IMORL: Incremental Multiple-Object Recognition and Localization

Haibo He, *Member, IEEE*, and Sheng Chen, *Student Member, IEEE*

***Abstract*—This paper proposes an incremental multiple-object recognition and localization (IMORL) method. The objective of IMORL is to adaptively learn multiple interesting objects in an image. Unlike the conventional multiple-object learning algorithms, the proposed method can automatically and adaptively learn from continuous video streams over the entire learning life. This kind of incremental learning capability enables the proposed approach to accumulate experience and use such knowledge to benefit future learning and the decision making process. Furthermore, IMORL can effectively handle variations in the number of instances in each data chunk over the learning life. Another important aspect analyzed in this paper is the concept drifting issue. In multiple-object learning scenarios, it is a common phenomenon that new interesting objects may be introduced during the learning life. To handle this situation, IMORL uses an adaptive learning principle to autonomously adjust to such new information. The proposed approach is independent of the base learning models, such as decision tree, neural networks, support vector machines, and others, which provide the flexibility of using this method as a general learning methodology in multiple-object learning scenarios. In this paper, we use a neural network with a multilayer perceptron (MLP) structure as the base learning model and test the performance of this method in various video stream data sets. Simulation results show the effectiveness of this method.**

***Index Terms*—Adaptive learning, concept drifting, feature representation, incremental learning, multiple-object learning.**

I. Introduction

OBJECT recognition and scene analysis are fundamental issues for developing high-level intelligence because they play an essential role in perception, reasoning, action, and goal-oriented behaviors [1]–[3]. Over the past decades, this problem has attracted much attention in the machine intelligence, computer vision, and pattern recognition community. Many new theories, algorithms, and practical tools have been developed and successfully applied to a wide range of application domains. However, there is still a long way to go to achieve the long-term goal of understanding and developing self-adaptive systems that can replicate certain levels of brain-like intelligence. Recently, Wang presented a nice survey on the problem of time dimension for scene analysis from the computational point of view [4]. The binding problem, which means the capability of grouping different elements of a perceived scene into coherent objects, was analyzed in detail in that paper. Time dimension and temporal correction theory was discussed, and various issues related to figure-ground separation and scene segmentation were also addressed. Interested readers can refer to [4] for a comprehensive review of this problem.

In this paper, we focus on multiple-object learning for machine intelligence. This problem has recently attracted growing attention due to increased interest and the demand for many real-world applications, such as surveillance, mobile sensor networks, robotics, homeland security and defense, and many others. Fig. 1 shows an estimated number of publications in this domain over the past ten years based on the IEEE database.

Much research has been reported in the literature on multiple-object learning. In [5], a connectionist model for recognizing multiple objects was presented. This method provided insight into network structures for recognizing multiple objects related to memory-based reasoning, and built a connection between probabilistic measures and the connectionist learning paradigm. In [6], a lobula giant movement detector (LGMD)-based neural network was proposed for collision detection in complex dynamic scenes. Offline simulation tests and real-time robotics experiments illustrated the robustness and reliability of this method. In [7], a mixture model based on the probability density estimation of data points with scale and shift parameters was proposed for recognition of multiple objects in an image plane. An expectation–maximization (EM) algorithm was proposed to estimate such parameters. To effectively learn from multiple objects, an approach consisting of a growing network and an attention network was proposed in [8], in which the growing network separates the objects under attention from the background and the attention network provides further attention information for the growing networks. Experimental results demonstrated that this method can effectively learn from gray-scale images. Multiple moving object analysis was studied in [9] and [10]. For instance, in [9], feature trajectories in the image sequences were first analyzed by the cost minimization method using 2-D Hopfield networks, and then they were segmented into different moving objects by detecting neighboring correspondences. In [10], a method with two distinct cameras was proposed to detect moving objects in a video stream. Frame difference was used to identify interesting objects from the background, and then such information was analyzed by a clustering algorithm to identify the number, the size, and the position of moving objects. Both indoor and outdoor experiments showed that this system can provide improved discrimination under certain conditions. Recently, the scale invariant feature transform (SIFT) has demonstrated great success for object recognition from static images [11], [12]. The key idea of SIFT is to transform an image into a collection

Manuscript received December 17, 2007; revised April 4, 2008; accepted June 19, 2008. First published September 26, 2008; current version published October 8, 2008.

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: hhe@stevens.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2001774

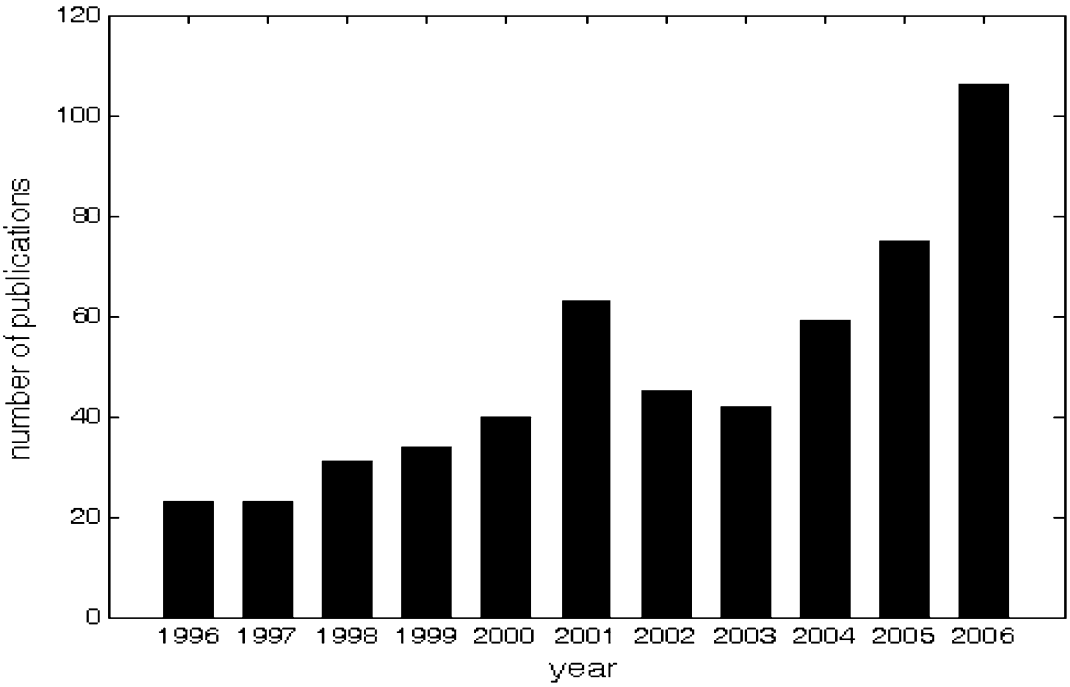

Fig. 1. Number of publications on multiple-object learning (based on IEEE database).

of local feature vectors that are invariant to image translation, scaling, and rotation. Extension work based on SIFT was also developed. For instance, in [13], principal components analysis (PCA) was integrated into standard SIFT representation, and a method named PCA–SIFT was proposed. In this method, PCA was applied to the normalized gradient patch to provide a more distinctive, robust, and compact performance of the image registration and object recognition. In [14], PCA–SIFT was used to detect and localize multiple objects for a humanoid vision system. This approach considered each video frame an independent observation of feature descriptors, and included a separate training and testing stage. During the training stage, PCA–SIFT features were used to learn the representations of an object, and then such features were used to detect and localize multiple interesting objects during the testing stage. In [15], a generative model was proposed for multiple-object classes detection. This method uses a codebook representation for image recognition and a probabilistic model for multiple-object detection in the same image. Simulation results over a wide range of scales, in-plane rotations, background clutter, and partial occlusions illustrated the effectiveness of this method. An important problem in multiple-object learning is to identify the interesting parts of an image (corresponding to individual objects) and the irrelevant clutter (not associated with any object). In [16], an empirical study of the bottom-up attention base method to extract useful information regarding location, size, and shape of multiple objects was presented. Given the set of reference views, a probabilistic model for automatic object recognition and segmentation was proposed in [17]. A neural network model for learning location invariance was considered in [18], where a feedforward network was used for object identification while a feedback network was developed for localization.

