Palmprint recognition using HMAX model and Support Vector Machine classifier

, Sara Motamed3, Karim Faez2 Mahboubeh Yaqubi1

1Electrical and Computer Engineering Department, Islamic Azad University, barageen Avenue, myaqubi@gmail.com

2Electrical Engineering Department, Amirkabir university of technology, kfaez@aut.ac.ir

3Electrical and Computer Engineering Department, Islamic Azad University, barageen Avenue, samotamed@yahoo.com

Abstract: Support vector machine (SVM) and HMAX model are two powerful recent techniques. SVMs are classifiers which have demonstrated high generalization capabilities in many different tasks, including the object recognition problem. HMAX is a feature extraction method and this method is motivated by a quantitative model of visual cortex. In this paper we combine these two techniques for the palmprint verification problem. The Hong Kong Polytechnic University (PolyU) palmprint database that this database includes 600 palmprint images from 100 persons, with 6 images per person, is exploited to test our approach. Experimental results using the combination HMAX model and support vector machine (SVM) classifier obtains higher recognition rate than those obtained with HMAX model and k-nearest neighbors (KNN) classifier in identity verification system based on palmprint, and also demonstrated that the HMAX model, compared with PCA method, not only obtains higher recognition rate, but also this method is scale and rotate invariant, whereas PCA method provide high recognition rate only in closely controlled conditions.

Keywords: Biometrics, Palmprint, Feature extraction, HMAX model, support vector machine (SVM).
1 - Introduction

Biometrics makes use of the physiological or behavioral characteristics of people such as fingerprint, iris, face, palm-print, gait, and voice, for personal identification [1], which provides advantages over non-biometric methods such as password, PIN, and ID cards. It is an essential technology for many mission-critical applications, such as homeland security, e-commerce, banking, etc. Palm-print is the unique inner surface pattern of human hand, including a number of discriminating features, such as principal lines, wrinkles, ridges, minutiae points, singular points, texture, etc. Compared with other biometric traits, the advantages of palm-print are the availability of large palm area for feature extraction, easy of capture and high user acceptability. An important issue in palm-print recognition is to extract palmprint features that can discriminate an individual from the other. There are two popular approaches to palm-print recognition. One approach transforms palm-print images into specific transformation domains. Among the works that appear in the literature are eigen-palm [2], Gabor filters [3], Fourier Transform [4], and wavelets [5]. Another approach is to extract principal lines and creases from the palm [6],[9],[16],[17]. However, this method is not easy because it is sometimes difficult to extract the line structures that can discriminate every individual well. Besides, creases and ridges of the palm are always crossing and overlapping each other, which complicates the feature extraction task.

PCA has been widely used for dimensionality reduction in computer vision. Result shows that PCA also performs well in various recognition tasks[14],[15]. Eigen-palms provide high recognition rate only in closely controlled conditions. Indeed, even a slight amount of rotation can cause a significant drop in system performance and in unattended systems rotations occur very frequently.

In this paper, we propose the use of an HMAX model to extract features from palm images. In this method, we introduce a novel set of features for robust palm recognition. Each element of this set is a complex feature obtained by combining position- and scale-tolerant edge detectors over neighboring positions and multiple orientations. This system’s architecture is motivated by a quantitative model of visual cortex [12], which summarizes in a quantitative way what most visual neuroscientists agree on: the first few hundreds milliseconds of visual processing in primate cortex follows a mostly feed-forward hierarchy. At each stage, the receptive fields of neurons (i.e., the part of the visual field that could potentially elicit a neuron’s response) tend to get larger along with the complexity of their optimal stimuli (i.e., the set of stimuli that elicit a neuron’s response). In its simplest version, the standard model consists of four layers of computational units where simple S units, which combine their inputs with Gaussian-like tuning to increase object selectivity, alternate with complex C units, which pool their inputs through a maximum operation, thereby introducing gradual invariance to scale and translation. The model has been able to quantitatively duplicate the generalization properties exhibited by neurons in infero temporal monkey cortex (the so-called view-tuned units) that remain highly selective for particular objects (a face, a hand, a toilet brush) while being invariant to ranges of scales and positions. The model originally used a very simple static dictionary of features (for the recognition of segmented objects) although it was suggested in [12] that features in intermediate layers should instead be learned from visual experience. (see Fig. 1).
The rest of the paper is organized as follows: for an input image, we first compute a set of features learned from the positive training set (see section 2). We then run a standard classifier on the vector of features obtained from the input image (section 3). The resulting approach is simpler than the aforementioned hierarchical approaches: it does not involve scanning over all positions and scales, it uses discriminative methods and it does not explicitly model object geometry. Yet it is able to learn from very few examples. Section 3 compare this approach with PCA method and section 4 conclude the paper.

