An Approach to Novelty Detection Applied to the Classification of Image Regions

Sameer Singh, *Senior Member*, *IEEE*, and Markos Markou, *Student Member*, *IEEE*

Abstract—In this paper, we present a new framework for novelty detection. The framework evaluates neural networks as adaptive classifiers that are capable of novelty detection and retraining on the basis of newly discovered information. We apply our newly developed model to the application area of object recognition in video. This paper details the tools and methods needed for novelty detection such that data from unknown classes can be reliably rejected without any a priori knowledge of its characteristics. The rejected data is postprocessed to determine which samples can be manually labeled of a new type and used for retraining. In this paper, we compare the proposed framework with other novelty detection methods and discuss the results of adaptive retraining of neural network to recognize further unseen data containing the newly added objects.

Index Terms—Scene analysis, neural networks, adaptive classifiers, novelty detection.

1 Novelty Detection

Novelty detection [38] is aimed at finding novel events or data. Novelty detection can be based on the difference between actual and perceived external stimulus [20]. Nairac et al. [29] state: "For novelty detection, a description of normality is learned by fitting a model to a set of normal examples, and previously unseen patterns are then tested by comparing their novelty score (as defined by the model) against some threshold." The main challenge is the definition of an appropriate model of known data and a threshold with which outliers can be detected for a given application. A number of methodologies for outlier detection have been developed over the years. Almost all of the developed methods so far have the following weaknesses:

1. difficulty in automated threshold determination,
2. failure to specify a generic methodology that is applicable across applications without a priori knowledge of known data distributions, and
3. failure to specify an effective incremental classifier retraining procedure.

In this paper, we address some of these issues. First, we briefly review some of the well-known novelty detection approaches.

Novelty detection methods based on statistics can be classed as those using parametric models to describe the data or those using nonparametric models. Parametric models have been found to be unsuitable in a range of applications as they require extensive knowledge of the problem and do not necessarily fit real data. Several studies have discussed the relationship between rejection thresholds and classification error for Gaussian data with single classifiers [6], [14] and multiple experts [12]. Some efforts have also been made recently on setting different thresholds for different classes [13]. Nonparametric methods based on Parzen windows, *k*-nearest neighbors and Gaussian mixture models have been widely used to find outliers in test data. Some attempts have also been made using data space partitioning into self/nonself regions and using a set of rules to assign test data into these self and nonself regions [8]. Unfortunately, these rule-based methods of outlier detection are not fairly robust unless proper boundaries of training data distributions are determined.

The Parzen window method is a simple kernel-based estimator with a small amount of free parameters that is easy to implement and train. The main idea is to place a Gaussian kernel for each pattern within the pattern vector determining the position of the center of the kernel. The Gaussian kernels have a parameter that controls the smoothness (height) of the kernels to give appropriate classification performance. The smoothing parameter is normally global to the entire set of Gaussian kernels [30], [49]. Bishop [3] selects the smoothing parameter as the average distance of 10 nearest-neighbor points averaged over the whole training set. A simple threshold is used to determine whether the data point belongs to the Gaussian kernel. The main limitation of the Parzen window method is that it is computationally expensive as the number of kernel points is the same as the number of patterns in the training set. In addition, the method models the noise of the training set [48]. The kernel width is fixed for all training data points, which means that if it is too large, it covers regions of space where the estimate is oversmoothed; if it is small, then it induces classification problems.

Gaussian Mixture Models (GMM) define general distributions with kernel width defined by the spread of data in each of the input features [35], [48], [49]. GMM uses fewer kernels than the number of patterns in the data set. For test data, it can be evaluated to see if it comes from any of these kernels or their combinations that generate training data. Radial Basis Function networks estimating GMM have been used as novelty detectors [3]. The main limitation of GMM is that given high data dimensionality, a large number of

• *The authors are with the Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK. E-mail: {s.singh, m.markou}@exeter.ac.uk.*

Manuscript received 1 Aug. 2001; revised 26 Sept. 2002; accepted 3 Feb. 2003. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 114634.

training samples are needed for modelling, and the computational effort is more than that of Parzen windowing technique [3], [49]. Also, there are too many free parameters when training data is limited [30].

One of the advantages of using neural networks for novelty detection is that during training, no a priori information is needed on data distribution and no specific parameters related to data need to be set. Traditional neural network approaches to novelty detection can be categorized as: *approaches based on auto-associators*, *approaches based on Kohonen Self Organizing Networks*, *approaches based on multilayer perceptrons*, *approaches based on support vectors*, and other exotic methods. Auto-associator is a feed-forward neural network that tries to reconstruct the input data as output [17], [42], [54]. The hidden node activations can be thought of as nonlinear principal components. For novelty detection, such a network is trained to minimize the output error to the least possible. Unseen test data is passed through the network with the output nodes recreating the patterns. If the input sample is similar to the known model of normality, then the recreation (output) error is low. However, if the network receives data from unknown distribution, then the recreation error is high [9], [36], [43], [44], [45], [46], [54]. A predetermined threshold can be used to detect novel (previously unknown data distribution) patterns. Streifel et al. [44] suggest that the threshold can be applied in the form of a hypersphere placed around data of known distributions. However, this approach is often inefficient as data has different spread along feature axes and threshold can be estimated reliably by using additional data presented to the trained network [17].

Kohonen Self Organizing Networks (SOM) can also be used for novelty detection. The SOM architecture is significantly different to multilayer perceptrons and it is primarily an unsupervised technique that identifies clusters in a data set and, in effect, moves the position of the neurons in feature space to represent these identified clusters. When a SOM is trained on normal data, it will generate a kernel-based representation (as each neuron represents a single kernel) of normality which can be used for novelty detection [15], [19], [55]. After training, when new data is passed, the Euclidean distance of output nodes representing data clusters can be thresholded to determine novelty. The main limitation of this method, as with others, is the selection of an appropriate threshold. There have been some suggestions on how automated threshold setting can be achieved using information mined from the "overlooked" two-dimensional lattice [15], [50]. Clustering SOMs have been used for novelty detection for robotic applications. Each node of the clustering network is attached to an output neuron that habituates with use. Habituating SOMs have been shown to have good novelty detection properties and new nodes can be added to include the newly detected class [22], [23], [24], [25], [26].

Neural networks in the form of multilayer perceptrons that map inputs to class outputs can also be used for novelty detection. If the output of the network for a given input pattern represents low confidence, i.e., all outputs show that there is no single clear winning node (class), then the input pattern can be deemed to be novel. If the output of the network is modified using a softmax function, then the posterior probabilities are so modified to take uncertainty into account and the thresholding process becomes easier [33]. The selection of threshold is an important consideration. According to Li et al. [21], the best threshold should ideally maximize a function based on the log of the actual output of the network and the threshold output. They show that using 0.5 as a threshold on a given output node is not a good way since test data may lie outside the input data space. The output of a neural network can also be modified by training it to reduce mutual information [31]. In this process, the output distribution is mapped to a Gaussian distribution with a circular decision boundary that is easy to threshold for detecting novelty. Another approach aimed at finding novelty using neural network outputs is based on using bootstrap type rejection mechanism [51], [52]. It is suggested that spurious patterns can be recognized by training the network with negative examples of random patterns. The objective is to create attractors in the pattern space representing the rejection class such that the spurious patterns are assigned to a rejection class rather than a known class. The problem is that random patterns may not represent the portions of input space desirable to be considered as rejection areas. To solve this problem, when the neural network rejects certain patterns with high confidence, these can be used as data of a rejection class that reinforces the network to discriminate between known and unknown classes. The rejection decision is made when the responses of all of the output units are very low (close to 0), or some (or most) of the outputs are high (close to 1). In either case, the rejection is performed since the difference between the largest output did not reach a minimum defined certainty level. The work of Vasconcelos [52] shows that, on a character recognition problem, this process gives better results for rejecting novel patterns.

