Comparison of ARTMAP Neural Networks for Classification for Face Recognition from Video

M. Barry and E. Granger[1]
Laboratoire d'imagerie, de vision et d'intelligence artificielle
École de technologie supérieure, Montréal, Canada
mbarry@livia.etsmtl.ca, eric.granger@etsmtl.ca

Abstract

In video-based of face recognition applications, the What-and-Where Fusion Neural Network (WWFNN) has been shown to reduce the generalization error by accumulating a classifier's predictions over time, according to each individual in the environment. In this paper, three ARTMAP variants – fuzzy ARTMAP, ART-EMAP (Stage 1) and ARTMAP-IC – are compared for the classification of faces detected in the WWFNN. ART-EMAP (stage 1) and ARTMAP-IC expand on the well-known fuzzy ARTMAP by using distributed activation of category neurons, and by biasing distributed predictions according to the number of time these neurons are activated by training set patterns. The average performance of the WWFNNs with each ARTMAP network is compared to the WWFNN with a reference k-NN classifier in terms of generalization error, convergence time and compression, using a data set of real-world video sequences. Simulations results indicate that when ARTMAP-IC is used inside the WWFNN, it can achieve a generalization error that is significantly higher (about 20% on average) than if fuzzy ARTMAP or ART-EMAP is used. Indeed, ARTMAP-IC is less discriminant than the two other ARTMAP networks in cases with complex decision bounderies, when the training data is limited and unbalanced, as found in complex video data. However, ARTMAP-IC can outperform the others when classes are designed with a larger number of training patterns.

1. Introduction

Face recognition has received considerable attention over the past decade because of the wide range of commercial and law enforcement applications and the availability of affordable technology. In addition, face acquisition does not depend on the cooperation of individuals. As a result, face recognition remains a powerful tool in spite of the existence of other very reliable characteristics for biometric recognition such as iris scans and fingerprint analysis [13].

As shown in Fig.1, a video-based face recognition system applied to the identification of individuals will first perform segmentation to locate and isolate regions of interest (ROI) in successive video frames. Then, invariant and discriminant features will be extracted from the ROIs, and used by the recognition system to assign a class label to individuals. Face recognition from video is however a very challenging problem since frames provided by video sequences are typically low quality and generally small. Furthermore, images acquired in uncontrolled environments presents several technical challenges such as change in illumination, poses and occlusion.

A typical approach to recognizing faces in video consist in applying techniques developed for static images, once face detection has been performed. Over the last few years, several techniques have been proposed to recognize faces in static images. These approaches yield a high level of accuracy when operational environments are said to be constrained, where many assumption may be made about pose, illumination, facial expression, orientation and occlusion to provide accurate recognition. However, the performance of these techniques can degrade considerably when applied in unconstrained environments, as found in many video surveillance applications. This may be tied to the limited amount of training data from which recognition systems are designed.

More recently, some authors [1, 7, 11, 14] have attempted to exploit both spatial and temporal information contained in video sequences to provide a higher level of accuracy in unconstrained environment. For instance, a time series states space has been proposed by Zhou *et al.* [14] to fuse temporal information in video, which simultaneously characterizes the kinematics and identity of individuals in a probabilistic framework. A distributed sensor network (DSN) is proposed by Foresti [7] as a solution

[1]**Corresponding author**: École de technologie supérieure, 1100 Notre-dame Ouest, Montréal, Québec, H3C 1K3, Canada, eric.granger@etsmtl.ca, phone:1-514-396-8650, fax:1-514-396-8595

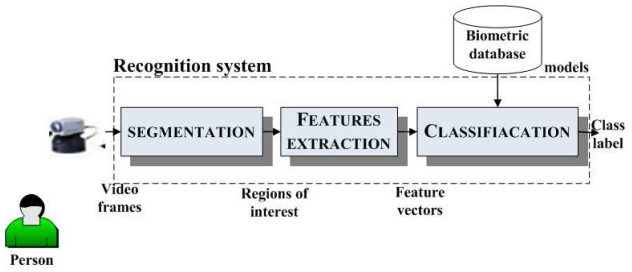


Figure 1. General system for face recognition in video

to the problem of partial occlusion that occurs in dynamics environments. Li and Chellappa [11] have introduced a face verification system from video sequences that exploits the trajectories of Gabor facial features. Finally, Barry and Granger [1] have applied the What-and-Where fusion neural network (WWFNN) to video-based face recognition. The WWFNN performs recognition by accumulating the responses of a classifier over time according to each individual in the environment. The prediction of the WWFNN is therefore the result of one or multiple responses by the classifier. From previous experiments on real video data [1], such accumulation has been shown to significantly reduce the generalization error.

In previous work, a particular realization of the WWFNN was considered for video-based face recognition. It used the fuzzy ARTMAP neural network for classification, and a bank of Kalman filters for tracking. This paper compares three ARTMAP variants – fuzzy ARTMAP, ART-EMAP (Stage 1) and ARTMAP-IC – for classification of faces detected in video sequences within the WWFNN. ARTMAP refers to a family of neural network architectures that can perform fast, stable, on-line, supervised or unsupervised, incremental learning, and classification. An attractive feature of ARTMAP neural networks is the ability to learn new data incrementally, without having to retrain on all cumulate training data, as would be the case with the Multi-Layer Perceptron. ARTMAP networks can also represent individual class by one or more prototypes, and lend themselves well to high speed processing, which make them suitable for both resource-limited and real-time face applications.

The performance of ARTMAP neural networks are assessed through computer simulation on complex real-world video data. The data set has been collected by the National Research Council of Canada [8], and corresponds to video sequences that display the face of a single person under different scenarios such as partial occlusion, pose, facial expression, motion, resolution and proximity. The average performance is assessed in terms of resources required during training and the generalization error during operation. A WWFNN with k-Nearest-Neighbor classifier is also included for reference.

This paper is organized as follow. In Section 2, the WWFNN applied to face recognition is briefly described. In Section 3, the ARTMAP neural network and the three variants considered in this study are briefly reviewed. In Section 4, the experimental methodology (data set, protocol and performance measures) employed to evaluate and compare performance are presented. Finally, in Section 5, simulations results are presented and discussed.

2. What-and-Where Fusion Neural Network

The What-and-Where fusion neural network [1] applied to face recognition from video is presented in Fig.2. It is composed of 3 modules: a classifier, a tracker and an evidence accumulation module.

During operations, the recognition system receives information provided by ROIs of successive video frames, which is then partitioned into two data streams called *What* and *Where*, and fed to the classification and tracking systems, respectively. The *What* parameters of an ROI characterizes the intrinsic properties of a face. In this paper, the *What* parameters are represented by the vectorized form of ROIs, $\mathbf{I} = [I_1, \ldots, I_l, \ldots, I_p]$, where I_i correspond to the gray level intensity of a pixel, and where $p = w$xh is the total number of pixels in a ROI of height h and of width w. In contrast, the *Where* data stream of an ROI indicates the position of the face in an environment, and it is represented by a vector $\mathbf{b} = [C, S]$, where C and S are the centroid and the size of the blob. The centroid $C(x, y)$ of a blob in a frame is defined by its 2D spatial coordinates, x and y, and the size $S(w, h)$ of a blob by its width w and its height h. It is important to note that the *Where* parameters are useless for face identification but necessary to resolve ambiguity such as occlusion in complex scene.

