# Partial Fingerprint on Combined Evaluation using Deep Learning and Feature Descriptor

Chrisantonius[1,2], Tri Kuntoro Priyambodo[1], Farchan Hakim Raswa[1,2], and Jia-Ching Wang[2]
[1]Department of Computer Science and Electronic, Universitas Gadjah Mada, Indonesia
[2]Department of Computer Science and Information Engineering, National Central University, Taiwan

***Abstract*— Partial fingerprint recognition has become crucial to identifying a user's authenticity in mobile device transactions. As a result, developments are increasing for more effective and accurate identification and authentication of a user using a scanner that captures a small fingerprint image. However, there is a reduction in the number of features from a full fingerprint to a partial fingerprint image during partial to partial fingerprint matching. Therefore, we propose a method combining deep learning and feature descriptors for partial fingerprint recognition. The matching score is obtained by the weighted combination of the scores from deep learning and feature descriptors. Experiments have been carried out with data variations such as the image size, epoch numbers and dataset types. The proposed method of combining deep learning and feature descriptors in the matching score evaluation process has obtained good results for the FVC2002 DB1, DB2 and DB3 datasets.**

***Keywords—partial fingerprint, deep learning, convolutional neural network, feature descriptor, combined matching evaluation***

## I. Introduction

A personal identification system generally involves a certain degree of manual operations. For example, a banking officer checks the customer's photo identifications against the customer's appearance. Likewise, manual verification of a person's identity is required upon the submission of insurance claims. Such operations are ineffective, and there is a need for a faster and more effective personal identification method. Moreover, manual verification may be inaccurate due to human errors. Therefore, a system for accurate and automatic personal identification will be beneficial for verification-required transactions. Biometrics such as iris, face, fingerprint, or hand gestures can be used to identify or recognize individuals. Fingerprint recognition has been considered one of the most practical ways to verify a user on smartphones, tablets, finger scanners and so on. As the demand for effectiveness and accuracy increases in current technological developments, the devices are expected to match partial to partial fingerprints in the identification process. However, the sensing area of the fingerprint scanner is relatively small and may only capture a partial fingerprint [1]. As a result, there are fewer minutiae present in the partial fingerprint compared to a full fingerprint. Therefore, there is a need for a more effective and accurate partial fingerprint identification on small devices such as mobile phones, tablets or laptops.

In deep learning, the convolutional neural network (CNN) can establish good modelling by modifying the hyperparameters of the model. The hyperparameters determine the network structure (for example, the hidden layer unit number and neuron) and variables determining how the workforce is trained (for example, the learning rate). The hyperparameters are initialized before the training process, that is, before optimizing the weights and bias. Several methods are used to perform a hyperparameter search, namely the manual search, grid search, random search and Bayesian optimization [2]. However, these methods require considerable processing time as well as high computational resources. Nevertheless, they have been applied to handle the performance optimizer, loss function, dropout [3], activation function, and regularization. Furthermore, the optimization of the CNN model has also been carried out using regularize activation function combination [4], filter learning combination [5], and low-rank matrix representation through convexification [6].

The Siamese neural network sometimes referred to as a twin neural network, is an artificial neural network containing two identical subnetworks with the same configurations, parameters and weights. A Siamese architecture can be used for verification, recognition, and feature representation [7]. In [8], the Siamese neural network uses a Recurrent Neural Network (RNN) to learn a dissimilarity metric from pairs of signatures and the resultant Recurrent Siamese network is used for online signature verification. In [9], the Siamese network is used for human identification by recognizing a person's gait. Additionally, the Siamese network is used in recognitions of other images, such as the palm print [10], as well as fingerprint contactless and base contact images [11]. The Siamese network is also used in feature representation to implement a multi-view for a 3D fingerprint from the PolyU2D Database [12]. Furthermore, deep learning is not restricted to being used for image or audio data. The Siamese network is also used in the textual match of resumes based on the job descriptions [13].

