Fingerprint Feature Extraction Using CNNs

Qun Gao∗ and George S. Moschytz∗

Abstract — Feature Extraction is an important step in fingerprint-based recognition systems. In this paper, a CNN Fingerprint Feature Extraction Algorithm is presented. It is applied to thinned fingerprints which have been previously obtained from real gray-scale, noisy fingerprints in the Image-Preprocessing stage, also by using CNNs. Examples are given to demonstrate the functionality of the proposed algorithm.

1 Introduction

Cellular neural networks (CNNs) [1, 2] have demonstrated their usefulness in fingerprint-based personal identification for civilian applications, especially for portable applications [3–5]. We have previously successfully accomplished the Image-Preprocessing stage in fingerprint recognition by means of CNNs [3]. This has encouraged us to tackle the next stage: Feature Extraction. The successful results are described in this paper.

2 Fingerprint Recognition

As shown in Fig. 1, a typical fingerprint-based recognition system [6, 7] operates in two distinct modes: enrollment and recognition. The purpose of the enrollment mode is to create a database. During this mode, an enrollee fingerprint is captured and processed in three stages: Fingerprint Reading, Image Preprocessing, and Feature Extraction. After the Feature-Extraction stage, a set of representative features of the enrollee fingerprint (the minutia template) is stored in the database. During the recognition mode, a fingerprint to be recognized undergoes the same three processing steps as in the enrollment mode. The result, a test minutia template, is compared with a minutia template from the database in the Feature-Matching stage. A match score which measures the degree of the similarity between the two minutia templates is calculated. Higher values indicate higher confidence in a match. In short, the main problem in fingerprint recognition is to decide how similar two fingerprints, i.e., their minutia templates, are.

In this paper, we concentrate on the Feature-Extraction unit.

3 Fingerprint Feature Extraction

The goal of the Feature-Extraction unit is to extract distinguishable features in fingerprints, as well as their attributes, in order to guarantee the next unit: Feature Matching.

Figure 1: Block diagram of a typical fingerprint-based recognition system

There are two main features in a fingerprint image: endings and bifurcations, as shown in Fig. 2. In a thinned fingerprint, an ending is the end point of a line, while a bifurcation is the junction point of three lines. For matching purposes, these so-called “minutiae” are usually denoted by their type, their location, and the direction of the adjacent ridge.

Figure 2: Fingerprint features: endings and bifurcations

Due to noisy original images and artifacts produced in the Image-Preprocessing stage, spurious minutiae will always be present. Therefore, after the minutiae are extracted, an important step, minutia reduction, is performed to eliminate the false minutiae in the Feature-Extraction stage. The spurious minutiae are normally eliminated by using empirically determined thresholds. In the following section, a CNN Feature-Extraction Algorithm which can accomplish these operations is presented.

Throughout the paper, original, simple continuous-time CNNs [1] are employed. Keeping VLSI implementation of CNN chips in mind, the template neighborhood radius is restricted to 1 and all templates used are spatially invariant, linear, and as sparse as possible. The latter implies the use of several sparse templates rather than a smaller number of less sparse templates, because it has been shown [8] that this improves robustness with regard to the relatively large template tolerances that are inevitable with IC-implementation.

4 CNN Fingerprint Feature-Extraction Algorithm

The proposed algorithm consists of 4 main processing units - Ending Detection, Bifurcation Detection, False Feature Elimination, and Direction Detection - in addition to several simple CNN operations, see Fig. 3. Each
of these 4 processing units accomplishes a specific task, as its name implies, and will be explained separately in detail in the subsequent subsections.

Figure 3: Block diagram of Fingerprint Feature Extraction

First, the endings and bifurcations will be extracted in parallel from the thinned input fingerprint image in the Ending-Detection and Bifurcation-Detection units. As a result, an ending image and a bifurcation image are obtained. These two images are added together in the Logic OR operation [9], and the output fed to the False Feature-Elimination unit. Here, false feature pairs, i.e., ending and bifurcation pairs resulting from so-called spurs [7] can be deleted, based on an empirically determined threshold, see Section 4.3.

With the subsequent Logic AND [9] we obtain a new ending/bifurcation image in which false ending/bifurcation pairs resulting from spurs have disappeared. In the following two False Feature-Elimination operations, false endings and false bifurcations are eliminated separately. The corresponding thresholds are presently determined empirically, see Section 4.3. The remaining endings and bifurcations are regarded as real features and result in an ending feature image and a bifurcation feature image.

