A Convolutional Neural Network Model Robust To Distorted Fingerprints

Ogechukwu Iloanusi, Chimdalu Okide

Abstract: The greatest challenge in fingerprint recognition is verifying distorted fingerprints. Distortion in fingerprints may arise from errors introduced while acquiring fingerprints, the nature of the fingerprints or from how there were deposited (in the case of latent fingerprints). In this paper, two convolutional neural networks were trained using different approaches. The first was trained in a regular pattern while the second was trained with an approach that minimizes errors that arise from verifying distorted fingerprints, and hence proposed for training models to be robust to distorted fingerprints. The trained models were evaluated on good and distorted data-sets. Results are modest and show better performance in the second model, compared to the first.

Index Terms: biometrics, convolutional neural network, distortion, fingerprint.

1 INTRODUCTION

Fingerprint is the most common biometric characteristic and is widely deployed in civil and commercial verification and identification systems. However, distortion is one of the greatest challenges in fingerprint recognition [1]–[3]. Distortion occurs when fingerprint ridges are rotated, smudged, partial or occluded. At times, distortion is due to poor fingerprint image acquisition, scanner errors or persons with challenging fingerprints. A 2D fingerprint scanner is used to sense a pattern of 3D fingerprint made up of ridges and furrows. The pressure of a fingertip on a scanner could make the sensed fingertip a dense, smudged or incomplete fingerprint. A rotated fingerprint results in translation in fingerprint minutiae locations and orientations. Fingerprint ridges and minutiae are displaced, deformed or missing in distorted fingerprints and lead to high errors in fingerprint recognition. Distortion is the sepulcher of fingerprint biometrics. In phones or devices that utilize iTouch application, rotation of user's fingerprint(s) is not a problem because, a fingerprint is registered in the device by enrolling all possible orientations and corners of the fingerprint in the device. This is however, impossible with large-scale enrolment like in civil applications where persons are enrolled in thousands. Fingerprint scanners used for large-scale enrolment do not support short over-detailed enrolment as in iTouch devices. A typical civil application is in VISA enrolment. Images of a fingerprint that were acquired in our Department with varying orientations, pressure on scanner and moisture on fingertip are shown in. Impression in (a) is clear, while impressions in (b), (c) and (d) are partial, rotated and smudged.

Fingerprint distortion has been demonstrated to have a linear limiting effect on performance in [4]. The impact of fingerprint distortion has been analysed in [5]–[7]. Several approaches to minimize distortion in fingerprints in the literature are either traditional, involving structures or local features for selection of feature descriptors for fingerprint recognition, or neural network based.

- *Ogechukwu Iloanusi -University of Nigeria Nsukka. ogechukwu.iloanusi@unn.edu.ng*
- *Chimdalu Okide -University of Nigeria, Nsukka. chimdalu.okide@unn.edu.ng*

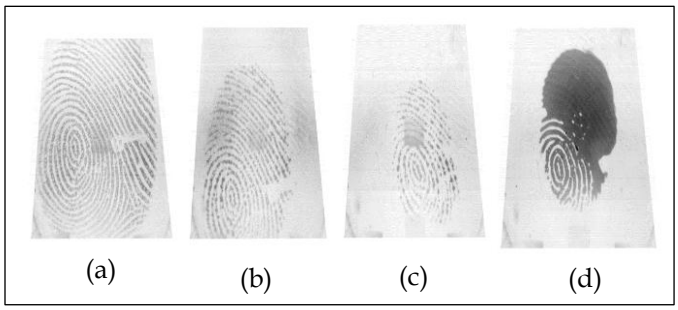


Fig. 1: *Fingerprint impressions of a subject clear in (a) partial in (b) rotated and partial in (c) and rotated and smudged in (d).*

Traditional methods are handcrafted and are often times not scalable, due to the limitations of feature descriptors in each fingerprint's feature set. Artificial neural networks are better but still do not well represent images, because image pixels in an image are vectorised prior to training an ANN or using an ANN for recognition, thereby deforming the pixels' locations in the image. Deep learning, using convolutional neural networks (CNN) is the best approach in image processing and analysis because images are used as they are, maintaining the spatial locations or properties of pixels in a CNN, unlike artificial neural networks (ANN). A CNN's structure is well adapted to classifying images unlike ANN that requires image pixels vectorised for adaptation to an ANN. Basically, in deep learning using a CNN, 2D or 3D filters or kernels or weights are used in respectively convolving a 2D or 3D image, and the responses are transferred to feature or activation maps connected to the kernels. This operation, amongst other dimensionality reduction or normalization methods, is repeated several times depending on the depth of the CNN. Filter sizes of $3 \times 3 \times C$, $5 \times 5 \times C$ or $7 \times 7 \times C$, where C is the number of channels, are typically used.

