Deep Learning technologies used for fingerprint feature extraction

Fingerprint recognition is a pattern recognition problem to start with. But it is difficult due to factors like high intra-class variability and high interclass similarity and practical issues including uncooperative data subjects, elastic distortion during scanning, inconsistent moisture conditions, and damaged fingerprints. The core problem here is the extraction of minutiae.

We can consider minutiae extraction as a machine learning problem and thus can be implemented using a deep neural network. Previously, Deep neural networks have been used in various approaches in the domain of fingerprint recognition. A neural network can be, used to increase the efficiency of a comparison algorithm, used to compare minutiae from partially overlapping fingerprints, and used to compare fingerprints based on a constructed n-parameter feature vector.

This paper proposed a convolutional neural network model MENet, for Minutiae Extraction Network using the conjunction of the automated supervised training procedure and the post-processing. MENet has as output two softmax normalized probabilities corresponding to ‘yes’ and ‘no’. The softmax normalization function gives the probability that the central pixel belongs to one of these classes, indicating the predicted presence or absence of a minutiae point, respectively.

The resultant probability map captures an estimate, over the entire fingerprint of minutiae point positions. In order to capture the precise location of minutiae points from this map, a post-processing procedure is followed. The output probability map is smoothed using a 3 × 3median filter. This is thresholded at iteratively lower thresholds in order to extract blobs, the centers of mass of which are the minutiae points.

The orientation of these minutiae is calculated by a standard method of local orientation estimation using the principal axis of variation of the image gradients. Minutiae orientations are crucial for fingerprint comparisons. MENet was constructed with 5 convolutional layers, followed by 2 fully connected layers of 1024 nodes, and finally a softmax output layer. Each convolutional layer consists of32 5×5 filters, and the first two of these layers use pooling. All units apart from the output layer uses ReLU activation functions. The input to MENet was obtained by moving a 30 × 30 sliding windows over the full input fingerprint. TensorFlow was used for the implementation.