Although deep learning algorithms have been used to solve fingerprint recognition problems, not many research works have used partial fingerprint datasets [14]. As a result, many research studies still use hand-crafted algorithms for partial fingerprint recognition. Global orientation modeling reconstructs the global topology representation from a partial fingerprint dataset and searches for a minimal candidate for the pair-wise fingerprint matching [15]. Level 2 fingerprint features, especially the minutiae subset, are utilized in partial fingerprint processing since the ridge ending, and bifurcation are similar to those used in full fingerprint processing [16]. The major challenge in partial fingerprint matching lies in the insufficient numbers of minutiae within the enrollment fingerprint obtained from the small scanning area. As a result, the accuracy of minutiae-based approaches is decreased in partial-to-partial fingerprint matching. In [17], a matching algorithm that utilizes multi-scale texture descriptors for partial fingerprint images is proposed. Nevertheless, features apart from the minutiae can be extracted from a partial fingerprint dataset. The ridge shape features (RSFs) represent the small ridge segments where concave and convex edge shapes are detected. Various approaches have been proposed for different datasets, such as the Accelerated KAZE (AKAZE) method used in [17] and the scale-in-variant feature transform (SIFT) used in [18].

A multitude of methods exists for the matching and evaluation of fingerprints. A match is made by comparing one query image to several images and calculating the match value for each compared image. The number of comparisons between the false and genuine images depends on the number of datasets during the training process [19][20][21]. The evaluation process can be carried out according to the used models and methods, as well as by combining several features and models. In [17], the evaluation match score is obtained by combining the weighted values of the texture and topology scores. The texture score is obtained from the average Hamming distance, and the topology score is obtained from the average topology score is normalized by the image diagonal. In [22], global matching and minutiae-based matching are combined. The method focuses on the high-level patterns with the minutiae features and the low-level details with the global features.

## II. Background Study

### A. Deep Learning - CNN Architecture

The Convolutional Neural Network (CNN) is a type of neural network with a grid-like topology, and it is specifically used for data processing. It is used to process data such as the time-series data, considered as a 1D grid sampled at regular intervals, and image data, considered as 2D pixel grids. Convolutional neural networks have been successfully applied in various fields. The name implies that the network uses convolution operations, which are a type of linear operations. Convolutional neural networks use convolution instead of matrix multiplication with at least one network layer. The input of a CNN is an image with a certain height and width. The first step is a convolution of the image with a kernel or filter. Next, the output from the activation function is used as the input for the subsampling or pooling process. The pooling process produces a feature map whose size is dependent on the used pooling mask. In the final step, the produced feature map is used as an input for the fully connected neural network to obtain the classification result.

### B. Feature Descriptor - SIFT

The Scale-Invariant Actual Transform (SIFT) is an algorithm used to detect and describe the local features of an image. The features generated by SIFT are referred to as keypoints. A keypoint contains the location coordinates, orientation, and descriptors of a feature on an image. The resulting features are resistant to changes in the scale and orientation of the objects, variations in the light intensities and 3D viewing angles [23]. A keypoint descriptor is generated according to the following steps: the search of a keypoint candidate, selection of a keypoint, orientation determination the of keypoint and finally, the generation of a keypoint descriptor.

The K-Nearest Neighbor (KNN) algorithm is used for the matching of keypoints. An image in the database has multiple SIFT feature clusters. A cluster is created automatically using the KNN algorithm and indexed by the KD tree. The KNN algorithm looks for the smallest distance between a feature vector and the vectors in the clusters.

## III. Implemented Method

In this work, we have built a general description model according to the steps shown in Fig.1. The first process is based on the template input and query, where the CNN predicts the score based on the training model's results. Afterwards, the feature description extracted by the SIFT is based on the template and query dataset. Then the results are matched using the matching score obtained by the KNN algorithm. The deep learning model and the feature descriptor have the same number of queries and templates for the partial image matching process in matching processes carried out by the deep learning model and the feature descriptor. Hence, the comparisons process made is comparable or equal. Finally, the scores obtained from the two methods are combined by an empirically weighted function to produce an output match percentage.

