# On-Line Model Modification for Adaptive Object Recognition

**Peter W. Pachowicz**[1] and **Sung W. Baik**[2]

**Abstract.** This paper presents and validates a method for adaptive object recognition in image sequences under dynamic perceptual conditions, and consequently, under changing object characteristics. The approach builds a close-loop interaction between object recognition and model modification systems. Object recognition applies a modified RBF classifier in order to recognize objects on a current image of a sequence. The feedback reinforcement generation mechanism evaluates the classification results when compared to the previous images and activates classifier modification, if needed. Classifier modification selects a strategy and employs five behaviors in adapting the classifier's structure and parameters. These behaviors include Accommodation, Translation, Generation, Extinction, and Prediction applied to selected classifier components. Accommodation modifies the component's boundary/spread. Translation shifts a given component over the feature space. Generation creates a new component of the RBF classifier. Extinction eliminates components that are no longer in use. Prediction further advances the evolution effects of three basic behaviors. The evolved RBF model is verified in order to confirm applied model modifications. Experimental results are presented for indoor and outdoor image sequences.

## 1. INTRODUCTION

Adapting a visual system to time varying environments, an integration of computer vision processes with on-line learning/adaptation processes is required. This paper presents an on-line adaptation mechanism for a RBF classifier. The developed approach introduces a feedback mechanism for an adjustment of classifier parameters and structure according to perceived differences between a model and the reality. Experimental results are presented for the texture recognition problem in indoor and outdoor environments.

Most relevant research is focused on the on-line manipulation of a model structure and parameters. This includes, self-organization applied to supervised learning with Gaussian potentials [8], dynamic link architecture for position-invariant object recognition [2], self-organization of dynamic links [7]. This work is in line with the model evolution research [4,10,11,12] where an on-line model manipulation is applied to adapt an object recognition system to new unseen appearances of an object.

## 2. MODEL EVOLUTION METHODOLOGY OVERVIEW

The approach builds an Adaptive RBF Classifier on top of the traditional Image Analysis (see Figure 1). An off-line trained classifier is used for image data classification. Classification results are analyzed and compared with the results from previous image(s). If a significant negative discrepancy in the recognition/confidence level is detected, the system modifies the classifier. This modification is executed through feedback reinforcement by gradually evolving classifier parameters and structure. The modified classifier is then re-applied to the same image to verify the adaptation process. If the classifier performs according to the expected gain, the system proceeds to the next incoming image.

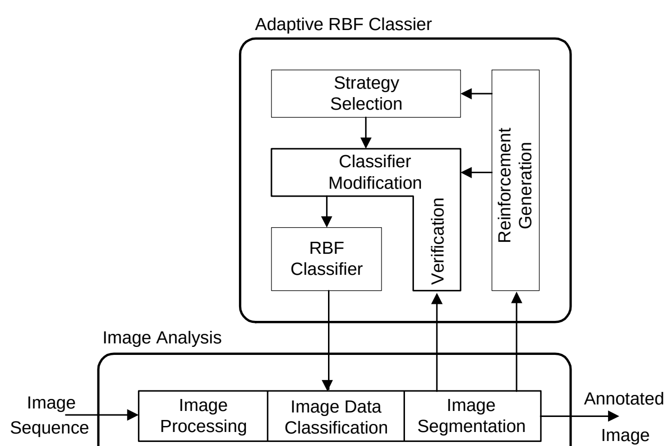


Figure 1: System architecture

A modified RBF classifier [1,9], with Gaussian distribution as a basis, was chosen for texture data modeling and classification. This is a well-known classifier widely used in pattern recognition and well-suited for engineering applications. Its well-defined mathematical model allows for further modifications and on-line

[1] Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030, USA, ppach@gmu.edu
[2] Datamat Systems Research, Inc., McLean, VA 22102, USA, sbaik@dsri.com

manipulation with its parameters and structure. The RBF classifier models a complex multi-modal data distribution through its decomposition into multiple independent Gaussians. Sample classification provides a class membership along with a confidence measure of the membership.