Approaches based on support vectors have also been used in the context of novelty detection [47]. Support vectors are used to generate a minimum volume bounding hypersphere around the data. The method allows for some objects to be outside and some to be inside the sphere using slack variables. The radius of the sphere is determined using the trade off between simplicity (or volume of sphere) and the number of errors (target points outside the sphere) using the data available. To determine if a test point is within the sphere, the distance to the center of the sphere is calculated and thresholded to see if the object is outside or inside the boundary. In order to address the problem that data is not spherically distributed, one can use different kernels [39] and rather than solving a quadratic programming problem to find the solution, linear programming-based methods are used [5].

A number of other approaches using neural networks have also been used for novelty detection, e.g., product of experts framework using Boltzmann machine by Murray [28], elliptical basis function neural nets by Brotherton et al. [4], a self-organizing generalized RBF network by Albrecht et al. [1] and Hopfield networks approach by Crook and Hayes [7]. In addition, clustering methods can also be used for novelty detection. For example, fuzzy c-means clustering [2] can be used to cluster training data into clusters and

rather than thresholding the Euclidean distance of a data point from the centroid, we can threshold its fuzzy membership for each cluster [32].

In this paper, we propose a novel approach for novelty detection using neural networks and implement it for recognizing natural objects in videos. Our approach attempts to make the process mostly independent of any parameter setting to generate a robust solution. The application area of video recognition serves to illustrate that, for difficult problems such as natural object recognition, the proposed model is very well suited compared to existing approaches. In Section 2, we describe the basics of the proposed model and discuss the methodology of novelty detection using our framework. Section 3 details the video recognition application area for finding novel unknown objects in a video stream. In Section 4, we describe experimental set-up and details. Section 5 shows the results obtained with a video sequence containing data on which the neural network had no former training. On the basis of data description of novel objects, the neural network is incrementally retrained on an enlarged set of classes and its recognition performance on a second video sequence containing now known, but previously unknown data is shown. The conclusions are presented in Section 6.

2 A Model of Novelty Detection

Our proposed model of novelty detection is shown in Fig. 1 where the training and test phases are shown separately. In the training phase, a neural network is trained on a given set of ground truth data for known objects (classes). In the context of recognizing natural objects in video streams, the training features are generated as follows: Training data images, which can be derived from training videos or still images, are first segmented. Image segmentation process groups homogeneous regions, where each region is ground truthed to belong to a given object (class). In our study, we use a region growing algorithm for image segmentation. At the end of image segmentation task, for every region in every image, we store its raw pixel values on the basis of which we extract color and texture features, and ground-truth (label or assign) data to known classes. Section 4 details the various features computed on the pixel data of regions. A feature selection algorithm based on selecting N best features on the basis of maximizing Bhattacharyya distance is used which reduces data dimensionality before data is input to the neural networks. Neural networks are trained on the training data and their architecture (number of hidden units, learning parameters) are optimized on the basis of maximizing unseen test result on a 10-fold cross-validation task, where training data is recursively split into 90 percent training and 10 percent test sets in a disjoint manner.

The testing phase is similar in the first half, i.e., generating test data features, except for the fact that no ground-truth labeling is performed. It is the classifier's task to generate object labels for objects in images. If test data comes from video, then the video sequence must be split into frames before each frame is analyzed as a separate image. Once the test data features are extracted, they are presented to the trained network for classification. A rejection filter is developed that categorizes test data samples as either *known* or *unknown*. The known samples are classified by the neural network into one of the known classes on the basis of a "winner takes all" strategy. The rejected samples are collected in a "bin" (a file) for further processing.

In our further discussion, we will make references to input data as the input to the neural network and output data as the output of the network. The following terms are used to simplify the explanation. Data distributions for a trained network can be characterized as α, or γ distributions. The α distribution describes the input data distribution based on scaled raw features. If each of the samples in the original training data, i.e., α distribution data, is presented to the trained neural network, then its output node distribution is called γ distribution. Our experiments show that data of different classes is fairly well clustered in the γ distributions as opposed to α distribution since the neural network forces data belonging to the same classes to lie in the same output space.

Our rejection filter is based on the concept of using data from a reject class when training. This concept is explained in Fig. 2 for a simple two-class problem with two features. Consider two class distributions A and B. We generate a set of random data points uniformly distributed surrounding the two distributions called "random rejects" that now define the space where the classifier should output a low score for both classes A and B. Generalizing the problem to n-dimensional spaces with k classes, each class corresponding to one output node of the neural network, we train the network with training data containing both samples of known distribution and random rejects. The random rejects are given an output class of all 0 values on output nodes. The algorithm for the rejection filter is defined as follows which is in the spirit of the approach used by [52]. The main difference is that Vasconcelos [52] uses outliers generated from training data itself to define random rejects, whereas we artificially generate such data.

Algorithm Rejection Filter

1. Given data of known classes $(c_1, c_2, \ldots, c_k)$. The target output of a sample s_i, where $1 \leq i \leq N$ for a total of N samples in the training data set is given as a binary string $(b_1, b_2, \ldots, b_k)$, where $b_j = 1$, if s_i belongs to class c_j, $1 \leq j \leq k$, otherwise $b_j = 0$.
2. Clean the training data by removing outliers. This avoids the problem of excessive empty space around a distribution within the surrounding hypercuboid so that a more compact set of random rejects can be generated. Outliers are removed using a simple strategy based on ordering the Euclidean distance of samples from their class centroid. We remove between 2-5 percent of samples as outliers per class.
3. For each feature i, determine its mean and variance parameters (μ_i, σ_i) as well as minimum and maximum $(\min_i, \max_i)$. Generate random numbers within the range $(\mu_i - 2.5^*\sigma_i, \mu_i + 2.5^*\sigma_i)$ and remove those that lie within $(\min_i, \max_i)$ range. The remaining random rejects are now within a hypercuboid tube. We choose 2.5 standard deviations since nearly 95 percent of the data (assuming it to be normal) lies within 1.96 standard deviations and a difference of around .56 standard deviations generates

