An Incremental Learning Algorithm for Face Recognition⋆

O. Déniz ⋆⋆, M. Castrillón, J. Lorenzo, and M. Hernández

Instituto Universitario de Sistemas Inteligentes
y Aplicaciones Numéricas en Ingeniería (IUSIANI)
Universidad de Las Palmas de Gran Canaria
Edificio Central del Parque Científico-Tecnológico
Campus Universitario de Tafira
35017 Las Palmas - SPAIN
{odeniz,mcastrillon,jlorenzo,mhernandez}@dis.ulpgc.es

Abstract In face recognition, where high-dimensional representation spaces are generally used, it is very important to take advantage of all the available information. In particular, many labelled facial images will be accumulated while the recognition system is functioning, and due to practical reasons some of them are often discarded. In this paper, we propose an algorithm for using this information. The algorithm has the fundamental characteristic of being incremental. On the other hand, the algorithm makes use of a combination of classification results for the images in the input sequence. Experiments with sequences obtained with a real person detection and tracking system allow us to analyze the performance of the algorithm, as well as its potential improvements.

Keywords: face recognition, incremental learning, face sequences

1 Introduction

The face recognition problem has generated a huge amount of research work in the last years. Although it is a very difficult task, some systems have been able to achieve an acceptable performance under restricted conditions. However, most of the published papers present experiments carried out under non-realistic conditions, like using only one image for the recognition decision. Some authors have shown that the information available in a video sequence can improve significantly the performance of the system, in comparison with the use of only one image. Some of the presented systems modify the representation space or the classifiers to take into account the information of the sequence. Others simply resort to a fusion of the classification results. With respect to the former, in [1]

faces are characterized by trajectories in a representation space, obtained from sequences in which a head is rotating in front of the camera, for example. The recognition decision is made after comparing the trajectory corresponding to the test sequence with those of prototype sequences. In [2] a subspace is generated with the input image sequence, and is also compared with subspaces generated during the training stage. This led to an improvement in the robustness of the system under expression and pose changes. With respect to the verification problem, the use of a set of images allowed up to a 40% error reduction, see [3]. Also, it was observed that this reduction was larger in the first images of the sequence, and then it began to saturate. With regard to the systems that resort to a fusion of the classification results, many fusion rules are possible, the most used being the maximum rule [4, 5, 6], the mean [7] and the sum [8]. In practical systems a fusion of the classification results is generally preferred.

In a problem such as face recognition any piece of available information can be of great value. The life cycle of a practical recognition system would be divided in two stages: classifier training (with a set of training images) and recognition itself. From a computational viewpoint it is not practical to generate a new classifier every time new information is gathered in the recognition stage, because the cost of this operation usually depends on the number of considered samples. In [9] an automatic learning system for face recognition is described. This system does not use supervised information but the output of the system itself to update its internal representation. If the system works with low error, this method will perform acceptably. However, if the system makes a wrong decision frequently it will degenerate. On the other hand, there are many cases in which supervised information is available. For example, the individual in front of the camera can identify himself voluntarily, or the recognition system can identify the individual by other means. For these applications, a solution is described in [10], where a decision tree for high-dimensional spaces is used. Each node of the tree represents a space obtained with PCA, and the tree is dynamically updated by forgetting and controlling its growth. In [11] the *Argus* system for visitor recognition is described. The goal of *Argus* is to detect and recognize people in front of a door, and also to notify the arrival to those people in the building related to the visitor. When *Argus* makes a wrong identification, the person in the building can provide the system with the identity of the person in front of the door, or confirm the decision in case it is correct. *Argus* uses stored images and the nearest-neighbour classifier. Thus, the information provided can be easily used to update the system. In [12], a fusion method is described in the authentication context. Client-impostor measures given by many experts in different moments are fused by a supervisor. Normality is assumed in its theoretic development and temporal fusion is achieved by using the history of errors made by the experts.

In this paper, a method for taking advantage of any available supervised information is described. This supervised information, gathered while the system is running, is used to improve the classification results. The main characteristics of the algorithm presented are the use of a combination of classification results and its incremental nature. In Section 2 the basis of the proposed method is

explained. The corresponding algorithm is described in Section 3. In Section 4 experiments which show the performance of the algorithm are described and finally, in Section 5 the most important conclusions are outlined.