Multiple-object tracking also attracted much attention in the recent literature. For instance, a linear programming relaxation approach for multiple-object tracking was proposed in [19]. This method considers the tracking as a multipath searching problem by explicitly modeling track interaction and object mutual occlusion. Experimental results show that this method is more robust in tracking objects with complex interactions in video streams. In [20], a framework for detecting and tracking multiple objects with the consideration of fragmentation and grouping was proposed. By using an inference graph and a generic model based on spatial relations, this method can effectively track multiple interacting objects. A kernel particle filter (KPF) method was investigated in [21] for visual tracking by forming a continuous estimate of the posterior density function and allocating particles based on the gradients. The hidden Markov model (HMM) is an important technique for object tracking. For instance, by maximizing the joint probabilities between the state sequences and the observation sequences, an HMM-based approach for multiple-object tracking was proposed in [22] that includes an observation model, a foreground mask, and an object detection map. By considering both the temporally and spatially significant occlusions, a tracking method performed at the region level as well as the object level was proposed [23]. The region level includes a genetic algorithm for an optimal region search, whereas the object level uses spatial distributions and an interocclusion relationship for object localization.

Most of the existing methods of multiple-object learning assume that all training data, such as video streams, are available at training time. However, in many real-world applications, the image data sets are continuously available in small chunks over an indefinitely long (possibly infinite) learning life. In such a situation, traditional approaches are mainly based on one of the

following two ideas. The first one uses a "compute, storage, and retrieve" paradigm, which means it keeps accumulating and storing all the data streams and then develops the decision hypothesis based on such data. However, this approach may not be feasible in many cases due to limited computation or memory resources, or simply because we no longer have access to previous data sets. The second approach is based on the simple ensemble learning idea, which means whenever a new chunk of images is available, either a single new hypothesis or a set of new hypotheses are retrained to predict the testing data. However, this method considers each chunk of video streams separately; therefore, there is no experience accumulation or knowledge transformation from the old data to the new data. Unlike such traditional approaches, the proposed framework in this paper aims to target this problem based on the adaptive incremental learning principle. The critical issue is how to develop effective methods that are able to learn from continuous video streams, while accumulating previous knowledge at the same time. This raises the fundamental dilemma of "stability-plasticity," from the psychological point of view [24], [25]. Furthermore, in the incremental learning scenario, new concepts, such as new interesting objects, are frequently introduced during the learning life. Therefore, the learning algorithm should be able to self-adapt to such information (concept drifting issues).

This paper aims to address the issue of incremental learning for multiple-object recognition and localization. To our best knowledge, this is the first study of incremental mining for multiple-object learning that specifically focuses on experience accumulation, knowledge integration, and the decision making process. One of the major contributions of this work is that we present a common framework for adaptive learning from continuous video streams and develop a practical algorithm to inherit such advantages. The key idea of the proposed IMORL approach is that it uses an adaptive learning principle to accumulate previous experience and automatically passes such knowledge to facilitate learning from future raw data. In addition, we also investigate concept drifting and analyze how fast and how accurately such a method can self-adapt to the introduction of new concepts during the learning life. Various experiments based on video streams from YouTube [26] are used to demonstrate the learning capabilities of this approach.

The rest of this paper is organized as follows. Section II presents the details of the IMORL approach for incremental multiple-object learning. A system-level framework and a learning algorithm are proposed in this section. In Section III, various experimental results based on different video data sets are presented to show the effectiveness of this method. Finally, a conclusion and a brief discussion on future research directions are outlined in Section IV.

II. Incremental Multiple-Object Recognition and Localization Methodology

A. IMORL Framework

We first formulate the multiple-object recognition and localization problem addressed in this paper. Let D_{t-1} represent the video streams received between time $t-1$ and t. Assume a hypothesis h_{t-1} has been developed based on the available image data sets D_{t-1}. How should the system adaptively and incrementally learn information when a new chunk of image sets D_t is presented?

Conventionally, this problem can be addressed by using one of the "stability" or "plasticity" characteristics. For instance, one can simply discard h_{t-1} and develop a new hypothesis based on all the available image sets accumulated thus far: $\{D_{t-1}, D_t\}$. However, this approach loses all previous experiences, and therefore, it suffers "catastrophic forgetting" [24], [25]. The proposed IMORL framework can adaptively accumulate previous knowledge to benefit future learning and prediction processes. In addition to the video stream learning we discussed in this paper, we believe the proposed learning method can also provide new insight into the general incremental learning capability for brain-like intelligent systems development, therefore potentially benefiting research and development in the machine intelligence community.

Another possible way to handle this situation is to use ensemble learning systems: whenever a new chunk of images D_t is available, either a single new hypothesis h_t, or a set of new hypotheses, are developed based on the raw data. Finally, a voting mechanism can be used to combine all the decisions from different hypotheses to achieve better learning accuracy. However, this method considers each chunk of video streams separately during the learning stage. Therefore, the essential problem of incremental image learning, the accumulation of experience and its usage in facilitating future learning and decision making, is simply bypassed.

In this paper, we propose an IMORL framework, as illustrated in Fig. 2. Compared to traditional learning schemes, the most innovative idea of the proposed method is that it can adaptively learn from stream video data, accumulate experiences, and use such knowledge to help future learning and prediction processes. Similar to [14], we also consider each incoming video frame (image) as an independent observation of the domain knowledge, which includes multiple interesting objects with different characteristics, such as gestures, orientations, and sizes. We also assume that the training image will become available incrementally during the learning life. IMORL involves a training stage and testing stage. During the training stage, each chunk of images D_t will go through a preprocessing procedure to segment and identify potential interesting regions. Then, each region will be transformed to a feature representation, which will be the training data sets. The proposed IMORL framework can identify the locations and recognize different objects for the testing video stream based on the knowledge it learned. We present this system in detail in the following sections.

B. Feature Representation

Image preprocessing is used to provide the representations of different interesting objects (classes) for learning and prediction. Various methods can be adopted in IMORL for this purpose. For instance, the SIFT feature represented by the local keypoints in an image can be used [11]–[14]. Various segmentation methods, such as threshold techniques, edge-based techniques, region-based techniques, and connectivity-preserving relaxation methods can also be used to provide such representations. In our current study, we first transform the

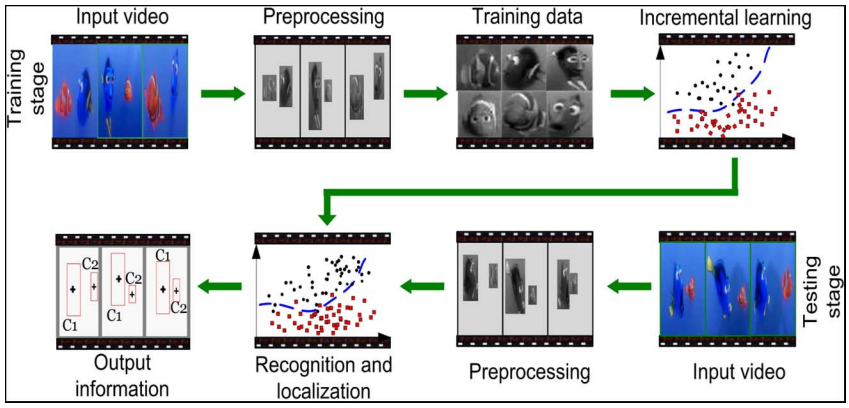

Fig. 2. IMORL framework for video data analysis.