The structure of the classifier is shown in Figure 2. Each group of nodes corresponds to a different class. The combination of nodes is weighted. Each node is a Gaussian function with a trainable mean and spread. Classification decision yields class $C_i$ of the highest $F_i(x)$ value for a sample vector $x$.

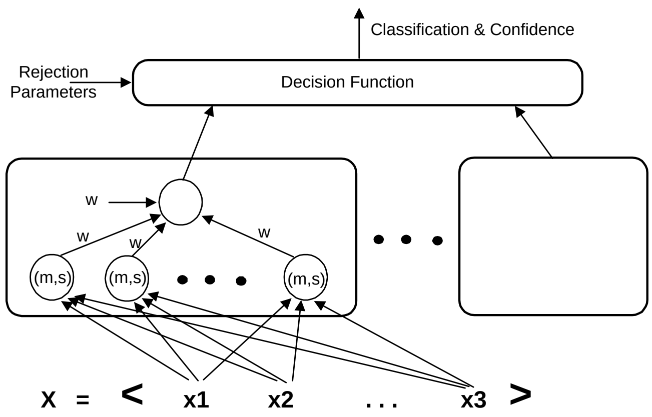


Figure 2: A modified RBF classifier

Traditional RBF classifier has been modified to deal with situations of low or confusing confidence. The *rejection area* [3] concept is introduced. Rejection area eliminates situations of (1) a low confidence and (2) a low separability from another class. The first situation implies that the confidence value for class $C_i$ must be meaningful. The second situation implies that the sample $X$ is classified to a class $C_i$ only if there is enough confidence when compared to the runner-up class $C_j$. Otherwise, the classification is delayed and the pattern is regarded as a rejection class $C_0$ (background).

A feedback reinforcement mechanism is designed to provide feedback information and control for the on-line adaptation of the classifier. This feedback exploits classification results on the next segmented image of a sequence. Reinforcement is generated over the following three steps: (1) *Sample Selection*, (2) *Sample Categorization*, and (3) *Reinforcement Parameters Generation*.

Sample Selection is performed in an unsupervised manner. A given size window (15 x 15 pixels - meaningful size of a texture patch) randomly moves over the segmented image. When all pixels in the window have the same class membership, the system understands that the window is located within a homogeneous area. Whenever such a window is found, a pixel position corresponding to the center of the window is picked up for feature data extraction. Redundant multiple overlapping windows are eliminated by rejecting samples of the highest deviation of classification confidence over the window [10]. Sample Categorization allocates selected data samples into groups of different similarity levels based on their confidences. Samples within each similarity group are generalized and described by reinforcement parameters expressing the direction and magnitude of a shift from the current RBF model.

## 3. DYNAMIC MODIFICATION OF THE CLASSIFIER

Reinforcement parameters are analyzed in relation to the structure and parameters of the classifier. First, the system selects strategies (called behaviors) for the classifier modification. Second, it binds reinforcement data to the selected behaviors. Finally, the behaviors are executed. There are four behaviors for the RBF classifier modification that can be selected and executed independently: (1) *Accommodation*, (2) *Translation*, (3) *Generation*, and (4) *Extinction*. Each behavior is implemented separately using mathematical rules transposing reinforcement parameters onto actions of RBF modification. Figure 3 illustrates four concepts of RBF modification behaviors.

Accommodation and Translation behaviors modify the classifier parameters only. This modification is performed over selected nodes of the net. The basis for Accommodation is to combine reinforcement parameters with the existing node parameters. The result of Accommodation is adjusted function spread. The node center does not change/shift through the feature space. The goal for Translation is to shift the node center in the direction of reinforcement without modifying the spread of the function. Combining Accommodation and Translation, the system can fully modify an existing node of the classifier.

