Hidden Markov Models for Recognition Using Artificial Neural Networks

V. Bevilacqua, G. Mastronardi, A. Pedone, G. Romanazzi, and D. Daleno

Dipartimento di Elettrotecnica ed Elettronica, Polytechnic of Bari,
via E. Orabona, 4,70125, Bari, Italy
bevilacqua@poliba.it

Abstract. In this paper we use a novel neural approach for face recognition with Hidden Markov Models. A method based on the extraction of 2D-DCT feature vectors is described, and the recognition results are compared with a new face recognition approach with Artificial Neural Networks (ANN). ANNs are used to compress a bitmap image in order to represent it with a number of coefficients that is smaller than the total number of pixels. To train HMM has been used the Hidden Markov Model Toolkit v3.3 (HTK), designed by Steve Young from the Cambridge University Engineering Department. However, HTK is able to speakers recognition, for this reason we have realized a special adjustment to use HTK for face identification.

1 Introduction

Real world process generally produced observable outputs which can be considered as signals. A problem of fundamental interest is characterizing such real world signals in terms of signal models. In primis, a signal model can provide the basis for a theoretical description of a signal processing system which can be used in order to provide a desiderated output. A second reason why signal models are important is that they are potentially capable of characterising a signal source without having the source available. This property is especially important when the cost of getting signals from the actual source is high. Hidden Markov Models (HMM) are a set of statistical models used to describe the statistical properties of a signal [3][8]. HMM are characterised by two interrelated processes:

1. an unobservable Markov chain with a finite number of states, a state transition probability matrix and an initial state probability distribution. This is the principal aspect of a HMM;
2. a set of probability density functions for each state.

The elements that characterized a HMM are:

- $N=|S|$ is the number of states of the model. If S is the set of states, then $S = \{s_1, s_2, ..., s_N\}$. $s_i \in S$ is one of the states that can be employed by the model. To observe the system are used T observation sequences, where T is the number of observations. The state of the model at time t is given by $q_t \in S$, $1 < t < T$.

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCS 4113, pp. 126 – 134, 2006.

- M=|V| is the number of different observation symbols. If V is the set of all possible observation symbols (also called the *codebook* of the model), then $V = \{v_1, v_2, ..., v_M\}$.
- $A = \{a_{ij}\}$ is the state transition probability matrix, where a_{ij} is the probability that the state i became the state j:

$$a_{ij} = p(q_t = s_j \mid q_{t-1} = s_i) \qquad 1 \le i, j \le N \tag{1}$$

- $B = \{b_j(k)\}$ the observation symbol probability matrix, $b_j(k)$ is the probability to have the observation k when the state is j:

$$b_j(k) = p(o_t = v_k \mid q_t = s_j) \quad 1 \le j \le N\ ,\ 1 \le k \le M \tag{2}$$

- $\prod = \{\pi_1, \pi_2, ..., \pi_N\}$ is the initial state distribution, where:

$$\pi_i = p(q_1 = s_i) \qquad 1 \le j \le N \tag{3}$$

Using a shorthand notation, a HMM is defined by the following expression:

$$\lambda = (A, B, \prod) \tag{4}$$

2 Hidden Markov Models for Face Recognition

Hidden Markov Models have been successfully used for speech recognition where data are essentially one dimensional because the HMM provide a way of modelling the statistical properties of a one dimensional signal. To apply the HMM also to process images, that are two dimensional data, we consider temporal or space sequences: this question has been considered in [2][6][7], where Samaria suggests to use a space sequence to model an image for HMM. For frontal face images, the significant facial regions are 5: hair, forehead, eyes, nose and mouth [1][5].