Deep learning has been applied to fingerprint minutiae extraction, segmentation or enhancement in the literature, but a distortion invariant CNN has never been trained in the literature. However, artificial neural networks have been trained for recognition of occluded fingerprints. In this paper, two fingerprint CNNs are trained adopting a regular training technique in the first CNN, while using a method that minimizes errors due to distorted fingerprints in the second. These CNNs would be referred to as regular CNN (***R-CNN***) and proposed CNN (***P-CNN***) in this paper. ***P-CNN*** is trained

using an approach that minimizes errors in recognition due to distorted fingerprints. This approach is recommended for training fingerprint CNNs to be robust to distorted fingerprints. The two CNNs trained in this paper have the same architecture and training data.

This paper is organised as follows: Section 1 introduces the paper while existing works on distortion were reviewed in Section 2. Architecture, training and approach used for ***R-CNN*** and ***P-CNN*** are discussed in Section 3 and evaluated on test sets in Section 4.

2 Review of Related Works

Several approaches have been applied in studying or minimizing distortion in fingerprints [8]–[11].

In [12], they propose a scheme for handling distorted fingerprints while identifying people in the watch-list database. This scheme includes a combination of acquired live scanned good quality fingerprints and acquisition of rolled fingerprints followed by an image enhancement stage for distorted fingerprints and both type and consistently checked and minutiae files are manually rectified if distortion exists in any fingerprint. A method for image content fingerprinting based on sparse coding and dictionary learning is proposed in [13] for good as well as distorted fingerprint feature selection and matching. A pre-processing step for distortion correction in fingerprints was proposed in [14]. A region of interest selected around the core point of a reference fingerprint is used to detect a region of interest in a probe image. Transformation parameters estimated based on a reference and probe's region of interest are used to minimize distortion in matching. A Malsburg learning neural network approach was proposed in [15] for recognizing occluded fingerprints. A hybrid artificial neural network was proposed for pre-processing and image restoration of distorted fingerprints. The network was tested by simulations in MATLAB / Simulink environment. Minutiae cylinder code fingerprint descriptors has also been proposed for feature selection in challenging fingerprints [16]. A system for minimizing distortion in fingerprints is proposed in [17] for detecting mutilated fingerprints. A method based on minutiae and ridge features, for recognizing fingerprints that are good and also distorted, is proposed in [18]. An alignment method for fuzzy vault technique used for fingerprint template protection is proposed for good and challenging fingerprints in [19].

So far, the methods applied in addressing distortion problems in fingerprints are traditional or ANN based. Therefore, deep learning is explored as an option for handling this challenge in this paper.

3 Regular and Proposed Fingerprint CNNs

3.1 Fingerprint CNNs' Architecture

Architecture designed for both CNNs in ***Fig. 2*** has a total of 26 layers including convolutional, rectified linear unit, pooling, fully connected, normalization and classification layers. Convolutional layers are 8 while fully connected layers are 3. CNN has an input size of 32 × 32 pixels, implying that sizes of all fingerprint images used in this paper were normalized to this input size. All fingerprints are grayscale images.

Input layer: 32 × 32 × 1
Convolution layer 1: [3 × 3 × 64]
Rectified linear unit layer 1
Convolution layer 2: [3 × 3 × 64 × 128]
Max pooling layer 1: [Stride: 2 × 2]
Rectified linear unit layer 2
Convolution layer 3: [3 × 3 × 128 × 256]
Rectified linear unit layer 3
Convolution layer 4: [3 × 3 × 256 × 256]
Rectified linear unit layer 4
Convolution layer 5: [3 × 3 × 256 × 512]
Rectified linear unit layer 5
Convolution layer 6: [3 × 3 × 512 × 512]
Rectified linear unit layer 6
Convolution layer 7: [3 × 3 × 512 × 512]
Max pooling layer 2: [Stride: 2 × 2]
Rectified linear unit layer 7
Convolution layer 8: [3 × 3 × 512 × 1048]
Rectified linear unit layer 8
Fully connected layer 1: [3 × 3 × 1048 × 1148]
Rectified linear unit layer 9
Fully connected layer 2: [3 × 3 × 1148 × 1148]
Rectified linear unit layer 10
Fully connected layer 3: [1 × 1 × 1148 × 1129]
Normalization layer: Softmax function
Classification layer: [1129 classes]