2- HMAX model

It is important to stress that biology imposes strong constraints on our system architecture: consistent with the standard view in neuroscience, our architecture is feed-forward and does not involve image scanning over all positions and sizes, the standard approach in computer vision. It also limits the basic operations that can be performed by individual units.

Our system is summarized in follow. In This model, the first two layers correspond to primate primary visual cortex, V1, i.e., the first visual cortical stage, which contains simple (S1) and complex (C1) cells [8]. The S1 responses are obtained by applying to the input image a battery of Gabor filters, which can be described by the following equation:

\[
G(x, y) = \exp \left( -\frac{(X^2 + \gamma^2Y^2)}{2\sigma^2} \right) \times \cos \left( \frac{2\pi X}{\lambda} \right)
\]

Where:

\[ X = x \cos \theta + y \sin \theta \]
\[ Y = -x \sin \theta + y \cos \theta \]

and adjusted the filter parameters, i.e., orientation \( \theta \), effective width \( \sigma \), and wavelength \( \lambda \), so that the tuning profiles of S1 units match those of V1 parafoveal simple cells. This was done by first sampling the space of parameters and then generating a large number of filters. We applied those filters to stimuli commonly used to probe V1 neurons [8] (i.e., gratings, bars and edges). After removing filters that were incompatible with biological cells [8], we were left with a final set of 16 filters at 4 orientations [13] (see table 1). The next stage – C1 – corresponds to complex cells which show some tolerance to shift and size: complex cells tend to have larger receptive fields (twice as large as simple cells), respond to oriented bars or edges anywhere within their receptive field [8] (shift invariance) and are in general more broadly tuned to spatial frequency than simple cells [8] (scale invariance). modifying the original Hubel & Wiesel proposal for building complex cells from simple cells through pooling [8], Riesenhuber & Poggio proposed a max-like pooling operation for building position- and scale tolerant C1 units. In the meantime, experimental evidence in favor of the max operation has appeared [18, 19]. Again pooling parameters were set so that C1 units match the tuning properties of complex cells as measured experimentally [13].

Generally, after given an input image, this approach is performed in following steps that this steps are summarized from pervious paragraph:

**S1**: Apply a battery of Gabor filters to the input image. The filters come in 4 orientations \( \theta \) and 16 scales \( s \). Obtain \( 16 \times 4 = 64 \) maps \( (S1)s^\theta \) that are arranged in 8 bands (e.g., band 1 contains filter outputs of size 7 and 9, in all four orientations,
band 2 contains filter outputs of size 11 and 13, etc.

**C1:** For each band, take the max over scales and positions: each band member is sub-sampled by taking the max over a grid with cells of size \( N \times N \) first and the max between the two scale members second, e.g., for band 1, a spatial max is taken over an 8×8 grid first and then across the two scales (size 7 and 9). Note that in [7] do not take a max over different orientations, hence, each band \( (C_1)^{2} \) contains 4 maps.

**During training only:** Extract \( K \) patches \( P_i = 1, ..., K \) of various sizes \( n_i \times n_i \) and all four orientations (thus containing \( n_i \times n_i \times 4 \) elements) at random from the \( (C_1)^2 \) maps from all training images.