For each ROI, the evidence accumulator receives the output activation pattern $\mathbf{y}^{ab}$ from the classifier and the track number h furnished by the tracking sub-system. Based on these two responses, the accumulator will provide the most likely identity $\mathbf{y}^e$ of the faces detected in a scene. The tracking system uses *Where* parameters to pursue faces in a given scene, over successive frames. While color information, appearance, shape and facial features have been used to track faces in video, a blob-based tracking scheme has been used to pursue faces in this work. A new track is initialized for each newly-detected blob, and deleted whenever a person leaves the scene. The tracking system computes the most likely position of each face in the next frame according to previous observations. For each new frame, the tracker computes the distance between estimated and actual blob coordinates, and then provides the track number h associated with that face.

Prior to operations, a neural network classifier is trained in supervised learning mode with a representative data set. This data set consists of a variety of ROIs extracted from

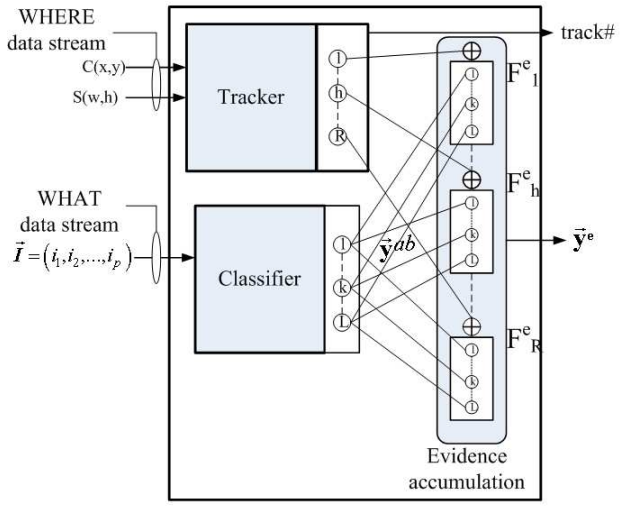


Figure 2. A What-and-Where fusion neural network for face recognition.

video frames for each individual. During operations, the classifier uses *What* parameters to predict the class associated with an input ROI. The output $\mathbf{y}^{ab}$ is a binary pattern of activity.

Predictions of the What-and-Where fusion neural network are provided via evidence accumulation. The accumulation module exploits the result of the tracking module to accumulate the responses of neural network classification. That is, the evidence accumulation module accumulate the classifier's responses over time according to each track. The prediction provided by the What-and-Where fusion neural network are therefore the result of one or multiple responses by the classifier. Evidence accumulation is implemented by means of identical evidence accumulation fields $F_1^e, F_2^e, \ldots, F_R^e$, where each field F_h^e is connected to a track h, and replicates the neural network classifier's output field, that is, contains L nodes, one per class. The classifier's output nodes are linked to their respective response nodes in all fields $F_h^e, h = 1, 2, \ldots, R$. Each field F_h^e is characterized by a field accumulation pattern $\mathbf{T}_h^e = (T_{h1}^e, T_{h2}^e, \ldots, T_{hL}^e)$. Upon initiation of track h, $\mathbf{T}_h^e$ is set equal to $\mathbf{0}$. When track $h = H$ is assigned to a ROI, F_H^e becomes active. The activity pattern $\mathbf{y}^{ab}$ output by the classifier accumulates onto F_H^e according to:

$$(\mathbf{T}_H^e)' = \mathbf{T}_H^e + \mathbf{y}^{ab} \tag{1}$$

Accumulation of ROI activity patterns in F_h^e continues until track h is deleted. For a given ROI, the activity pattern $\mathbf{y}^e$ output from evidence accumulation is equal to $\mathbf{T}_H^e$, and the individual is to be:

$$\mathbf{K}^e = \arg\max_{k^e}\{\mathbf{T}_{Hk^e}^e : k^e = 1, 2, \ldots, L\} \tag{2}$$

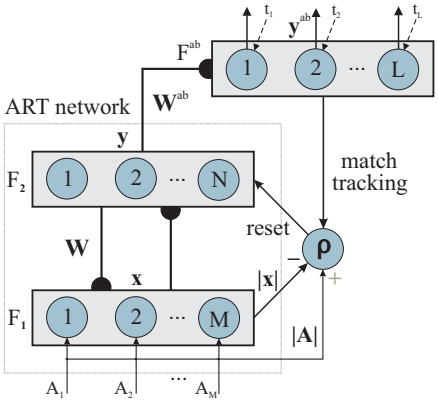


Figure 3. ARTMAP neural network.

3. Classification with ARTMAP Networks

ARTMAP network self-organize stable recognition categories in response to arbitrary sequences of input patterns [2]. The ARTMAP is often applied using the simplified version shown in Fig.3. It is obtained by combining an ART unsupervised neural network with a map field. The ART neural network consists of two fully connected layers of nodes: an M node input layer, F_1, and an N node competitive layer, F_2. A set of real-valued weights $\mathbf{W} = \{w_{ij} \in [0,1] : i = 1, 2, ..., M;\ j = 1, 2, ..., N\}$ is associated with the F_1-to-F_2 layer connections. Each F_2 node j represents a recognition category that learns a prototype vector $\mathbf{w}_j = (w_{1j}, w_{2j}, ..., w_{Mj})$. The F_2 layer is connected, through learned associative links, to an L node map field F^{ab}, where L is the number of classes in the output space. A set of binary weights $\mathbf{W}^{ab} = \{w_{jk}^{ab} \in \{0,1\} : j = 1, 2, ..., N;\ k = 1, 2, ..., L\}$ is associated with the F_2-to-F^{ab} connections. The vector $\mathbf{w}_j^{ab} = (w_{j1}^{ab}, w_{j2}^{ab}, ..., w_{jL}^{ab})$ links F_2 node j to one of the L output classes.

During training, ARTMAP classifiers perform incremental supervised learning of the mapping between training set vectors $\mathbf{a} = (a_1, a_2, ..., a_m)$ and output labels $\mathbf{t} = (t_1, t_2, ..., t_L)$, where $t_K = 1$ if K is the target class label for $\mathbf{a}$, and zero elsewhere. The following algorithm describes the operation of an ARTMAP classifier in learning mode:

1. **Initialization:** Initially, all the neurons of F_2 are uncommitted, all weight values w_{ij} are initialized to 1, and all weight values w_{jk}^{ab} are set to 0. Values of parameters α, ϵ, β and $\bar{\rho}$ are also set.

2. **Complement coding:** When a training pair $(\mathbf{a}, \mathbf{t})$ is presented to the network, $\mathbf{a}$ undergoes preprocessing, and yields pattern $\mathbf{A} = (A_1, A_2, ..., A_{M'})$. The vigilance parameter ρ is reset to its baseline value $\bar{\rho}$.

