



# A CNN BASED ROTATION INVARIANT FINGERPRINT RECOGNITION SYSTEM

Tuba Çelik MAYADAĞLI<sup>1</sup>, Ertuğrul SAATÇI<sup>2</sup>, Rifat EDİZKAN<sup>1</sup>

<sup>1</sup> Department of Electric-Electronic Engineering, Faculty of Engineering, Osmangazi University, Eskişehir, Turkey

<sup>2</sup> Department of Electrical-Electronic Engineering, Faculty of Engineering, İstanbul Kültür University, İstanbul, Turkey

tubacelikmayadagli@gmail.com, e.saatci@iku.edu.tr, redizkan@ogu.edu.tr

---

**Abstract:** This paper presents a Cellular Neural Networks (CNN) based rotation invariant fingerprint recognition system by keeping the hardware implementability in mind. Core point was used as a reference point and detection of the core point was implemented in the CNN framework. Proposed system consists of four stages: preprocessing, feature extraction, false feature elimination and matching. Preprocessing enhances the input fingerprint image. Feature extraction creates rotation invariant features by using core point as a reference point. False feature elimination increases the system performance by removing the false minutiae points. Matching stage compares extracted features and creates a matching score. Recognition performance of the proposed system has been tested by using high resolution PolyU HRF DBII database. The results are very encouraging for implementing a CNN based fully automatic rotation invariant fingerprint recognition system.

**Keywords:** Fingerprint, Cellular Neural Networks, Rotation Invariant, Fingerprint Recognition System.

---

## 1. Introduction

The use of biometrics is an evolving component in today's society. Fingerprints has been widely used in forensics applications such as criminal identification and prison security. Also fingerprint recognition technology has been widely adopted in civilian applications such as electronic banking, ecommerce, and access control because fingerprints are simple to get via fingerprint reader. Fingerprint recognition continues to be one of the most widely used biometric systems for personal identification. Aim of this work is to propose a hardware implementable CNN based system for fingerprint recognition. In the literature, although many methods exist for fingerprint recognition, there are relatively few CNN based methods [1, 2, 3]. Also these methods are not rotation invariant like the one proposed here.

The concept of CNN was introduced by L.O. Chua and L. Yang [4] and many papers have been published about the CNN and its application in image processing [5, 6, 7, 8]. CNN is a 2D grid of identical and regularly spaced cells. Therefore, the topology of CNN is well suited for image processing applications, image pixels can be mapped directly on to the array of CNN cells for processing. In other words, each cell in the CNN corresponds to a pixel in the image. Each cell

communicates with the cells in its local neighborhood only. The CNN cells are very simple circuit nodes.

Thus, CNNs are amenable to implementation in VLSI and FPGA [9, 10, 11, 12] technology, a feature that is extremely important for building fast image processing hardware. Different image processing tasks can be performed by changing the template coefficients of the CNN.

The proposed CNN based rotation invariant fingerprint recognition system consists of four stages: preprocessing, feature extraction, false feature elimination and matching. The preprocessing stage includes contrast stretching, Gabor-type filtering, lowpass filtering and grayscale to binary thresholding. The feature extraction stage includes core point detection, ridgeline thinning, minutiae point extraction and it creates a 2D feature vector. The false feature elimination stage includes removing the four types of false minutiae points. The matching stage includes comparing the extracted features with the ones in the database and it creates a matching score. The extracted 2D feature vector is used in the fingerprint matching. The False Match Rate (FMR), False Non-Match Rate (FNMR) and Receiver Operating Characteristic (ROC) curves are also calculated by changing the threshold value to reflect the full system performance for a possible application. Recognition performance of the proposed system has been tested by using PolyU HRF DBII database [21].

## 2. CNN Model for Gray Scale Image Processing

The normalized time nonlinear differential equation for cell  $(i,j)$  in the original Chua-Yang CNN model (conventional CNN) [4] for neighborhood radius  $r$  is given as:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{k,l=-r}^r a_{k,l} y_{i+k,j+l}(t) + \sum_{k,l=-r}^r b_{k,l} u_{i+k,j+l} + I \quad (1)$$

$$y_{ij}(t) = \frac{1}{2} (|x_{ij}(t) + 1| - |x_{ij}(t) - 1|) \quad (2)$$

Where  $A = [a_{k,l}]_{k,l=-r}^r$  and  $B = [b_{k,l}]_{k,l=-r}^r$  are the feed-back and feed-forward templates,  $u_{i,j}$ ,  $x_{i,j}$ ,  $y_{i,j}$

and  $I$  are the input, the state, the output and the bias term, respectively. Because of the thresholded activation function at the output given in Eq. (2), conventional stable CNN can only provide binary output which does not carry color information more than two. In order to represent gray scales at the output, a linear CNN is required, which can be defined as a purely linear version of conventional CNN. In other words, the piecewise output nonlinearity in the conventional CNN is set to the identity to obtain a linear CNN. Equivalently, it can be assumed that the input to the conventional CNN is sufficiently small and the dynamics evolve in a such way so that the outputs of the CNN never enter the saturation region of the piecewise linear output nonlinearity. By removing the output nonlinearity, the cell equation for such CNN is obtained from Eq. (1) as follows:

$$\frac{dx_{ij}(t)}{dt} = -x_{i,j}(t) + \sum_{k,l=-r}^r a_{k,l} x_{i+k,j+l}(t) + \sum_{k,l=-r}^r b_{k,l} u_{i+k,j+l} + I \quad (3)$$