Fig. 2: *Architecture of convolutional neural networks –* ***R-CNN*** *and* ***P-CNN***

All filter kernels used in this paper have a size of $3 \times 3 \times C$. The quantity of feature maps used in this architecture rise from 64, through 128, 256, 512 to 1148. The normalization layer in ***Fig. 2*** named a SoftMax layer, performs a normalized logistic function that quantizes values to either 0 or 1. Network was trained for 1129 classes. However, in most trained CNNs in the literature, the focus is not to classify objects on a limited number of known objects, but rather to adapt the CNN to recognize unforeseen objects or even identify persons' biometric characteristics. Hence, features in a CNN used to identify persons should be salient enough to distinguish among persons. Hence, in a CNN, features or activations can be extracted after a preferred convolutional layer. A feature matrix or vector corresponds to a feature level in a CNN, and varies depending on the level in the CNN, where it is extracted. In this paper, all feature vectors used for fingerprint verification are extracted after the second fully connected layer. Hence feature descriptors' set is a vector of size, 1148.

3.2 FINGERPRINT CNN TRAINING

3.2.1 Training Dataset

Two convolutional neural networks (CNN)s were trained in this Section. Dataset used in this paper comprises two proprietary databases from [4] and three standard databases [21]–[23]. The first proprietary database has 3744 ten-print fingerprints from 95 individuals, at 4 impressions per individual. The second proprietary dataset has 400 index

fingers of 200 individuals. The standard databases are the Fingerprint Verification Competition (FVC) training sets B of sub databases 2000 [21], 2002 [22] and 2004 [23].
The distribution of fingerprints in the first database is detailed in Table 1. Training set constituted all 3744 ten-prints of the first database, index fingerprints of the first 100 individuals of the second database and FVC set B sub databases. The remaining fingerprints of the other 100 individuals in the second database made up the test-set in this paper.
The total number of classes used in the training is 1129. This huge number of classes results from the 10 possible classes for an individual in the first database given that the 10 fingers of an individual are unique.

Table 1: *Fingerprints in the ten-print database*

Left	**Wed**	**Index**	**Middle**	**Thumb**	**Tiny**
Subjects' number	94	95	94	92	94
Fingerprints' number	376	380	376	368	376
Right	**Wed**	**Index**	**Middle**	**Thumb**	**Tiny**
Subjects' number	94	95	94	92	95
Fingerprints' number	376	380	376	368	380

Nine (9) sub databases were used in the FVC database namely: FVC 2000 DB1B, DB2B and DB4B; FVC 2002 DB2B, DB3B and DB4B; FVC 2004 DB2B, DB3B and DB4B. Each of the FVC training set B database has 80 fingerprints from 10 individuals. In training ***P-CNN***, each fingerprint impression was additionally rotated to −30°, 15°, 30°, 45° and 60° orientations in order to represent majority of images in the training set as distorted. In order to have as much training images in ***R-CNN***, images were replicated to the number of impressions in ***P-CNN***. Total number of training images in ***R-CNN*** and ***P-CNN*** are 32,132 and 32,032 respectively.

3.2.2 Training parameters

Prior to training, training sets and labels were randomly shuffled. Convolution kernels or weights were randomly initialized. Basic learning rate and momentum were set to 0.001 and 0.9 respectively. Mini-batch size was 128. Both models converged on training. The training parameters such as the total number of iterations, final mini-batch loss and accuracy and total training time to convergence of ***R-CNN*** and ***P-CNN*** are given in Table 2.

Table 2: *Training parameters for* ***R-CNN*** *and* ***P-CNN****.*

The parameters show that ***R-CNN*** and ***P-CNN*** converged and have mini-batch loss of 0.00% and mini-batch accuracy of 100.00%.

