Asian Journal of Information Technology 5 (11): 1177-1179, 2006

Multi-Agent Based Facial Recognition System Using RETSINA

[1]K.L. Shunmuganathan and [2]V. Vasudevan,
[1]Department of Computer Science and Engineering,
Noorul Islam College of Engineering, Kumarakoil, Kanniyakumari, India
[2]Arulmigu Kalasalingam college of Engg, India

Abstract: We propose an appearance-based facial recognition system based on multi-agents. By this method a person can be identified and verified using his face. Each RETSINA agent performs the facial recognition with its own small training sets and its stops other agents when it finds a match by communicating through ACL. Experimental results suggest that the proposed multi-agent based Facial recognition method provides a better recognition rate and achieves less time complexity. The worst case analysis shows proposed multi-agent based facial recognition is 80% better than ordinary facial recognition.

Key words: Multi-agents, agent communication language, RETSINA, biometrics, principal component analysis, eigenvectors, feature extraction, covariance matrix

INTRODUCTION

Though there are many biometric recognition techniques, the time complexity is always an important issue. Many researches are working for this by taking various biometrics like face, ear, iris, gait etc. and by implementing various recognition techniques. From ATMs to logging into a network or your personal computer, biometrics is being used throughout the world. In facial recognition, it is mandatory that the training set should contain 'n' images of each individual. For example if the training set consists of 10 images of each individual, then for 1000 individuals it will contain 10000 images. This makes the system more complex and requires lot of time to process. This difficulty can be avoided using Multi-agents. The proposed system uses Multi-Agents along with PCA and Euclidean distance method for feature extraction and feature matching respectively

Principal component analysis: Let X denote the n-dimensional unitary column vector of the an image, where

$$X = \begin{bmatrix} x1 \\ x2 \\ x3 \\ . \\ . \\ xn \end{bmatrix} \quad (1)$$

and x1,x2,….,xn are the common pixel in each of the n training images. This one vector represents one common pixel in all n images.

If the images are of size M X N, there will be a total of K=MN n-dimensional vectors after all the pixels are represented in this manner. We can write them on a line of text simply by expressing them as $X = (x_1, x_2, \ldots\ldots x_n)^T$ where 'T' indicates the transpose. The mean vector of the population is defined as

$$M_x = (1/K) \sum_{k=1..n} X_k \quad (2)$$

The covariance matrix could be approximated from the samples as follows

$$C_x = (1/K) \sum_{k=1..n} (X_k X_k T - M_x M_x T) \quad (3)$$

where k= 1.. n and C ii is the referred to as the variance of C_x and C_{ij} is the referred to as the co-variance of C_x.

The eigen values of an n x n matrix C_x can be found by solving the relation

$$C_x e_i = \lambda_i e_i \quad (4)$$

where λ_i is the eigen value and e_i is the eigen vectors. Since the eigen vectors have the same dimension as the original images, they are referred to as eigen pictures in[1] and eigen vectors in[2]. Form a new matrix; A whose rows are formed from the eigenvectors (principal components) of C_x, ordered so that the first row of A is

Corresponding Author: K.L. Shunmuganathan, Department of Computer Science and Engineering, Noorul Islam College of Engineering, Kumarakoil, Kanniyakumari, India

the eigenvector corresponding to the largest eigen value and the last row is the eigenvector corresponding to the smallest eigen value.

Feature matching: First, the new image is transformed into its principal component. The Euclidean distance between two weight vectors provides a measure of similarity between the corresponding images. If the Euclidean distance exceeds on average some threshold value, there is no match. Weight vector Ωnew is formed from the resulting weights.

$$W_k = e_i^T (\Gamma_{new} - M_x) \text{ where } k=1..n \quad (5)$$

$$\Omega_{new} = [W_1 W_2..W_n] \quad (6)$$

where e_i is the eigen vector, Γ_{new} is the new unknown image and Mx is the average image. Euclidean distance between two weight vectors $d(\Omega_i,\Omega_j)$ provides a measure of similarity between the corresponding images i and j.

$$d(x_i, x_j) = (\Sigma_{r=1..n}(ar(x_i) - ar(x_j))^2)^{1/2}$$

where ar(x) denotes the value of the rth attribute of instance x.