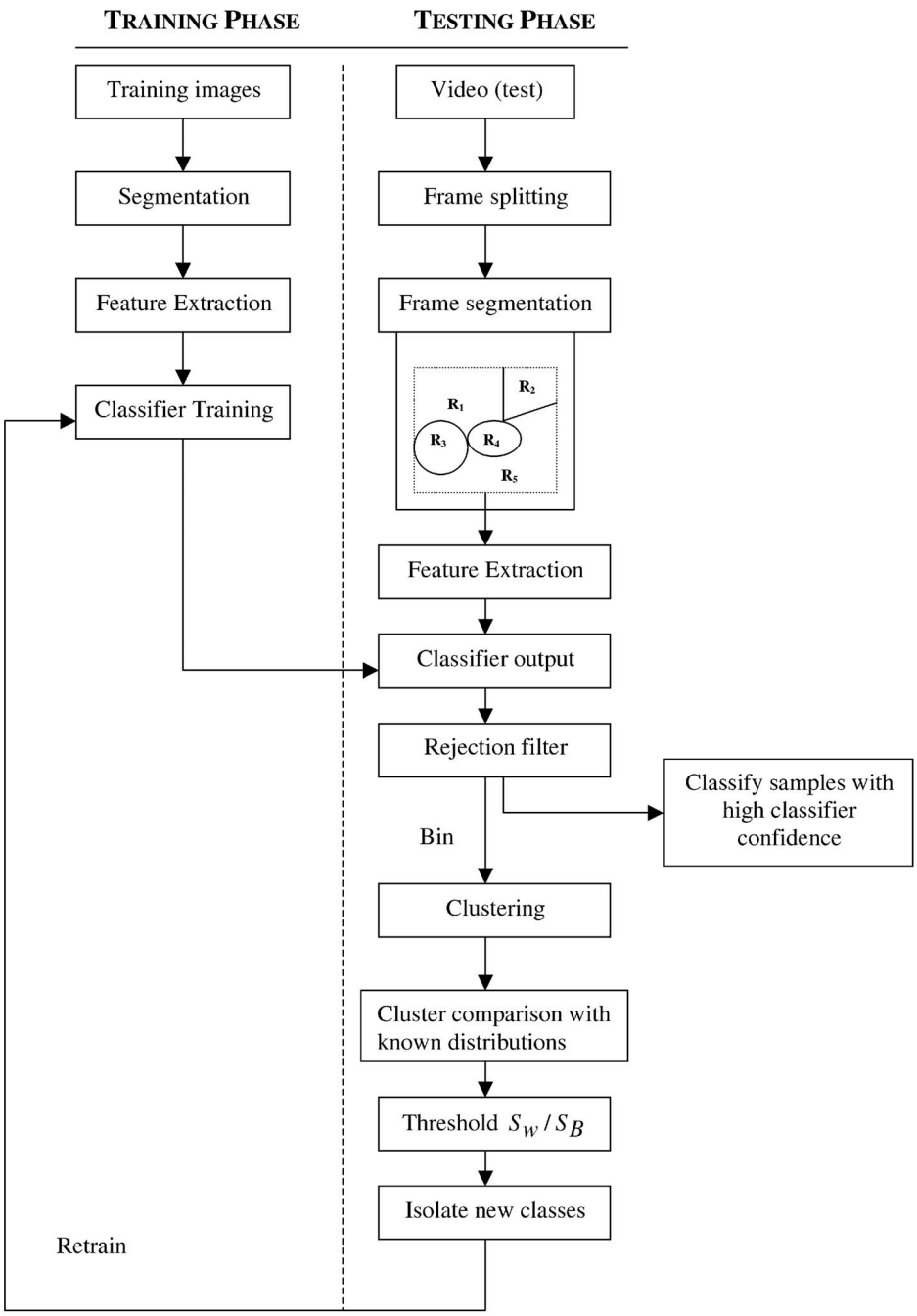

Fig. 1. A flowchart of the adaptive model.

a tight boundary; one can choose a greater number (>.56) for generating thicker random reject populations. Our experiments show that any thicker boundary for random rejects does not help improve the novelty detection ability.

4. Generate a large number of random rejects within the newly created ranges on each feature and remove those samples that lie within the distribution of known classes. Finally, we are left with a number of random samples outside of known distributions.
5. Train the neural network with random rejects by assigning them output of $[0, 0, \ldots, 0]$.
6. The rejection filter is based on the following strategy: Given a sample s_i and its test output using the trained neural network as $(a_1, a_2 \ldots, a_k)$, reject the sample as unknown if for all output nodes $a_j < 0.5$, $1 \leq j \leq k$.
7. Optimize the choice of training parameters of the neural network on the basis of maximizing measure Z.

$$Z = \left(\frac{P_{bin}^{new}}{P^{new}}\right) * \partial_1 - \frac{1}{C}\sum_{i=1}^{C}\left(\frac{P_{bin}^{i}}{P^{i}}\right) * \partial_2$$

where, P_{bin}^{new} is the total number of patterns of the new class (random rejects) in the bin, P^{new} is the total number of patterns of the new class (random rejects), P_{bin}^{i} is the

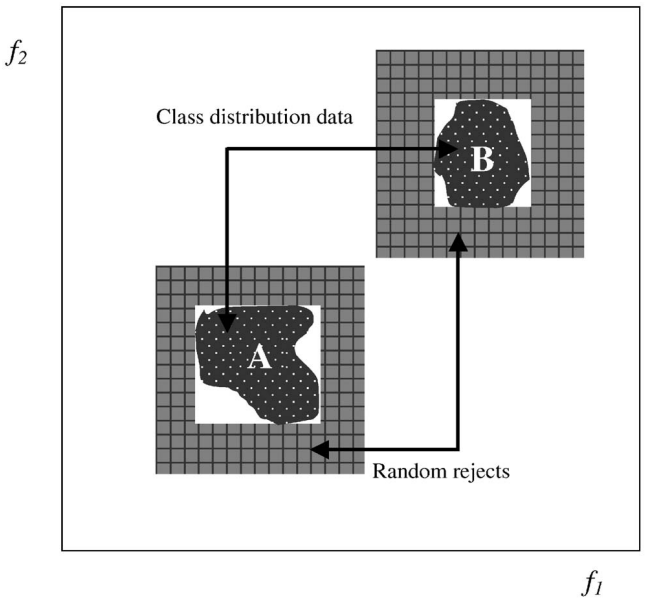

Fig. 2. The generation of random rejects.

total number of patterns of known class i, ∂_1, ∂_2 are coefficients, and C is the total number of known classes $1 \leq i \leq C$ and $i \neq newclass$. The Z metric originally lies between $[-1, +1]$ and is scaled to lie between $[0, 1]$ as follows: $Z_{scaled} = \frac{(Z+1)}{2}$. We quote in future Z as the Z_{scaled} value.

The first term in the calculation of Z calculates the percentage of samples from the new class that are put in the bin. Ideally, this term should be as close to 1 as possible. The second term calculates the percentage of each known class samples that are rejected to the bin and averages them out over all classes. Ideally, this should be as close to 0 as possible. So, the measure rewards good performance showing high values of Z and penalizes a poor performance showing low values of Z. The importance of the two terms is controlled by two weighing coefficients that vary within the $[0, 1]$ range. For simplicity, they are set equal to 1.

In Fig. 1, rejection filtering (described above) is followed by further analysis of the rejected data and classification of unrejected data into known classes. The composition of the bin now represents a challenging task to determine its components. An ideal filter should have rejected most of the unknown distribution samples but also some of the outliers of known distributions. Theoretically, the outliers of known classes that were rejected represent a good case to modify the labelling of such data for retraining a network. For example, if data of grass in a shade is thrown out as outlier, then during retraining, it may be useful to class grass as of two types: grass under normal lighting condition and grass under shade. An important step in "bin" analysis is to find clusters of data which can be labeled and used for retraining the network. In our study, we use Fuzzy c-means clustering [2] followed by the use of Davies Bouldin index [53] to determine the appropriate number of clusters in the "bin." This analysis is performed in the α distribution space. The main reason for using α distribution space as opposed to γ distribution space is that all outliers generate similar outputs for the neural network (they are all squashed in a narrow output space representing confusion), irrespective of their true class. On the other hand, data in the α distribution space is still separable and it can be clustered. We also use Self-Organizing Map [53] for clustering.