In this case, each cell on the array is simply a linear spatial filter whose input/output behavior is described by the  $A$  and  $B$  templates. To process a gray scale image composed of  $M \times N$  pixels with an  $M \times N$  linear CNN, the gray scales of the image should be normalized into the allowed input range  $[-1,1]$ . Hence in linear CNN, gray scale images are represented by values in the range  $[-1,1]$ ; e.g. black by  $-1$  and white by  $+1$ , and gray scales in between. Provided that the network parameters are such that the linear CNN is completely stable, after the transient has decayed to zero the state will settle to an equilibrium point.

### 2.1. Discrete Time CNN Model

In order to discretize the normalized time differential equation of a CNN cell by sampling, let us define  $t = nT$  then for neighborhood radius  $r$  we obtain:

$$\frac{dx_{ij}(nT)}{d(nT)} = -x_{ij}(nT) + \sum_{k,l=-r}^r a_{k,l} y_{i+k,j+l}(nT) + \sum_{k,l=-r}^r b_{k,l} u_{i+k,j+l} + I \quad (4)$$

where  $T$  is a constant time-step. After choosing a numerical integration method, the resulting discrete time system can be implemented in the time domain through difference equations. If Euler-forward approximation formula is used  $\frac{dx_{ij}(nT)}{d(nT)} \cong \frac{x_{ij}((n+1)T) - x_{ij}(nT)}{T}$  then

substituting  $x(n) = x(nT)$ ,  $y(n) = y(nT)$  gives the discrete CNN model as

$$x_{ij}(n+1) = (1-T)x_{ij}(n) + T \left[ \sum_{k,l=-r}^r (a_{k,l} y_{i+k,j+l}(n) + b_{k,l} u_{i+k,j+l}) + I \right] \quad (5)$$

Recall that Eq. (5) is space invariant and for each pair  $(i, j)$ ,  $i = 1, \dots, M$ ,  $j = 1, \dots, N$ , it defines a system of  $M \times N$  difference equations.

## 3. CNN Based Fingerprint Preprocessing

A fingerprint image enhancement algorithm receives a poor quality fingerprint image, applies a set of processes to the input fingerprint and outputs the improved quality fingerprint image. Fingerprints have been enhanced by using the method proposed in [13]. According to the proposed method, the preprocessing stage processes the fingerprint images segment by segment and consists of four substages: contrast stretching, CNN Gabor-type filtering, lowpass filtering and grayscale to binary thresholding. This section briefly describes these preprocessing steps.

### 3.1. Contrast Stretching

An image histogram is a chart that shows the relative distribution of intensities in an image. Contrast stretching maps a section of the image intensities to the full output intensity range (for CNN  $[-1,1]$ ) linearly. By using this operation, low contrast images can be improved [13].

### 3.2. Gabor-type Filtering

In a gray level and small fingerprint image segment, ridges and valleys form a slowly varying sinusoidal-like shaped plane wave which has a well-defined spatial frequency and orientation. Therefore, an orientation selective bandpass filter tuned to the corresponding local spatial frequency and orientation can effectively remove the undesired noise and preserve the true ridge and valley structures. As bandpass filters, Gabor filters have both frequency and orientation selective properties and have optimal joint resolution in both spatial and frequency domains [14]. Hence, they are an appropriate choice for processing of local structures. In this study, CNN Gabor-type filters [15] was used as the orientation selective bandpass filters.

### 3.3. Lowpass Filtering

Due to the effect of noise, the most enhanced image obtained by CNN Gabor-type filtering does not always accurately represent the ridge features. Therefore, after filtering a spatial smoothing operation on the whole image is necessary. Since ridge orientation varies slowly compared to the noise on the image, a lowpass filter should be used to modify the incorrect ridges. The CNN templates given in [16] was used as the lowpass filter.

### 3.4. Gray-Scale to Binary Thresholding

The feature extraction method proposed for fingerprint images are based on image binarization. Therefore, after spatial smoothing, grayscale to binary thresholding is applied to the enhanced grayscale fingerprint image by using the CNN templates given in [17].

## 4. CNN Based Fingerprint Feature Extraction

This section describes the proposed fingerprint feature extraction procedure, which is based on core point detection, ridgeline thinning, minutiae point extraction and minutiae point angle calculation.

