feature vector of size 88 looks to give the best outcome.

The trend in Figure 8 shows that the correct prediction is steady for a k value greater than 10. The trend is similar to that of Figure 6. The feature matrix of size 1616 demonstrated a similar result as size 44. Both of the features could separate linearly male fingerprints and female fingerprints. However, the correct prediction for the male fingerprint is lower than for the female prediction.

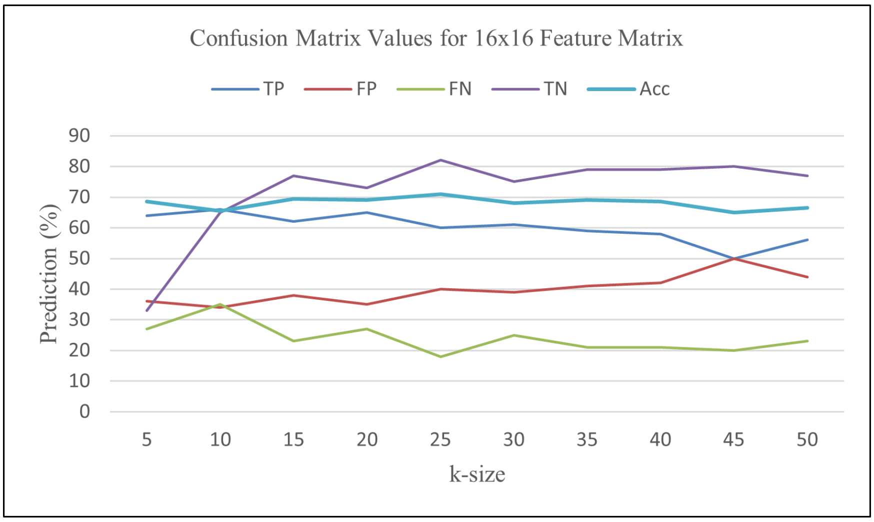

Figure 6. of Confusion Matrix Values for 1616 Feature Matrix

Table 4. Confusion Matrix Elements of KNN with the input of dimension 8 × 8

k	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)
5	55	45	27	73	64
10	57	43	32	69	62.5
15	47	53	20	80	63.5
20	53	47	23	77	65
25	42	58	20	80	61
30	47	53	24	76	61.5
35	44	56	18	82	63
40	45	55	19	81	63
45	42	58	14	86	64
50	44	56	14	86	65

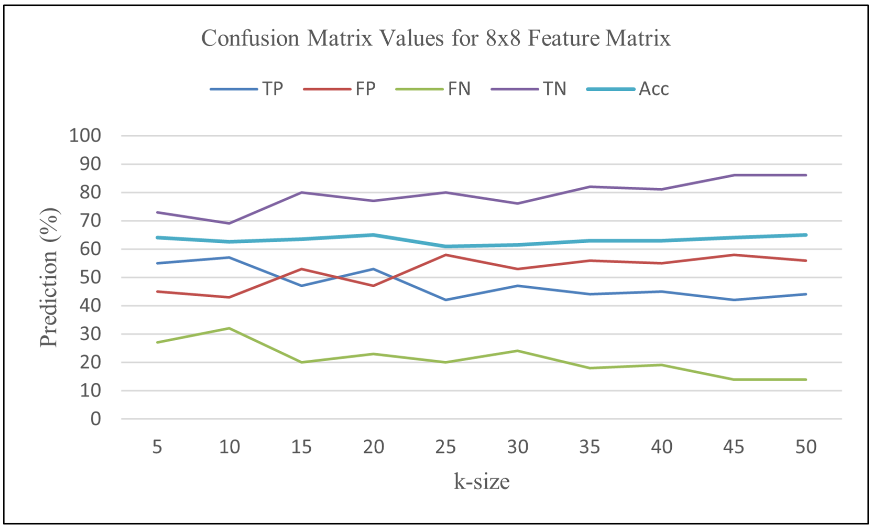

Figure 7. Chart of Confusion Matrix Values for 8 × 8 Feature Matrix

Table 5. Confusion Matrix Elements of KNN with the input of dimension 4 × 4

k	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)
5	62	38	30	70	66
10	69	31	37	63	66
15	61	39	26	74	67.5
20	60	40	29	71	65.5
25	53	47	19	81	67
30	56	44	20	80	68
35	52	48	17	83	67.5
40	58	42	21	79	68.5
45	54	46	20	80	67
50	56	44	20	80	68