Finally, the direction of ridges leaving endings and bifurcations is extracted separately in the Direction-Detection units.

Thus, with our Fingerprint Feature-Extraction Algorithm, we obtain four feature images from a thinned fingerprint input: ending image, ending direction image, bifurcation image and bifurcation direction image. [Note that, unlike with digital methods, we do not have the location of the features.] These four feature images are then used as a basis for comparison in the next stage, Feature Matching. In what follows, these four steps are briefly described separately.

4.1 Ending-Detection Algorithm

Ridge endings are the termination points of lines in a thinned fingerprint image. This implies that an ending point has only one black neighbour in its nearest $3 \times 3$ neighbourhood, see Fig. 4 a). Thus, we can find these points by first eliminating all isolated black points, and then finding all black points that have at most one black neighbour, see Fig. 5. The template for the first operation is given by

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma & \gamma & \gamma \\ \gamma & 8\gamma & \gamma \\ \gamma & \gamma & \gamma \end{pmatrix}, \quad I = \gamma, \quad (1)$$

and the second by

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma & \gamma \\ \gamma & 6\gamma & \gamma \\ \gamma & \gamma \end{pmatrix}, \quad I = 3\gamma, \quad (2)$$

where $\gamma$ is a scaling factor. Higher $\gamma$ accelerates the convergence. An example is shown in Fig. 4.

Figure 4: Example of Ending Detection: a) input, b) after Isolated Point Elimination, c) output (Ending image)

4.2 Bifurcation-Detection Algorithm

Bifurcations are the junction points of three lines in a thinned fingerprint image. Thus, a bifurcation has only 3 black neighbours in its nearest $3 \times 3$ neighbourhood. But not all points which have only 3 black neighbours are bifurcations, see Fig. 6 a). This is taken into account by our algorithm (see Fig. 7).

Figure 6: Example of Bifurcation Detection: a) input, b) after Junction Point Extraction, c) after Isolated Point Detection, d) after Junction Point Extraction in T- and Corner-form, e) output

Figure 7: Block diagram of the Fingerprint Bifurcation-Detection Algorithm

Referring to Fig. 7, in “Junction Point Extraction”, the black pixels which have at least 3 black neighbours in their nearest neighbourhood are extracted. The corresponding template set is given by

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma & \gamma & \gamma \\ \gamma & 6\gamma & \gamma \\ \gamma & \gamma \end{pmatrix}, \quad I = 3\gamma, \quad (3)$$

However, this will still leave T- and Corner-forms (see Fig. 6b) from which we have to extract real bifurcations. For this, the resulting image goes through two different operations in parallel: Isolated Point Extraction [9] and Junction Point Extraction in T- and Corner-form.
The first extracts black pixels which are alone in an input image, the second retains only black pixels which have at least 2 black neighbours among their north, east, south, and west neighbours. The corresponding template set is given by

\[
A = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix},
B = \begin{pmatrix}
0 & \gamma & 0 \\
\gamma & 4\gamma & \gamma \\
0 & \gamma & 0
\end{pmatrix},
I = 3\gamma.
\] (4)

The results of these operations can be seen in Fig. 6 c) and d). Finally, we add the resulting images to obtain only the bifurcations, see Fig. 6 e).

4.3 False Feature-Elimination Algorithm

For the reasons mentioned in Section 3, false minutiae - endings and bifurcations - are not always avoidable after Ending Detection and Bifurcation Detection. It is necessary to eliminate them in order to facilitate the comparison of two fingerprints in the Feature-Matching stage.

Experience shows that two false minutiae - two false endings, or two false bifurcations, or one false ending and one false bifurcation - are normally close to each other [7]. As a consequence, we have developed a CNN False Feature-Elimination algorithm that can eliminate two black points with a distance smaller than or equal to \( n \) pixels. Here distance means the number of white pixels separating two black pixels. Presently, we have determined \( n \) empirically for different cases. To delete two false endings, \( n \) (referred to as \( n_e \)) should be smaller than the minimum of the distance between two real endings. The same holds for two false bifurcations (\( n_h \)), and for a pair consisting of one false ending and one false bifurcation (\( n_{eh} \)). In addition, we require that \( n_s < \min(n_e, n_h) \). Ultimately, all thresholds will need to be fixed for a general-purpose CNN-based fingerprint recognition system, unless an adaptive learning process can be found to find the optimum thresholds.