The output of the clustering process is a set of clusters with each bin data sample assigned uniquely to a single cluster. The next step is to determine which clusters are truly distinguishable from the known data distributions in the γ distribution space. A cluster containing mostly outliers of the known class does not represent a cluster of a truly unknown object and could be disregarded from further analysis. On the other hand, the identification of clusters with data from important new classes is a significant step before retraining the neural network with data containing this novel object(s). The cluster comparison with training data distributions of known classes is performed in the γ distribution space. The main reason, as mentioned earlier, is that data of different classes is fairly well clustered in the γ distribution space as opposed to α distribution space since the neural network forces data belonging to the same classes to lie in the same output space. The comparison is based on the S_w/S_B metric [10]. For a given cluster, it is matched with data of each of the known k classes as follows: Consider cluster i data to be compared with input distribution of a given class j. The average intercluster distance S_w is computed by considering all data samples in both clusters. If S_{wi} represents the intercluster distance of i and S_{wj} of j, then $S_w = (\ell_1.S_{wi} + \ell_2.S_{wj})/(\ell_1 + \ell_2)$ where ℓ_1 and ℓ_2 are the number of samples in the two clusters. Similarly we compute the intra or between cluster distance S_B by considering the distance between the point pairs across clusters. The S_w/S_B metric shows how separable cluster i data is from input distribution of a given class j and it is bounded within the limits $[0, \infty]$. A low value indicates that the two data sets are separable, and a value greater than 1 indicates that one data set superimposes the other data set. Intuitively, we can consider that two data sets are separable provided that the intercluster distance is at least half of the intracluster distance, i.e., separability is determined by $S_w/S_B < 0.5$. Clusters which have a $S_w/S_B < 0.5$ with all known class data, can be labeled as "novel" and now used for manual labeling. After manually labeling clusters representing novel classes, we can now proceed to retrain the neural network. One of the important issues related to novelty detection is how to retrain the neural network. Incremental rather than complete retraining is important to decrease the overall computational load in this context. Constructive neural networks [18] and other approaches have appeared in literature to incrementally increase the number of classes and hidden nodes without completely new retraining.

Our adaptive model for retraining a multilayer perceptron is shown in Fig. 3. In the training phase, the data of all known classes (e.g., ω_1 ,ω_2 ,ω_3) is trained with an optimized number of hidden nodes $(h_1 \ldots h_n)$. Once novel classes have been identified (e.g., two new classes $\omega_{new}(1)$, $\omega_{new}(2)$), new hidden nodes $(\lambda_1 \ldots \lambda_m)$ are created that are fully linked to the input layer and the new output nodes. In this approach,

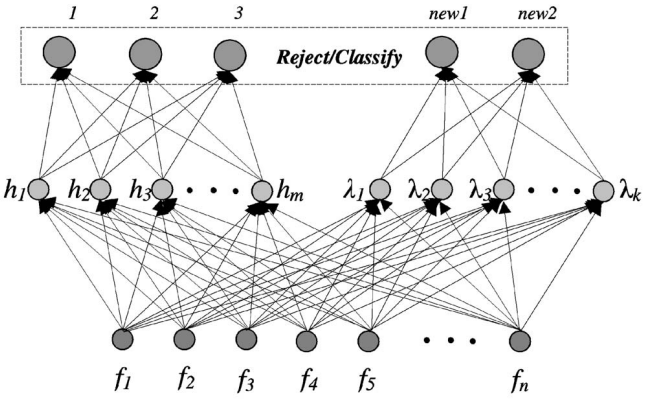

Fig. 3. Our model of an incremental network for training.

during retraining only the links between the input later, hidden nodes $(\lambda_1 \ldots \lambda_m)$ and new outputs $\omega_{new}(1)$, $\omega_{new}(2)$ are weight optimized. The training data for this retraining is now comprised of only new class training data that was filtered out surrounded by random rejects. Test samples are classified on the basis of winner takes all approach at the output layer. This approach has the distinct advantage of retaining earlier knowledge (weights) for classifying data and generating new subnetworks that can be trained much faster than complete retraining. For this reason, we do not perform complete retraining. In addition, we also do not perform partial retraining where the old weights of the earlier trained network are transferred to a new network for further training because of the herd effect [11] since it cannot be guaranteed that the old weights will not change.

3 Video Recognition

Novelty detection in video is an important topic of research with a growing demand. For example, in military applications, the detection of new objects in infrared video sequences can be used to make decisions on what to do [41]. Similarly, for security applications using robotic vehicles with vision capabilities, new events or objects can represent intrusion [37]. CCTV video analysis represents another important area where new faces or objects can be recognized leading to increased awareness of the environment. Novelty detection in video can also be applied to a range of problems within the smart rooms technology to determine abnormal visual behaviour of people. In a range of applications dealing with video, the demands of novelty detection mechanism are far more stringent than when dealing with ordinary data or signals. The large amount of data that requires image processing with video leads to a very high-computational load during the image segmentation, feature extraction, and classification stages. A number of issues related to these are discussed in Section 6. In this paper, we focus primarily on video sequences of outdoor environment. In order to understand what information is present in the video, we aim to train a classifier on naturally occurring objects and then by including a synthetic object in the outdoor environment, we test whether the proposed model can identify this novelty. Some example images are shown in Fig. 5. This is a challenging task since outdoor environment has variable lighting conditions and the same object may appear quite different under different conditions. We model the following objects from the outdoors: trees, grass, sky, and road. In our video sequence, we take an unseen set of samples from these known objects and two synthetic objects: a brown colored briefcase and a gray colored piece of clothing (fleece). The synthetic objects are chosen to have unique texture and color information that has sufficient overlap with natural objects (e.g., fleece appears similar to road in color and the briefcase's texture is similar to sky). We generate two video sequences V_1 (containing all known and two unknown objects) and V_2 (containing some of the known and one unknown object). The aim is to find whether unknown objects can be filtered out reliably from V_1 and after retraining the network with these, a good identification is possible for objects in V_2. The experimental details of how this is achieved are presented in the next section.

4 Experimental Details

The training data is obtained as a sequence of still images from the Minerva benchmark [40] (www.dcs.ex.ac.uk/minerva). A total of 357 still images of size 512×512 pixels containing natural objects are chosen. The video sequences are gathered on a different date and with different environmental conditions compared to the training data (available at: http://www.dcs.ex.ac.uk/research/pann/master/web2/newisa.htm. The video sequence V_1 is generated at a rate of 10 frames per second and it contains 203 frames. The video sequence V_2 is also generated at a rate of 10 frames per second and it contains a total of 141 frames. The first video sequence contains all of the known objects and two unknown objects: brown-colored "briefcase" and gray-colored "fleece." The second video sequence contains one new object: "briefcase." For each object region, a total of 109 features were extracted. These include correlogram features [16], color moment features [27], Color space features [34], and wavelet features (four features including standard deviation, mean, skewness, and kurtosis are extracted from the three subbands of the wavelet coefficient distribution on red, green, and blue channels separately). Feature selection was performed by maximizing Bhattacharya distance which gives a feature set of 50 features. The features selected on a per grouping basis are as follows: correlogram 3/4 (3 out of 4), moments 2/5, color space 36/37, and wavelets 9/36.