3. **Prototype selection:** Pattern $\mathbf{A}$ activates layer F_1 and is propagated through weighted connections $\mathbf{W}$

to layer F_2. Activation of node j in the F_2 layer is determined by the *choice function*. The F_2 layer produces a binary, winner-take-all pattern of activity $\mathbf{y} = (y_1, y_2, ..., y_N)$ such that only node $j = J$ with the greatest activation value remains active $(y_J = 1)$. Node J propagates its prototype vector w_J back onto F_1 and the *vigilance test* is performed. This test compares the degree of matching between $\mathbf{w}_J$ and $\mathbf{A}$ to the vigilance parameter $\rho \in [0, 1]$. If this test is satisfied, node J remains active and resonance is said to occur. Otherwise, the network inhibits the active F_2 node and searches for another node J that passes the vigilance test. If such a node does not exist, an uncommitted F_2 node become active and undergoes learning (Step 5).

4. **Class prediction:** Pattern **t** is fed directly to the map field F^{ab}, while the F_2 activity pattern $\mathbf{y}$ is propagated to the map field via associative connections $\mathbf{W}^{ab}$. The latter input activates F^{ab} nodes according to the *prediction function*, and the most active F^{ab} node K yields the class prediction $(K = k(J))$. If node K constitutes an incorrect class prediction, a *match tracking* signal raises vigilance just enough to induce another search among F_2 node (Step 3). This search continues until either an uncommitted F_2 node becomes active (learning ensues at Step 5), or a node J that has previously learned the correct class prediction K becomes active.

5. **Learning:** Learning input **a** involves updating prototype vector $\mathbf{w}_J$, and, if J corresponds to a newly-committed node, creating a permanent associative link to F^{ab}. A new association between F_2 node J and F^{ab} node $K(k(J) = K)$ is learned by setting $w^{ab}_{jk} = 1$ for $k = K$, where K is the target class label for **a**.

Once the weights ($\mathbf{W}$ and $\mathbf{W}^{ab}$) have converged for the training set patterns, ARTMAP can predict a class label for an input pattern by performing Steps 2, 3 and 4 without any testing. A pattern $\mathbf{a}$ that activates node J is predicted to belong to the class $K = k(J)$.

Although the first ARTMAP [2] classifier is limited to processing binary-valued input patterns, the ART-EMAP (stage 1) [4], ARTMAP-IC [3] and fuzzy ARTMAP [5] can process both analog and binary-valued input patterns by employing fuzzy ART as the ART network. With the fuzzy ART network, the input patterns goes through a transformation called complement coding, which doubles their number of components and becomes therefore, $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) = (a_1, a_2, ..., a_M; a^c_1, a^c_2, ..., a^c_M)$, where $a^c_i = (1 - a_i)$, and $a_i \in [0, 1])$. With complement coding and fast learning $(\beta = 1)$, fuzzy ART represents category j as hyperrectangle R_j that just encloses all the training patterns $\mathbf{a}$ to which it has been assigned.

ART-EMAP (stage 1) and ARTMAP-IC are extensions of fuzzy ARTMAP that produce a binary winner-take-all pattern $\mathbf{y}$ when training, but uses distributed activation of coded F_2 nodes when testing. ARTMAP-IC is more extended by biasing distributed test set predictions according to the number of times F_2 nodes are assigned to training set patterns[2]. Table 1 presents the equations used by the three ARTMAP networks for different steps of the algorithm.

4. Experimental Methodology

In order to compare the performance of WWFNNs using fuzzy ARTMAP, ART-EMAP (stage 1) and ARTMAP-IC classifiers, computer simulations were performed using a complex real-world data base of video streams. Prior to simulation trials, this dataset was normalized using the min-max technique, and partitioned into two parts – training and test subsets. Each subset contains an equal proportion of images from each class. The experimental protocol used in this paper is 10-fold cross validation. This strategy partition the training subset into 10 equal subsets. Over 10 trials, each fold is successively used as a validation subset, while the 9 remaining folds are used for training. Fuzzy ARTMAP, ART-EMAP (stage 1) and ARTMAP-IC neural networks are trained using the particle swarm optimization learning strategy [9]. This learning strategy select the network parameters values to minimize generalization error. The 4-dimensional search space for PSO learning strategy was set to the following range: $\beta \in [0, 1]$, $\alpha \in [0.00001, 1]$, $\bar{\rho} =\in [0, 0.999]$, and $\epsilon =\in [-1, 1]$. Each simulation trial was performed with 60 particles, and ended after 100 iterations, or when the best particle position remains unchanged for 5 consecutive iterations. Finally, a bank of Kalman filters was used for tracking faces in the WWFNN.

Average results, with corresponding standard error, are always obtained, as a result of the 10 independent simulation trials. The non-parametric k-Nearest-Neighbor (k-NN) [6] classifier is included for reference. For each computer simulation, the value of k employed with k-NN was selected among 1 through 10, using 10-fold validation. During each simulation trial, the performance of the k-NN classifier, the fuzzy ARTMAP, the ART-EMAP (stage 1) and the ARTMAP-IC neural networks are compared from a perspective of different ROI scaling sizes. To assess the impact on performance of ROI normalization, the size of the ROI containing the face furnished by the detection is scaled using a bi-linear interpolation, from an ROI of 10x10 to an ROI of 60x60.

The dataset used for computer simulations was collected by the National Research Council (NRC) [8]. It contains 22

[2] ARTMAP-IC also involves Negative Match Tracking (MT^-), where the match tracking parameters takes small negatives values, $\epsilon \leq 0$. MT^- has not been included in this study since it has been associated to a higher generalization error in some applications [10].

Table 1. Equations used by the three ARTMAP networks: $|\cdot|$ is the norm operator ($|\mathbf{w}_j| \equiv \sum_{i=1}^{2M} |w_{ij}|$), $\wedge$ is the fuzzy AND operator ($(\mathbf{A} \wedge \mathbf{w}_j)_i \equiv min(A_i, w_{ij})$), α is the *choice parameter*, β is the *learning rate parameter*, ϵ is the *match tracking parameter*, $\bar{\rho}$ is the *baseline vigilance parameter*, Q is the number of F_2 category with the greated activation T_j, and c_j is the number of training pattern that activate F_2 node j.