To find the templates (1) (7), we applied the exact, analytical design approach in [10].

\[
A = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix},
B = \begin{pmatrix}
0 & \gamma & \gamma \\
\gamma & \gamma & \gamma \\
0 & \gamma & \gamma
\end{pmatrix},
I = 8\gamma.
\] (5)

\[
A = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix},
B = \begin{pmatrix}
0 & 0 & \gamma \\
\gamma & 0 & \gamma \\
0 & \gamma & 0
\end{pmatrix},
I = 2\gamma.
\] (6)

\[
A = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix},
B = \begin{pmatrix}
\gamma & 0 & 0 \\
0 & \gamma & 0 \\
0 & 0 & \gamma
\end{pmatrix},
I = 2\gamma.
\] (7)

After this, the two resulting images will be added together in the Logic OR operation, and finally the Isolated Point Extraction operation is applied.

The key point using these two Erosion operations instead of a conventional Erosion operation for all directions is to guarantee that after \( n/2 \) Dilation operations, and \( n/2 \) Erosion operations, two points separated by \( n \) pixels on a diagonal remain connected, so that they can be eliminated in the following operations. An example is shown in Fig. 9 with \( n = 2 \). As can be seen, three pairs of points, whose distance is equal to two, were deleted. Finally, only two points, whose distance to any other point is larger than two, remained.

![Figure 9: Example of False Feature Elimination](image)

To find the template sets (1) (7), we applied the exact, analytical design approach in [10].

4.4 Direction-Detection Algorithm

In digital fingerprint-based recognition systems, the direction of ridges leaving endings and bifurcations is represented by a vector parallel to the ridges leaving minutiae. In a planar CNN system, this cannot be done, because the output of the system is an image.

In order to extract the direction of ridges leaving endings and bifurcations, we use the CNN operation Figure Reconstruction [9]. With this operation, we can reconstruct one or several figures from an input image by using a suitable initial state. If we wish to reconstruct only part of a figure, we must stop this CNN operation after an appropriate number, \( \tau \), of CNN integration steps.
With a suitable \( \tau \), this CNN operation can restore a short section of the ridge leaving an ending, and a short section of each of the three ridges leaving a bifurcation.

Fig. 10 shows the result of the Figure-Reconstruction operation applied to the first images in Fig. 4 and Fig. 6 by using these two images as the input of the operation, and the last images in Fig. 4 and Fig. 6 as the initial state of the operation, respectively. In this example, we chose \( \tau \) equal to 12. These short sections of the ridges leaving feature points can be regarded as the so-called direction information that we need.

![Figure 10: Example of Direction Detection: a) Ending Directions, b) Bifurcation Directions](image)

**4.5 Examples**

In order to demonstrate the performance of the proposed CNN Fingerprint Feature-Extraction Algorithm, we applied it to a thinned fingerprint input which has been obtained by applying our Fingerprint Image-Preprocessing algorithm [3] onto a gray-scale fingerprint image of size \( 256 \times 256 \), as shown in Fig. 11 a) and b).

Figs. 11 c) - f) illustrate the “ending” image and the “ending direction” image, the “bifurcation” image and the “bifurcation direction” image, corresponding to the region within the dashed line in the thinned fingerprint, respectively. The parameters used in these examples are \( n_s = 2, n_e = n_b = 6, \) and \( \tau = 22 \).

The four feature images obtained will be saved in the database in the enrollment mode, while in the recognition mode they will be used for comparison purposes in the Feature-Matching stage.

![Figure 11: Examples: a) gray-scale fingerprint, b) thinned fingerprint, c) “ending” image, d) “ending direction” image, e) “bifurcation” image, f) “bifurcation direction” image](image)

**5 Conclusions**

We have presented a CNN algorithm which accomplishes the Feature-Extraction stage in fingerprint recognition. It is able to extract almost all genuine endings and bifurcations and their corresponding direction attributes, and to eliminate the spurious endings and bifurcations which may result from the noisy original fingerprint, and the previous processing operations. Together with our Fingerprint Image-Preprocessing algorithm [3], the proposed algorithm will facilitate the next important stage in fingerprint recognition, i.e., the Feature-Matching stage. This latter stage is presently being completed and will be reported on shortly.

**References**