The composition of the training data and test data sets V_1 and V_2 is shown in Table 1, including the details of how many random rejects are inserted into the training data. Before the data can be presented to the neural network, it should be scaled between $[0, 1]$ range. This is because the range of the original data is well outside the range of the logistic activation function used in neural networks. One of the problems while scaling is that test data, when scaled using distribution parameters obtained from the training data, can still lie outside this range. The scaling for each feature is defined as follows:

Algorithm Scaling Data

1. Given, for a feature f_i, where $1 \leq i \leq M$, for a total of M features, the mean and variance defined for training data across all classes as (μ_i, σ_i).

TABLE 1
Data Composition for Training the Network Based on Minerva Benchmark Images and Testing Them on Video Sequence V_1 (Phase 1), and Retraining the Network with Original Training Data and Newly Learned Classes to Classify Unseen Test Data of Video Sequence V_2 (Phase 2)

	1st phase		2nd phase	
	Training	Testing	Training	Testing
Data	*Minerva*	V_1	Minerva+V_1	V_2
Grass	379	169	379	78
Tree	315	45	315	0
Sky	397	33	397	0
Road	215	38	215	63
Random Rejects	2000	-	2000	-
Briefcase	-	52	52	78
Fleece	-	36	36	0

2. The distance between a sample to be normalized x_{old}, irrespective of whether it is training or test data sample, from the mean is given as $z = \frac{x_{old} - \mu_i}{\sigma_i}$ standard deviations away.
3. Scale the data to a new value x_{new} as follows: $x_{new} = 1 - \left(\frac{0.5}{1+z}\right)$, if $x_{old} \geq \mu_i$; $x_{new} = \left(\frac{0.5}{1-z}\right)$, if $x_{old} < \mu_i$
4. The scaled data value lies within the range $[0, 1]$ for all observations.

The experiments are now carried out in two phases: In the first phase, a neural network (multilayer perceptron using backpropagation with optimized learning and momentum parameters) is trained with feature data obtained from Minerva benchmark samples and applied on test data obtained from V_1 video sequence. The main aim of this analysis is to investigate the quality of the rejection filter and the ability of the bin clustering mechanism for isolating new classes. In the second phase, we generate a new training data set that contains the original training data and the data of novel objects determined from phase one. The retraining is performed incrementally and the retrained network is tested on video sequence V_2 that contains the newly added objects. The evaluation of this phase is based on the correct classification of objects in V_2 as ascertained by the test confusion matrix.

In the first phase, the neural network is optimized for the number of hidden nodes and the maximum number of epochs on the basis of Z measure on training data (the number of hidden nodes is varied between five and 100 and the best performing network on a validation set is chosen). For all networks, we find that none of the random rejects are classified as of known classes and, therefore, the optimization of Z measure is equivalent to minimizing the number of samples of known classes that are rejected. We select a neural network with the architecture 50 input nodes, 10 hidden nodes, and four output nodes corresponding to known classes Grass, Tree, Sky, and Road. The following section details the results obtained.

5 Results

We discuss the following results:

1. classification results using video sequence V_1 with and without using the rejection filter,
2. identification of novel objects,
3. classification results using video sequence V_2 after retraining the network, and
4. comparison with other solutions.

5.1 Classification of Video Sequence V_1

The trained neural network using Minerva benchmark data and random rejects is tested on the unseen video sequence V_1 (phase 1). During training, we apply the rule of thumb that the number of random rejects must be at least equal to the number of data points of known classes and the further addition of such random data does not improve network learning any further. This is confirmed with our experiments that show the addition of random rejects more than the number of data points does not improve rejection performance (in our experiment, the total number of data points is 1,306 and we finally use a total of 2,000 random rejects). The value of parameter Z for increasing number of random rejects R can be shown as follows in the format (R, Z): $(100, .09)$, $(200, .57)$, $(500, .28)$, $(1000, .95)$, $(1500, .97)$, $(2000, 1.0)$, $(2500, .94)$, $(3000, 1.0)$. The data composition of the training and test sets is shown in Table 1. If we do not use the rejection filter assuming that the test data has the same number of objects as the training data, a recognition rate of 72.1 percent correct classification is obtained. This accuracy is obviously dependent on the number of samples from unknown classes and becomes worse as samples of novel classes are used as test patterns on which the network has not been trained. The confusion matrix is shown in Table 2a. Some of the tree samples are confused as grass, all of the fleece samples are assigned to mostly road or some to sky, and all briefcase samples are classified as grass.

If the rejection filter is applied, then all of the fleece and briefcase and none of the known class samples are rejected. If we classify only the test data that has not been rejected, then a classification accuracy of 94.3 percent is achieved as shown in Table 2b. The only mistake made by the classifier is the assignment of a few tree samples to grass. These results show that the rejection filter is of very high quality and improves the classification results considerably.

5.2 Identification of Novel Objects

The identification of novel objects is based on the contents of the bin. The first step is the identification of clusters in

TABLE 2
Confusion Matrix Based on Classifying Video Sequence V_1
(a) without Using Rejection Filter and
(b) Using the Rejection Filter

	Grass	Tree	Sky	Road
Grass	**169**	0	0	0
Tree	16	**29**	0	0
Sky	0	0	**33**	0
Road	0	0	0	**38**
Fleece	0	0	6	30
Briefcase	52	0	0	0
Recognition rate 72.1%				

(a)

	Grass	Tree	Sky	Road
Grass	**169**	0	0	0
Tree	16	**29**	0	0
Sky	0	0	**33**	0
Road	0	0	0	**38**
Recognition rate 94.3%				

(b)

the bin. There is no a priori information about the number of clusters present. Hence, we vary the number of clusters from two to 10 and monitor the Davies Bouldin index of cluster validity. The original feature data for the contents of the bin is clustered using Fuzzy C-Means clustering (we also applied a Self-Organizing Map [53] and got exactly the same result). The number of clusters with the minimum DB index is chosen; this is equal to two in our case. From our ground-truth a priori information, we know that the first cluster is composed entirely of "briefcase" patterns and the second cluster is composed entirely of "fleece" patterns. The next step is to automatically determine which of these clusters represents truly new object data as opposed to a cluster composed of outliers of known data. For this, we use output activations of these clusters to compute their separability in γ distribution. The metric we compute is S_w/S_B as the ratio of average intercluster to intracluster distance. As mentioned earlier, if a given cluster has $(S_w/S_B) \leq 0.5$, when compared with all known data distributions, then that cluster is assumed to come from a totally new class. The results of this analysis are shown in Table 3a that shows that both clusters represent totally new objects.

5.3 Network Retraining and Video Sequence V_2 Classification

The adaptive neural network model shown in Fig. 3 is retrained. The data composition for phase 2 is shown in Table 1. The training is incremental rather than completely new retraining as discussed earlier. We also compare this mode of training with complete retraining in which case random rejects are used for retraining the network and we duplicate the samples of newly added class "briefcase" and "fleece" three times since it has less number of samples in training than other known classes.