2   IRDB

The proposed method, which we call IRDB (Incremental Refinement of Decision Boundaries), is applied over a decision scheme like that represented by the following rule:

$$\text{If } m = \arg \max_{i=1,\ldots,z} d_i(\boldsymbol{x}) \Rightarrow \boldsymbol{x} \in C_m, \tag{1}$$

where $d_i(\boldsymbol{x})$ is the estimation provided by the classifier, and classes are represented as $C_1, \ldots, C_z$. When the system is running a set of n labelled samples is gathered $(\boldsymbol{x}^j; d_1^j, \ldots, d_z^j)$, with $j = 1, \ldots, n$. In order to take advantage of this information a new decision rule will be used:

$$\text{If } m = \arg \max_{i=1,\ldots,z} F_i(\boldsymbol{p}_i; d_i(\boldsymbol{x})) \Rightarrow \boldsymbol{x} \in C_m, \tag{2}$$

where the functions F_i act as modifiers of the classifier outputs and have as parameters the vectors $\boldsymbol{p}_i$, of P elements. The best parameter vectors will be those that conform to the relationship: $d_i^j = F_i(\boldsymbol{p}_i; d_i(\boldsymbol{x}^j))$, with $j = 1, \ldots, n$. These vectors $\boldsymbol{p}_i$ can be assigned one by one in a suboptimal way with:

$$\boldsymbol{p_i}^* = \arg \min_{\boldsymbol{p_i}} [\hat{P}(error|\boldsymbol{x} \in C_i, \boldsymbol{p_i}) - \hat{P}(correct|\boldsymbol{x} \notin C_i, \boldsymbol{p_i})] \tag{3}$$

for $i = 1, \ldots, z$, where the probabilities $\hat{P}$ must be estimated from the available labelled samples. Figure 1 shows the operation of the proposed system.
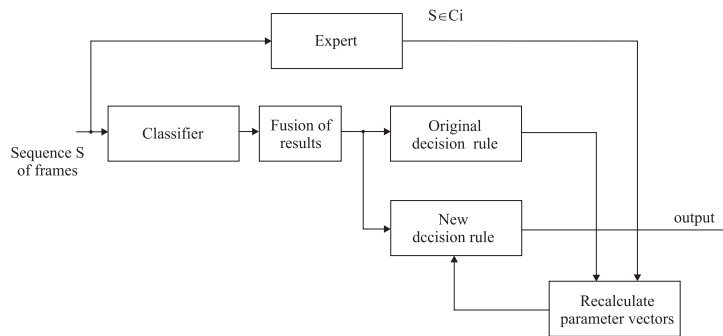
Fig. 1. Operation of the IRDB method. The calculation of the parameter vectors is accomplished only when some entity (expert) provides labels for the input sequences.

If the total number Q of labelled samples used to estimate the probabilities is small, the result can be worse than that obtained with the original decision rule. However, the method will tend to provide better results as new samples are accumulated and considered. The objective is therefore to obtain the parameter vectors $\boldsymbol{p_i}$ which conform to the labelled samples, in an incremental way and in a fixed time, for the operation would have to be completed on-line. This can be accomplished by the algorithm described in the next section.

3   Algorithm

The concrete instance of the IRDB method studied in this paper corresponds to the use of the functions $F_i = d_i(\boldsymbol{x}) + p_i$, which means that an additive weight modifies each class output. Using weights is equivalent to the use of a decision threshold, and therefore for two classes the method can be thought of as a continuous search for a better position in the system ROC (*Receiver Operating Characteristic*) curve. On the other hand, the inputs d_i used by the method are both the classifier outputs and the temporal fusion of classifier outputs. This decision is prompted by one of the implicit objectives of our recognition system: improving the identification rate of sequences, not individual frames. In particular, we seek to improve the correct identification rate as a function of the number of considered frames. In a real situation the number of frames that feed the classifier is not known (though it can be fixed beforehand. This performance can be measured by the area under the curve which represents identification rate against number of frames of the sequence, calculated as the sum S of the percentage of test subsequences correctly identified for each number of considered frames. Here we are assuming that the number of frames in an input sequence is a priori unknown and is between 1 and a maximum value established beforehand (which can be related to the maximum response time of the system). In order to facilitate the explanation of the algorithm, from now on we will assume that the number of classes z is 2.