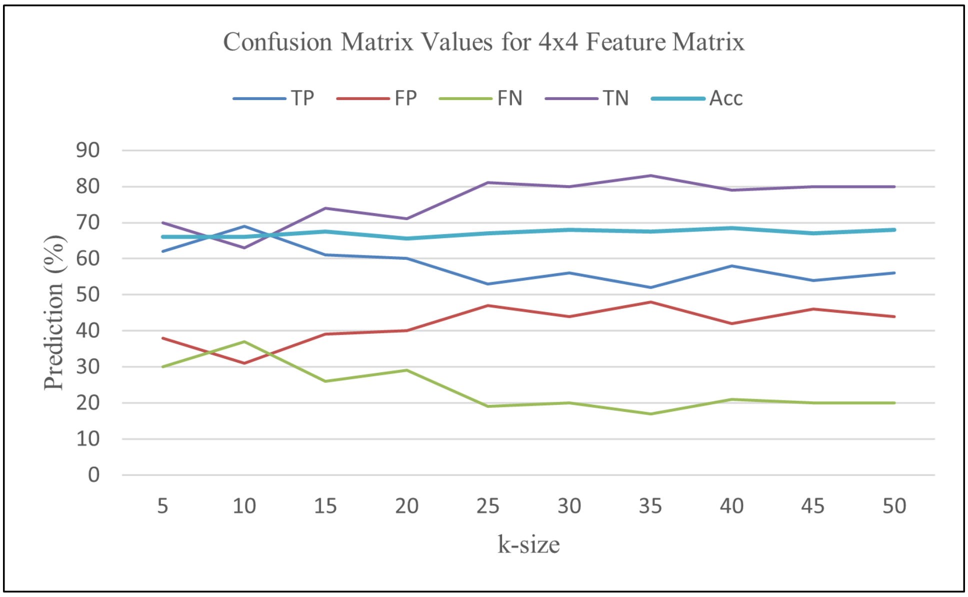

Figure 8. Chart of Confusion Matrix Values for 4 × 4 Feature Matrix

Table 6. Confusion Matrix Elements of KNN with the input of dimension 2 × 2

k	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)
5	60	40	36	64	62
10	67	33	37	63	65
15	58	42	33	67	62.5
20	66	34	35	65	65.5
25	63	37	27	73	68
30	61	39	31	69	65
35	54	46	28	72	63
40	61	39	26	74	67.5
45	56	44	24	76	66
50	57	41	28	72	65.5

Compared to other charts in this experiment, the graph in Figure 9 and Figure 5 has more variability for a k value of less than 20. However, the trend shows a consistent prediction value for a k value of more than 20. Another finding of this chart is that the feature size of 2 × 2 could not give a more acceptable result. Theoretically, the 2 × 2 matrix is too low compared to the original size. It means that much information is lost in the pooling process.

After examining all the experimental findings, the proposed method showed inferior results compared to the CNN model. For example, research on fingerprints in [15, 19] and [20] showed accuracy up to 95.05%, 96.64%, and 97.98%, respectively. Meanwhile, [21, 4, 8, 9] and [10] give accuracies above 95% for general object classification. Meanwhile, the proposed method can only achieve up to 71.5% accuracy. However, compared to CNN, the method simplifies the procedure by avoiding convolution computation and the ReLu function. As a result, the computation is much faster.