### 4.1. Core Point Detection

Finding a reference point is very important in advanced matching algorithms because it can be used as the location of origin for marking minutiae points. In this work, core point was used as a reference point and was found by porting the algorithm defined in [18] to the CNN framework. Core point is a global structure (i.e. overall pattern of the ridges and valleys) of a fingerprint image and has special symmetry properties. It can be found by its strong response to complex filters designed for rotational symmetry extraction [18]. The method used is a complex filtering approach. Complex filters, applied to the orientation field in multiple resolutions, are used to detect rotational symmetry. Although these filters can detect arch-type and delta-type symmetries, only the arch-type symmetry is utilized in this work. Moreover, this complex filtering approach was implemented in the CNN framework. The complex filter used to find the core point is given as in Eq. (6).

$$h = (x + iy)g(x, y) = re^{i\phi}g(x, y) \quad (6)$$

where  $g(x, y)$  is a Gaussian function. In this work, there is a modification to the original theory, instead of Gaussian filter, a CNN implementable Gaussian-like filter is used. The CNN templates for Gaussian-like lowpass filter is given as in Eq. (7). By using these templates, the CNN templates corresponding to the complex filter used to find the core point can be obtained as in Eq. (8)

$$A = \begin{bmatrix} 0 & \gamma^2 & 0 \\ \gamma^2 & -(1+4\gamma^2) & \gamma^2 \\ 0 & \gamma^2 & 0 \end{bmatrix}, \quad (7)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0$$

$$\tilde{A} = (x + iy)^* \begin{bmatrix} 0 & \gamma^2 & 0 \\ \gamma^2 & -(1+4\gamma^2) & \gamma^2 \\ 0 & \gamma^2 & 0 \end{bmatrix}, \quad (8)$$

$$\tilde{B} = (x + iy)^* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0$$

where "\*" is the convolution operator. Complex filters are not applied directly to the original fingerprint image. Instead, they are applied to complex images, i.e. the complex valued orientation tensor field image given in Eq. (9) in different scales

$$z(x, y) = (f_x + i f_y)^2 \quad (9)$$

where  $f_x$  is the derivative of the original image in the  $x$  direction and  $f_y$  is the derivative in the  $y$  direction. With the help of the Sobel operator, CNN templates for  $f_x$  and  $f_y$  can be found as in Eq. (10):

$$f_x : A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad (10)$$

$$I = 0$$

$$f_y : A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad (10)$$

$$I = 0$$

The complex filter response is  $c = \mu e^{i\alpha}$ , where  $\mu$  is a measure of symmetry and  $\alpha$  is the orientation of the symmetric pattern. In the core type symmetry point, the filter gives a strong response. Then, multiscale filtering is used to extract core point more robustly and precisely. Therefore, for the fingerprint shown in Figure 1, the complex orientation field  $z(x, y)$  is represented by a four level CNN Gaussian-like pyramid as shown in Figure 2. Level 3 has the lowest, and level 0 has the highest resolution. In the multiscale filtering, only the angle of the complex orientation field is used, i.e. the magnitude is set to one in  $z(x, y)$ . Then, the core point is found for each resolution since the pattern of the orientation field around

core point is the same at different resolutions. The CNN complex filter response at different resolution levels is denoted by  $c_k$  where  $k = 3, 2, 1$  and  $0$  are the resolution levels. Figure 3 shows the magnitude of the filter responses of filter  $h$  at levels  $k = 0, 1, 2, 3$ .



Figure 1. Original Fingerprint Image

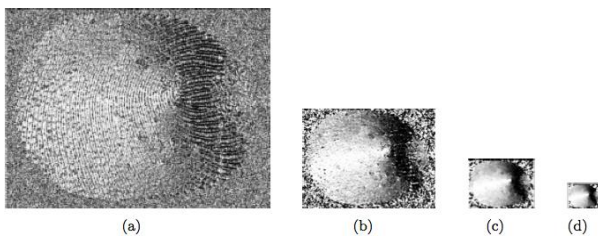


Figure 2. Different levels of CNN Gaussian-like pyramid (a) Level 0, (b) Level 1, (c) Level 2, and (d) Level 3

Additionally, a point tracking algorithm given in [18] is used to find the position of a possible core point in a fingerprint image. Algorithm starts by finding the maximum filter response in Level 3. To get even further precision in the localization of the maximum, a new search is done in lower levels of the pyramid i.e. in Level 2, Level 1 and Level 0. This point tracking algorithm is terminated in a point computed in the highest level (highest resolution) Level 0. Finally, coordinates of the point found in Level 0 corresponds to the core point coordinates. Results of the core point detection algorithm are shown in Figure 4.

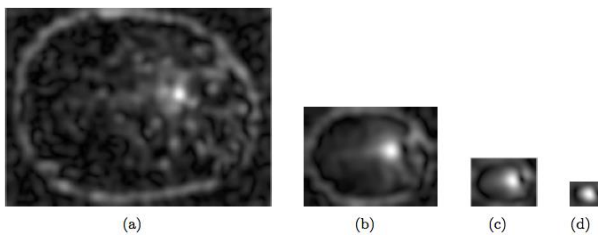


Figure 3. Magnitude of the CNN Complex filter responses for (a) Level 0, (b) Level 1, (c) Level 2, and (d) Level 3

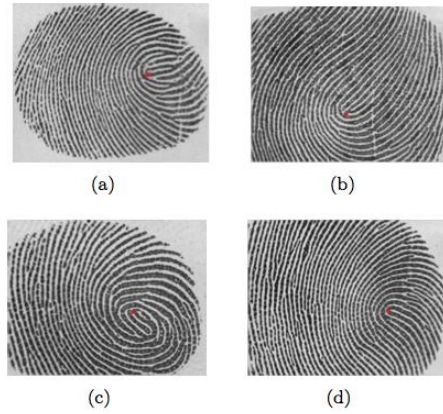


Figure 4. Results of the core point detection algorithm

### 4.2. Ridgeline Thinning

Bifurcations and ridgeline endings are the local structures of the fingerprint images. In order to extract these features, each ridgeline should be made one pixel thick. There are many methods for thinning. In this section, skeletonization method given in [17] is used. The method uses eight different CNN templates to thin the ridgelines and the result is shown in Figure 5(b).