CNN Type	**Number of Iterations**	**Mini-batch loss**	**Final mini-batch accuracy**	**Training time (seconds)**
R-CNN	19,800	0.0000	100.00%	11,784.3
P-CNN	32,000	0.0000	100.00%	19,031.7

4 EXPERIMENTS, RESULTS AND DISCUSSIONS

All experiments in this paper were carried out in MATLAB 2017b environment. ***R-CNN*** and ***P-CNN*** were trained in MATLAB environment.

4.1 Test sets

Test sets were evaluated on ***R-CNN*** and ***P-CNN*** in this Section. Test sets comprise the last 200 fingerprints of 100 subjects in the second dataset not used in the training. This is to ensure that training and test sets are disjointed. Eight more groups of the test sets were created by using rotating the probe set images in the following orientations: 10°, 15°, 20°, 25°, 30°, 35°, 40°, and 45°. The reference set was not rotated. The rotation introduced increasing forms of distortion in the test sets. The reason for the varying rotations is to see how their performances vary on ***R-CNN*** and ***P-CNN***. Hence, they were 9 test sets with 200 fingerprints each, where query are 100 fingerprints of 100 individuals, and references are 100 fingerprints of 100 of individuals. It is hopefully expected that performance of the distorted fingerprints would be closer to that of the normal sets, and not significantly poorer, on ***P-CNN***. Hence, performance on ***P-CNN*** should be better than that on ***R-CNN***.

4.2 Fingerprint matching protocol for genuine and impostor matches

A match is a comparison between two biometric impressions which could be genuine (when between those of same subject) or impostor (when between those of different subjects). In the verification exercise, a query print is compared against all 100 references to seek a match. Since there are 2 impressions per individual, there should be only 1 matching impression, $\binom{2}{2} = 1$, for a query's comparison, that is genuine, while other 99 should be impostors. The total number of genuine matches for 100 queries will be 100. The number of possible impostor matches amongst a subject's impression and other non-matching subjects' impressions are 99. Total number of impostor matches are $99 \times 100 = 9{,}900$. This can also be computed from the total number of matches, 10,000 minus 100 genuine matches. In this paper, 100 genuine and 9,900 impostor scores were computed for each of the 9 test sets. Genuine and impostor scores are dissimilarity scores.

4.3 Performance metrics of 9 test sets on R-CNN and P-CNN

A biometric recognition system takes a two-valued decision depending on a set threshold. A score is classified as either genuine or impostor depending on this set threshold. 'Genuine' is affirmed when a score is on or above the threshold, signifying a true match and 'impostor' affirmed when the score is below the threshold, meaning there is no match. Genuine accept rates (GAR), false accept rates (FAR) and false reject rates (FRR) are then determined from the scores. False accepts are false positives errors while false rejects are true negatives errors. A false accept occurs in biometric recognition when a person's biometric is wrongly matched to another's person's biometric. FAR at a particular threshold, ***T***, is determined as follows:

$$FAR|_T = \frac{number\ of\ false\ accepts\ at\ T}{total\ impostor\ scores} \times 100\% \quad (1)$$

A false reject occurs when a person's biometric is erroneously rejected by a biometric system or matcher.

$$FRR|_T = \frac{number\ of\ false\ rejects\ at\ T}{total\ genuine\ scores} \times 100\% \quad (2)$$

Genuine accept rate (GAR) at ***T*** may be determined as follows:

$$GAR|_T = 100\% - FAR|_T \quad (3)$$

In applications requiring biometric security, the threshold can be set by the user or administrator. In biometric evaluations, varying thresholds are used to determine the false accept, genuine or true accept and false reject rates. Varying thresholds were used in this paper. Accuracy of biometrics recognition systems or algorithms is determined from their error rates. These are termed the performance metrics of biometric systems. The less the biometric error rate, the more accurate the system. Test sets were evaluated using known biometric performance metrics, namely, false accept rate (FAR), false reject rate (FRR), genuine accept rate (GAR) and receiver operating characteristics curves. Receiver operating characteristic curve is a plot of false accept versus genuine accept rates, showing the trade-off between the two sets of values. Area under the receiver operating characteristic was finally used as a metric, since it will be difficult to show 16 curves, with narrow gaps between them, in a single graph. A perfect area and best performance is 1 square unit while the worst case is 0. Area under the ROC curve (AUC) was determined in this paper using Riemann's sums of approximation from rectangles. As thresholds can span in hundreds, the approximation rectangles are thin enough to approximate the area under the curve. Area under the curve for the 9 test sets evaluation on ***R-CNN*** and ***P-CNN***, at varying rotations, are shown in
Table ***3***. In the evaluation with ***R-CNN*** in
Table ***3***, AUC significantly declines in value as value of fingerprint orientation increases. Performance at 0° in ***R-CNN*** declined from 0.646 to 0.549 at 35°. The decline in the evaluation of the test sets on ***P-CNN*** in
Table ***3*** is not as pronounced as in ***R-CNN***. Values showing degradation in performances would better represent the degradation in performances in evaluations on ***R-CNN*** and ***P-CNN***. Using the AUC at 0° as reference, degradation in performance for each evaluation in either ***R-CNN*** or ***P-CNN***, at a particular orientation, $\theta°$, $DP_{\theta°}$, is calculated in this paper as follows:

$$DP_{\theta^\circ} = \frac{AUC_{0^\circ} - AUC_{\theta^\circ}}{AUC_{0^\circ}} \times 100\% \quad (4)$$

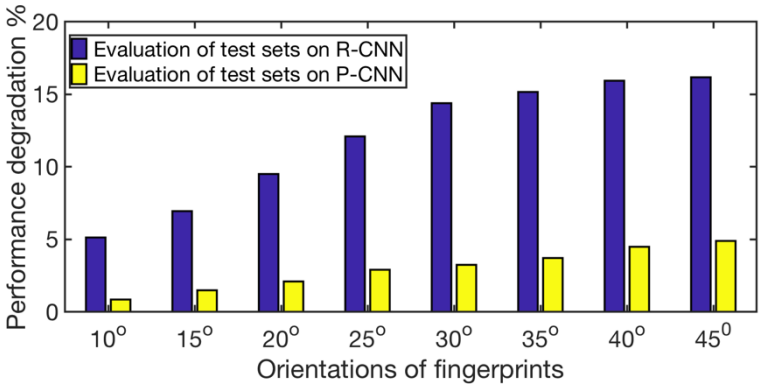
Therefore, for ***P-CNN*** where $AUC_{0^\circ} = 0.732$ and $AUC_{20^\circ} = 0.717$,

$$DP_{20^\circ} = \frac{0.732 - 0.717}{0.732} \times 100\% = 2.1\% \quad (5)$$

DP_{0° is not determined since it is the normal orientation.
DP_{θ° is determined for the evaluations on ***R-CNN*** and ***P-CNN*** based on
Table ***3*** and is plotted in a bar chart in ***Fig. 3***.

Table 3: *Area under the curves for* ***R-CNN*** *and* ***P-CNN*** *at varying rotations:* 0°, 10°, 15°, 20°, 25°, 30°, 35°, 40°, *and* 45°.

CNN Type	0°	10°	15°	20°	25°	30°	35°	40°	45°
R-CNN	0.646	0.613	0.602	0.585	0.568	0.553	0.549	0.543	0.542
P-CNN	**0.732**	**0.726**	**0.721**	**0.717**	**0.711**	**0.708**	**0.705**	**0.699**	**0.696**



θ°	*R-CNN*	*P-CNN*
10°	5.10%	**0.83%**
15°	6.93%	**1.47%**
20°	9.49%	**2.10%**
25°	12.08%	**2.89%**
30°	14.39%	**3.22%**
35°	15.15%	**3.71%**
40°	15.94%	**4.48%**
45°	16.17%	**4.87%**

Fig. 3: *Degradation in performance,* $DP_{\theta^{\circ}}$, *for* ***R-CNN*** *and* ***P-CNN*** *at varying rotations:* 0°, 10°, 15°, 20°, 25°, 30°, 35°, 40°, *and* 45°.

Degradation in performance in ***P-CNN*** is much less than that of ***R-CNN*** for all rotations of fingerprints. This shows that ***P-CNN*** was trained in a manner that minimizes errors due to fingerprint distortion. Performance is even better in ***P-CNN***.
In ***Fig. 3,*** the bars of ***R-CNN*** are much taller than those of ***P-CNN*** at all orientations of fingerprints. Performance degradation in ***P-CNN*** at 45° is 4.87% compared to 16.17% of ***R-CNN*** at 45°.