**S2:** For each C1 image \( (C_1)^2 \), compute:

\[
Y = \exp(-\gamma ||X - P_i||_2) \quad \text{for all image patches } X \text{ (at all positions) and each patch } P \text{ learned during training for each band independently. Obtain } S_2 \text{ maps } (S_2)^2_i.
\]

**C2:** Compute the max over all positions and scales for each \( S_2 \) map type \( (S_2)^2_i \) (i.e., corresponding to a particular patch \( P_i \) ) and obtain shift- and scale-invariant \( C_2 \) features \( (C_2)_i \), for \( i = 1, ..., K \).

S1 units come in 16 scales \( s \) arranged in 8 bands \( \Sigma \). For instance, consider the first band \( \Sigma = 1 \). For each orientation, it contains two S1 maps: one obtained using a filter of size 7, and one obtained using a filter of size 9. Note that both of these S1 maps have the same dimensions. In order to obtain the C1 responses, these maps are sub-sampled using a grid cell of size \( N^2 \times N^2 = 8 \times 8 \). From each grid cell obtain one measurement by taking the maximum of all 64 elements. As a last stage in [7] took a max over the two scales, by considering for each cell the maximum value from the two maps. This process is repeated independently for each of the four orientations and each scale band. In this new version of the standard model that described in [7] the subsequent S2 stage is where learning occurs. A large pool of \( K \) patches of various sizes at random positions are extracted from a target set of images at the C1 level for all orientations, i.e., a patch \( P_i \) of size \( n_i \times n_i \) contains \( n_i \times n_i \times 4 \) elements, where the 4 factor corresponds to the four possible S1 and C1 orientations. In this simulations [7] they used patches of size \( n_i = 4, 8, 12 \) and 16 but in practice any size can be considered. The training process ends by setting each of those patches as prototypes or centers of the S2 units which behave as radial basis function (RBF) units during recognition, i.e., each S2 unit response depends in a Gaussian like way on the Euclidean distance between a new input patch (at a particular location and scale) and the stored prototype. This is consistent with well-known neuron response properties in primate infero temporal cortex and seems to be the key property for learning to generalize in the visual and motor systems [11]. When a new input is presented, each stored S2 unit is convolved with the new \( (C_1)^2 \) input image at all scales (this leads to \( K \times 8 \) \( (S_2)^2 \) images, where the \( K \) factor corresponds to the \( K \) patches extracted during learning and the 8 factor, to the 8 scale bands). After taking a final max for each \( (S_2) \) map across all scales and positions, we get the final set of \( K \) shift- and scale-invariant \( C_2 \) units. The size of final \( C_2 \) feature vector [7] thus depends only on the number of patches extracted during learning and not on the input image size. This \( C_2 \) feature vector is passed to a classifier for final analysis. That we used of two classifier k-nearest neighbor(KNN) and support vector machine (SVM).

3- Experiments

3.1- Palmprint database and preprocessing

The Hong Kong Polytechnic University (PolyU) palm-print database that this database includes 600 palm-print images from 100 persons, with 6 images per person, is exploited to test our ap-
That first three images of each subject are used for training, and the last images three is used for testing.

**Table 1:** Model Parameters

<table>
<thead>
<tr>
<th>Band Σ</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>filters sizes s</td>
<td>7 &amp; 9</td>
<td>11 &amp; 13</td>
<td>15 &amp; 17</td>
<td>19 &amp; 21</td>
<td>23 &amp; 25</td>
<td>27 &amp; 29</td>
<td>31 &amp; 33</td>
<td>35 &amp; 37</td>
</tr>
<tr>
<td>effective width σ</td>
<td>2.8 &amp; 3.6</td>
<td>4.3 &amp; 5.4</td>
<td>6.3 &amp; 7.3</td>
<td>8.2 &amp; 9.2</td>
<td>10.2 &amp; 11.3</td>
<td>12.3 &amp; 13.4</td>
<td>14.6 &amp; 15.8</td>
<td>17.0 &amp; 18.2</td>
</tr>
<tr>
<td>wavelength λ</td>
<td>3.5 &amp; 4.6</td>
<td>5.6 &amp; 6.8</td>
<td>7.9 &amp; 9.1</td>
<td>10.3 &amp; 11.5</td>
<td>12.7 &amp; 14.1</td>
<td>15.4 &amp; 16.8</td>
<td>18.2 &amp; 19.7</td>
<td>21.2 &amp; 22.8</td>
</tr>
<tr>
<td>grid size N²</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>orientation θ</td>
<td>0; 2π/3; 4π/3; 5π/3; 8π/3; 10π/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>patch sizes u₀</td>
<td>4 x 4; 8 x 8; 12 x 12; 16 x 16 (x 4 orientations)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the preprocessing step the palm-print images are cropped to a size of 384x284 pixels. The normalization step includes geometric normalization, masking and photometric normalization. In this phase all the images are scaled to a standard 150x150 size. Next all non-palmprint areas are masked. Different levels of masking are experimented for finding the best one to get as good performance as possible for the algorithm. Finally the images are normalized for illumination. Then, this images are given to HMAX model to extract of palm-print features vector which HMAX model explained in section 2.