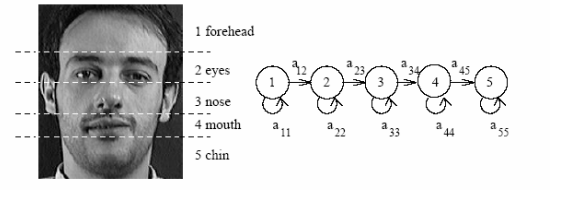


Fig. 1. The significant facial regions

Each of these facial regions (facial band) is assigned to a state in a left to right 1D continuous HMM. The Left-to-right HMM used for face recognition is shown in previous figure.To recognize the face k we must trained the following HMM:

$$\lambda^{(k)} = \left(A^{(k)}, B^{(k)}, p^{(k)}\right) \tag{5}$$

To train a HMM we have used 4 different frontal face gray scale image for any person. Each face image of width W and height Y is divided into overlapping blocks of height L and width W. The amount of overlap between consecutive blocks is M. The number of blocks extracted from each face image equals the number of observation vectors T and is given by:

$$T = \frac{(Y - L)}{(L - M)} + 1 \tag{6}$$

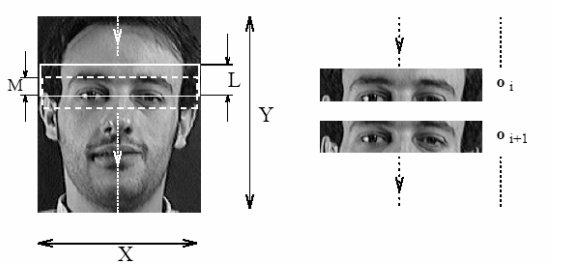


Fig. 2. The facial regions overlapped

The choice of parameters M and L can significantly affect the system recognition rate. A high amount of overlap M significantly increases the recognition rate because it allows the features to be captured in a manner that is independent of the vertical position. The choice of parameter L is more delicate. An insufficient amount of information about the observation vector could arise from a small value of the parameter L, while large L increases the probability of cutting across the features. However, the system recognition rate is more sensitive to the variations in M than in L, for this reason is used $M \leq (L-1)$.

We have considered X=92, Y=112, L=10, M=9 [1], then:

- T=103
- X x Y=10304 pixel
- X x L=920 pixel
- X x M=828 pixel

The observation sequence has T element, each of them is characterised by a window X x L=920 pixel. Using the pixel as elements of an observation sequence is the cause of a high complexity computing and a high sensitive to the noise. In this work is presented a new approach based on Artificial Neural Networks (ANNs) with the main goal to extract the principal characters of an image for reducing the complexity of the problem. To train HMM has been used The Hidden Markov Model Toolkit v3.3 (HTK) [4], designed by Steve Young from the Cambridge University Engineering Department. However, HTK is able to speech recognition, for this reason we have realized a special adjustment to use HTK for face identification.

3 Recognizing

After the HMM training, it's possible to recognize a frontal face image using the Viterbi's algorithm finding the model M_i that computes the maximum value $P(O \mid M_i)$, where O is the sequence of observation arrays that it's need to recognize. For HTK, the recognition is implemented by the Token Passing Model, an alternative formulation of the Viterbi's algorithm. For recognizing an image is used the tool "HVite":

```
HVite -a -iresult -I transcripts.mlf dict hmmlist foto1
```

-i means that the results will be stored into file "result", while "foto1" is the frontal face image to recognize for HTK. "transcrpts.mls","dict","hmmlist" are text files.

Fig. 3. Transcripts.mlf , "foto1" is the name of the image

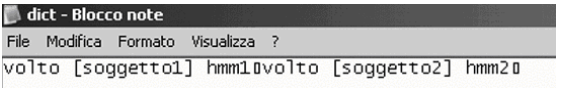


Fig. 4. Dict

In Fig. 4 "soggetto1" and "soggetto2" are the names of the frontal face images to recognize, the following *hmm#* is the associated HMM. "hmm2" e "hmm1" are the files stored by the tool "HRest": in the first case has been used the pixels for the observation sequences, while for "hmm1" has been applied the ANN approach introduced by this work.

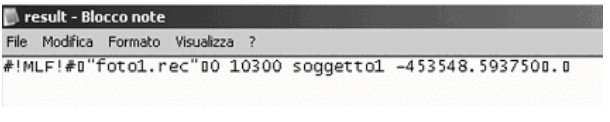


Fig. 5. Result

The view of the file asserts that the "foto1" has been recognize as "soggetto1" with total logarithmic probability "-453548.593750", for each observation sequence the average probability is the same value divided by T.