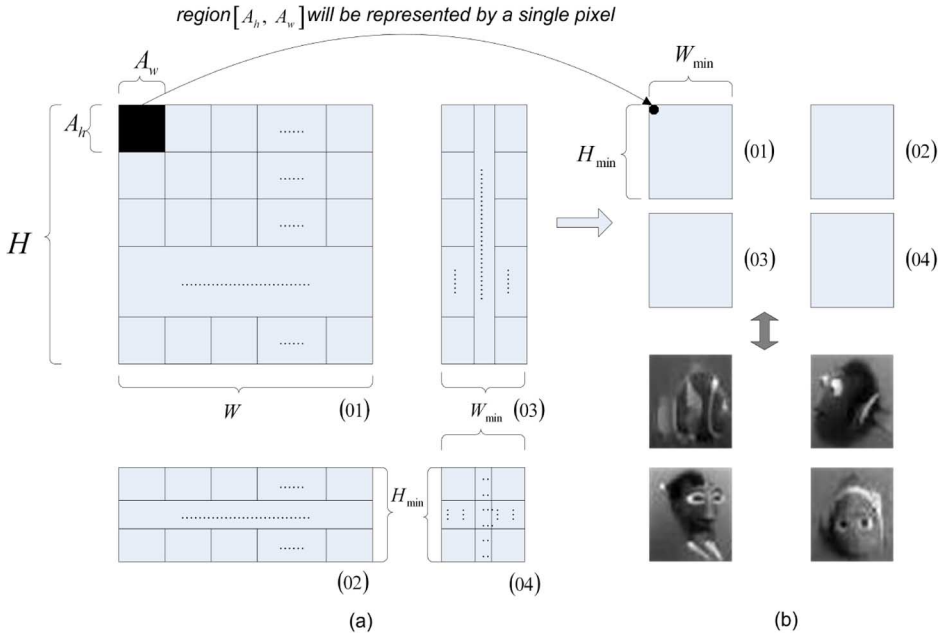

Fig. 3. Feature representation: (a) original representation; (b) scaled representation.

original image to a gray-scale representation. Then, edge-based segmentation, followed by dilation, morphological stuffing, and grain elimination, is used to identify the corresponding centroid and bounding box of each potential interesting object. A detailed analysis on image segmentation can be found in [27].

Because different interesting class objects may have different sizes after the feature representation process, we scale all objects (defined by the bounding box region) to be the same size to facilitate the training and testing process. To do this, we first find the minimum height $H_{\min}$ and width $W_{\min}$ of all potential interesting objects based on the training data ($H_{\min}$ and $W_{\min}$ may or may not be decided by the same object), then we scale all representations to the same size of $H_{\min} \times W_{\min}$. To do this, we define

$$\frac{H}{H_{\min}} = A_H + \text{residual}_H \tag{1}$$

$$\frac{W}{W_{\min}} = A_W + \text{residual}_W \tag{2}$$

where residual_H and residual_W are the remainders of division. To retain all the information in the scaled representation, we randomly distribute the number of residuals across all A_H and A_W

$$A_h = A_H + \text{rand}[0 \quad 1] \tag{3}$$

$$A_w = A_W + \text{rand}[0 \quad 1]. \tag{4}$$

In this way, the difference of the number of the containing pixels in all the rectangle areas will be no greater than 1. Fig. 3 illustrates this process with some training data representations after the preprocessing step.

C. Incremental Learning for Image Streams

After the preprocessing stage, presented in Section II-B, each interesting object can be represented as a pair of $\{\boldsymbol{x_i}, y_i\}$, where $\boldsymbol{x_i}$ is an instance in the $H_{\min} \times W_{\min}$ dimensional feature space $\boldsymbol{X}$, and $y_i \in Y = \{1, \ldots, C\}$ is the class identity label associated with $\boldsymbol{x_i}$. From now on, we refer to *image* as the original

video frame and *training data* as the $\{\boldsymbol{x}, y\}$ representation of interesting objects.

Motivated by the adaptive boosting (AdaBoost) algorithms [28]–[30], we propose an incremental learning procedure. The major objective of this incremental learning algorithm is twofold: *to automatically accumulate previous experience and use such knowledge to benefit learning from new data and support the decision making process*. Again, assume at time instant t a new set of training data D_t (based on the preprocessing steps in Section II-B) are available. The previous knowledge in this case includes hypothesis h_{t-1}, which is developed based on applying the distribution function P_{t-1} to the data set D_{t-1}. Here, the distribution function P can either be *a sampling probability function* or *a weights distribution* for different instances in the stream data: difficult examples that are hard to learn will carry higher weights compared to those examples that are easy to learn [28]–[30]. In this way, this method can automatically shift the decision boundary to be more focused on the difficult examples. According to different types of base learning algorithm, the representation of P can be expressed in different formats, such as a numerical weight for each training example or a sampling probably used to sample the training data. For the first chunk of received training data, if there is no given prior knowledge, the initial value of P_1 can be set to a uniform distribution across all the object representations, because no experience has been attained yet. The main learning procedure is presented as follows.

[IMORL-Learning Algorithm]

Information and knowledge from previous time step:

—training data chunk D_{t-1} with m examples, which can be represented as $\{\boldsymbol{x}_i, y_i\}, (i = 1, \ldots m)$, where $\boldsymbol{x}_i$ is an instance in the $H_{\min} \times W_{\min}$ dimensional feature space $\boldsymbol{X}$ and $y_i \in Y = \{1, \ldots, C\}$ is the class identity label associated with $\boldsymbol{x}_i$;

—a distribution function P_{t-1};

—a hypothesis h_{t-1} developed by the data examples based on D_{t-1} with P_{t-1};

—a testing data representation as D_{te}.

Current input:

—a new chunk of data D_t with m' examples: $\{\boldsymbol{x}_j, y_j\}, (j = 1, \ldots m')$ (m' may or may not be the same as m).

Procedures:

1) Calculate the DistanceMap (DM) between (D_{t-1}, D_t), return $[I, Q]$, where $I = [I_j] \in \{1, \ldots, m\}$ is the index of the nearest neighbor in D_{t-1} for each data in D_t, and $Q = [Q_j] \in [0, \infty)$ is the corresponding distance value. The DM is defined as the Euclidean distance in n-dimensional space. Therefore, for each $j, j = 1, \ldots, m'$, one can get the $[I, Q]$ matrix according to

$$\mathrm{DM}_{ji} = \sqrt{\sum_{k=1}^{n} (\boldsymbol{x}_{jk} - \boldsymbol{x}_{ik})^2} \tag{5}$$

$$I_j = \arg \min_{i \in \{1, \ldots, m\}} (\mathrm{DM}_{ji}) \quad \text{and} \quad Q_j = \min(\mathrm{DM}_{ji}). \tag{6}$$

2) Scale the distance according to

$$\hat{Q} = 1 - e^{-Q} \tag{7}$$

$$Q_s = 1/e^{\hat{Q}} \tag{8}$$

where $Q_s \in (1/e, \quad 1]$.

3) Update the initial distribution function for D_t

$$\hat{P}_t = \frac{P_{t-1}(I) \times Q_s}{Z_t} \tag{9}$$

where Z_t is a normalization constant so that $\hat{P}_t$ is a distribution.

4) Apply hypothesis h_{t-1} to D_t and calculate the error of h_{t-1}

$$\epsilon_t = \sum_{j: h_{t-1}(\boldsymbol{x_j}) \neq y_j} \hat{P}_t(j) \tag{10}$$

where $\epsilon_t \in [0, \quad 1]$.

5) Update the distribution function for D_t

$$P_t = \frac{\hat{P}_t}{Z'_t} \times \begin{cases} \epsilon_t, & \text{if} \quad h_{t-1}(\boldsymbol{x_j}) = y_j \\ 1, & \text{otherwise} \end{cases} \tag{11}$$

where Z'_t is a normalization constant so that P_t is a distribution

6) Repeat the procedure when the next chunk of new data sets D_{t+1} is received.

Output—the final hypothesis:

$$h_{\text{final}}(\boldsymbol{x}) = \arg \max_{y \in Y} \sum_{T:\ h_T(\boldsymbol{x}) = y} \log(1/\epsilon_T) \tag{12}$$

where T is the set of incrementally developed hypotheses in the learning life.