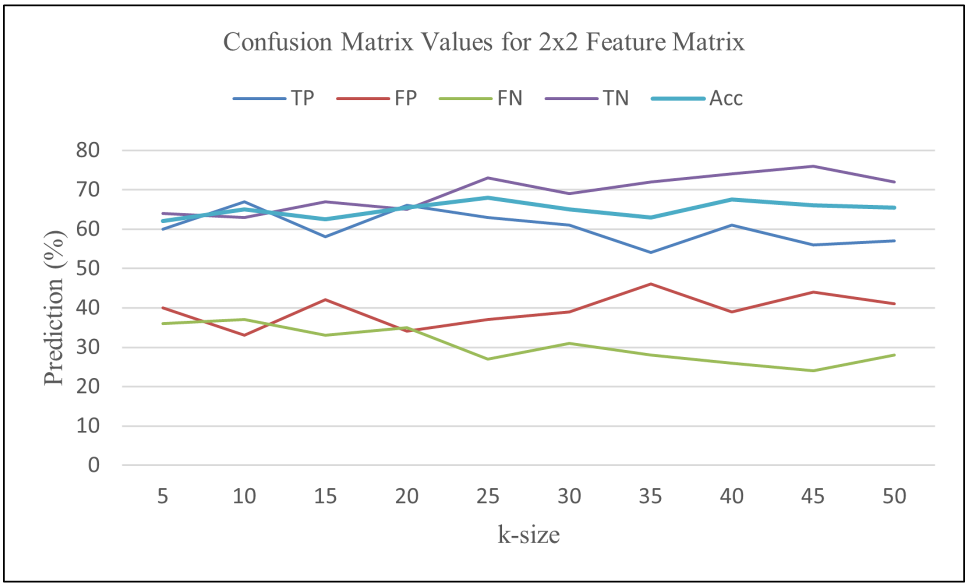

Figure 9. Chart of Confusion Matrix Values for 2 × 2 Feature Matrix

4. CONCLUSION

The empirical findings of this study provide a new viewpoint on the appropriate feature size for the input of the KNN algorithm. The most recognized conclusion from this model is that the pooling method could be implemented directly on the input data and that the execution time was fast. These results indicate that the three pooling layer methodsmax-pooling, average-pooling, and min-poolingcould be applied in succession. The main drawback of this study is its low accuracy, which is less than 70%. However, despite its limitations, the study certainly adds to our understanding of the pooling method that could simplify the feature generation of fingerprints. The efficiency of the arrayed order of the polling levels requires further research.

5. DECLARATIONS

AUTHOR CONTIBUTION

The first author carried out the research processes and wrote the manuscript draft. The second author helped to check cited literature and improve the language as well as the wholeness of the writings.

FUNDING STATEMENT

This research is funded by the Faculty of Information Technology, Duta Wacana Christian University.

COMPETING INTEREST

The author states that there is no conflict of interest with editors or reviewers regarding the publication of this article.

REFERENCES

[1] P. Gnanasivam and R. Vijayarajan, “Gender Classification from Fingerprint Ridge Count and Fingertip Size Using Optimal Score Assignment,” *Complex & Intelligent Systems*, vol. 5, no. 3, pp. 343–352, 2019.

[2] T. Philipp, D. Naser, B. Andreas, and K. Arjan, "Minutiae-Based Gender Estimation for Full and Partial Fingerprints of Arbitrary Size and Shape," in *Asian Conference on Computer Vision*, vol. 11361 LNCS, 2019, pp. 171–186.

[3] R. B. Ali, R. Ejbali, and M. Zaied, "A Deep Convolutional Neural Wavelet Network for Classification of Medical Images," *Journal of Computer Science*, vol. 14, no. 11, pp. 1488–1498, 2018.