As explained in Section 2, in order to assign the additive weights it is necessary to obtain the estimations: $\hat{P}(error|\boldsymbol{x} \in C_i, \boldsymbol{p_i})$ and $\hat{P}(correct|\boldsymbol{x} \notin C_i, \boldsymbol{p_i})$, or in another form, $\hat{P}(error|\boldsymbol{x} \in C_i, F_i - d_i(\boldsymbol{x}))$ and $\hat{P}(correct|\boldsymbol{x} \notin C_i, F_i - d_i(\boldsymbol{x}))$. These estimations will be represented by two histograms, $\boldsymbol{H}_i = (H_{i,1}, \ldots, H_{i,nb})$ and $\overrightarrow{HNO}_i = (HNO_{i,1}, \ldots, HNO_{i,nb})$, both characterized by the ranges $(r_0, \ldots, r_{nb})$, nb being a parameter fixed a priori. These histograms, 2 for each class i, are calculated as follows. Given a new labelled sample $\boldsymbol{x}$, if it belongs to class i (for i between 1 and z) and the original decision rule is wrong (that is, if $\boldsymbol{x} \in C_i$ and $\arg\max_{j=1,\ldots,z} d_j(\boldsymbol{x}) \neq i$), the value $p = \max_{j=1,\ldots,z}(d_j(\boldsymbol{x})) - d_i(\boldsymbol{x})$ is calculated, which is the difference between the output for the winner class and the output for class i. This value p is then added to the histogram $\boldsymbol{H}_i$: if $r_k \leq p < r_{k+1} \Rightarrow H_{i,k+1} = H_{i,k+1} + 1$. If on the contrary $\boldsymbol{x}$ does not belong to class i and the original decision rule is right (that is, if $\boldsymbol{x} \in C_l, l \neq i$ and $\arg\max_{j=1,\ldots,z} d_j(\boldsymbol{x}) = l$), the value $p = \max_{j=1,\ldots,z}(d_j(\boldsymbol{x})) - d_i(\boldsymbol{x})$ is calculated and added to the histogram $\overrightarrow{HNO}_i$. Once all the new information has been added

to the histograms weights are assigned. In order to obtain the weight to apply to a class i equation (3) is used, which is equivalent to calculating the maximum of the subtraction of the histograms $\boldsymbol{H}_i$ and $\overrightarrow{HNO}_i$. That is, the weight assigned is the one that will remove a large number of errors while not losing many correct decisions. The assigned weight is thus one of the values r_0 to r_{nb}. Once a weight p_i is assigned, and before assigning the next one, it is necessary to update all the histograms to keep the coherence of the process. This could be done by recalculating the histograms, considering each labelled sample again (and using the assigned weight p_i). However, this will make the process non incremental. An incremental update can be accomplished by modifications to the histograms if the condition $r_i + r_j = r_k$ holds, for $i, j, k = 0, \ldots, nb$ and j and i such that $r_i + r_j \leq r_{nb}$. In that case the modifications needed are simple shifts. For example, if $r = (0, 0.2, 0.4, 0.6, 0.8, 1)$ and the values of p obtained for class 1 are $0.1, 0.3, 0.5, 0.1, 0.5, 0.1$, the histogram $\boldsymbol{H}_1$ would be $\boldsymbol{H}_1 = (3, 1, 2, 0, 0)$. After assigning the weight $p_1 = 0.2$, the new $\boldsymbol{H}_1$ would be $\boldsymbol{H}_1 = (1, 2, 0, 0, 0)$. The value 3 will go to $\overrightarrow{HNO}_2$, for it corresponds to mistakes of the class 1 that after the assignation will turn into correct decisions for class 2. Using the multiclass notation of Section 2, this condition can be represented as $F_i(\boldsymbol{r}_j, \boldsymbol{r}_k) = \boldsymbol{r}_l$, for $j, k, l = 0, \ldots, nb$ and j and k such that $F_i(\boldsymbol{r}_j, \boldsymbol{r}_k) \leq \boldsymbol{r}_{nb}$. For the particular case of $F_i = d_i(\boldsymbol{x}) + p_i$, it is easy to see that this condition holds if all the bins of the histograms have the same size.

As for the computational cost of the algorithm, it depends on the number of histogram bins (nb), on the number of classes z, and on the generation of subsequences. If we want to use all the available information, all the possible combinations of sequence frames should be generated. As the number of possible combinations can be too large a number, other option must be chosen, like for example using only a fixed number of the possible combinations. The only effect of this would be a delay in the learning process, for less information would be used at each step. On the other hand, using all the information in the input sequence, which belongs to a single class, would unbalance the histograms and the results would be incorrect. In order to avoid this, the information obtained in previous steps is replicated for the other classes. Alternatively, weights could be updated only after having the same number of samples for every class. The storage cost of the method is $O(z \cdot nb^{P(z-1)})$ (P is the number of elements of the parameter vectors) which can be a limitation depending on the number of classes of our problem.