Fig. 4 visualizes the entire learning process. There are two mechanisms in the IMORL approach to enable incremental learning. First, whenever a new chunk of training data D_t is received, an initial estimation of the distribution function for D_t, denoted as $\hat{P}_t$, is calculated in (9) based on the relationship between D_{t-1} and D_t [represented by the numerical mapping functions (5) and (6), and the distance scaling functions (7) and (8)] and the previous weight distribution function P_{t-1}.

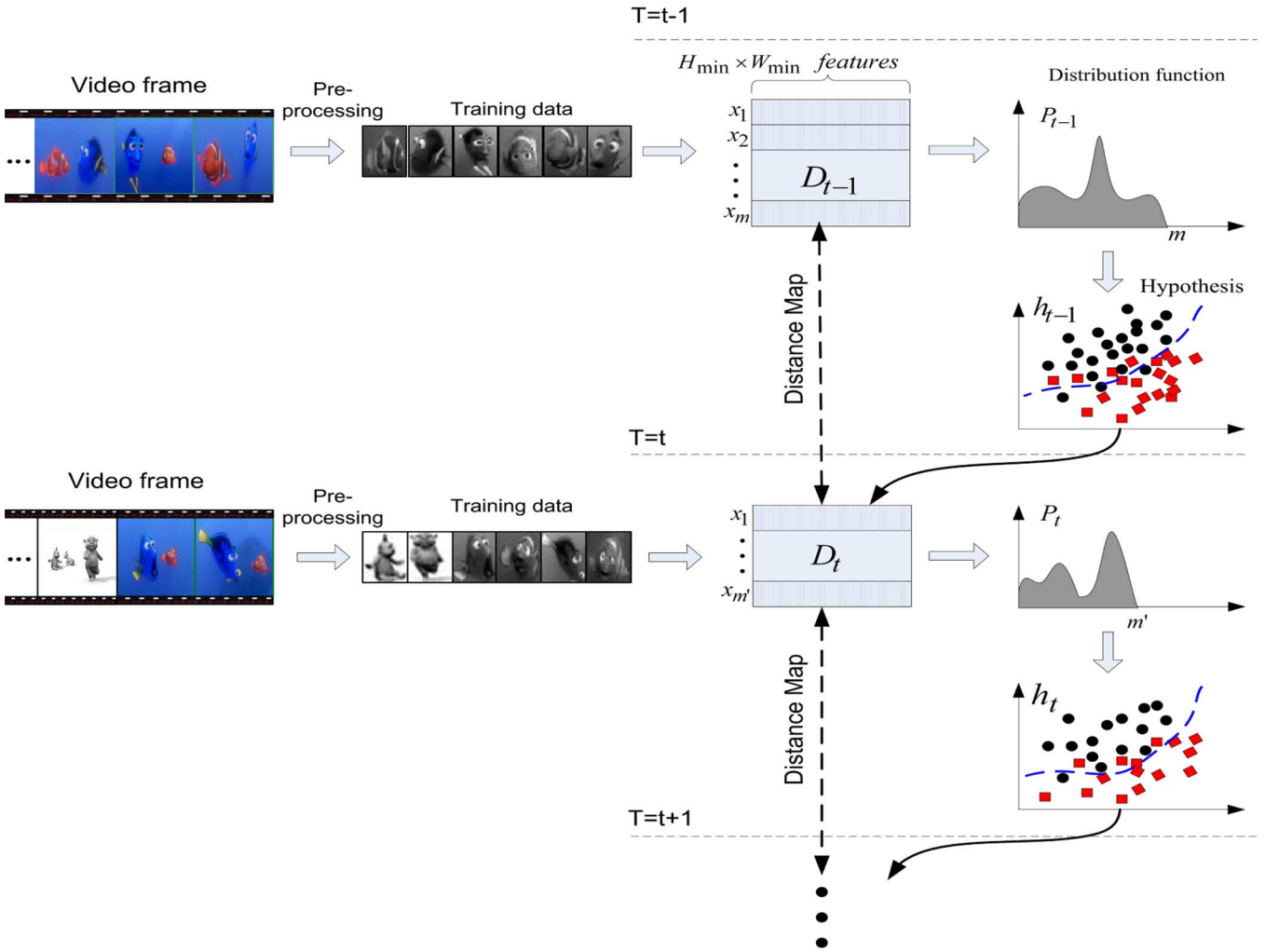

Fig. 4. Data flow of adaptive incremental learning.

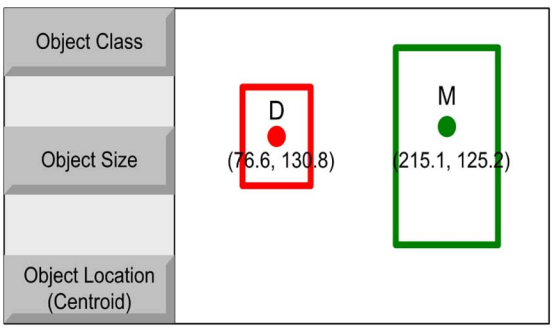

Fig. 5. IMORL simulator output information.

Because P_{t-1} reflects the learning capability for D_{t-1}, this estimation automatically passes previous knowledge to facilitate learning from the new data D_t, which provides a connection from previous experience to the new data. The scaling functions (7) and (8) are introduced to *properly map the distance and keep the data distribution characteristics to update weights*. For example, let us consider the case that an instance in D_t (represented by A) is the same as one of the instances in D_{t-1} (represented by B), therefore the distance value returned by (5) and (6) will be 0. In this case, the scaled distance by (7) and (8) will give a value of 1. Therefore, in (9), this will pass the same weight of B to the instance A, which should be the case because these two instances have the same distribution. The major purpose of (9) is to provide a mechanism to pass the previous knowledge to the new data to facilitate incremental learning. When the boosting idea is applied to traditional static learning problems, the weights can be updated iteratively based on the static training data. However, in the incremental learning scenario, one cannot directly obtain/update such weights when a new chunk of data stream is received. Equation (9) provides such a connection. In summary, the first mechanism [the procedures 1)–3) in the algorithm] is to build the connection from previous knowledge to the new data and provide an initial estimation of the distribution function to facilitate experience accumulation. Second, the previous hypothesis h_{t-1} is used to estimate its learning ability for the new data D_t (10), and the error measurement is used to refine the distribution function of P_t (11). In this way, the knowledge from the previous hypothesis h_{t-1} is used to help the learning from the new data D_t: more weights will be assigned to those difficult examples in D_t based on the previous knowledge in h_{t-1}. In this way, this method inherits the advantage of the adaptive boosting