The eight templates thin a fingerprint image in the eight directions (North, Northeast, East, Southeast, South, Southwest, West and Northwest). The thinned ridgelines can now be effectively used to extract the feature points before the feature matching process.



Figure 5. (a) The enhanced image, and (b) the result of the used skeletonization method

### 4.3. Minutiae Point Extraction

As mentioned before, the individuality of the fingerprint images is due to the arrangement of the bifurcation and ending points and their angles. Once the ridgelines are thinned, these features can be extracted. Bifurcation points have been extracted by using the method proposed in [2].

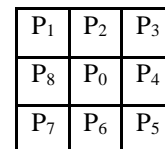


Figure 6. Neighbors of pixel  $P_0$

As shown in Figure 6,  $P_0$  is the pixel that we want to analyze. The point  $P_0$  can only be a bifurcation point, if its value is -1. Here, -1 represents the black pixels, and 1

represents the white pixels. Then, the bifurcation points can be extracted by using the function given in Eq. (11):

$$\sum_{k=1}^8 |P(k+1) - P(k)| = 12, \quad P(9) = P(1) \quad (11)$$

After applying the function in a  $3 \times 3$  neighborhood, if the result is 12, this means that the point  $P_0$  is a bifurcation point. As an example, assume that the values of  $P_3$ ,  $P_5$ , and  $P_8$  are -1, and the others are 1. Application of the function gives a result equal to 12. It is possible to implement this function in the CNN framework. The CNN templates for the function can be given as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_8 & U_{ij} & b_4 \\ b_7 & b_6 & b_5 \end{bmatrix}, \quad (12)$$

$$I = -12; \quad b_i = |b_{i+1} - b_i|, \quad i = 1, \dots, 8$$

In the light of the theory given for bifurcation point extraction above, a new function is proposed to extract the ending points. The function can be given by Eq. (13).

$$\sum_{k=1}^8 |P(0) - P(k)| = 14 \quad (13)$$

After applying the function, the result will be 14, if the point  $P_0$  is an ending point. The point  $P_0$  can only be an ending point, if its value is -1. As an example, assume that only one of the pixels except  $P_0$  in the  $3 \times 3$  neighborhood has a value of -1 and the others is 1. Application of the function gives a result equal to 14. Similar to the previous case, the CNN templates for this operation is proposed as in Eq. (14).

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_8 & U_{ij} & b_4 \\ b_7 & b_6 & b_5 \end{bmatrix}, \quad (14)$$

$$I = -14; \quad b_i = |U_{ij} - b_i|, \quad i = 1, \dots, 8$$

#### 4.4. Calculation of Minutiae Point Angles

Angles of the ending points are calculated over a  $13 \times 13$  pixels window located around the ending point as shown in Figure 7.

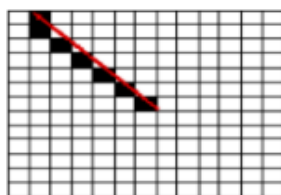


Figure 7.  $13 \times 13$  pixels window located around the ending point in the thinned fingerprint

As can be seen from the figure, the ending point is located to the center of the window. Undesired groups of pixels outside the fingerprint ridge line under consideration are removed from the window. After removal of unwanted pixels, a line is assumed between the window center (ridge ending point) and the closest pixel to the window border. Angle between the horizontal axis and the straight line connecting these two points is used as the ending point angle. This angle is calculated by using the slope of the straight line.

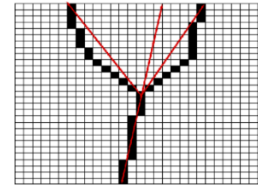


Figure 8.  $29 \times 29$  pixels window located around the bifurcation point in the thinned fingerprint

Similarly, bifurcation point angles are calculated over a  $29 \times 29$  pixels window located around the bifurcation point as shown in Figure 8. Again, the bifurcation point is located to the center of the window. Undesired groups of pixels outside the fingerprint ridge lines forming the bifurcation point are removed from the window. Then, three straight lines are formed between the window center (bifurcation point) and the closest pixels to the window border. The three angles between the horizontal axis and the three straight lines are calculated by using the slope of these lines. After that, angles between the straight lines are calculated and the bifurcation point angle is set as the smallest angle between three line segments. The result of ridgeline thinning is shown in Figure 9(a) and the extracted feature points and their corresponding angles to be used in the feature matching process are shown in Figure 9(b).