Algorithmic step	**Training phase**	**Testing phase**
1. Initialization:	$\alpha > 0, \beta \in [0,1], \bar{\rho} = 0, \epsilon = 0^+$	N/A
2. Complement coding:	$\mathbf{A} = (\mathbf{a}, \mathbf{a^c})(\mathbf{M'} = \mathbf{2M})$	$\mathbf{A} = (\mathbf{a}, \mathbf{a^c})(\mathbf{M'} = \mathbf{2M})$
3. Prototype selection:		
- choice function	$T_j(\mathbf{A}) = \lvert\mathbf{A} \wedge \mathbf{w}_j\rvert/(\alpha + \lvert\mathbf{w}_j\rvert)$	$T_j(\mathbf{A}) = \lvert\mathbf{A} \wedge \mathbf{w}_j\rvert/(\alpha + \lvert\mathbf{w}_j\rvert)$
- vigilance test	$\lvert\mathbf{A} \wedge \mathbf{w}_j\rvert > \rho M$	N/A
- F_2 activation	$y_j = 1$ only if $j = J$	$y_j = 1$ only if $j = J$
4. Class prediction:		
- prediction function		
- FAM	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$
- ART-EMAP	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$	$S_k^{ab}(\mathbf{y}) = \frac{\sum_{j\in Q} w_{jk}^{ab} T_j}{\sum_{k=1}^{L}\sum_{j\in Q} w_{jk}^{ab} T_j}$
- ARTMAP-IC	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$	$S_k^{ab}(\mathbf{y}) = \frac{\sum_{j\in Q} w_{jk}^{ab} c_j T_j}{\sum_{k=1}^{L}\sum_{j\in Q} w_{jk}^{ab} c_j T_j}$
- match tracking	$\rho' = (\lvert\mathbf{A} \wedge \mathbf{w}_J\rvert/M) + \epsilon$	N/A
5. Learning:		
- prototype update	$\mathbf{w}'_J = \beta(\mathbf{A} \wedge \mathbf{w}_J) + (1-\beta)\mathbf{w}_J$	N/A
- instance counting	$c_J = c_J + 1$	N/A

video sequences captured with an Intel webcam mounted on a computer monitor. Each sequence have an average duration of 12 seconds, and contains an average of 300 frames. It contains the face of one among eleven individuals sitting in front of a computer and exhibiting a wide range of facial expressions, pose and motions. The detection process[3] yields 300 ROIs - between 29x18 and 132x119 ROIs per individual. There are two sequences per individual, one dedicated to training and the other to testing. The video sequences are taken under approximately the same illumination conditions (no sunlight, only ceiling light evenly distributed over the room), the same setup and almost the same background, for all persons in the data base. Furthermore, the video capture have two different resolutions 160 x 120 and 320 x 240 and each face occupies between $\frac{1}{4}$ to $\frac{1}{8}$ of the image. Finally, this dataset contains a variety of challenging scenarios such as low resolution, motion blur, out of focus factor, facial orientation, facial expression and occlusion.

The average performance of classifiers was assessed in terms of resources required during training and its generalization error on the test set. The amount of resources required during training is measured by compression and convergence time. *Compression* refers to the average number of training patterns per category prototype created in the competition layer F2. *Convergence time* is the number of epochs required to complete learning for a learning strategy. It does not include presentations of the validation subset used to perform hold-out validation. *Generalization error* is estimated as the ratio of incorrectly classified test subset patterns over all test set patterns. The combination of compression and convergence time provides useful insight into the amount of processing during the training phase to produce its best asymptotic generalization error. Compression is directly related to memory resources required for recog-

[3]Note that faces in frames were detected using the Haar coefficient approach [12].

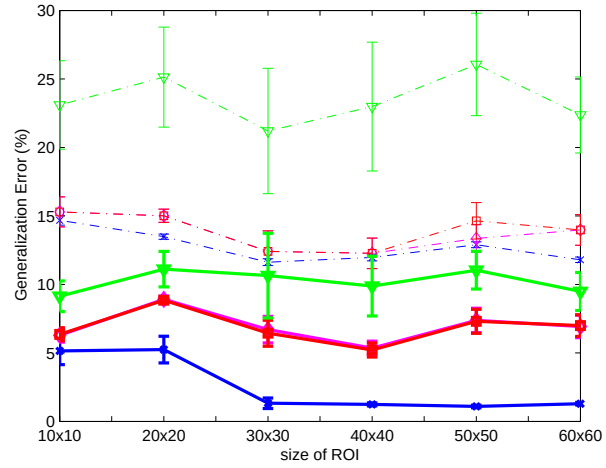
nition, and to the computational time during operation.

5. Simulations Results

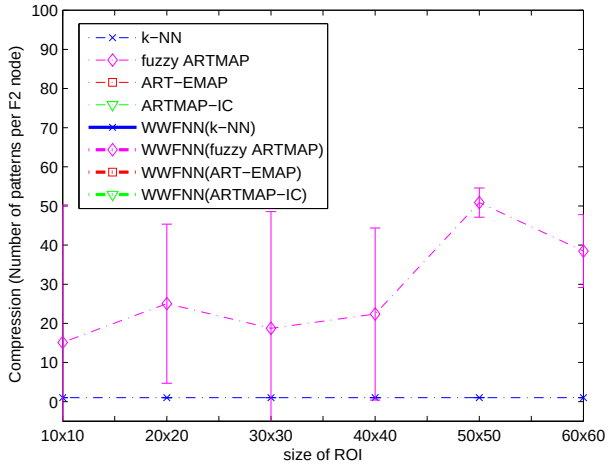
Fig.4 presents the average performance of fuzzy ARTMAP, ART-EMAP (stage 1) and ARTMAP-IC neural networks and k-NN classifier, alone and within the WWFNN, as a function of the ROI scaling size. Simulation trials (not presented in this paper) have shown that, with this dataset, the Q-max rule [3] for distributing F_2 layer activation in ART-EMAP and ARTMAP-IC classifiers gives better results than the threshold rule [4]. The choice of Q that was found to give good results for the video data is $Q = \min\left[\frac{N_c}{L}, 3\right]$, where $L = 11$ is the number of classes and N_c is the number of committed F_2 nodes. In particular, higher values of Q tend to degrade performance considerably. Fig.4(a) indicates that fuzzy ARTMAP and ART-EMAP perform better than ARTMAP-IC for this data set, and yield an average generalization error of about 13% over ROI size, in comparison to 22% for ARTMAP-IC. The k-NN yields the lowest average generalization error of about 12%. On average, the three ARTMAP networks converge after about 552 epochs of the PSO training strategy (60 particles x 9.2 iterations x 1 epoch) and, give a compression that varies between 15 (for an ROI of 10x10) and 50 (for an ROI of 60x60) training patterns per F_2 nodes. ARTMAP networks represent a good alternative when the amount of ressouces and computational time is limited. For example, with an ROI of 60x60, ARTMAP corresponds to about 40 times less memory and matching operations comparatively to k-NN.

The WWFNN allows to achieve an average generalization error that is significantly lower than one of the classifiers alone. This error is about 6% for fuzzy ARTMAP and ART-EMAP, which correspond to a 50% improvement in accuracy over these networks alone when the ROI scaling size is 60x60. In contrast, the generalization error tends toward 1% for k-NN as the ROI size grows. Finally, the WWFNN with ARTMAP-IC yields an error of about 10%. This lower accuracy of ARTMAP-IC is cause by the use of the frequency information (c_j values) when computing predictions. Note that the dataset used to evaluate the performance is unbalanced (*i.e.*, the training set has a different number of ROIs per class, due to the limitations in the detection process).