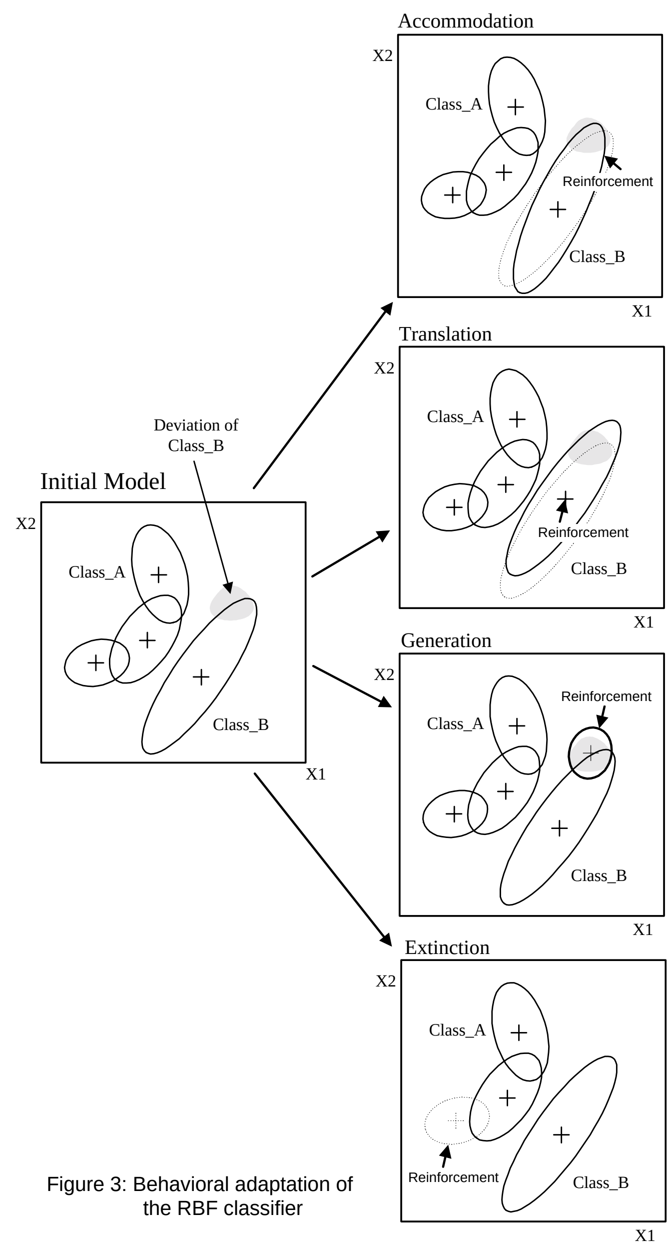


Figure 3: Behavioral adaptation of the RBF classifier

Generation and Extinction behaviors modify the classifier structure by expanding or pruning the number of nodes. The basic idea of Generation is to create a new node. A node is generated when there is (1) a significant progressive shift in function location and/or (2) an increase in complexity of feature space, for example, caused by the increase in the multi-modality of data distribution. The goal of Extinction is to eliminate useless nodes from a classifier. Extinction is activated by the utilization of classifier nodes in the image classification process. Nodes, which constantly do not contribute to the classifier, are disposed. This allows for controlling the complexity of the classifier over time.

Additional *Prediction* behavior has been developed to progress the effects of accommodation and translation. Prediction magnifies the adjustments applied to the node boundary and node position in the feature space. This behavior is applied when there is a directional and persistent change in object characteristics. The effects of prediction are illustrated in Figure 4.