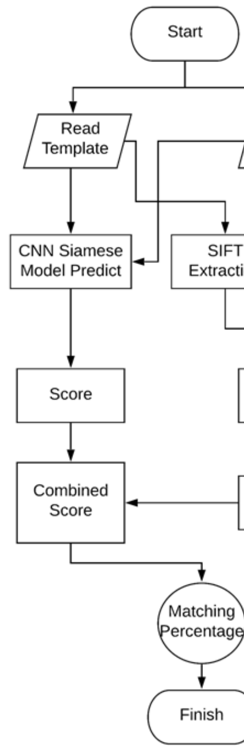



Fig. 1. Steps in the general description model built for partial fingerprint recognition.

### A. Siamese CNN

The preprocessed datasets are used in the next stage to train the Siamese CNN model. The Siamese CNN architecture used in this study consists of a convolution layer and a fully connected layer. The CNN architecture is used in this work. The training model uses the binary cross-entropy as a loss function and the ReLU as an activation function during the training process. Two images with a partial image size of 184x184 pixels are used as input for the Siamese CNN, and the input images are compared. Both images are used for the convolution layers which is processed in convolution feature and convolution model using the convolution operation on the CNN architecture. The convoluted image is then be subtracted from each input result to produce a fully connected layer.

### B. SIFT

In partial fingerprint recognition, the feature extraction process obtains the keypoints for each partial fingerprint. The keypoints are used to match the template and the query image at a later stage. The SIFT feature extraction is divided into four stages: searching for keypoint candidates, selecting the keypoint, determining the keypoint orientation and forming the keypoint descriptor.

Fig. 2. shows the steps for extracting the feature descriptor. The feature extraction stage involves several steps

to calculate the keypoint used to differentiate the images. The keypoint is represented as an image descriptor that will be used at a later stage. Next, the KNN method is used to search for suitable features or keypoint between the images. The value of $k$ used in this study is 2. This stage matches features in the form of a matrix image with another image that has been obtained previously by the SIFT algorithm.

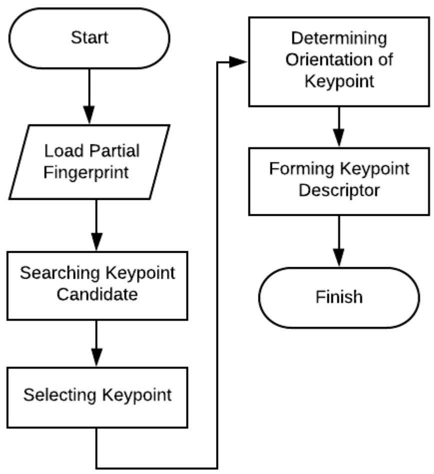



Fig. 2. Flowchart for the feature descriptor.

### C. Combined Matching Score Evaluation

The evaluation is performed by comparing the partial image fingerprint based on the template and the query images. In this work, the matching scores from the Siamese CNN model and the Feature Descriptor (obtained by SIFT) are combined by a weighted function. The comparison is performed with a genuine and an imposter image. We used FVC2002 as a dataset including DB1, DB2, and DB3. Each dataset contains 4000 partial fingerprint images (100 fingers index and each finger index have 40 impressions). The 4000 partial fingerprint images give a total comparison for the genuine and the impostor, 44450 for the imposters and 70300 for the genuine comparisons. The final evaluation score is obtained by a combination of the matching scores, $Score_{DL}$ and $Score_{FD}$. A weight ($wt$) is used in the weighted function to combine the matching scores as shown in Equation (1), where $wt$ has been empirically set to 0.02.

$$Final\ Score = (wt * Score_{FD}) + (1 - wt) * Score_{DL} \quad (1)$$

## IV. RESULTS

The matching evaluation results are shown in Table I. The results are compared with the ridge features and RGHE method. The table shows that the proposed method has a lower EER value compared to the other two methods.

TABLE I. MATCHING EVALUATION FOR DB1.

| FVC2002 DB1 | EER(%) |
|---|---|
| **Proposed** | **3.9** |
| Ridge Feature[24] | 5.57 |
| RGHE[25] | 4.36 |

We have compared the proposed method with two other methods in Table I.