Tables 2 and 3 reveal that ARTMAP-IC tends to favor classes with more training patterns. These tables present the average confusion matrices associated with k-NN, fuzzy ARTMAP, ART-EMAP and ARTMAP-IC classifiers alone (Table 2), and when used inside the WWFNN (Table 3). In these tables, most of the test patterns are assigned to classes 3, 5, 7 and 10, have at least 160 training patterns. Note that in Table 2, for the classes 2 and 8, the average generalization error per class are 81% and 42.1%, respectively, and both



(a) Generalization Error vs ROI size



(b) Compression vs ROI size

Figure 4. Average performance of fuzzy ARTMAP, ART-EMAP, ARTMAP-IC and k-NN classifiers, alone and within the WWFNN, versus the ROI scaling size for all NRC training data. Note that fuzzy ARTMAP, ART-EMAP and ARTMAP-IC provide the same compression and convergence time. (error bars are standard error).

have been trained with less than 100 patterns. However, given the cost of data collection process, and shortcomming of face detection (segmentation) algorithms, limited and unbalanced data sets may be common. Moreover, the data set used for training is relatively small when one considers the complexity of the environment, and the number of features typically used with video data. Biasing class predictions in video data with unbalanced training data classes does not appear to yield any performance improvement.

Fig.5 shows the occurrence of prediction errors in time associated with the WWFNN using the three ARTMAP neural networks in this study, for an ROI size of 60x60, and for videos 2 and video 8. The classes corresponding to these two video are designed with fewer training pat-

terns (36 patterns for class 2 and 90 patterns for class 8) than the other videos. In Fig.5(a)(b) and Fig.5(c)(d), the fuzzy ARTMAP and ART-EMAP give erroneous prediction on few frames. For example, as shown on Fig.5(b) and Fig.5(d), in video 8, the individual presents abrupt motions between frames 60 to 100. The impact of these patterns are reduced by the WWFNN, and completely attenuated in the case of video 2 as shown in Fig.5(a) and Fig.5(c). However, as shown in Fig.5(e)(f), ARTMAP-IC misclassifies most of the test patterns over time. Since WWFNN prediction is based on the accumulation of ARTMAP-IC predictions according to track, successive prediction errors by ARTMAP-IC will tend to accumulate errors over time.

Although weighting predictions according to the number of training set patterns may not appropriate in certain contexts, it may be useful when the training data is larger and more representative of the environment. As shown in Table 2 and Table 3, ARTMAP-IC perform better than the two other ARTMAP networks for videos 7 and 10. The classes defined by these two videos are designed with a high number of training patterns. Fig.6 presents the occurrence of errors in time associated with the WWFNN using the three ARTMAP networks, for an ROI size of 60x60, and for videos 7 and 10. In these two videos, the individuals vary proximity to the camera, show different facial expression and orientation, and out of focus factor. The individual presents abrupt motions yielding partial face patterns in video 10. As shown in Fig.6, ARTMAP-IC considerably reduces the impact of this variability. For video 7, the ARTMAP-IC prediction error is almost 0%. ARTMAP-IC provides lower generalization than fuzzy ARTMAP and ART-EMAP when the classes are well defined by training data.

6. Conclusion

A particular realization of the What-and-Where Fusion Neural Network was previously applied to face recognition in video sequences. In this paper, three ARTMAP neural networks – fuzzy ARTMAP, ART-EMAP (Stage 1) and ARTMAP-IC – have been compared for for classification of detected faces within this framework. Their performance has been assessed in terms of resources during training and the generalizations error achieved during operations, through computer simulation on complex real-world video data.

Simulations results indicate that fuzzy ARTMAP and ART-EMAP (Stage 1) yield a significantly lower generalizations error than ARTMAP-IC, over a wide range of ROI scaling sizes. The distributed activation of coded F_2 category nodes has no significant effect on the test set predictions on this data set, since ART-EMAP performs as well as fuzzy ARTMAP. However, the instance counting procedure used in the ARTMAP-IC increases the generalizations error considerably on the video data. A more detailed analysis of errors has revealed that this is linked to the use of limited and unbalanced training data set. In fact, ARTMAP-IC tends to outperform the others when classes are designed with a larger number of training patterns.

References

[1] M. Barry and E. Granger. Face recognition in video using a what-and-where fusion neural network, to appear, 2007. *Int'l Joint Conference on Neural Network*, Orlando, USA, 12-17 August.

[2] G. A. Carpenter, S. Grossberg, and J. H. Reynolds. Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4(1):565–588, 1991.

[3] G. A. Carpenter and N. Markuzon. Artmap-ic and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, 11(1):323–336, 1998.

[4] G. A. Carpenter and W. D. Ross. Art-emap: A neural network architecture for object recognition by evidence accumulation. *IEEE Trans. on Neural Networks*, 6(4):805–818, 1995.

[5] S. Chen, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen. Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Trans. on Neural Networks*, 3(1):698–713, 1992.

[6] T. M. Cover and P. E. Hart. Nearest neighbor patterns classification. *IEEE Trans. Information Theory*, 13:21–27, 1967.

[7] G. L. Foresti. Detecting multiple objects under partial occlusion by integrating classification and tracking approaches. *Int'l Jounal of Imaging Systems and Technology*, 11(1):263–276, 2000.

[8] D. Gorodnichy. Video-based framework for face recognition in video, 2005. *Computer and Robot Vision*, 330–338, Victoria, Canada, 9-11 May.

[9] E. Granger, P.Henniges, R. Sabourin, and L. S. Oliveira. Particle swarm optimisation of fuzzy artmap parameters, 2006. *Int'l Joint Conference on Neural Network*, Vancouver, Canada, 16-21 July.

[10] P. Henniges, E. Granger, R. Sabourin, and L. S. Oliveira. Impact of fuzzy artmap match tracking strategies on the recognition of handwritten digits, 2006. *Artificial Neural Networks in Engineering Conf.*, St-Louis, USA, 5-8 November.

[11] B. Li and R. Chellappa. Face verification through tracking facial features. *Jounal of the Optical Society of America*, 18(1):530–544, 2001.

[12] R. Lieanhart and J. Maydt. An extended set of haar-like features for rapid object detection. *Int'l Conf. on Image Processing*, 1(1):900–903, 2002.

[13] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. survey*, 35(4):399–458, 2003.

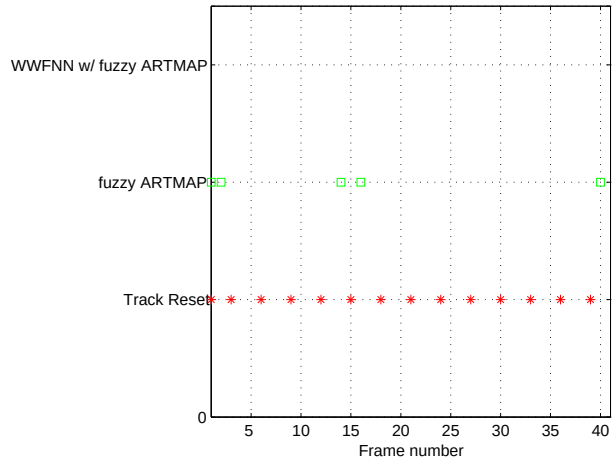
[14] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. on Image Processing*, 13(11):263–276, 2004.

Table 2. Average confusion matrix associated with fuzzy ARTMAP, ART-EMAP (stage 1), ARTMAP-IC and *k*-NN classifiers obtained on test data for an ROI size of 60x60.