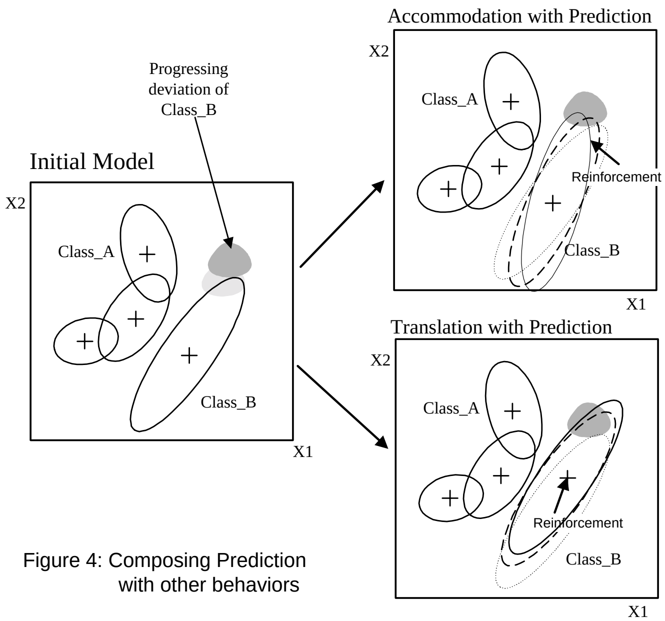


Figure 4: Composing Prediction with other behaviors

Behavioral modification of the classifier structure and parameters is verified over the same and previous images. The purpose of classifier verification is (1) to confirm the progress of classifier modification and (2) to recover from eventual errors. Classifier verification is absolutely required because behavioral modification of the classifier is performed in an unsupervised manner. If errors occur and are not corrected, they would seriously confuse the system when working over the next images of a sequence. There are two possible causes of errors: 1) incorrect reinforcement generation, and 2) incorrect selection of modification behavior. Classifier verification compares the classification and image segmentation results on the same image. If the expected improvement is not reached, then the classifier structure and parameters are restored. Classifier modification is repeated with a different choice of behaviors and/or less progressive reinforcement.

## 4. EXPERIMENTAL RESULTS

Figure 5 shows the first image of a sequence for (1) an indoor texture scene and (2) an outdoor scene used for experimentation. Image sequences were acquired by a b&w camera. The distance was gradually decreased between the camera and the scene. The first sequence has 22 images of the scene containing four fabrics (class A, B, C and D). The second sequence has 8 images of the scene containing three objects (bush, grass, and wooden fence).