[4] M. Harahap, A. P. S. Pasaribu, D. R. Sinaga, and R. Sipangkar, "Classification of Tuberculosis Based on Lung X-Ray Image With Data Science Approach Using Convolutional Neural Network," *SinkrOn*, vol. 7, no. 4, pp. 2193–2197, 2022.

[5] M. O. Arowolo, M. O. Adebiyi, E. P. Michael, H. E. Aigbogun, S. O. Abdulsalam, and A. A. Adebiyi, "Detection of COVID-19 from Chest X-Ray Images using CNN and ANN Approach," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 754–759, 2022.

[6] N. Alsharman and I. Jawarneh, "GoogleNet CNN neural network towards chest CT-coronavirus medical image classification," *Journal of Computer Science*, vol. 16, no. 5, pp. 620–625, 2020.

[7] R. Davuluri and R. Rengaswamy, "Improved Classification Model using CNN for Detection of Alzheimer's Disease," *Journal of Computer Science*, vol. 18, no. 5, pp. 415–425, 2022.

[8] V. Venkatesh, N. Yallappa, S. U. Hegde, and S. R. Stalin, "Fine-Tuned MobileNet Classifier for Classification of Strawberry and Cherry Fruit Types," *Journal of Computer Science*, vol. 17, no. 1, pp. 44–54, 2021.

[9] S. Jatmika and D. E. Saputra, "Rice Plants Disease Identification Using Deep Learning with Convolutional Neural Network Method," *SinkrOn*, vol. 7, no. 2, pp. 2008–2016, 2022.

[10] A. A. Gomaa and Y. M. El-Latif, "Early Prediction of Plant Diseases using CNN and GANs," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 514–519, 2021.

[11] K. Singh, R. Scholar, A. Mahajan, and V. Mansotra, "1D-CNN based Model for Classification and Analysis of Network Attacks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 604–613, 2021.

[12] B. K. Triwijoyo, "Model Fast Tansfer Learning pada Jaringan Syaraf Tiruan Konvolusional untuk Klasifikasi Gender Berdasarkan Citra Wajah," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 211–221, 2019.

[13] B. K. Triwijoyo, A. Adil, and A. Anggrawan, "Convolutional Neural Network With Batch Normalization for Classification of Emotional Expressions Based on Facial Images," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 1, pp. 197–204, 2021.

[14] T. A. Zuraiyah, S. Maryana, and A. Kohar, "Automatic Door Access Model Based on Face Recognition using Convolutional Neural Networ," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 22, no. 1, pp. 241–252, 2022.

[15] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 61–68, 2020.

[16] I. Boucherit, M. O. Zmirli, H. Hentabli, and B. A. Rosdi, "Finger Vein Identification Using Deeply-Fused Convolutional Neural Network," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 646–656, 2022.

[17] H. Hariyanto, S. Madenda, S. A. Sudiro, and T. M. Kusuma, "Fingerprint Authenticity Classification Algorithm based-on Distance of Minutiae using Convolutional Neural Network," *Jurnal Telekomunikasi dan Komputer*, vol. 11, no. 3, pp. 243–253, 2021.

[18] C. Lin and A. Kumar, "A CNN-Based Framework for Comparison of Contactless to Contact-Based Fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 662–676, 2019.

[19] M. Aritonang, I. D. Hutahaean, H. Sipayung, and I. H. Tambunan, "Implementation of Fingerprint Recognition Using Convolutional Neural Network and RFID Authentication Protocol on Attendance Machine," *ACM International Conference Proceeding Series*, pp. 151–156, 2020.

[20] B. Rim, J. Kim, and M. Hong, “Gender Classification from Fingerprint-images using Deep Learning Approach,” *ACM International Conference Proceeding Series*, pp. 7–12, 2020.

[21] W. Alakwaa, M. Nassef, and A. Badr, “Lung Cancer Detection and Classification with 3D Convolutional Neural Network (3D-CNN),” *International Journal of Biology and Biomedical Engineering*, vol. 11, no. 8, pp. 66–73, 2017.