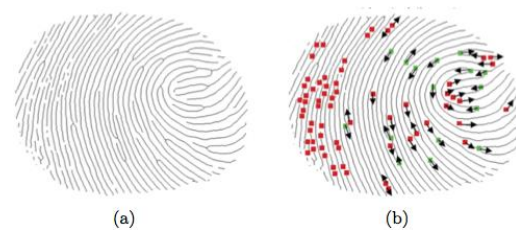


Figure 9. (a) The result of the skeletonization, and (b) the extracted feature points (ending points in red and bifurcation points in green) and their angles

#### 5. False Feature Elimination

False minutiae points will significantly decrease the accuracy of matching if they are simply regarded as genuine minutiae. Therefore, in order for removing false minutiae points, some mechanisms are essential to increase the fingerprint verification system performance. Four types of false minutiae are specified in the literature [19] is shown in Figure 10. Case m1 is a spike piercing into a valley, m2 is a spike falsely connects two ridges, m3 has two near bifurcations located in the same ridge and m4 has

two ridge ending points and they have nearly the same orientation and short distance. The procedure for removing the false minutiae points can be found in [19].

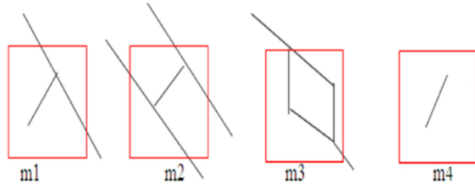


Figure 10. False Minutiae Points

The extracted feature points and their angles are shown in Figure 11(a) and the feature points and their angles obtained after the false minutiae points removal procedure are shown in Figure 11(b).

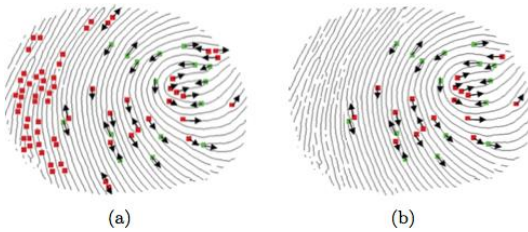


Figure 11. Extracted feature points and their angles, (a) before and (b) after the false minutiae points removal procedure

### 6. Matching

Fingerprint feature points are defined with their coordinates, angles and types. The most popular minutiae point matching algorithms prefer to use only the coordinates and angles of the points [20]. In this work, every feature point is represented with a vector  $\mu = \{x, y, \theta\}$  where  $x$  and  $y$  are the coordinates and  $\theta$  is the angle of the feature point. If  $Q$  and  $I$  denote the features of the unknown fingerprint and the fingerprint in the database, respectively, all the feature points can be represented mathematically by Eq. (15).

$$Q = \{\mu'_1, \mu'_2, \dots, \mu'_m\}, \quad \mu'_j = \{x'_j, y'_j, \theta'_j\}, \quad j = 1 \dots m$$

$$I = \{\mu_1, \mu_2, \dots, \mu_n\}, \quad \mu_i = \{x_i, y_i, \theta_i\}, \quad i = 1 \dots n$$
(15)

where  $m$  and  $n$  denote the number of extracted features from the fingerprints  $Q$  and  $I$ , respectively. Figure 12 shows the coordinates and angle values of ending and bifurcation points. Matching the two fingerprint image by solely using these features is not possible. During the fingerprint image acquisition, a fingerprint image can be translated as much as  $\Delta x$  and  $\Delta y$  in the  $x$  and  $y$  directions and can be rotated with respect to the original fingerprint by an angle equal to  $\Delta \theta$ . In order to match the two fingerprints,

$\Delta x$ ,  $\Delta y$  and  $\Delta \theta$  parameters should be found or at least should be compensated before matching.

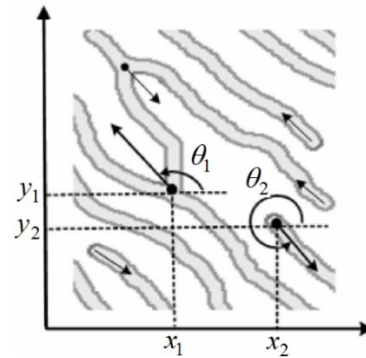


Figure 12. Coordinate and angle values of feature points

A rotation invariant matching is possible if a reference point like core point is used. In this approach, new features are defined by using the Euclidean distance between the core point and the extracted feature point and also the relative angle between the feature point and the line connecting the feature point to the core point.

After the feature points are found, each feature point is expressed in polar coordinate system and the relative angle between the feature point and the line connecting the feature point to the core point is found as in Eq. (16).

$$Q = \{\mu'_1, \mu'_2, \dots, \mu'_m\}, \quad \mu'_j = \{r'_j, \phi'_j\}, \quad j = 1 \dots m$$

$$I = \{\mu_1, \mu_2, \dots, \mu_n\}, \quad \mu_i = \{r_i, \phi_i\}, \quad i = 1 \dots n$$
(16)

In these equations,  $r$  and  $\phi$  describe the Euclidean distance between the reference point and feature point, and the relative angle between the feature point and the line connecting the feature point to the core point as shown in Figure 13, respectively.