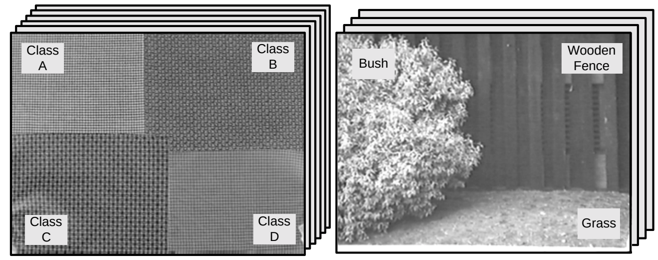


Figure 5: Image sequences used for experimentation

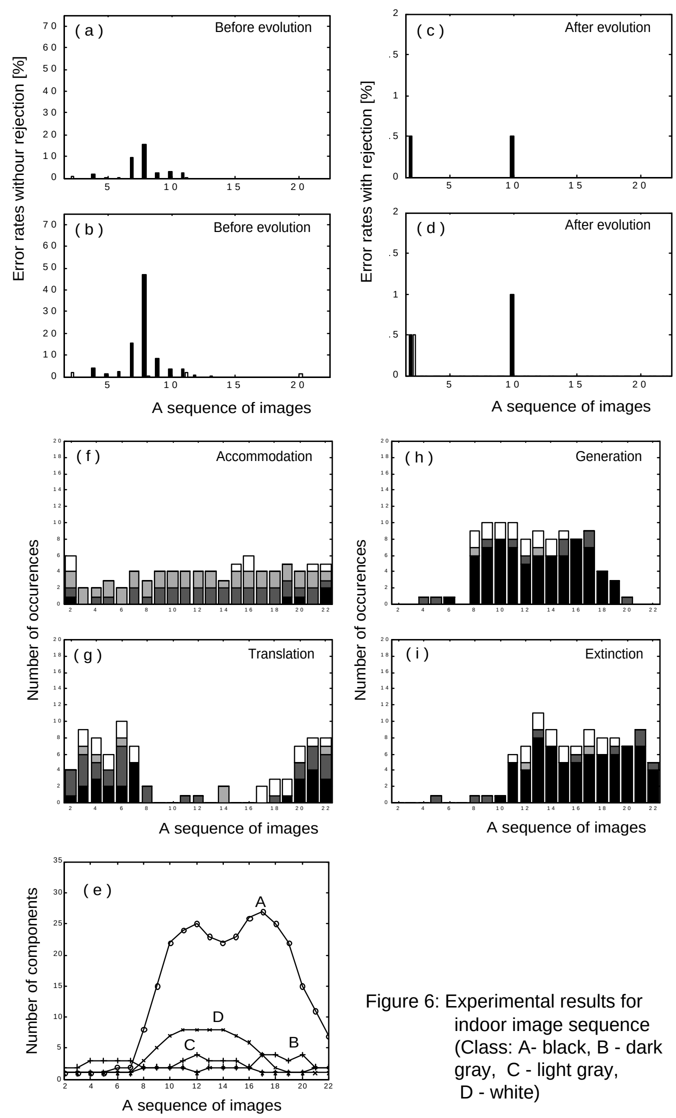


Figure 6: Experimental results for indoor image sequence (Class: A- black, B - dark gray, C - light gray, D - white)

Each incoming image is processed to extract texture features through the following three steps: (1) Gabor spectral filtering [5], (2) local 7x7 averaging of filter responses to estimate local energy response of the filter, and (3) local non-linear spatial filtering. Non-linear filtering is used to eliminate a smoothing effect between distinctive homogenous areas. The filter computes standard deviation over five windows spread around a central pixel. The mean for the lowest deviation window is returned as the output. Values of each texture feature are subject to a normalization process [13] to eliminate negative imbalances in feature distribution.

Figure 6 shows experimental results with the indoor image sequence. There are two types of error rates registered: 1) error rate without rejection, and 2) error rate with rejection. Error rates with rejection provide a better analysis of experimental results. Classification errors are registered for each new incoming image *I(i+1)* before the RBF classifier is modified (see diagrams a-b) and after it is modified over the *I(i+1)* image (see diagrams c-d). Because the system goes through every image of a sequence, the modified classifier over the *I(i+1)* image is then applied to the next image. The results show a dramatic improvement in both error rates. Both error rates achieve almost zero level after the classifier is evolved over images of a sequence.

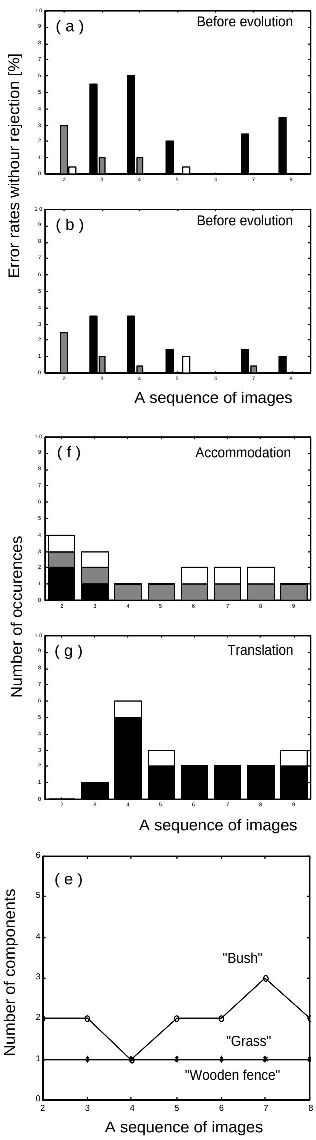


Figure 7: Experimental results for outdoor image sequence (Class: "Bush" - black, "Wall fence" - light gray, "Grass" - white)