4 Experiments

In order to analyze the performance of the IRDB method, experiments were made with real face sequences. These sequences were obtained with the DESEO system [13]. DESEO is a hardware-software system that can detect and follow people in real time, using motion and/or skin colour information. The images that DESEO provides are processed to confirm that they contain a relatively frontal face, and if so, normalize them. The whole process is described in detail in [14]. The net

result is a set of face images, normalized and ready to be recognized, see Figure 2. On the one hand, a 2 class problem was studied: 10 sequences, one for each individual, each one with 167 frames. The 2 classes are: people related to our laboratory and people that work in the laboratory but are not directly related to the laboratory. On the other hand, experiments were made using 5 classes: one sequence per indidividual, the objective being his/her identification. Due to a lack of space, only the results for the two class problem are presented here. All the images used in the experiments are 39x43 pixels in size. For each sequence, 3 images were used for training the classifier, 50 as supervised information for the algorithm and the rest for test. PCA *(Principal Component Analysis)* was applied to the set of training images. Each experiment was made ten times, each time changing the order of the images in the sequence randomly. The final results presented here are the mean of those ten results. The test images from each sequence were extracted n at a time from the complete sequence (with overlap: frame 1-frame 2, frame 2-frame 3,...), with n between 1 and 10. The generation of subsequences for the IRDB algorithm was made in the same way. In one case, the nearest-neighbour classifier was used (taking the mean as prototype and euclidean distance) and in the other an SVM *(Support Vector Machines)* classifier, with radial basis function kernel. The parameter nb was set to 20 in all the experiments.

Fig. 2. Two examples of the normalized face sequences used in the experiments.

The results obtained without IRDB are shown in Table 1. All the curves representing correct identifications against number of frames were monotonous and increasing. As fusion strategies the mean and the majority vote rules were used. Also, to convert the output values of the classifier to the needed [0,1] range, the mapping function $y = 1/(1 - e^{-\left(\frac{x-\mu}{\sigma}\right)})$ was used, where μ and σ are respectively the mean and standard deviation of the values obtained for the training set.

The results obtained using IRDB are shown in Tables 2 and 3. In the first column (F) appears the number of accumulated frames, and in the second column (C) the class of the accumulated frames. With respect to the second line of the table, n is the maximum number of frames of the generated subsequences. N is the maximum possible value, in this case max(frames of the input sequence,10). From the results presented it can be seen that the IRDB method improves the performance of the recognition, and that the improvement increases with the

Table 1. Numerical results for the sum of percentage of correct decisions (S) and maximum percentage of correctly identified sequences achieved (MAX).

Classifier	Fusion	S	MAX
Nearest-neighbour	Mean	732.40	74.88
	Majority	721.48	74.55
SVM Classifier	Mean	782.05	79.96
	Majority	770.85	79.26

number of accumulated frames. Also, it can be observed the positive effect of generating subsequences with n=N, in comparison with the use of n=1 (for n=1 the method is no fusing classification results).

Table 2. Results obtained with the IRDB algorithm, using the mean rule.

F	C	Nearest neighbour				SVM classifier			
		n=N		n=1		n=N		n=1	
		S	MAX	S	MAX	S	MAX	S	MAX
4	1	711.83	72.67	709.05	72.39	760.35	77.78	770.45	78.47
	2	724.33	74.38	722.09	73.80	812.13	83.36	793.87	81.23
6	1	729.37	75.03	719.21	73.60	815.02	84.02	798.17	81.80
	2	734.28	75.71	724.26	74.26	815.19	83.89	812.54	83.33
8	1	734.28	75.71	722.46	74.01	814.46	83.77	815.02	83.71
	2	743.08	76.77	728.02	74.59	813.52	83.39	814.59	83.66
25	1	748.22	76.87	732.57	74.96	815.89	84.05	814.58	83.49
	2	751.87	77.54	728.96	74.56	820.87	84.81	805.46	82.33
50	1	747.59	76.94	732.78	74.89	820.98	84.46	810.38	83.06
	2	751.02	77.50	736.47	75.59	822.60	84.89	817.82	83.84

5   Conclusions and Future Work

In the last years it has been empirically shown that the temporal combination of the classification results in a sequence improves the performance of the recognition system. In practical systems simple combination rules are generally used, such as the mean, the maximum or the majority vote rules. On the other hand, for certain applications there is supervised information available that, given the complexity of the problem to solve, should not be discarded. Both aspects, of practical interest, have been considered in the incremental learning method proposed. From a computational viewpoint, the method does not degenerate with the number of accumulated frames. Possible improvements to the algorithm would include those related to the storage cost for a large number of classes. This limitation could be alleviated by the use of virtual memory and sparse matrices. On the other hand, some classifiers are extended to the multiclass problem

Table 3. Results obtained with the IRDB algorithm, using the majority vote rule.