4 Artificial Neural Networks to Observe an Image for Hidden Markov Models

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. All connections among neurons are characterized by numeric values (*weights*) that are updated during the training. The ANN is trained by a supervised learning process: in the training phase the network processes all the pairs of input-output presented by the user, learning how to associate a particular input to a specific output trying to extend the information acquired also for cases that don't belong to the *training set spectrum*. Any pair of data in the training set is presented to the system a quantity of time determined by the user *a priori*. The learning step is based on the Error *Back Propagation* (EBP) algorithm. The weights of the network are updated sequentially, from the output layer back to the input layer, by propagating an error signal backward along the neural connections (hence the name "back-propagation") according to the gradient-descent learning rule:

$$\Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \qquad 0 < \eta < 1 \tag{7}$$

For this work an ANN is used to compress a bitmap image in order to represent it with a number of coefficients that is smaller than the total number of pixels. Using these coefficients instead of the pixels has been realized a robust facial face recognition system that can operate under a variety conditions, such as varying illuminations and background. The ANN, using the EBP algorithm, extracts the main features from the image to store them in a sequence of 50 bits reducing the complexity computing of the problem. The image is a facial feature of a frontal face image; from this area we consider 103 segments of 920 pixels that represent the observable *states* of the model. Now all of these sections are divided into features of 230 pixels, that are the input of the network. The first layer is formed by 230 neurons, each neuron per pixel, the hidden layer is composed by 50 units and the last layer by 230 neurons. After the training, the network is able to work as a pure linear function, the input of the first layer must be the same of the output of the last layer. The compressed image is described by 50 bits that are the outputs of a hidden layer consisting of a Heaviside function processing elements. For any window of 230 pixels we have an array of 50 elements, so a section of 920 pixels is compressed in a 4 sub windows of 50 binary value array each. The matrix weights, referred to the connections between the inputs and the hidden layer, codifies the image bitmap, while the matrix weights associated to the connections between the hidden layer and the outputs, decodes the sequence of bits.

The ANN has been trained using 10 frontal face image of different persons, after the training phase, the matrix weights is stored and finally the ANN is tested with other images that are similar, but not the same to the training set feature. Finally, we

have a face image compressed into an *observation* vector of 103 element of 200 binary (1/0) values that will be computed by the Hidden Markov Models (HMM).

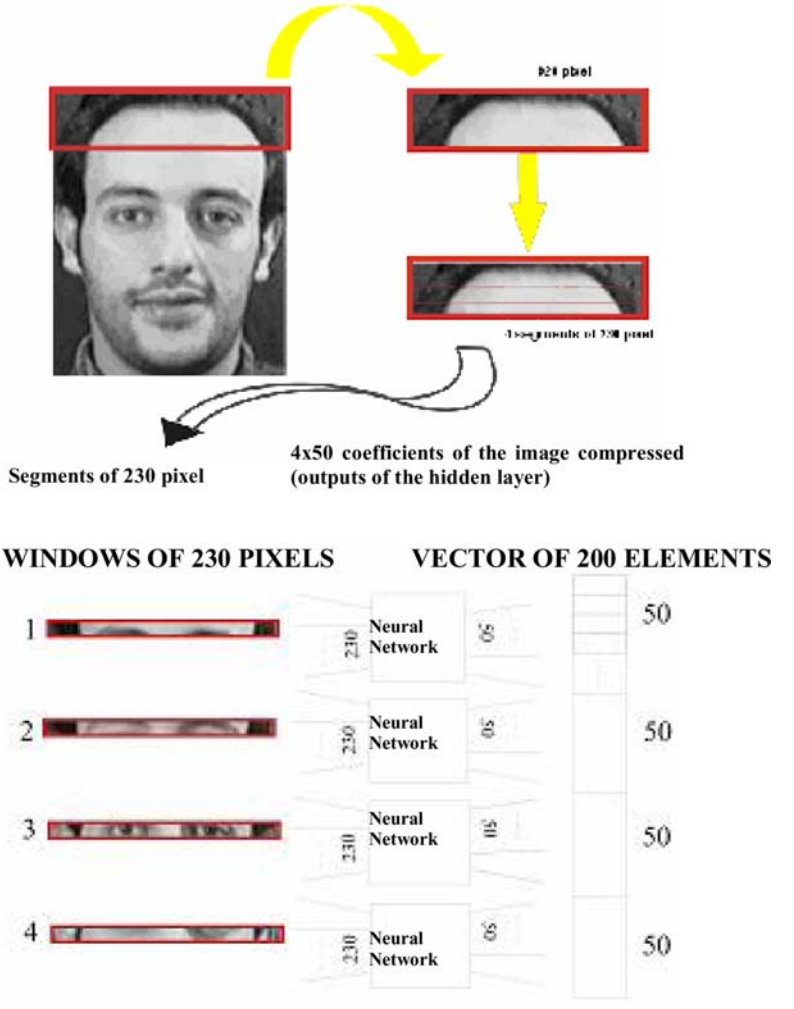
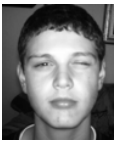