Figure 6e shows the change in classifier complexity over the evolution process. The number of nodes for class A rapidly increases beginning from image #8 and reaches a maximum of 27 nodes when going through image #17. After that, it rapidly decreases to 9 nodes at the end of the image sequence. Other classes have relatively simpler structure than class A. The change in classifier complexity is confirmed by the frequency of the behaviors applied (see diagrams f-i). Generation is frequently applied over the mid-range images causing the complexity of the RBF net to grow substantially. Also, Generation is most frequently applied to the model of class A. With a small delay, Extinction eliminates unused nodes keeping the classifier complexity under control. Accommodation is applied occasionally. Translation dominates at the beginning and the end of image sequence. This indicates that texture characteristics change more rapidly for the mid-range images that was confirmed by feature distribution study.

Similar results were obtained for the experiment using outdoor image sequence. These results are shown in Figure 7.

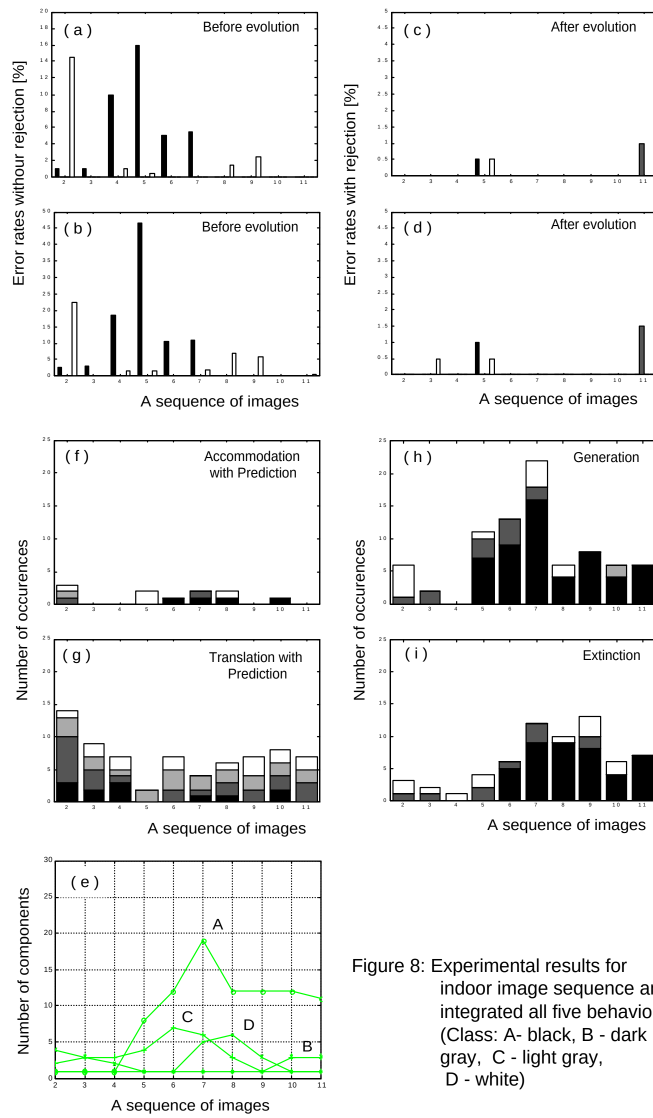


Figure 8: Experimental results for indoor image sequence and integrated all five behaviors (Class: A- black, B - dark gray, C - light gray, D - white)

The decrease in classification errors is moderate due to the increased noisiness of the texture of natural objects. Classifier complexity is low over the entire sequence of images. Translation is the most frequently applied behavior while Generation is infrequent. It is seen that the characteristics of class "Grass" change significantly that is reflected in frequently used Translation and Generation behaviors for this class. The other two classes are slightly adjusted at almost every image of the sequence - see the application of Accommodation. Extinction is applied only two times due to a low complexity of the classifier over the entire evolution process.