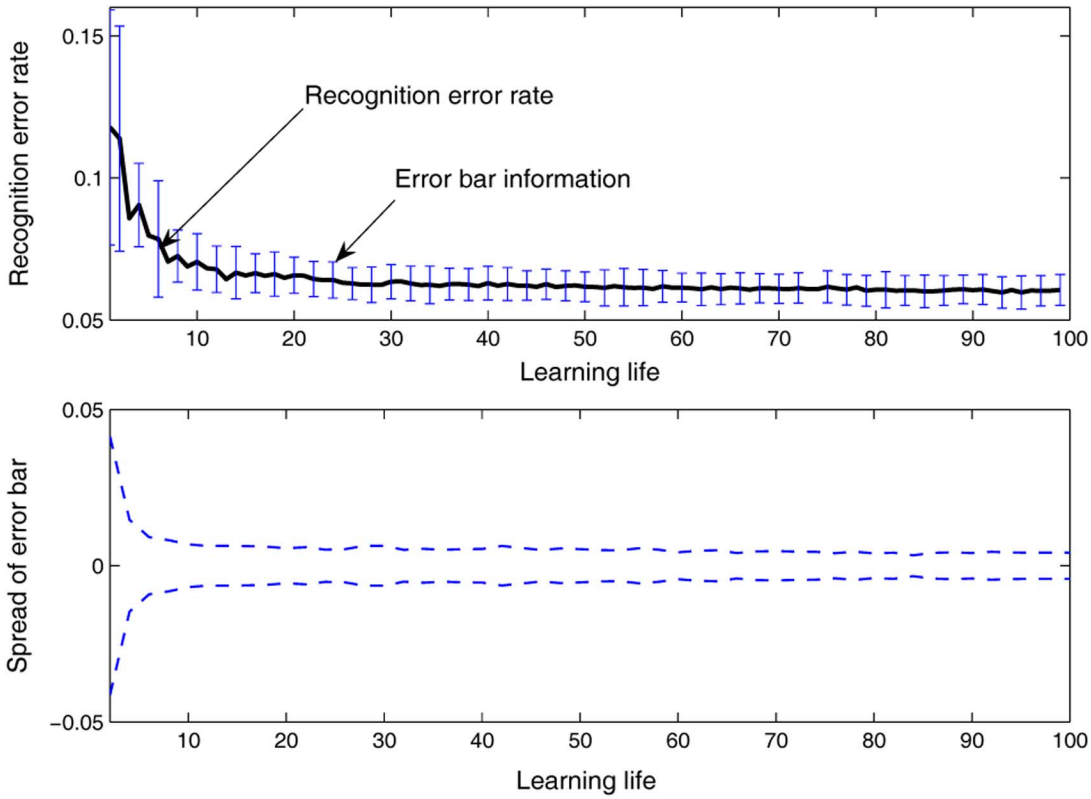

Fig. 6. Testing error reduction during incremental learning life.

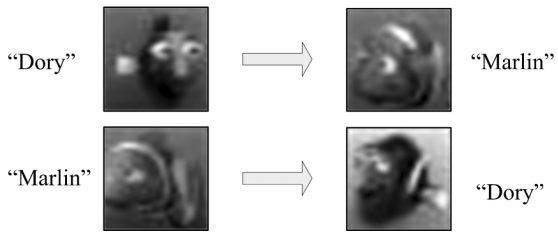

Fig. 7. Some misclassified examples in the incremental learning life.

method in an incremental fashion: it automatically shifts the decision boundary to those difficult examples in the stream data sets. One should note that IMORL can handle the variations of different data chunk sizes in the learning life, which can be seen from (5) and (6), as there are no restrictions on the data size of D_{t-1} and D_t.

Concept drifting is an important issue in incremental learning. In multiple-object learning scenarios, it is not uncommon for new interesting objects to be introduced during the learning life. Assume at time instance t, the received training data can be represented by $D_t = \{D_s, D_n\}$, where D_s represents examples from previously observed classes, whereas D_n represents those examples that the hypothesis have not learned so far (new objects). Under this situation, examples in D_n will all be misclassified by hypothesis h_{t-1}. Therefore, the weights for D_n examples will be increased based on (10) and (11). In this way, the IMORL approach automatically assigns higher weights to those newly introduced class examples to aggressively learn the new object information.

III. Experimental Results

We test the IMORL approach on different video data sets from YouTube [26]. The first video clip is "Finding Nemo" with two classes of interesting objects: Dory and Marlin, denoted "D" and "M," respectively. Based on the data preprocessing method presented in Section II-B, we extract a total of 4000 image data examples. Each data example is represented by a feature vector of 600 dimensions. We randomly select 2000 examples to train the system and use the remaining 2000 examples to test the performance. Meanwhile, we assume that the training data will become available incrementally in 100 chunks, each with 20 examples. In our current simulation, we use a neural network of multilayer perceptrons (MLPs) with one hidden layer as the base learning algorithm. The number of hidden layer neurons is set to ten, and the input neuron and output neuron equal to the number of features and the number of classes, respectively. We adopt the logistic function, and backpropagation with 500 iterations is used to train the network. Fig. 5 shows a snapshot of the prediction results for the testing image, where the highlighted region represents that an interesting object has been recognized and localized in the testing image with the corresponding location coordinates and class identity labels.

To see whether the IMORL can incrementally learn information and accumulate experience, we test the recognition error performance with the learning life. Fig. 6 shows the error reduction rate based on an average of 20 random runs, and Fig. 7 shows some misclassified examples in this experiment. Here, the error rate is measured by the correctly identified instances over all the testing instances at different learning stages. Fig. 6

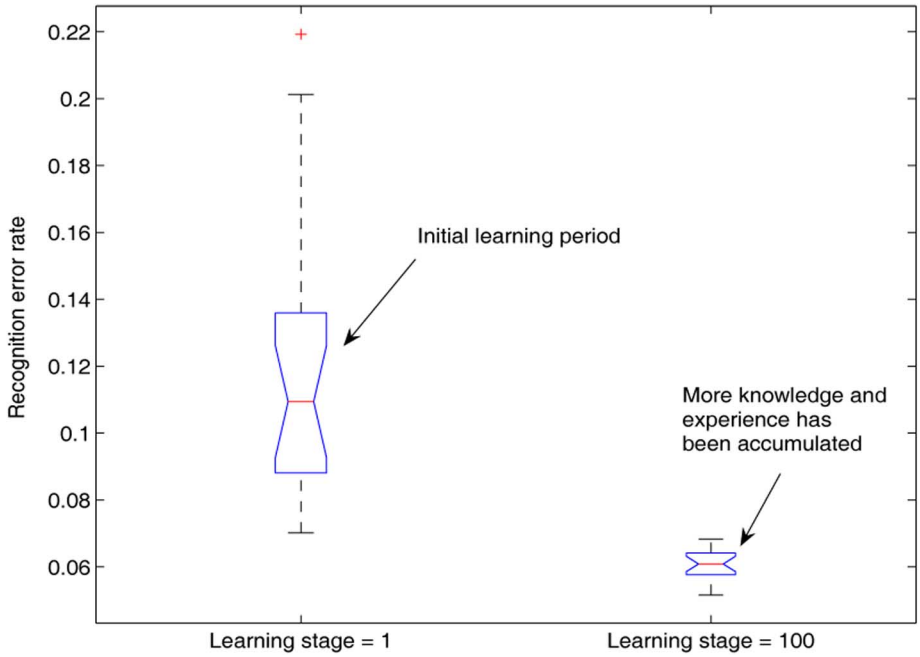

Fig. 8. Visualization of testing error statistics during different learning stages.

TABLE I
NUMERICAL CHARACTERISTIC OF THE 20 RANDOM RUNS AT DIFFERENT LEARNING STAGES

	Learning stage = 1	Learning stage = 100
largest non-outlier	0.2012	0.0683
upper quartile	0.1359	0.0641
median	0.1094	0.0609
lower quartile	0.0881	0.0576
smallest non-outlier	0.0702	0.0516
upper outlier	0.2193	

also illustrates the error bar information for the 20 runs at different learning stages. The reduction of the spread of the error bar over the learning life indicates that the system can continuously accumulate knowledge to be more stable and robust. To compare statistical error information during the incremental learning life, Fig. 8 visualizes the box plot for the 20 random runs at different learning stages, and Table I shows the corresponding numerical values. The box plot can illustrate groups of numerical data by presenting their five-number characteristics: minimum and maximum range values, the upper and lower quartiles, and the median [31]. By comparing the statistical information reflected by the box plot at the beginning of the learning life (learning stage $= 1$) and the more knowledgeable stage (learning stage $= 100$), it also confirms that the system can incrementally learn from the stream video data and accumulate knowledge to improve learning performance.

In the second type of experiment, we investigate the performance of the IMORL approach to handle the situation when new interesting objects are introduced during the incremental learning life (concept drifting). To do this, we combine the video frames of "Finding Nemo" with a new video clip, "Baby Shrek." There are three types of interesting objects in "Baby Shrek": Shrek Jr. 1, Jr. 2, and Jr. 3, which are represented by J1, J2, and J3, respectively. Similar to the first experiment, we assume the training data becomes available incrementally in 100 chunks, each with 20 data examples. The performance is also based on the average of 20 random runs.