- The ridge features are used for fingerprint matching. It combines the minutiae and ridge features. The ridge features contain the ridge count, ridge length, ridge accuracy and ridge curvature direction. The result from the ridge features is a block around each minutia. The ridge feature is used to calculate the similarity of the number of minutiae pairs that have crossed the threshold.
- The RGHE method protects the minutiae-based fingerprint template. The proposed template is strongly non-invertible. The RGHE method retains the neighborhood structure and obtains the resultant binary code in the Euclidean space.

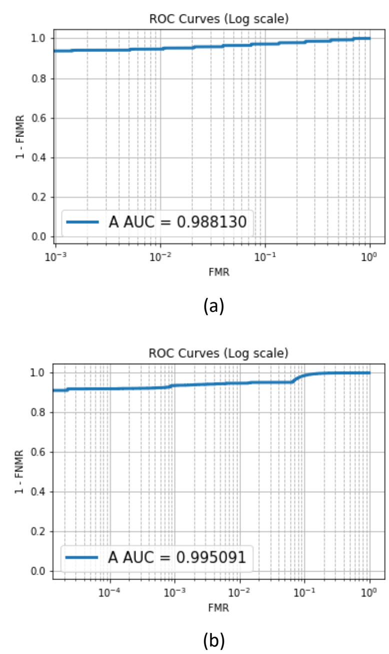
TABLE II. EVALUATIONS FOR DB1, DB2, AND DB3.

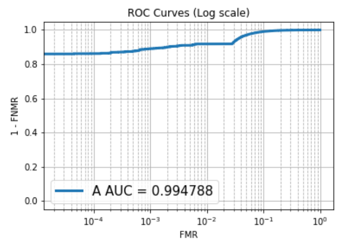
| FVC2002 | EER(%) |
|---|---|
| DB1 | 3.9 |
| DB2 | 4.8 |
| DB3 | 4.09 |

Table II presents the evaluation results for DB1, DB2, and DB3 for a 184x184 image of 184x184 using an adequate epoch, in this experiment adequate epoch is 2500. The EER values are close to 4% for DB1 and DB3 and 4.8% for DB2. Thus, even the same model and method are used, the evaluation results are not necessarily the same for every dataset.

The ROC curves are shown in Fig. 3, which shows that almost straight ROC curves can be obtained for a 184x184 image with AUV values of more than 90%.

The results indicate that the classifier has a high chance to distinguish between the genuine and impostor class values since the classifier can detect more FAR and FRR values.






(c)

Fig 3. ROC Curve of partial fingerprint matching from partial fingerprint dataset FVC2002 (a) DB1, (b) DB2, and (c) DB3.

## V. Conclusion

This research combines deep learning using the Siamese CNN and feature extraction with the SIFT for matching partial fingerprints. Evaluations have been carried out by combining the results from the deep learning process and feature descriptors with a weighted function. The evaluation is performed using the EER and AUC values and shows that the proposed method achieves reliable performance compared with the ridge features and RGHE methods.

## References

[1] W. Lee, S. Cho, H. Choi, and J. Kim, "Partial fingerprint matching using minutiae and ridge shape features for small fingerprint scanners," *Expert Syst. Appl.*, vol. 87, pp. 183–198, 2017, doi: https://doi.org/10.1016/j.eswa.2017.06.019.

[2] M. Claesen and B. De Moor, "Hyperparameter Search in Machine Learning," *CoRR*, vol. abs/1502.0, 2015, [Online]. Available: http://arxiv.org/abs/1502.02127.

[3] S. Reddy, K. T. Reddy, and V. Kumari, "Optimization of Deep Learning using Various Optimizers , Loss Functions and Dropout," 2019.

[4] R. Jie, J. Gao, A. Vasnev, and M.-N. Tran, "Regularized Flexible Activation Function Combinations for Deep Neural Networks," *CoRR*, vol. abs/2007.1, 2020, [Online]. Available: https://arxiv.org/abs/2007.13101.