5 Conclusion

Two (2) fingerprint convolutional neural network (CNN) models were trained in this paper. The first was trained in a regular pattern while the second was trained using an approach that minimizes errors due to distortions in fingerprints, when verifying fingerprints. Test sets not used in the training were evaluated on the 2 CNNs. The percentages in performance degradation obtained in the results in the CNNs show that the second model is more robust to distortion compared to the first model. The approach adopted for minimizing distortion is hence proposed for training fingerprint CNN models to be robust to distorted fingerprints.

References

[1] S. Gu, J. Feng, J. Lu, and J. Zhou, "Efficient Rectification of Distorted Fingerprints," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 1, pp. 156–169, Jan. 2018.

[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, 2nd ed. London: Springer-Verlag, 2009.

[3] A. M. Bazen and S. H. Gerez, "Achievements and Challenges in Fingerprint Recognition," in Biometric Solutions, Boston, MA: Springer US, 2002, pp. 23–57.

[4] O. N. Iloanusi and C. A. Ezema, "A quantitative impact of fingerprint distortion on recognition performance," Inf. Secur. J. A Glob. Perspect., vol. 26, no. 6, pp. 267–275, Nov. 2017.

[5] N. D. Kalka and R. A. Hicklin, "On relative distortion in fingerprint comparison," Forensic Sci. Int., vol. 244, pp. 78–84, Nov. 2014.

[6] H. D. Sheets, A. Torres, G. Langenburg, P. J. Bush, and M. A. Bush, "Distortion in Fingerprints: A Statistical Investigation using Shape Measurement Tools," J. Forensic Sci., vol. 59, no. 4, pp. 1113–1120, Jul. 2014.

[7] J. Wallis and J. Goulet, "Quantitative analysis of the distortion of friction ridge impressions according to three deposition pressure levels and horizontal movement," J. Forensic Identif., vol. 67, no. 2, pp. 259–277, 2017.

[8] C. S. Mlambo and Y. Moolla, "Complexity and Distortion Analysis on Methods for Unrolling 3D to 2D Fingerprints," in 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2015, pp. 103–109.

[9] C. S. Mlambo and M. B. Shabalala, "Distortion Analysis on Binary Representation of Minutiae Based Fingerprint Matching for Match-on-Card," in 2015 IEEE Symposium Series on Computational Intelligence, 2015, pp. 349–353.

[10] Qijun Zhao, A. Jain, and G. Abramovich, "3D to 2D fingerprints: Unrolling and distortion correction," in 2011 International Joint Conference on Biometrics (IJCB), 2011, pp. 1–8.

[11] 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, Nov. 2017.

[12] K. Madhavi and B. Sreenath, "Rectification of distortion in single rolled fingerprint," in 2016 International Conference on Circuits, Controls, Communications and Computing (I4C), 2016, pp. 1–4.

[13] Y. Li and L. Guo, "Robust Image Fingerprinting via Distortion-Resistant Sparse Coding," IEEE Signal

Process. Lett., vol. 25, no. 1, pp. 140–144, Jan. 2018.

[14] E. Derman and M. Keskinoz, “Normalized cross-correlation based global distortion correction in fingerprint image matching,” in 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), 2016, pp. 1–4.

[15] S. Kundu and G. Sarker, “A modified BP network using malsburg learning for rotation and location invariant fingerprint recognition and localization with and without occlusion,” in 2014 Seventh International Conference on Contemporary Computing (IC3), 2014, pp. 617–623.

[16] R. Cappelli, M. Ferrara, and D. Maltoni, “Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

[17] R. Shanthi and T. A. Kumaran, “Enhanced fingerprint distortion removal system,” in 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1859–1863.

[18] 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.

[19] J. Zouari and M. Hamdi, “Enhanced fingerprint fuzzy vault based on distortion invariant minutiae structures,” in 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2016, pp. 491–495.

[20] O. N. Iloanusi, “Fusion of finger types for fingerprint indexing using minutiae quadruplets,” Pattern Recognit. Lett., vol. 38, no. 1, pp. 8–14, 2014.

[21] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2000: fingerprint verification competition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 3, pp. 402–412, Mar. 2002.

[22] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2002: Second Fingerprint Verification Competition,” in Object recognition supported by user interaction for service robots, 2002, vol. 3, pp. 811–814.

[23] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2004: Third Fingerprint Verification Competition,” Springer, Berlin, Heidelberg, 2004, pp. 1–7.