Two learning scenarios are considered in this experiment. For scenario 1, only images from "Finding Nemo" are used to train the system throughout the entire learning life. For scenario 2, new objects ("Baby Shrek") are introduced at chunk 30, that is to say, T1 period (from chunk 1 to chunk 29) includes only images from "Finding Nemo" while T2 period (from chunk 30 to chunk 100) includes images from both video data streams. In both scenarios, the testing data sets are a mixture of half the images from one video clip and half from the other. Fig. 9 shows the testing recognition error rate versus the learning life period, where the solid line represents learning scenario 2 and the dotted line represents scenario 1. Theoretically speaking, the minimum error rate for scenario 1 will be bounded by 50%. This is because none of the "Baby Shrek" objects (half of the testing data) can be correctly recognized by the IMORL approach because it never learned this information during the entire learning life. On the other hand, when the new concepts are introduced at chunk 30 for scenario 2, the error rates begin to gradually decrease again. This improvement is because the IMORL approach can automatically learn the new object information, and use such knowledge to improve its recognition performance over the testing data.

To show how the IMORL approach can automatically handle the concept drifting issue, Fig. 10 shows a detailed view of the change of weight distributions from time t_{29} to t_{30} (just before and after the new concepts are introduced). The important thing here is to observe how the weights for each training data will be adjusted to facilitate learning from the new concepts. As one can see from Fig. 10, when the "Baby Shrek" images are introduced at time t_{30}, all these new objects tend to receive higher weights compared to those of the old objects. This is because the IMORL approach can automatically assign higher weights to the new

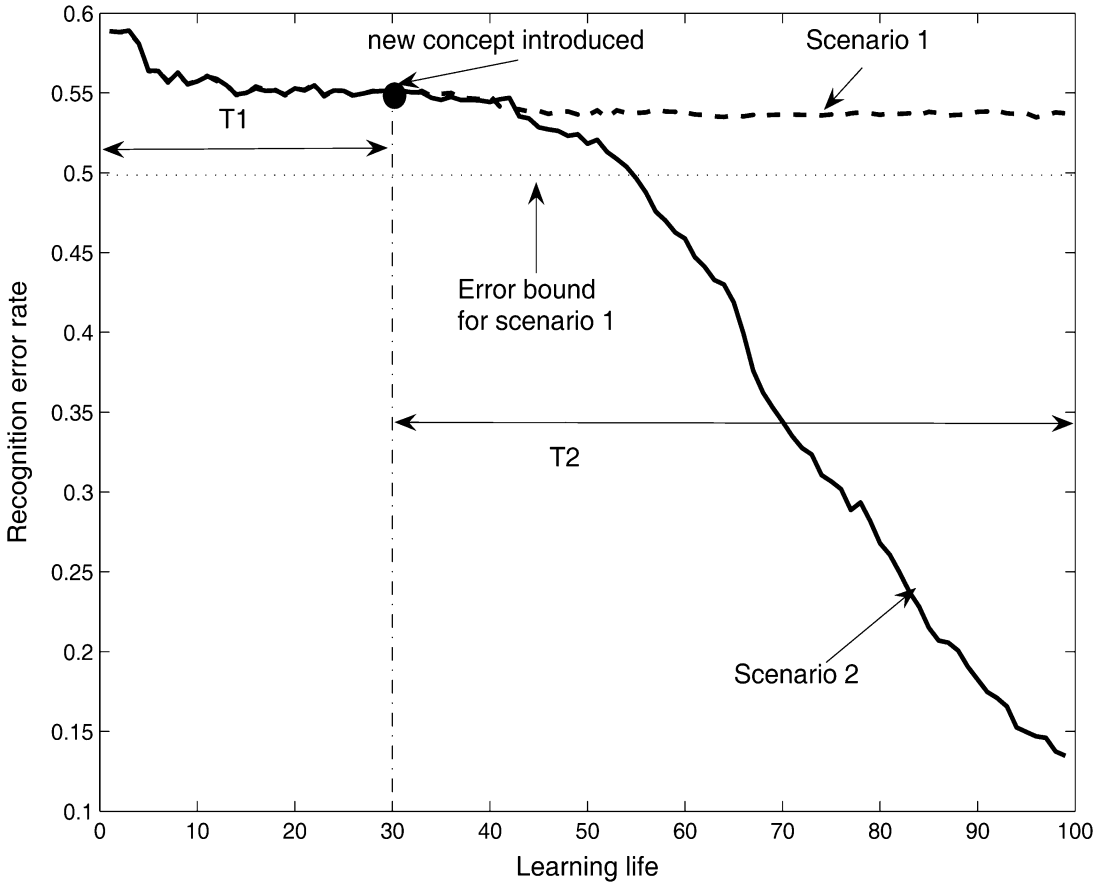

Fig. 9. Concept drifting: new objects introduced at chunk 30.

concept examples, as discussed in Section II-C. Therefore, this approach can adaptively push the decision boundary to those new objects to aggressively learn such new concepts.

Another interesting issue for concept drifting during the incremental learning life is how fast the system can self-adjust to the new concept information. From Fig. 9, we can see that, although at time t_{30} the new concepts have been introduced, the system cannot dramatically reduce its recognition error at this time instance yet. This is because at this time, the system will make decisions based on the extensive knowledge accumulated over the previous 29 chunks of training data in the T1 period plus the single chunk of new information. With the continuous learning of such new concepts over time in the T2 period, the system will gradually improve recognition performance based on such knowledge. These kinds of characteristics are reflected by the gradual reduction of error rates with the increase of learning life (learning scenario 2). This is similar to high-level intelligence in the human brain: a person with extensive knowledge in one field may initially resist new concept information. Instead, he or she may prefer to make decisions based on his/her well-established previous experience. With the continuous exposure to a new concept, he/she may gradually learn, adopt, and use such new information to help the decision making process.

Fig. 11 shows another example, when the new concepts are introduced at chunk 50. In this situation, because the learning life of new concepts (T2 period) is shorter compared to those of Fig. 9, the final recognition error is higher than that in Fig. 9. This raises an interesting and fundamental question for brain-like intelligence development: *When is a good/optimal time to introduce and learn new concepts during the incremental learning life?*

Generally speaking, the introduction of new concepts during the incremental learning life is an important issue for understanding the brain-like intelligence and potentially developing mechanisms to be able to replicate certain levels of biological intelligence. As the development of real intelligent machines remains one of the greatest unsolved scientific and engineering challenges [32], [33], the brain itself provides strong evidence of incremental learning from continuously active interaction with unstructured and uncertain environments to accomplish goals, and being able to self-adapt to new concepts and achieve certain degrees of global generalization. The big challenge is how to get closer to understanding the fundamental issues, developing mechanisms to potentially capture those capabilities, and trying to bring it closer to reality. Although various algorithms have been proposed for certain domain-specific applications [34], [35], there is no general framework that can self-adapt and learn new concepts. To this end, we hope the proposed IMORL framework also provides some insights into this issue for machine intelligence research. For instance, comparing Figs. 9 and 11, when the new concept is introduced at chunk 30 of the learning life, the system can achieve higher recognition rates (lower recognition error) compared to the sce-