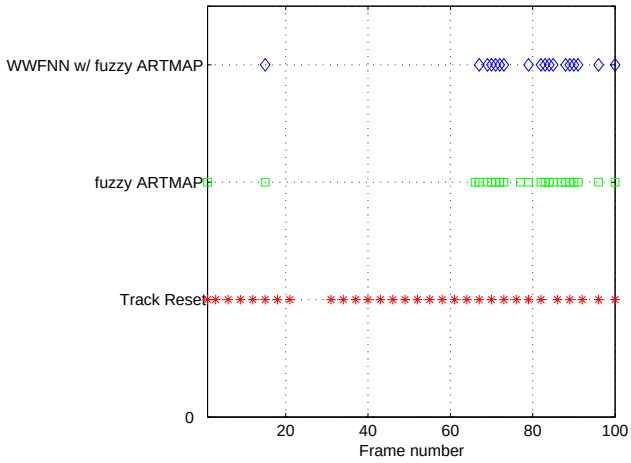
True Classes	Classifiers	Predicted classes											Error per class (%)
		1	2	3	4	5	6	7	8	9	10	11	
1	*k*-NN	135.7	2.1	0	0	0	0	0	1.7	0.3	0.1	2.1	4.4
	fuzzy ARTMAP	110.8	1.6	0	0.8	17.9	0.2	1.5	0.6	3.1	0	5.5	22
	ART-EMAP	108.3	1.6	0.1	0.8	17.7	0.3	1.7	1.3	3.6	0.1	6.5	23.7
	ARTMAP-IC	110.7	0.2	0.3	0.9	12.1	0.3	4.4	1.1	4.3	1	6.7	22
2	*k*-NN	0	40.6	0.1	0	0	0	0.1	0	0.2	0	0	1
	fuzzy ARTMAP	0	31.1	0.8	0	4.5	0	2.6	0	0.1	0	1.9	24.1
	ART-EMAP	0.2	36.2	0.7	0	2.8	0	0.8	0	0.1	0	0.2	11.7
	ARTMAP-IC	10	7.8	0.5	0	6.8	0	3.6	0.1	0.5	1.2	10.5	81
3	*k*-NN	0	0	156.1	0	0	0	0	3.9	0	0	0	2.4
	fuzzy ARTMAP	0	0	158.6	0	0.7	0	0	0.6	0	0.1	0	0.9
	ART-EMAP	0	0	158.1	0	0.8	0	0	0.7	0	0.4	0	1.2
	ARTMAP-IC	0	0	158.4	0	0.2	0.2	0.1	1.1	0	0	0	1
4	*k*-NN	0	1.8	0	81.6	0	0.1	3.1	0	0	0.8	43.6	37.7
	fuzzy ARTMAP	0	0	0	97.7	1.2	0.7	0	0	0	0	31.4	25.4
	ART-EMAP	0	0	0	97.8	0.4	0.9	0	0	0	0	31.9	25.3
	ARTMAP-IC	0.4	0.1	0	77.5	0	1	0	0	0.3	5.7	46	40.8
5	*k*-NN	0	0	0	0.1	168.2	0.2	19.9	0	5.6	0	0	13.2
	fuzzy ARTMAP	0	0	1.5	0	177.8	0	10.3	0	4	0.4	0	8.6
	ART-EMAP	0	0	0.6	0	177	0	12.4	0	4	0	0	8.8
	ARTMAP-IC	0	0	1.7	0	141.8	0.2	44.8	0	5.1	0.4	0	26.9
6	*k*-NN	1.6	7.2	0	0	0.4	116.3	3.6	0.2	0.1	2.1	2.5	13.2
	fuzzy ARTMAP	0	0.8	0	0.8	2.9	128.3	0.2	0	0	1	0	4.3
	ART-EMAP	0	0.2	0	1.7	3.4	127.2	0	0	0	1.5	0	5.1
	ARTMAP-IC	0	0.4	0.3	5.5	0.7	104.2	0.3	0	0.5	22.1	0	22.2
7	*k*-NN	1.9	9.6	0	0	1.6	0	178.7	1.1	1	0	0.1	7.9
	fuzzy ARTMAP	1.3	0.4	0	0.2	28	0	162.6	1.1	0.4	0	0	13.5
	ART-EMAP	0.7	0	0	0	25.1	0.2	166.2	1.5	0.3	0	0	11.6
	ARTMAP-IC	1	0	0	0.1	6.6	0	184.7	1.2	0.4	0	0	4.8
8	*k*-NN	0	1.3	0	0	0	0	0	98.7	0	0	0	1.3
	fuzzy ARTMAP	0	0	7.7	0	0.4	0	0.3	91.1	0.1	0.3	0.1	8.9
	ART-EMAP	0	0	11.4	0	0.1	0	0	88.2	0.2	0.1	0	11.8
	ARTMAP-IC	0	0.1	37	0	0	0	0	58.8	1.7	1.8	0.4	41.2
9	*k*-NN	4.7	0	1.1	0	0	0.1	0	0	180.3	0	1.8	4.1
	fuzzy ARTMAP	0.6	0	0	0.5	0	0.3	0	0	180.1	0.9	5.6	4.2
	ART-EMAP	0.6	0	0	0.2	0	0.4	0	0	181.7	0.6	4.5	3.4
	ARTMAP-IC	1.2	0	0	0.1	0	0.6	0	0	178.8	0.7	6.6	4.9
10	*k*-NN	1	8.6	3.1	0	0	40.4	0	2.1	0	110.6	3.2	34.6
	fuzzy ARTMAP	0	0.7	2.9	0	2.3	57.3	0	3.1	0	101.2	1.5	40.1
	ART-EMAP	0	0.3	4	0	2.2	56.5	0	3.7	0.4	100.5	1.4	40.5
	ARTMAP-IC	0	0.4	3.9	0	0	37.7	0	2.8	3.9	117.3	3	30.6
11	*k*-NN	0	0.4	0	0	0	0	2.1	0	0	0.1	144.4	1.8
	fuzzy ARTMAP	0	0	0	0	3.5	0	7	0	0	0	136.5	7.1
	ART-EMAP	0	0	0	0	4.6	0.1	7.2	0	0	0	135.1	8.1
	ARTMAP-IC	8.6	0.1	0.1	0	1.4	0.2	30.7	0	3.8	0.1	102	30.6

Table 3. Average confusion matrix associated with a What-and-Where fusion neural network using fuzzy ARTMAP, ART-EMAP (stage 1), ARTMAP-IC and k-NN classifiers obtained on test data for an ROI size of 60x60.