The results are shown in Table 3b and 3c. We see in Table 3b the confusion matrix generated by complete retraining with optimized number of hidden nodes. A classification accuracy

TABLE 3
(a) Cluster Comparison with Known Class Distributions: S_w/S_B Metric, (b) Complete Retraining Confusion Matrix Showing 94.1 Percent Accuracy, and (c) Adaptive Model Retraining Confusiom Matrix Showing 100 Percent Accuracy for the Testing Data in Video Sequence V_2

	Grass	Tree	Sky	Road
Cluster 1 (52 briefcase)	0.23	0.08	0.05	0.23
Cluster 2 (36 fleece)	0.16	0.08	0.07	0.33

(a)

	Grass	Road	Briefcase
Grass	**78**	0	0
Road	12	**51**	0
Briefcase	0	0	**63**

(b)

	Grass	Road	Briefcase
Grass	**78**	0	0
Road	0	**63**	0
Briefcase	0	0	**63**

(c)

of 94.1 percent is found on test data. The total time taken for this retraining is 5 minutes and 36 seconds (500 epochs) on a Pentium II 400 MHz machine running Linux operating system. Table 3c shows the result of the adaptive model of retraining which achieves 100 percent accuracy and the training time was reduced to 17 seconds (200 epochs) on the same machine. The results between road/grass confusion are improved in Table 3c compared to Table 3b since we use two different networks and the boundary generation problem is easier with less number of classes for the neural network (it has been observed in several studies that classification performance improves by using multiple classifiers, each trained on a subset of classes, called multistage classification, Parikh, 1977).

5.4 Comparison with Other Solutions

We compare our strategy with the following strategies:

1. novelty detection based on neural networks,
2. novelty detection using softmax function in neural networks,
3. novelty detection based on Gaussian Mixture Model (GMM), and
4. novelty detection using auto-associator.

All of these methods do not use random rejects. These experiments are performed on the recognition of video sequence V_1 as described in Section 5.1.

5.4.1 Novelty Detection with Neural Networks

In this approach, we simply threshold the winning output node of the network for a test sample presentation. Say, for example, if for a given test sample s, the winning node output is υ, which is found to be less than a fixed threshold τ, $0 \leq \tau \leq 1$, then the sample is assumed to be novel.

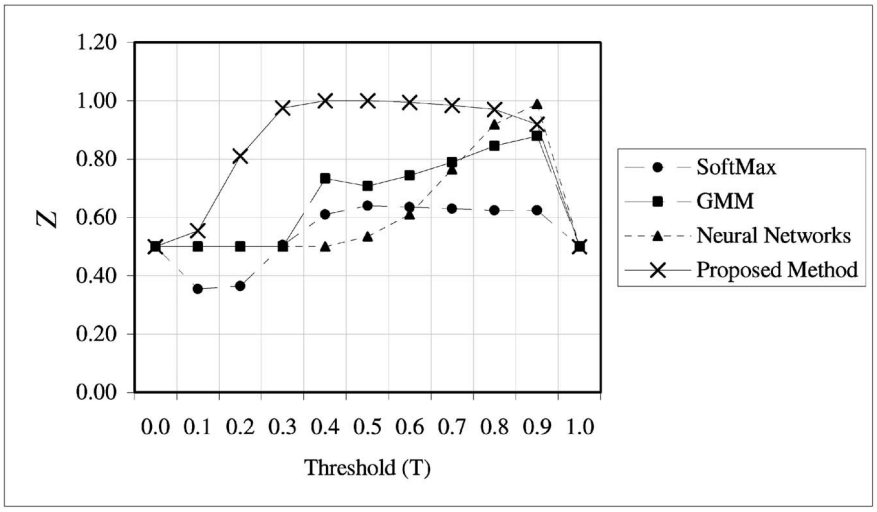

Fig. 4. Comparison of four strategies for rejecting novel samples during testing.

5.4.2 Novelty Detection With Softmax

Neural network outputs are not probabilities so we instead train a backpropagation neural network with architecture $50 \times 20 \times 4$ with the Softmax activation function for the output units and logistic activations for input and hidden units. The output of the network now is a probability value. Rejection was performed using our rejection method of thresholding the highest activation (posterior probability) [6] in a similar manner to Section 5.4.1.

5.4.3 Novelty Detection With GMM

A Gaussian Mixture Model with parameters estimated using the EM algorithm was used. A supervised training approach was followed where the probability density function of each class in the training data was modeled with three Gaussian components using only the training data belonging to that class. After training, the Bayes rule was applied to get a posterior for each training class. Rejection was performed based on the maximum posterior probability rejecting the test pattern if the probability is lower than a threshold τ. The technique is very similar to techniques proposed by authors including [3], [9], [47], [48], [49].

The results of these three approaches to novelty detection are compared with our approach in Fig. 4. The main aim is to maximize the Z measure. The figure shows that the random rejects serve an important role in novelty detection and lead to a value of $Z = 1$ for optimized threshold. The GMM and thresholding without random rejects also yields a reasonable performance if the threshold is properly optimized. Overall, our approach is a clear winner.

5.4.4 Novelty Detection With Auto-Associator

The auto-associator novelty detection system has been described in [17], [44], [46], [54]. Our auto-associator had 50 input nodes fully connected through weighted links to 30 hidden nodes, which, in turn, were fully connected to 50 output nodes. The hidden nodes and the output nodes had sigmoidal activation functions. The network was trained using backpropagation with momentum algorithm The objective of the network was to recreate the input pattern at the output layer. The test pattern was passed through the trained network and the Euclidean distance between the input pattern and the output response of the network was thresholded. Patterns belonging to known classes should yield small distance (error in recreation) unlike patterns belonging to unknown classes. Inspite of optimizing rejection threshold on test data, we found that all of the patterns belonging to the new class "briefcase" (52 in total) were rejected successfully, but no pattern from the unknown class "fleece" was rejected. Moreover, four patterns from the known class "Grass" were erroneously rejected.

6 Conclusions and Salient Observations

In this paper, we have proposed a new method of novelty detection. The method is based on the use of a robust rejection filtering mechanism and use of clustering techniques with the analysis of inter versus intracluster distances to determine which clusters represent data from new classes. This methodology is significantly different to the use of other novelty detection techniques and we have shown that this method performs better than other well-known solutions to classifier rejection and novelty detection. There are several issues that require important consideration, especially if any model of novelty detection is to be applied on video recognition tasks such as in our case. Some of the issues related to the image analysis part of the work include:

1. We find that it is important to have good image segmentation and feature extraction methods. In our study, we have used region growing segmentation method, however, these have to be chosen on a given application basis for the best results. Also, it is important that the quality of texture and color features used is good. We found that the neural