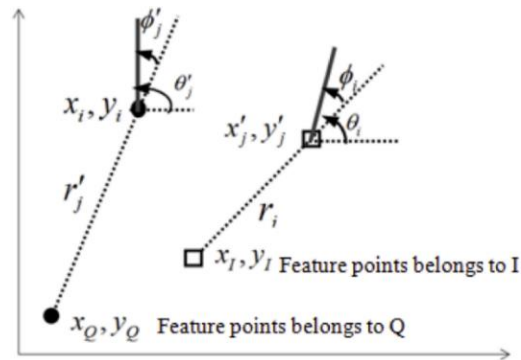


Figure 13. Demonstration of feature points in polar coordinates

These points can be obtained from Eqs. (17) and (18).

$$r_i = \sqrt{(x_i - x_I)^2 + (y_i - y_I)^2},$$

$$r'_j = \sqrt{(x'_j - x_Q)^2 + (y'_j - y_Q)^2}$$
(17)

$$\phi_i = \tan^{-1} \left( \frac{y_i - y_I}{x_i - x_I} \right) - \theta_i, \quad (18)$$

$$\phi'_j = \tan^{-1} \left( \frac{y'_j - y_Q}{x'_j - x_Q} \right) - \theta'_j$$

where  $(x_Q, y_Q)$  and  $(x_I, y_I)$  denote the coordinates of the feature points in the fingerprints  $Q$  and  $I$ , respectively. Calculation of the matching score of two fingerprint images can be done by using the matching function  $\Psi$  given in Eq. (19). This function returns 1 for two feature points in  $Q$  and  $I$ , if difference between the distances and angles are smaller than given tolerance values  $r_0$  and  $\phi_0$ , otherwise 0. These tolerance values are used to compensate the changes occurring in the  $\Delta x$ ,  $\Delta y$  and  $\Delta \theta$  parameters.

$$\Psi = \begin{cases} \mu_i, \mu'_j \\ 1 & |r_i - r'_j| \leq r_0 \text{ and } |\phi_i - \phi'_j| \leq \phi_0 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

For each feature pair in  $Q$  and  $I$  that satisfies the given condition above, the function returns 1, and the matching counter  $N_p$  is increased by one. The matching score between two fingerprint images is calculated by using the equation given in Eq. (20).

$$\text{Matchingscore} = \frac{100 \times N_p}{\max \{M, N\}} \quad (20)$$

where  $M$ ,  $N$  and  $N_p$  denote the number of extracted features in  $Q$  and  $I$  fingerprint images, and the number of matched features, respectively.

## 7. System Performance

In the biometric community, some widely accepted methodologies and protocols exist for testing and reporting the performance of a biometric system [22, 23, 24, 25]. To assess the performance of the proposed system, The Hong Kong Polytechnic University (PolyU) High-Resolution-Fingerprint (HRF) Database II [21] has been used. PolyU HRF DBII contains 1480 fingerprint images of 148 fingers collected in two sessions separated by two weeks. Each session has five sample images per finger. Resolution of each image is 1200 dpi and size of the images is 640×480 pixels. In order to evaluate the performance of the proposed system, the following experiments have been performed on the first session of the database:

**Genuine matching:** Each fingerprint in the database is matched against the remaining samples of the same fingerprint to compute the FNMR (also referred as False Rejection Rate - FRR). Hence, the total number

of genuine matching attempts can be given as:  $((5 \times 4)/2) \times 148 = 1,480$ .

**Imposter matching:** The first sample of each fingerprint in the database is matched against the first sample of the remaining fingerprints in the database to compute the FMR (also referred as False Acceptance Rate - FAR). Hence, the total number of impostor matching attempts can be given as:  $((148 \times 147)/2) = 10,878$ . In these matchings, if image  $f$  is matched to  $g$ , the symmetric match (i.e.,  $g$  against  $f$ ) is not executed to avoid correlation in the scores [24].

In the experiments, FMR, FNMR and ROC curves are calculated by changing the threshold value to reflect the full system performance for a possible application. There is a trade-off between FNMR and FMR, hence both of them cannot have the smallest possible values at the same time and they are not independent in reality. Both FMR and FNMR depend on the chosen threshold  $T$  and therefore they are a function of  $T$ . A ROC curve plots the FNMR versus FMR and eliminates the graphs dependence on threshold  $T$ . Thus a ROC curve shows performance of a system in different operating points. ROC curves provide for objective comparisons in decision systems. Hence, they can be applied when comparing biometric systems in general and fingerprint recognition systems in particular [20].

Both FMR and FNMR have been plotted versus threshold value as shown in Figure 14 and ROC curve has been drawn on Figure 15. As it can be seen from Figure 14, there is an intersection point called Equal Error Rate (EER) at which both FNMR and FMR values are equal. Also, the operating point of EER can be determined by the intersection of the ROC curve and the straight line where FNMR = FMR. Typically, value of this point is used as the threshold value of the system. But if a high security system is required, the threshold value can be chosen to be less than the EER point. This means that more rejections and less false acceptances will happen. Therefore, the choice of the threshold depends on the application. Consequently, EER value for the proposed system on the given fingerprint database is found as 0.1556.

EER is used to show biometric performance of a system, typically when operating in the verification task. Instead of comparing systems or methods directly, most of the time only the EER measure is used to compare performance of different biometric systems. In general, as the value of EER decreases, accuracy of the biometric system increases. So the EER measure is truly useful in comparing biometric system performance. Therefore, in this work, only one parameter namely EER is used to compare different methods.