F	C	Nearest neighbour				SVM classifier			
		n=N		n=1		n=N		n=1	
		S	MAX	S	MAX	S	MAX	S	MAX
4	1	721.48	74.55	723.29	74.771	770.85	79.26	770.85	79.26
	2	727.46	75.17	727.46	75.172	775.98	79.72	775.98	79.72
6	1	727.46	75.17	727.46	75.171	779.07	80.21	777.43	79.91
	2	727.46	75.17	726.80	75.092	777.62	80.03	778.78	80.19
8	1	727.46	75.17	727.46	75.171	779.07	80.21	779.07	80.21
	2	730.73	75.48	726.80	75.092	777.62	80.03	777.62	80.03
25	1	730.39	75.56	730.73	75.481	778.03	80.11	779.74	80.32
	2	731.75	75.70	730.73	75.482	777.62	80.03	776.82	79.99
50	1	731.39	75.57	730.73	75.481	777.62	80.03	778.52	80.16
	2	731.39	75.57	729.86	75.472	778.41	80.10	778.52	80.16

by using many 2-class solutions (i.e. those based in SVM), and therefore would not suffer this storage limitation. Another point of future interest is the use of other modifier functions F_i that can provide a greater learning ability. Finally, experiments with a real, practical implementation would be useful to establish the performance and limitations of the method precisely.

References

[1] Y., S. Gong, and H. Liddell. Exploiting the dynamics of faces in spatio-temporal context. In *Procs. The Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV2000)*, Singapore, December 2000.

[2] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 318–323, Nara, 1998.

[3] J. Kittler, J. Matas, K. Jonsson, and M.U. Ramos Sánchez. Combining evidence in personal identity verification systems. *Pattern Recognition Letters*, 18(9):845–852, 1997.

[4] A. J. Howell and H. Buxton. Towards unconstrained face recognition from image sequences. In *Procs. of the Second Int. Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, October 1996.

[5] H. Wechsler, V. Kakkad, J. Huang, S. Gutta, and V. Chen. Automatic video-based person authentication using the RBF network. In *First Int'l Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, Crans-Montana, Switzerland, 1997.

[6] T. Choudbury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal person recognition using unconstrained audio and video. Technical Report TR-472, MIT Media Lab, 1998.

[7] A. Senior. Recognizing faces in broadcast video. In *Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, Corfu, Greece, Sept. 1999.

[8] S. McKenna and S. Gong. Recognising moving faces. In *Procs. of the NATO ASI on Face Recognition: From Theory to Applications*, Stirling, UK, 1997.

[9] K. Okada and C. von der Malsburg. Automatic video indexing with incremental gallery creation: integration of recognition and knowledge acquisition. In *Procs. of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, pages 431–434, Adelaide, August 1999.

[10] J. Weng, C.H. Evans, and W.S. Hwang. An incremental learning method for face recognition under continuous video stream. In *Procs. of the Fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000.

[11] R. Sukthankar and R. Stockton. Argus: The digital doorman. *IEEE Intelligent Systems and their applications*, 16(2):14–19, 2001.

[12] E. Bigun, J. Bigun, B. Duc, and S. Fischer. Expert conciliation for multi modal person authentication systems by bayesian statistics. In J. Bigun, G. Chollet, and G. Borgefors, editors, *Audio and Video based Person Authentication - AVBPA97*, volume LNCS-1206, pages 291–300. Springer, 1997.

[13] F.M Hernández, J. Cabrera, M. Castrillón, and C. Guerra. DESEO: An active vision system for detection, tracking and recognition. In *Procs. of the Second International Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, October 1996.

[14] M. Castrillon, J. Lorenzo, M. Hernandez, and J. Cabrera. Before characterizing faces. In *IX Spanish Symposium on Pattern Recognition and Image Analysis*, Castellón, Spain, 2001.