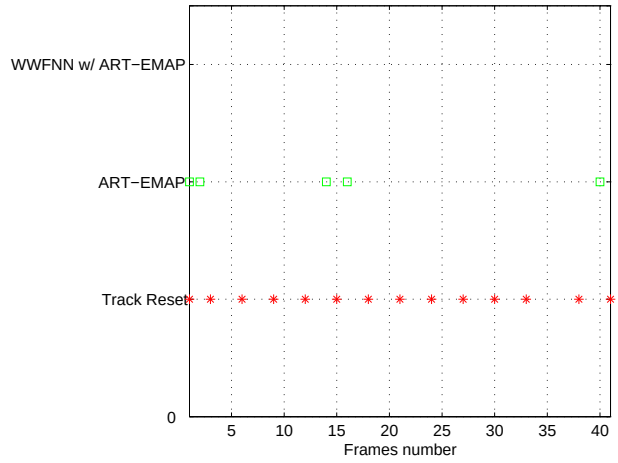
True Classes	Classifiers in WWFNN	Predicted classes											Error per class (%)
		1	2	3	4	5	6	7	8	9	10	11	
1	k-NN	130.1	5.2	0.8	1.7	0.8	0.1	0.8	0	0.2	0	2.3	8.4
	fuzzy ARTMAP	142	0	0	0	0	0	0	0	0	0	0	0
	ART-EMAP	142	0	0	0	0	0	0	0	0	0	0	0
	ARTMAP-IC	142	0	0	0	0	0	0	0	0	0	0	0
2	k-NN	0	40.9	0	0	0	0	0.1	0	0.1	0	0	0.2
	fuzzy ARTMAP	0	41	0	0	0	0	0	0	0	0	0	0
	ART-EMAP	0	41	0	0	0	0	0	0	0	0	0	0
	ARTMAP-IC	9.6	30.1	0.2	0	0	0	0	0	0.1	0	1	26.6
3	k-NN	0	0	157.6	0	0	0	0	1.9	0	0.5	0	1.5
	fuzzy ARTMAP	0	0	160	0	0	0	0	0	0	0	0	0
	ART-EMAP	0	0	160	0	0	0	0	0	0	0	0	0
	ARTMAP-IC	0	0	160	0	0	0	0	0	0	0	0	0
4	k-NN	0	0.1	0	109	0	0.3	0	0	0	3.2	18.4	16.8
	fuzzy ARTMAP	0	0	0	131	0	0	0	0	0	0	0	0
	ART-EMAP	0	0	0	131	0	0	0	0	0	0	0	0
	ARTMAP-IC	0.4	0.1	0	130	0	0	0	0	0	0	0.5	0.7
5	k-NN	0	0	0	0.2	179.6	0.1	10.5	0	3.6	0	0	7.4
	fuzzy ARTMAP	0	0	0.3	0	193.7	0	0	0	0	0	0	0.2
	ART-EMAP	0	0	0.5	0	193.5	0	0	0	0	0	0	0.3
	ARTMAP-IC	0	0	1.6	0	192.3	0	0.1	0	0	0	0	0.9
6	k-NN	0	1.3	0.3	0.8	0.2	121.5	4.7	0	2.4	1.3	1.5	9.3
	fuzzy ARTMAP	0	0	0	0.9	2.8	130	0	0	0	0.3	0	3
	ART-EMAP	0	0	0	1	2.7	130	0	0	0	0.3	0	3
	ARTMAP-IC	0	0.1	0.3	4.4	0.5	127.9	0	0	0.1	0.7	0	4.6
7	k-NN	0	0	0	0	2.1	0	191.8	0	0.1	0	0	1.1
	fuzzy ARTMAP	0.4	0	0	0	20.1	0.2	173.3	0	0	0	0	10.7
	ART-EMAP	0.5	0	0	0	19.5	0.2	173.8	0	0	0	0	10.4
	ARTMAP-IC	0.5	0	0	0	4.1	0	189.4	0	0	0	0	2.4
8	k-NN	0	0	0	0	0	0	0	99.5	0.5	0	0	0.5
	fuzzy ARTMAP	0	0	9.8	0	0	0	0	90.2	0	0	0	9.8
	ART-EMAP	0	0	9.8	0	0	0	0	90.2	0	0	0	9.8
	ARTMAP-IC	0	0.1	36.6	0	0	0	0.2	62.2	0.3	0.5	0.1	37.8
9	k-NN	0	0	0	0	0	0	0	0	183.9	0	4.1	2.2
	fuzzy ARTMAP	0.6	0	0	0.5	0	0.3	0	0	180.1	0.9	5.6	4.2
	ART-EMAP	0.6	0	0	0.2	0	0.4	0	0	181.7	0.6	4.5	3.4
	ARTMAP-IC	1	0	0	0	0	0.6	0	0	186.3	0	0.1	0.9
10	k-NN	0	1.7	0.4	0	0	1	0	3.3	0.7	159.2	2.7	5.6
	fuzzy ARTMAP	0	0.3	3.6	0	2	52	0	3.2	0.4	106.9	0.6	36.7
	ART-EMAP	0	0.3	3.6	0	1.7	52.1	0	0.3	0.4	107.1	0.5	36.6
	ARTMAP-IC	0	0.4	2.7	0	0	34	0	2.3	3.5	124.8	1.3	26.2
11	k-NN	0	0	0	0	0	0.4	0.1	0.4	0	0.2	145.9	0.7
	fuzzy ARTMAP	0	0	0	0	3.3	0	4.9	0	0	0	138.8	5.6
	ART-EMAP	0	0	0	0	3	0	4.9	0	0	0	139.1	5.3
	ARTMAP-IC	7.3	0	0.1	0	0.8	0.2	27.1	0	2.4	0	109.1	25.8