The proposed method has been compared with the state-of-the-art minutia-based methods (or only the minutiae based part of the fusion strategies) that use the PolyU HRF DBII database and recognition accuracy has been evaluated according to the EER. Table 1 lists the EERs of the three methods. According to EER listed in Table 1, the proposed method performs better than [27] and worse than [26]. These results are very encouraging for implementing a CNN based rotation invariant fingerprint recognition system.

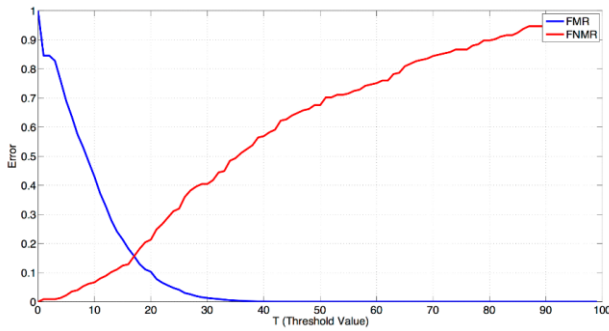


Figure 14. FMR (blue) and FNMR (red) versus threshold

Table 1. EERs of the three methods on the PolyU HRF DBII database

Method	EER of using only minutiae
Proposed method	15.56%
[26]	0.61%
[27]	57.80%

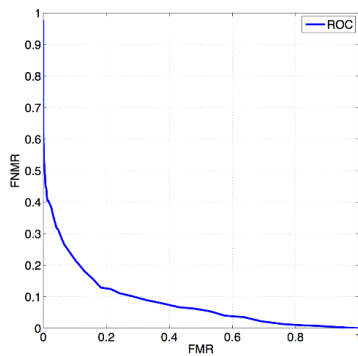


Figure 15. ROC curve on the PolyU HRF DBII database

## 8. Conclusion

In this work, by keeping the hardware implementability in mind, an effort has been put to propose a rotation invariant fingerprint recognition system in the CNN framework. CNNs have an extremely important feature for building fast image processing hardware, that is, they are suitable to implementation in VLSI and FPGA technology. Also, different image processing tasks can be performed easily by changing the template coefficients of the CNN. Thus, implementing the proposed system on CNN is expected to greatly decrease the computational time.

In the literature to date, there is no rotation invariant CNN based fingerprint recognition system. Moreover, ending point detection and core point detection are implemented in the CNN framework for the first time. The performance of the proposed system has been assessed by using the high resolution PolyU HRF DBII database. Furthermore, the system performance is analyzed by the help of FMR, FNMR and ROC curves for possible trade-off analysis in real life applications. The results are very encouraging for implementing a

CNN based fully automatic rotation invariant fingerprint recognition system.

Future work will include the use of Discrete-Time CNN framework to implement proposed system on a FPGA based platform and some performance adjustments. Moreover, in the continuation of the work, level 3 features (e.g. pores) are planned to be used for recognition. These level 3 features can only be reached on a high resolution database. Therefore, the proposed method was developed and tuned for high resolution fingerprint databases.

## 9. References

- [1] Q. Gao, S. Moschytz, "Fingerprint Feature Extraction Using CNNs", in Proceedings of European Conference on Circuit Theory and Design 2001, Espoo, Finland, 2001, pp. 28-31.
- [2] T. Su, Y. Du, Y. Cheng, Y. Su, "Fingerprint Recognition System Using Cellular Neural Network", in Proceedings of 9th International Workshop on Cellular Neural Networks and their Applications, Hsinchu, Taiwan, 2005, pp. 170-173.
- [3] I. Kale, R. Abrishambaf, H. Demirel, "A Fully CNN Based Fingerprint Recognition System", in Proceedings of 11th International Workshop on Cellular Neural Networks and Their Applications, Santiago de Compostela, Spain, 2008, pp. 14-16.
- [4] L. O. Chua, L. Yang, "Cellular neural networks: Theory", IEEE T CIRCUITS SYST, vol. 35, pp. 1257-1272, 1988.
- [5] M. D. Doan, M. Glenser, R. Chakrabaty, M. Heidenreich, S. Cheung, "Realization of a Digital Cellular Neural Network for Image Processing", in Proceedings of Third International Workshop on Cellular Neural Networks and Their Applications, Rome, Italy, 1994, pp. 85-90.
- [6] E. Saatci, "Image Processing Using Cellular Neural Networks", PhD Thesis, London South Bank University, London, UK, 2003.
- [7] L. O. Chua, L. Yang, "Cellular Neural Networks: Applications", IEEE T CIRCUITS SYST, vol. 35, pp. 1273-1290, 1988.
- [8] K. R. Crounce, L. O. Chua, "Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial", IEEE T CIRCUITS-I, vol. 42, pp. 583-601, 1995.
- [9] Z. Nagy, P. Szolgay, "Configurable multilayer CNN-UM emulator on FPGA", IEEE T CIRCUITS-I, vol. 50, pp. 774-778, 2003.
- [10] J. Javier Martinez, F. Javier Toledo, E. Fernandez, J. M. Ferrandez, "A retinomorph architecture based on discretetime cellular neural networks using reconfigurable computing", NEUROCOMPUTING, vol. 71, pp. 766-775, 2008.
- [11] K. Kayaer, V. Tavsanoğlu, "A new approach to emulate CNN on FPGAs for real time video processing", in Proceedings of 11th International Workshop on Cellular Neural Networks and Their Applications, Santiago de Compostela, Spain, 2008, pp. 23-28.
- [12] N. Yildiz, E. Cesur, K. Kayaer, V. Tavsanoğlu, M. Alpay, "Architecture of a Fully Pipelined Real-Time Cellular Neural Network Emulator", IEEE T CIRCUITS-I, vol. 62, pp. 130-138, 2015.
- [13] E. Saatci, V. Tavsanoğlu, "Fingerprint Image Enhancement Using CNN Filtering Techniques", INT J NEURAL SYST, vol. 13, pp. 453-460, 2003.
- [14] J. G. Daughman, "Uncertainty Relation for Resolution in Space, Spatial-Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters", J OPT SOC AM, vol. 2, pp. 1160-1169, 1985.