Further experiments with model evolution applied to the experimental domain shown in Figure 4 demonstrated problems with model evolution for lower frequency of images used. For example, the system was not able to adapt to the sequence of indoor images when every second image was used in the input to the system. It occurred that the difference in texture features distribution was not overlapping for consecutive images and caused system failure to adapt to the sequence at image #8.

Experiments have been repeated for fully integrated four behaviors and added Prediction behavior as explained in Section 3. The results were improved significantly when compared with Figure 6 and 7. Most importantly, the system was able to adapt to the indoor image sequence of lower number of images. These most interesting results are presented in Figure 8. Error rates after evolution were very well maintained at low levels. The utilization of Translation was visible for the entire sequence. The number of Generations was much higher than expected when compared with Figure 6. Finally, model complexity was handled much better - fewer model components were created. In summary, the experiment proved that Prediction is a very important model evolution behavior and should be include in an adaptive system working with image sequences.

## 5. CONCLUSIONS

The methodology developed has been tested on a variety of texture recognition problems in image sequences. The results demonstrated that on-line adaptation of the RBF classifier resulted in effective object classification over image sequences where object appearance was adversely affected by changing perceptual conditions such as resolution and lighting. Future work is focused on (1) the implementation of additional model evolution behaviors, (2) and the application of hybrid methods for image segmentation, and (3) the extension of the on-line model modification over other application domains.

## REFERENCES


[1] Baik, S., Model Modification Methodology for Adaptive Object Recognition in a Sequence of Images, Ph.D. Thesis, 1999.

[2] Buhmann, J. and J. Lange, "Object recognition with Garbor functions in the dynamic link architecture parallel implementation on a transputer network," Neural Networks For Signal Processing, Prentice-Hall International Editions, pp121-159, 1992.

[3] Fairhurst, M. C., Computer Vision for Robotic Systems, Prentice Hall, 1988.

[4] Goldfarb,L., "On the Foundation of intelligent Processes-I. An evolving model for pattern learning," Pattern Recognition, Vol.23, No.6, pp595-616, 1990.

[5] Jain, A.K. and S. Bhattacharjee, "Text Segmentation Using Gabor Filters for Automatic Document Processing," Machine Vision and Applications, No 5, pp.169-184, 1992.

[6] Haykin S., Neural Networks, Prentice Hall, 1999.

[7] Konen, W. K., T. Maurer, and Ch. Von Der Malsburg, "A fast dynamic link matching algorithm for invariant pattern recognition," Neural networks, Vol. 7, No. 6/7, pp.1019-1030, 1994.

[8] Lee, S. and B. Kosko, "Supervised Learning with Gaussian Potentials," Neural Networks for Signal Processing, Prentice-Hall International Editions, pp.189-227, 1992.

[9] Musavi, M. T., et al., "A Probabilistic Model For Evaluation Of Neural Network Classifiers," Pattern Recognition Vol. 25, No 10, pp.1241-1251, 1992.

[10] Pachowicz, P.W., "Invariant Object Recognition : A Model Evolution Approach," Proc. DARPA IUW, Washington, DC, pp.715 - 724, 1993.

[11] Pachowicz, P.W., "A Learning-based Semi-autonomous Evolution of Object Models for Adaptive Object Recognition," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 24, No 8, pp.1191 - 1207, 1994.

[12] Pachowicz, P.W. and A. Williams, "IA-Chameleon: A SAR Wide Area Image Analysis Aid," *Proc. ATR GW Workshop*, Baltimore, MD, July 1996.

[13] Theodoridis, S. and K. Koutroumbas, Pattern Recognition, Academic Press,1999.