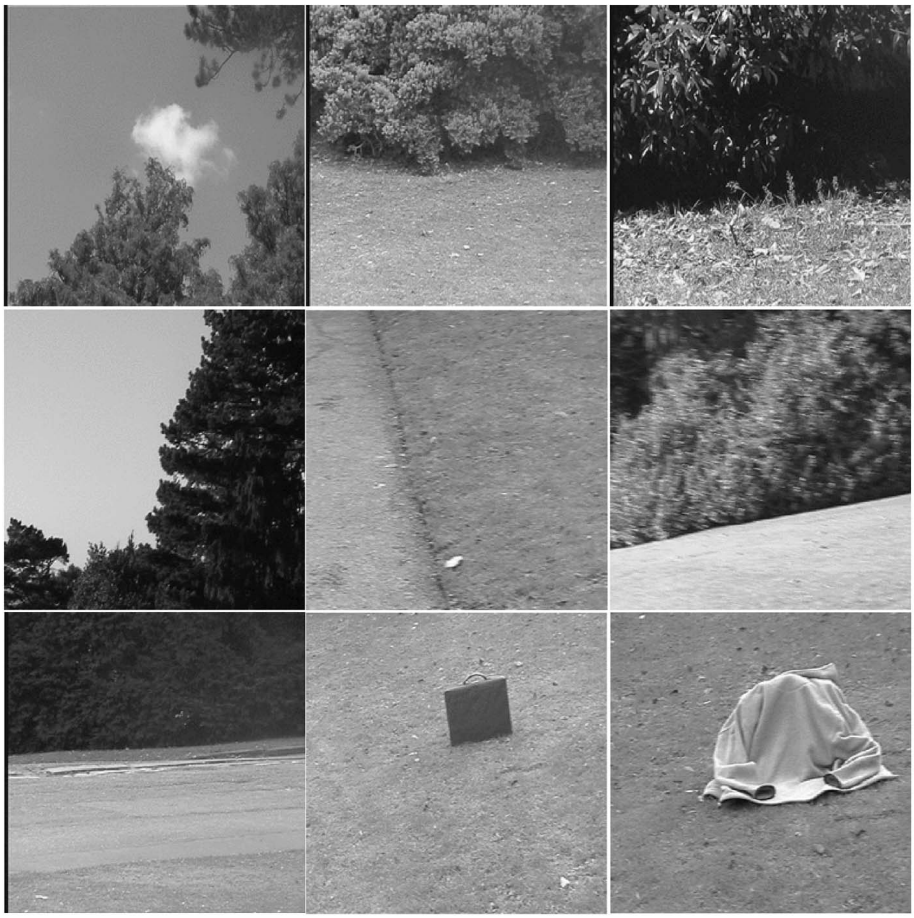

Fig. 5. Some example images used in our analysis. The last two images contain the nocel objects placed on the grass.

network trained better with a reduced set of selected features rather than the complete set.

2. Video analysis presents some novel challenges. In some frames, the amount of data of certain classes is not enough. For example, as we move the camera, some new objects entering the frames have only a very few pixels on the basis of which reliable features cannot be computed. We have discarded such frames, but a further study on solving this problem is needed. As a result of this problem, novel objects cannot be reliably detected unless they fully enter the frames.
3. Illumination changes in outdoor video recordings can have a major impact on the quality of images and, therefore, the features extracted from them. This needs to be addressed.

In addition to the issues related to image analysis, several important issues related to the pattern recognition part of the problem require consideration including how to train and develop neural network models. In this paper, we have attempted to be fairly explicit in how we have performed our analysis. One of the key aspects of our approach is that all training optimization must be performed in total ignorance of the test data available. Further work suggested in this area includes the development of networks using bagging and boosting, and implementation of constructive neural networks such as cascade correlation.

REFERENCES

[1] S. Albrecht, J. Busch, M. Kloppenburg, F. Metze, and P. Tavan, "Generalised Radial Basis Function Networks for Classification and Novelty Detection: Self-Organisation of Optimal Bayesian Decision," *Neural Networks,* vol. 13, pp. 1075-1093, 2000.

[2] J. Bezdek, R. Ehrlich, and W. Full, "FCM: The Fuzzy c-Means Clustering Algorithm," *Computers and Geosciences,* vol. 10, no. 2, pp. 191-203, 1984.

[3] C. Bishop, "Novelty Detection and Neural Network Validation," *Proc. IEE Conf. Vision and Image Signal Processing,* pp. 217-222, 1994.

[4] T. Brotherton, T. Johnson, and G. Chadderdon, "Classification and Novelty Detection Using Linear Models and a Class Dependent-Elliptical Basis Function Neural Network," *Proc. Int'l Joint Conf. Neural Networks,* May 1998.

[5] C. Campbell and K.P. Bennett, "A Linear Programming Approach to Novelty Detection," *Advances in Neural Information Processing Systems,* vol. 14, 2001.

[6] C.K. Chow, "On Optimum Rejection Error and Reject Tradeoff," *IEEE Trans. Information Theory,* vol. 16, no. 1, pp. 41-46, 1970.

[7] P. Crook and G. Hayes, "A Robot Implementation of a Biologically Inspired Method for Novelty Detection," *Proc. Towards Intelligent Mobile Robots Conf.,* 2001.

[8] D. Dasgupta and F.A. Gonzalez, "An Immunogenetic Approach to Intrusion Detection," Dept. of Computer Science, Univ. of Memphis, Report No. CS-01-001, May 2001.

[9] M.J. Desforges, P.J. Jacob, and J.E. Cooper, "Application of Probability Density Estimation to the Detection of Abnormal Conditions in Engineering," *Proc. Inst. of Mechanical Eng.,* vol. 212, pp. 687-703, 1998.

[10] R.O. Duda, R.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley, 2001.

[11] S.E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture," *Advances in Neural Information Processing Systems,* D.S. Touretzky, ed., vol. 2, pp. 524-532, 1990.

[12] P. Foggia, C. Sansone, F. Tortorella, and M. Vento, "Multiclassification: Reject Criteria for the Bayesian Combiner," *Pattern Recognition,* vol. 32, pp. 1435-1447, 1999.

[13] G. Fumera, F. Roli, and G. Giacinto, "Reject Option with Multiple Thresholds," *Pattern Recognition,* vol. 33, pp. 2099-2101, 2000.

[14] L.K. Hansen, C. Liisberg, and P. Salamon, "The Error-Reject Tradeoff," *Open Systems and Information Dynamics,* vol. 4, pp. 159-184, 1997.

[15] T. Harris, "Neural Network in Machine Health Monitoring," *Professional Eng.,* 1993.

[16] J. Huang, S. Kumar, M. Mitra, W.J. Zhu, and R. Zahib, "Image Indexing Using Colour Correlogram," *Proc. IEEE Conf. Computer Vision and Recognition (CVPR '97),* pp. 762-768, 1997.

[17] N. Japkowicz, C. Myers, and M. Gluck, "A Novelty Detection Approach to Classification," *Proc. 14th Int'l Joint Conf. Artificial Intelligence,* pp. 518-523, 1995.

[18] T. Kwok and D. Yeung, "Objective Functions for Training New Hidden Units in Constructive Neural Networks," *IEEE Trans. Neural Networks,* vol. 8 no. 5, pp. 1131-1148, 1999.

[19] J. Lamirel, M. Crehange, and J. Dulcoy, "NOMAD: A Documentary Database Interrogation System Using Multiple Neural Topographies and Novelty Detection," *Advances in Knowledge Organisation,* vol. 4, pp. 334-341, 1994.

[20] M.A. Lewis and L.S. Simo, "Certain Principles of Biomorphic Robots," *Autonomous Robots,* 2001.

[21] Y. Li, M.J. Pont, and N.B. Jones, "Improving the Performance of the Radial Basis Function Classifiers in Condition Monitoring and Fault Diagnosis Applications where 'Unknown' Faults May Occur," *Pattern Recognition Letters,* vol. 23, pp. 569-577, 2002.

