Fingerprint Verification with Siamese Networks

Cédric Vachaudez\(^2\), Baiyu Chen\(^1\), Luke Wang\(^1\), Meyesam Madadi\(^4\), Isabelle Guyon\(^2,3\), Sergio Escalera\(^2\), Bernhard Boser\(^1\)


Background

Biometrics play an increasingly important role in security to ensure privacy and identity verification. However, false negatives, in part due to poor image quality when fingers are wet or dirty, and false positives due to the ease of forgery are two prevailing issues. Optical sensors can be compromised with a simple fingerprint photo-copy. Capacitive sensors’ performance decreases when common contaminants are present, e.g. water, lotion, condensation, etc. The recently developed ultrasonic fingerprint reader [3] addresses the shortcomings of current technologies. It permits reading both the epidermis (surface) layer and the dermis (below the surface) layer of the finger, which makes it extremely hard to counterfeit a fingerprint. Furthermore, the technology is insensitive to humidity and dirt on fingers.

Siamese Network Structure

Our research uses “Siamese neural network” [2] to solve fingerprint verification. Proposed initially in the 1990’s for signature verification [2] and concurrently applied to optical fingerprint verification [3], this network structure presents distinctive qualitative advantages and its performances can be boosted with appropriate pre-processing.

Technical Details

Input images are preprocessed using standard signal processing steps:

1. Delimitation of fingerprints and application of enhancement: median filter, contrast enhancement, and other noise reduction;
2. Alignment of the two fingerprints (angle and position);
3. Creation of local orientation, frequency, and variance maps.

The maps thus computed, shown in Figure 2, are provided to the network. Although the convolutional network is, in principle, capable of computing such maps, the pre-processing steps saves some training effort. Two sides of the network are identical and have shared weights. The similarity between the final outputs from each side is calculated using a cosine similarity function. To assess our new method, we have assembled a large database of thousands of optical fingerprints from various public sources. Using a model of the ultrasound sensor, we can transform the optical fingerprints to closely resemble the ultrasound images captured by the fingerprint sensor, of which we have sample images. With a subset of the available data we obtained promising preliminary results.

Results and Discussion

The results of this Siamese network are shown through the progress of “Area Under the Curve” (AUC) of the Receiver Operating Characteristic (ROC). The neural network performs better than the preprocessed maps (although not very significantly yet). Further improvements could be gained by incorporating the pre-processing into the neural network architecture as convolutional trained layers and fine tuning the end-to-end system. In addition, the quality of the dataset we are presently using is not very high, and contains fingerprints often containing marks and smudges. We are working towards obtaining a large amount of simulated fingerprints to perform massive training. Figure 3 shows the activation maps from different inner layers of the network of Figure 1, showing that it is able to perform some form of feature extraction.

Reference:


Figure 1. Structure of the Siamese Neural Network

Figure 2. Raw Image and Preprocessing Output

Figure 3. Activation Maps in the various layers of the network of Figure 1.

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Activation map 1</th>
<th>Activation map 2</th>
<th>Activation map 3</th>
<th>Activation map 4</th>
<th>Activation map 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessed</td>
<td>0.484</td>
<td>0.535</td>
<td>0.707</td>
<td>0.784</td>
<td>0.553</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>0.834</td>
<td>0.754</td>
<td>0.876</td>
<td>0.792</td>
<td>0.626</td>
</tr>
<tr>
<td>Template</td>
<td>0.54</td>
<td>0.52</td>
<td>0.72</td>
<td>0.30</td>
<td>0.58</td>
</tr>
<tr>
<td>Ultrasonic</td>
<td>0.54</td>
<td>0.52</td>
<td>0.72</td>
<td>0.30</td>
<td>0.58</td>
</tr>
<tr>
<td>Random</td>
<td>0.73</td>
<td>0.69</td>
<td>0.72</td>
<td>0.70</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Table 1. Performance comparisons for different preprocessings before and after passing data through the neural network. Cosine is used as similarity function.