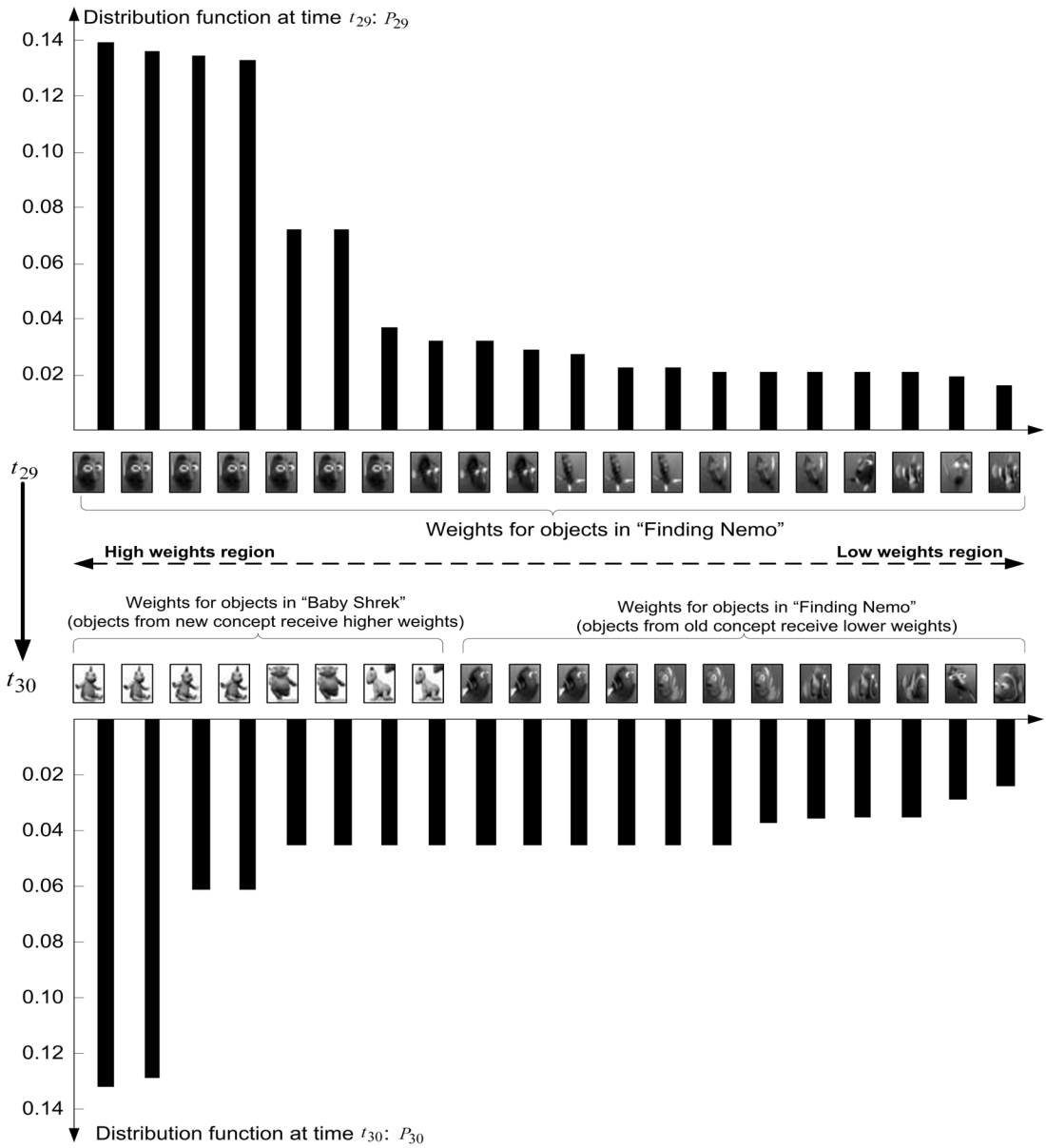

Fig. 10. Distribution function before and after new concepts are introduced.

nario when the new concept is introduced at chunk 50 (Fig. 11). This may suggest that the earlier the stage for introducing such new concepts, the better chance for the system to learn and use such knowledge to achieve goals. However, in a realistic learning environment, this may be difficult or even impossible for several reasons. First, certain types of knowledge may only appear during the middle of the human life, such as family issues. Second, the brain may have constrained resources, such as memory and functionalities, during the earlier stages of development. For instance, one cannot expect a one-year-old baby to remember all the vocabularies in the English or Chinese languages. Therefore, the resource limitations during the earlier development stages enforce certain levels of limitation on the learning capability. Third, in realistic learning scenarios from active interaction with the external environments, it is unavoidable to face new concepts during the learning life. On the other hand, if new concepts are introduced too late in the learning life, there may not be enough time (corresponds to short T2 period) for the intelligent system to explore or fully master such knowledge. Another interesting problem is that during the learning life, multiple new concepts or different types of new concepts may be introduced at the same time. Under such situations, although the proposed IMORL framework has no limitations on the numbers and types of new concepts it can handle, more sophisticated mechanisms may need to be incorporated into this algorithm to accumulate knowledge and experience to self-adapt to such new concepts. For instance, if the new concept is represented in different types of feature spaces, one may not be able to directly calculate the Euclidean distance to build the connection from past data to the new data. Therefore, instead of using the Euclidean distance-based mapping functions as in (5) and (6), more complicated mapping functions may need to be used to capture the power of this framework. Fully understanding the intelligent systems' behavior under such situations and, more important, developing mechanisms to be able to potentially capture at least part of

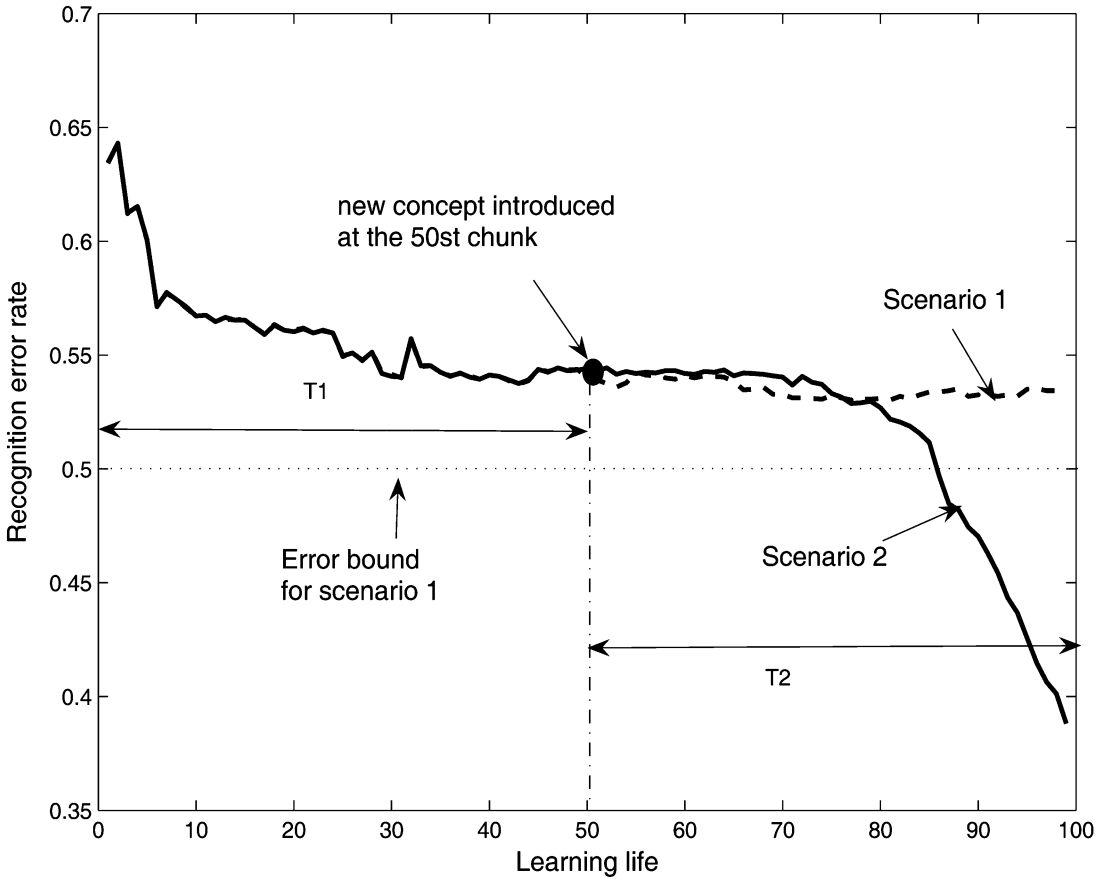

Fig. 11. Incremental learning when new concepts are introduced at chunk 50.

such a capability, are still very challenging problems. The detailed study of these issues is beyond the scope of this paper and will be addressed in separate research.