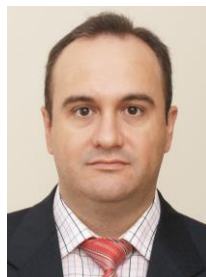
- [15] B. E. Shi, "Gabor-type filtering in space and time with Cellular Neural Networks", IEEE T CIRCUITS-I, vol. 45, pp. 121-132, 1998.
- [16] C. Rekeczky, "Dynamic spatio-temporal nonlinear filtering and detection on CNN architecture - theory, modeling and applications", PhD Thesis, Computer and Automation Institute Hungarian Academy of Sciences, Hungary, 1999.
- [17] K. Karacs, G. Cserey, A. Zarandy, P. Szolgay, C. Rekeczky, L. Kek, V. Szabo, G. Paziienza, T. Roska, "Software Library for Cellular Wave Computing Engines Version 3.1", Cellular Sensory and Wave Computing Laboratory of the Computer and Automation Research Inst., Hungarian Academy of Sciences and the Jedlik Laboratories of the Pazmany P. Catholic University, Budapest, Hungary, 2010.
- [18] K. Nilsson, J. Bigun, "Localization of corresponding points in fingerprints by complex filtering", PATTERN RECOGN LETT, vol. 24, pp. 2135-2144, 2003.
- [19] N. K. Jalutharia, "Fingerprint Recognition and Analysis", MSc Thesis, Thapar University, India, 2010.
- [20] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, "Handbook of fingerprint recognition", Second Edition, Springer, New York, USA, 2009.
- [21] The PolyU HRF Database II, Available at: [http://www4.comp.polyu.edu.hk/~biometrics/HRF/HRF\\_old.htm](http://www4.comp.polyu.edu.hk/~biometrics/HRF/HRF_old.htm) Accessed 10 May 2017.
- [22] P. J. Phillips, A. Martin, C. L. Wilson, M. Przybocky, "An introduction to evaluating biometric systems", COMPUTER, vol. 33, pp. 56-63, 2000.
- [23] UK Government's Biometrics Working Group, "Best practices in testing and reporting performance of biometric devices", v2.01, 2002.
- [24] D. Maio, D. Maltoni, R. Capelli, J. L. Wayman, A. K. Jain, "Fvc2000: Fingerprint verification competition", IEEE T PATTERN ANAL, vol. 24, pp. 402-412, 2002.
- [25] R. Capelli, D. Maio, D. Maltoni, J. L. Wayman, A. K. Jain, "Performance evaluation of fingerprint verification systems", IEEE T PATTERN ANAL, vol. 28, pp. 3-18, 2006.
- [26] Q. Zhao, D. Zhang, L. Zhang, N. Luo, "Adaptive fingerprint pore modeling and extraction", PATTERN RECOGN, vol. 43, pp. 2833-2844, 2010.
- [27] N. A. Mngenge, "An Adaptive Quality-Based Fingerprints Matching Using Feature Level 2 (Minutiae) and Extended Features (Pores)", M.Eng Degree Thesis, University of Johannesburg, South Africa, 2013.



is a PhD student at University of New Hampshire.

**Tuba Çelik Mayadağlı** received the B.S. degree from Uludag University and M.S. degree from Osmangazi University in 2006 and 2013, respectively.

She worked at Viko Electric Electronic A.S as a R&D engineer. Her research interests are image and signal processing, cellular neural networks, and embedded systems. Currently, she



cellular neural networks.

**Ertuğrul Saatçı** received the B.S. and M.S. degree from Istanbul University, Istanbul, Turkey, in 1993 and 1996, respectively, and Ph.D. degree from London South Bank University in 2003. He is currently an Assistant Professor at Department of Electronic Engineering, Istanbul Kultur University. His major research interests lie in the areas of signal and image processing, biometrics, and



processing, and embedded systems.

**Rifat Edizkan** received the B.S. and M.S. degree from Anadolu University, Eskisehir, Turkey, in 1987 and 1990, respectively, and Ph.D. degree from Eskisehir Osmangazi University in 1999. He is currently a Professor Doctor at Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University. His research interests are pattern recognition, image and signal