[5] J. Jeong, J.-H. Cho, and J.-G. Lee, "Filter combination learning for CNN model compression," *ICT Express*, vol. 7, no. 1, pp. 5–9, 2021, doi: https://doi.org/10.1016/j.icte.2021.01.001.

[6] Y. Zhang, P. Liang, and M. J. Wainwright, "Convexified Convolutional Neural Networks," *CoRR*, vol. abs/1609.0, 2016, [Online]. Available: http://arxiv.org/abs/1609.01000.

[7] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometric Recognition Using Deep Learning: {A} Survey," *CoRR*, vol. abs/1912.0, 2019, [Online]. Available: http://arxiv.org/abs/1912.00271.

[8] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Exploring Recurrent Neural Networks for On-Line Handwritten Signature Biometrics," *IEEE Access*, vol. 6, pp. 5128–5138, 2018, doi: 10.1109/ACCESS.2018.2793966.

[9] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," *2016 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2832–2836, 2016.

[10] D. Zhong, Y. Yang, and X. Du, "Palmprint Recognition Using Siamese Network," 2018.

[11] C. Lin and A. Kumar, "Multi-Siamese networks to accurately match contactless to contact-based fingerprint images," *2017 IEEE Int. Jt. Conf. Biometrics*, pp. 277–285, 2017.

[12] C. Lin and A. Kumar, "Contactless and partial 3D fingerprint recognition using multi-view deep representation," *Pattern Recognit.*, vol. 83, pp. 314–327, 2018, doi: https://doi.org/10.1016/j.patcog.2018.05.004.

[13] S. Maheshwary and H. Misra, "Matching Resumes to Jobs via Deep Siamese Network," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 87–88, doi: 10.1145/3184558.3186942.

[14] U. Soni and G. Mahesh, "A Survey on State of the Art Methods of Fingerprint Recognition," 2018.

[15] Y. A. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 72–87, Jan. 2011, doi: 10.1109/TPAMI.2010.73.

[16] S. Asha and C. Chellappan, "Partial Fingerprint Matching Using Minutiae Subset," in *Intelligent Computing, Networking, and Informatics*, 2014, pp. 445–452.

[17] S. Mathur, A. Vjay, J. Shah, S. Das, and A. Malla, "Methodology for partial fingerprint enrollment and authentication on mobile devices," in *2016 International Conference on Biometrics (ICB)*, 2016, pp. 1–8, doi: 10.1109/ICB.2016.7550093.

[18] A. Aravindan and S. M. Anzar, "Robust partial fingerprint recognition using wavelet SIFT descriptors," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 963–979, 2017, doi: 10.1007/s10044-017-0615-x.

[19] A. Bazen, "Fingerprint Identification - Feature Extraction, Matching and Database Search," *Form. Methods Syst. Des. - FMSD*, 2002.

[20] B. Bakhshi and H. Veisi, "End to End Fingerprint Verification Based on Convolutional Neural Network," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 2019, pp. 1994–1998, doi: 10.1109/IranianCEE.2019.8786720.

[21] C. Gottschlich, "Skilled Impostor Attacks Against Fingerprint Verification Systems And Its Remedy," 2015.

[22] F. Zhang, S. Xin, and J. Feng, "Combining global and minutia deep features for partial high-resolution fingerprint matching," *Pattern Recognit. Lett.*, vol. 119, pp. 139–147, 2019, doi: https://doi.org/10.1016/j.patrec.2017.09.014.

[23] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, p. 91, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[24] C.-C. Liao and C.-T. Chiu, "Fingerprint recognition with ridge features and minutiae on distortion," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2109–2113, doi: 10.1109/ICASSP.2016.7472049.

[25] Z. Jin, M.-H. Lim, A. B. J. Teoh, and B.-M. Goi, "A non-invertible Randomized Graph-based Hamming Embedding for generating cancelable fingerprint template," *Pattern Recognit. Lett.*, vol. 42, pp. 137–147, 2014, doi: https://doi.org/10.1016/j.patrec.2014.02.011.