After the features have been extracted, this becomes a general classification problem. Two pattern classification methods are applied to recognize the incoming palm-prints. The first method used is a support vector machine (SVM) classifier, The other method used is a k-nearest neighbor (KNN).

### 3.2- Experimental results

In order to compare between our method with PCA method, we run the “Eigen-Palm” and HMAX experiments on the subset of persons PolyU, and We used a simple multiple-class SVM and k-nearest neighbor as classifier. Experimental results using the combination HMAX model and support vector machine (SVM) classifier (with kernel=1) obtains higher recognition rate than those obtained with HMAX model and k-nearest neighbors (KNN) classifier in identity verification system based on palm-print, and the proposed method, compared with PCA method, not only obtains higher recognition rate, but also this method is scale and rotate invariant, whereas PCA method provide high recognition rate only in closely controlled conditions (table 2, table 3).

**Table 2.** Recognition rate using HMAX method

<table>
<thead>
<tr>
<th>Class Number</th>
<th>SVM (kernel=1)</th>
<th>SVM (kernel=2)</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>96.6%</td>
<td>96.6%</td>
<td>96.6%</td>
</tr>
<tr>
<td>20</td>
<td>93.3%</td>
<td>93.3%</td>
<td>88.3%</td>
</tr>
<tr>
<td>30</td>
<td>93.3%</td>
<td>93.3%</td>
<td>88.8%</td>
</tr>
<tr>
<td>40</td>
<td>94.1%</td>
<td>94.1%</td>
<td>90.8%</td>
</tr>
<tr>
<td>50</td>
<td>93.3%</td>
<td>93.3%</td>
<td>89.3%</td>
</tr>
<tr>
<td>60</td>
<td>92.2%</td>
<td>92.2%</td>
<td>88.3%</td>
</tr>
<tr>
<td>70</td>
<td>90.9%</td>
<td>90.4%</td>
<td>87.1%</td>
</tr>
<tr>
<td>80</td>
<td>89.5%</td>
<td>90%</td>
<td>87.0%</td>
</tr>
<tr>
<td>90</td>
<td>89.2%</td>
<td>88.8%</td>
<td>86.3%</td>
</tr>
<tr>
<td>100</td>
<td>89%</td>
<td>88.6%</td>
<td>86.6%</td>
</tr>
</tbody>
</table>

**Table 3.** Recognition rate using PCA method
4- Conclusion

This paper describes a new biologically-motivated framework for robust palm-print recognition: for this system first computes a set of scale- and translation-invariant C2 features from a training set of palm-print images and then runs a standard discriminative classifier on the vector of features obtained from the input image. also, we run HMAX model using of KNN classifier, and Experiment results show that the HMAX method, compared with PCA method, not only obtains higher recognition rate, but also this method is scale and rotate invariant. In addition to, Experimental results showed using of the combination HMAX method and support vector machine (SVM) classifier (with kernel=1) obtains higher recognition rate than those obtained with HMAX model and k-nearest neighbors (KNN) classifier.

REFERENCES

tern Recognition, 36(10), pp 24-29-2439 (2003).