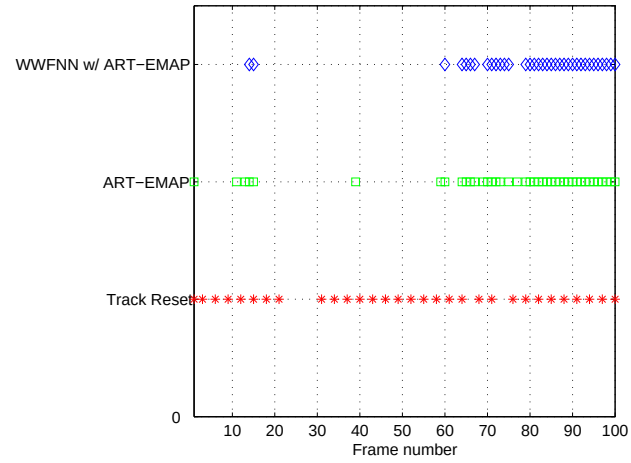
(a) Video sequence # 2



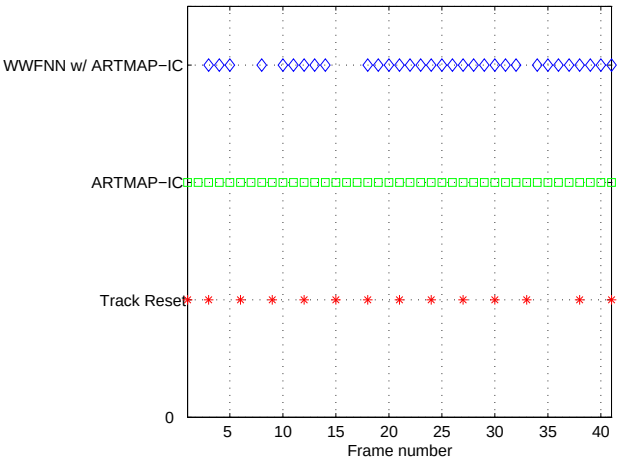
(b) Video sequence # 8



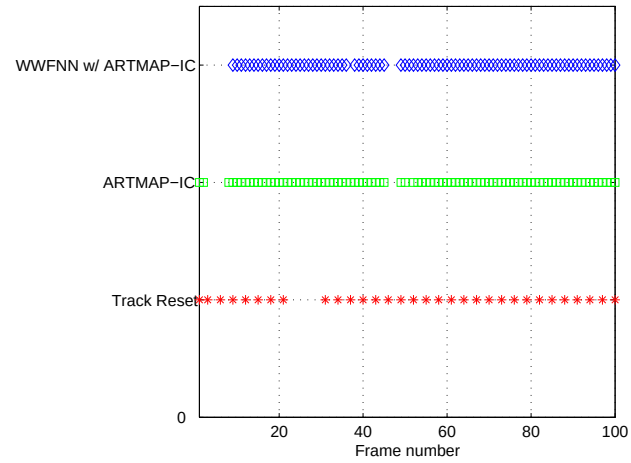
(c) Video sequence # 2



(d) Video sequence # 8

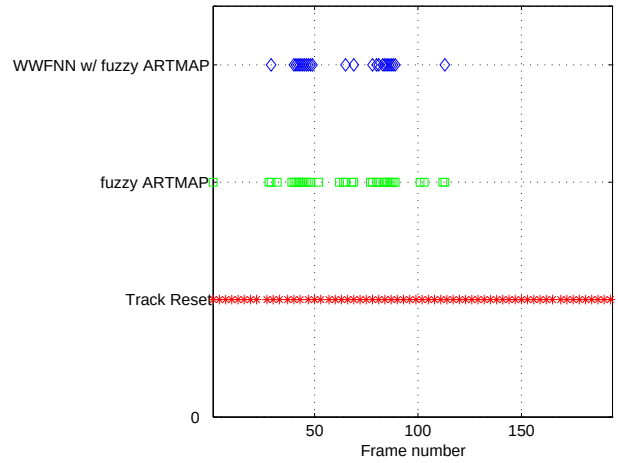


(e) Video sequence # 2

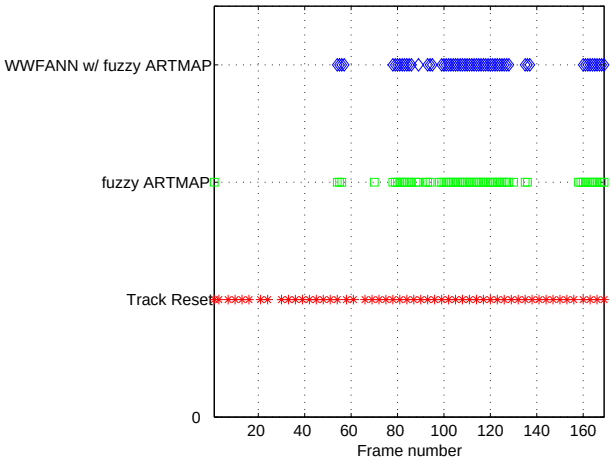


(f) Video sequence # 8

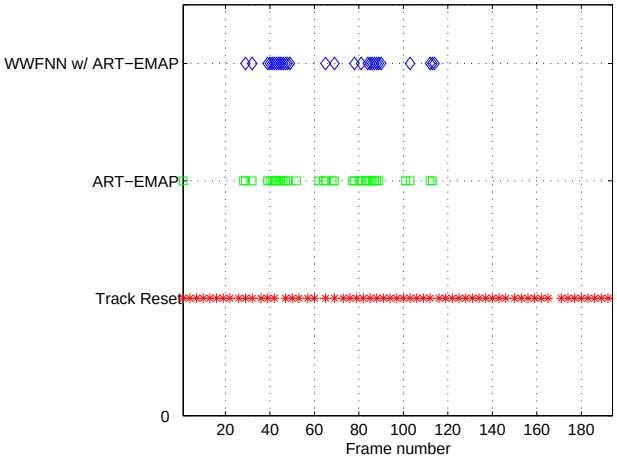
Figure 5. An example of the distribution of prediction errors over time with the WWFNN and the three ARTMAP variants alone when using an ROI scaling size of 60x60.



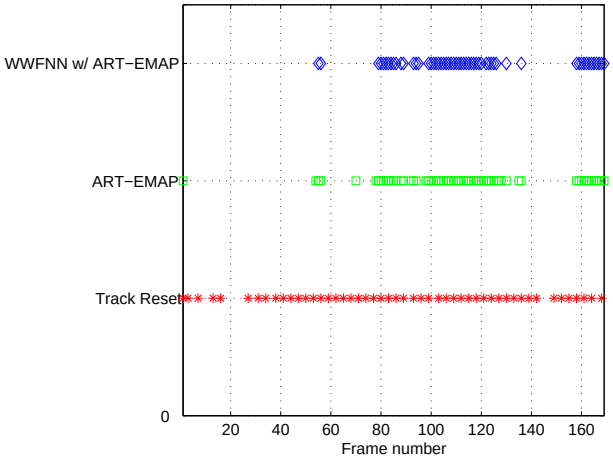
(a) Video sequence # 7



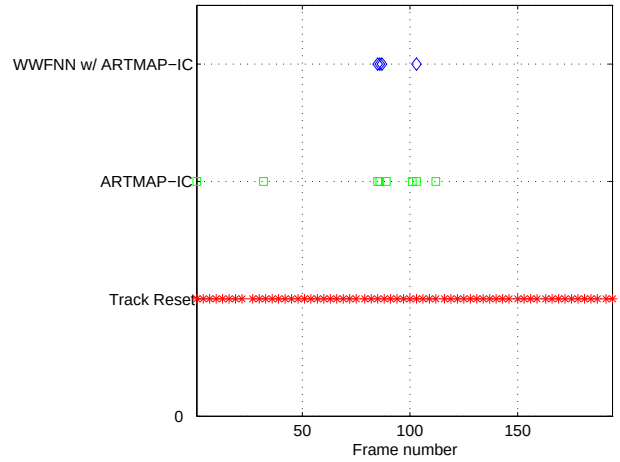
(b) Video sequence # 10



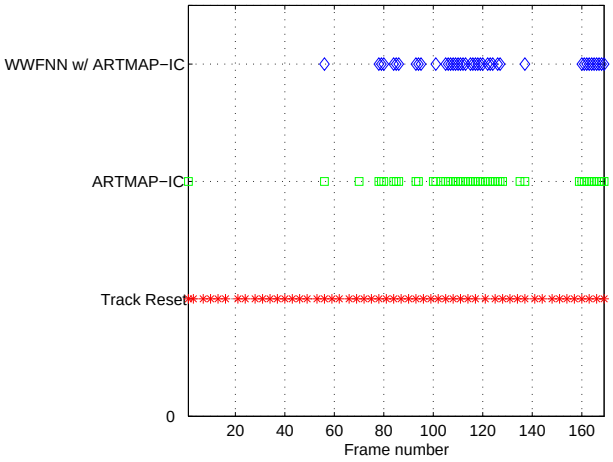
(c) Video sequence # 7



(d) Video sequence # 10



(e) Video sequence # 7



(f) Video sequence # 10

Figure 6. An example of the distribution of prediction errors over time with the WWFNN and the three ARTMAP variants alone when using an ROI scaling size of 60x60.