Introduction to multi-agents and RETSINA: Agent technology is a new concept within the Artificial Intelligence (AI). An agent can be used to denote a hardware or software-based computer system situated in a specific environment, which has some properties such as autonomous, pro-active, social ability and reactive capability. The agent paradigm in AI is based upon the notion of reactive, autonomous, internally motivated entities that inhabit dynamic, not necessarily fully predictable environments[3]. Autonomy is the ability to function as an independent unit over an extended period of time, performing a variety of actions necessary to achieve pre-designated objectives while responding to stimuli produced by integrally contained sensors[4].

Multi-agent Systems can be characterized by the interaction of many agents trying to solve a variety of problems in a co-operative fashion. Besides AI, intelligent agents should have some additional attributes to solve problems by itself in real-time; understand information; have goals and intentions; draw distinctions between situations; generalize; synthesize new concepts and/or ideas; model the world they operate in and plan and predict consequences of actions and evaluate alternatives. It may be obvious that finding a feasible solution is a necessity for an agent. Often local optima in decentralized systems are not the global optimum. This problem is not easily solved. The solution has to be found by tailoring the interaction mechanism or to have a supervising agent co-coordinating the optimization process of the other agents.

RETSINA (Reusable Environment for Task-Structured Intelligent Networked Agents) is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA (Foundation for Intelligent Physical Agents) standards for intelligent agents. RETSINA is a middleware that facilitates the development of multi-agent systems. It includes

- A runtime environment where RETSINA agents can "live" and that must be active on a given host before one or more agents can be executed on that host.
- A library of classes that programmers have to/can use (directly or by specializing them) to develop their agents.
- A suite of graphical tools that allows administrating and monitoring the activity of running agents.

Multi-agents using RETSINA: Each running instance of the RETSINA runtime environment is called a Container as it can contain several agents. The set of active containers is called a Platform. A single special Main container must always be active in a platform and all other containers register with it as soon as they start. It follows that the first container to start in a platform must be a main container while all other containers must be "normal" (i.e. non-main) containers and must "be told" where to find (host and port) their main container (i.e. the main container to register with).

RETSINA agents are identified by a unique name and provided they know each other's name, they can communicate transparently regardless of their actual location. Each RETSINA agent performs the facial recognition with its own small training sets and its stops other agents when it finds a match by communicating through ACL.

RESULTS

Fig. 1: Sample gallery images

Fig. 2: Mean image

Fig. 3: Eigenvectors

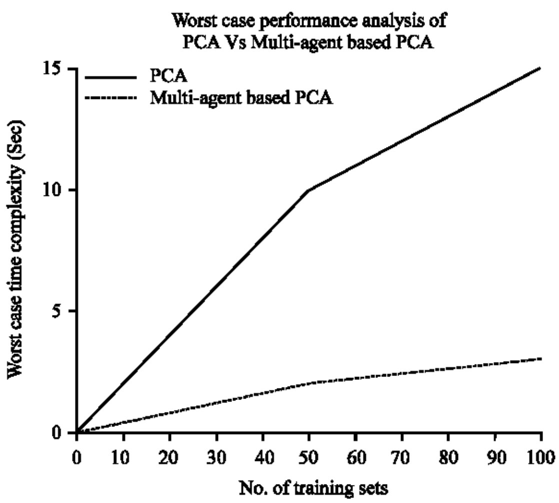

Fig. 4: Performance analysis

CONCLUSION

The proposed multi-agent based facial recognition system is 80% better than ordinary facial recognition systems in its worst case time complexity. This paves the way for using this system in applications which requires even very large training set. Recently many feature extraction techniques are proved to be good in recognition rate when compared with PCA. So an enhancement can be any other feature extraction techniques like Laplacian faces, Fisherfaces etc. can be used along with multi-agents.

REFERENCES

1. Turk, M. and A. Pentland, 1991. Eigenfaces for Recognition, J. Cognitive Neuroscience, 3: 71-86.
2. Rafael, C.G. and R.E. Woods, 2003. Digital Image Processing, Pearson Education, 2nd Edn., pp:675-679.
3. Anil, J., R. Bolle, S. Pankanti, Biometrics “Personal Identification in Networked Society”,Kluwer Academic Publishers, pp: 65-84, 273-284, 327-342.
4. Frances, M., T. Brazier, B.M. Dunin-Keplicz, R.N. Jennings and J. Treur, 1997. Desire: Modeling Multi-Agent Systems in a Compositional Formal Framework. Intl. J. Cooperative Inform. Sys., pp: 67-94.