[22] S. Marsland, U. Nehmzow, and J. Shapiro, "A Model of Habituation Applied to Mobile Robots," *Proc. Towards Intelligent Mobile Robots Conf.,* 1999.

[23] S. Marsland, U. Nehmzow, and J. Shapiro, "A Real-Time Novelty Detector for a Mobile Robot," *Proc. European Advanced Robotics Systems Conf.,* 2000a.

[24] S. Marsland, U. Nehmzow, and J. Shapiro, "Novelty Detection for Robot Neotaxis," *Proc. Second Int'l ICSC Symp. Neural Computation,* pp. 554-559, 2000b.

[25] S. Marsland, U. Nehmzow, and J. Shapiro, "Detecting Novel Features of an Environment Using Habituation," *Proc. Simulation of Adaptive Behaviour,* 2000c.

[26] S. Marsland, U. Nehmzow, and J. Shapiro, "Novelty Detection in Large Environments," *Proc. Towards Intelligent Mobile Robots Conf.,* 2001.

[27] F. Mindru, T. Moons, and L. VanGool, "Colour-Based Moment Invariants for Viewpoint and Illumination Independent Recognition of Colour Patterns," *Proc. Int'l Conf. Pattern Recognition (ICPR '98),* pp. 113-124, 1998.

[28] A.F. Murray, "Novelty Detection Using Products of Simple Experts—A Potential Architecture for Embedded Systems," *Neural Networks,* vol. 14, pp. 1257-1264, 2001.

[29] A. Nairac, T. Corbett-Clark, R. Ripley, N. Townsend, and L. Tarassenko, "Choosing an Appropriate Model for Novelty Detection," *Proc. Fifth Int'l Conf. Artificial Neural Networks,* pp. 227-232, 1997.

[30] A. Nairac, N. Townsend, R. Carr, S. King, P. Cowley, and L. Tarassenko, "A System for the Analysis of Jet Engine Vibration Data," *Integrated Computer Aided Eng.,* vol. 6, pp. 53-65, 1999.

[31] J.A. Parikh, "A Comparative Study of Cloud Classification Strategies," *Remote Sensing of the Environment,* vol. 6, pp. 67-81, 1977.

[32] L. Parra, G. Deco, and S. Miesbach, "Statistical Independence and Novelty Detection with Information Preserving Non-Linear Maps," *Neural Computation,* vol. 8, pp. 260-269, 1995.

[33] N.J. Pizzi, R.A. Vivanco, and R.L. Somorjai, "EvIdent: A Functional Magnetic Resonance Image Analysis System," *Artificial Intelligence in Medicine,* vol. 21, pp. 263-269, 2001.

[34] K.N. Plataniotis and A.N. Venetsanopoulos, *Color Image Processing and Applications.* Springer-Verlag, 2000.

[35] S.J. Roberts and W. Penny, "Novelty, Confidence and Errors in Connectionist Systems," *Proc. IEE Colloquium Intelligent Sensors and Fault Detection,* no. 1996/261, 1996.

[36] S.J. Roberts, "Novelty Detection Using Extreme Value Statistics," *IEE Proc. Vision, Image and Signal Processing,* vol. 146, issue 3, pp. 124-129, 1999.

[37] R. Ruotolo, C. Surace, and K. Worden, "Application of Two Damage Detection Techniques to an Off-Shore Platform," *Proc. SPIE,* vol. 3727, pp. 882-888, 1999.

[38] M. Saptharishi, K. Bhat, C. Diehl, C. Oliver, M. Savvides, A. Soto, J. Dolan, and P. Khosla, "Recent Advances in Distributed Collaborative Surveillance," *SPIE Proc. Unattended Ground Sensor Technologies and Applications (AeroSense 2000),* vol. 4040, pp. 199-208, 2000.

[39] R. Saunders and J.S. Gero, "The Importance of Being Emergent," *Proc. AI Design,* 2000.

[40] B. Schölkopf, R. Williamson, A. Smola, J.S. Taylor, and J. Platt, "Support Vector Method for Novelty Detection," *Neural Information Processing Systems,* S.A. Solla et al., eds., pp. 582-588, 2000.

[41] M. Singh and S. Singh, "Minerva Scene Analysis Benchmark," *Proc. Seventh Australian and New Zealand Intelligent Information Systems Conf.,* pp. 231-235, 2001.

[42] S. Singh, M. Markou, and J.F. Haddon, "Detection of New Image Objects in Video Sequences Using Neural Networks," *Proc. SPIE Electronic Imaging '2000,* pp. 204-213, 2000.

[43] H. Sohn, K. Worden, and C.R. Farrar, "Novelty Detection Using Auto-Associative Neural Network," *Proc. Symp. Identification of Mechanical Systems: Int'l Mechanical Eng. Congress and Exposition,* 2001.

[44] R.J. Streifel, R.J. Maks, and M.A. El-Sharkawi, "Detection of Shorted-Turns in the Field of Turbine-Generator Rotors Using Novelty Detectors—Development and Field Tests," *IEEE Trans. Energy Conversation,* vol. 11, no. 2, pp. 312-317, 1996.

[45] C. Surace and K. Worden, "Some Aspects of Novelty Detection Methods," *Modern Practices in Stress and Vibration Analysis,* pp. 89-94, 1997.

[46] C. Surace and K. Worden, "A Novelty Detection Method to Diagnose Damage in Structures: An Application to An Offshore Platform," *Proc. Eighth Int'l Conf. Off-Shore and Polar Eng.,* vol. 4, pp. 64-70, 1998.

[47] C. Surace, K. Worden, and G. Tomlinson, "A Novelty Detection Approach to Diagnose Damage in a Cracked Beam," *Proc. SPIE,* vol. 3089, pp. 947-953, 1997.

[48] D.M.J. Tax and R.P.W. Duin, "Support Vector Domain Description," *Pattern Recognition Letters,* vol. 20, pp. 1191-1199, 1999.

[49] L. Tarassenko, "Novelty Detection for the Identification of Masses in Mammograms," *Proc. Fourth Int'l Conf. Artificial Neural Networks,* vol. 4, pp. 442-447, 1995.

[50] L. Tarassenko, A. Nairac, N. Townsend, and P. Cowley, "Novelty Detection in Jet Engines," *Proc. IEE Colloqium Condition Monitoring, Imagery, External Structures and Health,* pp. 41-45, 1999.

[51] O. Taylor, J. Tait, and J. MacIntyre, "Improved Classification for a Data Fusing Kohonen Self Organising Map Using a Dynamic Thresholding Techniques," *Proc. 16th Int'l Joint Conf. Artificial Intelligence,* vol. 2, pp. 828-832, 1999.

[52] G.C. Vasconcelos, "A Bootstrap-Like Rejection Mechanism for Multilayer Perceptron Networks," *II Simposio Brasileiro de Redes Neurais,* pp. 167-172, 1995.

[53] G.C. Vasconcelos, M.C. Fairhurst, and D.L. Bisset, "Recognizing Novelty in Classification Tasks," *Proc. NIPS Conf. Novelty Detection, Adaptive Systems Monitoring,* 1994.

[54] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organising Map," *IEEE Trans. Neural Networks,* vol. 11, no. 3, pp. 586-600, 2000.

[55] K. Worden, "Structural Fault Detection Using a Novelty Measure," *J. Sound and Vibration,* vol. 201, issue 1, pp. 85-101, 1997.