Fig. 6. System developed

5 Conclusions and Results

Two recognizing systems have been developed: the first uses pixels as observation vectors and the second employs vectors containing the principal components of the images returned by the neural network. The system has been tested using 4 facial face image of 4 different persons:

Table 1. Training images

NOME	Image 1	Image 2	Image 3	Image 4
subject1				
subject2				
subject3				
subject4				

For each face recognized we have computed the total logarithmic probability.

Table 2. Results for subject3

Subject 3	Without ANN	Using ANN
	-436165.750000	-93947.101563
	-470681.968750	-99128.085938
	-478726.531250	-98591.703125
	-442659.375000	-95156.367188

Table 2. (*continued*)

	-485397.093750	-104245.335938
	-497308.093750	-103274.578125
	-498396.781250	-108224.695313
	-475043.062500	-99134.726563
	-491749.812500	-103642.601563
	-479394.687500	-102927.476563

Table 3. Results for subject 4

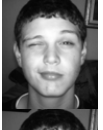
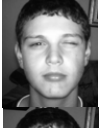
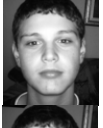
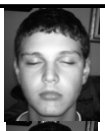
Subject 4	Without ANN	Using ANN
	-435390.937500	-95473.640625
	-430359.500000	-94148.359375
	-446048.250000	-98179.273438
	-441871.812500	-95641.335938
	-516868.187500	-107430.375000

Table 3. (*continued*)

	-443342.437500	-97138.390625
	-456531.875000	-99424.187500
	-609438.000000	-127188.664063
	-635798.500000	-122770.937500
	NOT RECOGNIZED	
	-658844.000000	-131893.875000
	NOT RECOGNIZED	

The logarithmic probability is the possibility that the bitmap will be recognized: it is clear that the system with ANN is more efficient than the system without. In the table 3., there are two images that are not recognized by the system without the neural network.

References

1. Nefian, A. V., Monson, H.: Hidden Markov Models for Face Recognition. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP98), Seatle (1998) 2721-2724
2. Samaria, F.: Face Identification using Hidden Markov Model. 1st Year Report Cambridge University Engineering Department, London (1992)
3. Rabiner, L., Huang, B.: Fundamentals of Speech Recognition. Englewood Cliffs, Prentice-Hall, New York (1993)
4. Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University Engineering Department, Cambridge, UK (2002)
5. Nefian, A. V., Monson, H.: Detection and Recognition Using Hidden Markov Models. International Conference on Image Processing, 1 (1998) 141–145
6. Samaria, F.: Face Recognition Using Hidden Markov Models. PhD Thesis, University of Cambridge, (1994)
7. Samaria, F., Young, S.: Face HMM Based Architecture for Face Identification. Image and Computer Vision, 12 (1994) 537-583
8. Cottrell, G. W., Munro, P., Zipser, D.: Learning Internal Representations from Gray Scale Image: An Example of Extensional Programming. In Ninth Annual Conference of the Cognitive Science Society, (1987) 462-473