IV. Conclusion and Future Work

We propose an IMORL approach in this paper. The objective of this method is to adaptively learn multiple objects from continuous video streams. Simulation results based on real video clips demonstrate the effectiveness of this method. One of the most important contributions of our work is the incremental learning capability and self-adaptation to concept drifting during learning life. The proposed adaptive learning algorithm can automatically accumulate previous experience and use such knowledge to benefit future learning and decision making. Furthermore, this approach can self-adapt to new concepts introduced during the learning life. A brief discussion regarding the relationship between the time of introduction of new concepts and learning efficiency is also presented in this paper.

There are some interesting future works along this direction. First, our current approach transforms the color image to gray-scale representation. It would be interesting to see the effects of integrating color information to this method to facilitate learning and the prediction process. For instance, one can develop three parallel hypotheses in Fig. 4, each for a color channel to learn different feature information. Second, our approach requires a good representation of the objects in the training and testing stage. Therefore, one can investigate different approaches for segmentation and feature representation, such as use of the SIFT approach [11], [12] in the preprocessing stage. Third, in our current study, similar to [14], we treat each incoming video frame as an independent observation and focus our attention on learning aspects. It will be interesting to see the integration of states transformation of objects in different frames for tracking purposes. Our group is currently investigating all these issues. Motivated by our initial results in this paper, we believe that IMORL may provide a powerful method for incremental multiple-object learning.

References

[1] Y. Cao and S. Grossberg, "A laminar cortical model of stereopsis and 3D surface perception: Closure and da Vinci stereopsis," *Spatial Vis.*, vol. 18, pp. 515–578, 2005.

[2] S. Grossberg, "How does the cerebral cortex work? Learning, attention and grouping by the laminar circuits of visual cortex," *Spatial Vis.*, vol. 12, pp. 163–185, 1999.

[3] S. Grossberg and P. D. Howe, "A laminar cortical model of stereopsis and three-dimensional surface perception," *Vis. Res.*, vol. 43, no. 7, pp. 801–29, 2003.

[4] D. L. Wang, "The time dimension for scene analysis," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1401–1426, Nov. 2005.

[5] J. Basak, C. A. Murthy, S. Chaudhury, and D. D. Majumder, "A connectionist model for category perception: Theory and implementation," *IEEE Trans. Neural Netw.*, vol. 4, no. 2, pp. 257–269, Mar. 1993.

[6] S. Yue and F. C. Rind, "Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 705–716, May 2006.

[7] S. Akaho, "The EM algorithm for multiple object recognition," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 5, pp. 2426–2431.

[8] S. D. You, "Preprocessing network for multiple objects," in *Proc. IEEE Int. Conf. World Congr. Comput. Intell.*, 1994, vol. 6, pp. 4149–4153.
[9] S. Thirumalai and N. Ahuja, "Parallel distributed detection of feature trajectories in multiple discontinuous motion image sequences," *IEEE Trans. Neural Netw.*, vol. 7, no. 3, pp. 594–603, May 1996.
[10] G. Costantini, D. Casali, and R. Perfetti, "Detection of moving objects in a binocular video sequence," in *Proc. Int. Workshop Cellular Neural Netw. Appl.*, 2006, pp. 1–5.
[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, Corfu, Greece, 1999, pp. 1150–1157.
[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 20, pp. 91–100, 2003.
[13] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. 506–513.
[14] S. Zickler and M. Veloso, "Detection and localization of multiple objects," in *Proc. Humanoids*, Genoa, Italy, 2006, pp. 20–25.
[15] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2006, pp. 26–36.
[16] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. II-37–II-44.
[17] I. Simon and S. M. Seitz, "A probabilistic model for object recognition, segmentation, and non-rigid correspondence," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.
[18] G. Kleij, F. Velde, and M. Kamps, "Learning location invariance for object recognition and localization," in *Lecture Notes in Computer Science*, ser. 3704. Berlin, Germany: Springer-Verlag, 2005, pp. 235–244.
[19] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[20] B. Bose, X. Wang, and E. Grimson, "Multi-class object tracking algorithm that handles fragmentation and grouping," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[21] C. Chang, R. Ansari, and A. Khokhar, "Multiple object tracking with kernel particle filter," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2007, vol. 1, pp. 566–573.
[22] M. Han, W. Xu, H. Tao, and Y. Gong, "An algorithm for multiple object trajectory tracking," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2004, vol. 1, pp. I-864–I-871.
[23] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 1051–1058.
[24] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory," in *The Handbook of Brain Theory and Neural Netw.*, M. A. Arbib, Ed., 2nd ed. Cambridge, MA: MIT Press, 2003, pp. 87–90.
[25] S. Grossberg, "Neural substrates of adaptively timed reinforcement, Recognition, and motor learning," in *Models of Action: Mechanisms for Adaptive Behavior*, C. D. L. Wynne and J. E. R. Staddon, Eds. Hillsdale, NJ: Erlbaum Associates, 1998, pp. 29–85.
[26] YouTube Video Data Sets [Online]. Available: http://www.youtube.com/
[27] A. P. Dhawan, *Medical Image Analysis*. New York: Wiley, 2003, pp. 175–210.
[28] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 148–156.
[29] Y. Freund and R. E. Schapire, "Decision-theoretic generalization of on-line learning and application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[30] Y. Freund, "An adaptive version of the boost by majority algorithm," *Mach. Learn.*, vol. 43, no. 3, pp. 293–318, Jun. 2001.
[31] K. Potter, "Methods for presenting statistical information: The box plot," in *Proc. Visualiz. Large Unstructured Data Sets, Lecture Notes in Informatics (LNI)*, H. Hagen, A. Kerren, and P. Dannenmann, Eds., 2006, vol. S-4, pp. 97–106.
[32] P. J. Werbos, "Backwards differentiation in AD and neural nets: Past links and new opportunities," in *Lecture Notes in Computational Science and Engineering*, H. M. Bucker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, Eds. Berlin, Germany: Springer-Verlag, 2006, vol. 50, pp. 15–34.
[33] P. J. Werbos, "Using ADP to understand and replicate brain intelligence: The next level design," in *Proc. IEEE Int. Symp. Approximate Dyn. Programm. Reinforcement Learn.*, Apr. 2007, pp. 209–216.
[34] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proc. Int. Conf. Knowl. Disc. Data Mining*, 2003, pp. 226–235.
[35] G. Forman, "Tackling concept drift by temporal inductive transfer," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 252–259.

Haibo He (M'06) received the B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, in 2006.

Currently, he is an Assistant Professor at the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ. His research interests include machine intelligence, self-adaptive systems, computational intelligence and applications, very large scale integration (VLSI), field-programmable gate array (FPGA) design, and embedded systems design.

Dr. He has served as a Session Chair/Co-Chair and Technical Program Committee Member for several premium international conferences. He has also been a Reviewer for several major international journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, the IEEE TRANSACTIONS ON POWER DELIVERY, etc. He also has been a Guest Editor for *Soft Computing* (Springer) and *Applied Mathematics and Computation* (Elsevier). He has delivered several invited talks including the IEEE North Jersey Section Systems, Man & Cybernetics invited talk on "Self-Adaptive Learning for Machine Intelligence." He was the recipient of the Outstanding Master Thesis Award of Hubei Province, China, in 2002. Currently, he is a committee member of the IEEE Systems, Man and Cybernetic Technical Committee on Computational Intelligence, and also an active member of the Association for Computing Machinery (ACM) and the Association for the Advancement of Artificial Intelligence (AAAI).

Sheng Chen (S'08) received the B.S. and M.S. degrees in control science and engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004 and 2007, respectively. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ.

His research interests include computational intelligence and applications, machine learning, data mining, reinforcement learning, and self-adaptive intelligent systems.