[Brief paper]

An Adaptive Resource-Allocating Network for Automated Detection, Segmentation, and Classification of Breast Cancer Nuclei

Topic area : Image Processing and Recognition

Kyoung-Mi Lee (*)
Center for Artificial Vision Research
Korea University
Seoul 136-701, Korea
TEL: +82-2-3290-3572
E-mail: kmlee@image.korea.ac.kr
Web: image.korea.ac.kr/~kmlee

W. Nick Street
Department of Management Sciences
The University of Iowa
Iowa City, IA 52242
1-319-335-1016
nick-street@uiowa.edu
dollar.biz.uiowa.edu/~street

List of Figures

List of Tables

Abstract

This paper presents a unified image analysis approach for automated detection, segmentation, and classification of breast cancer nuclei using a neural network, which learns to cluster shapes and to classify nuclei. The proposed neural network is incrementally grown by creating a new cluster whenever a previously unseen shape is presented. Each hidden node represents a cluster used as a template to provide faster and more accurate nuclei detection and segmentation. On-line learning gives the system improved performance with continued use. The effectiveness of the resulting system is demonstrated on a task of cytological image analysis, with classification of individual nuclei used to diagnose the sample. This demonstrates the potential effectiveness of such a system on diagnostic tasks that require the classification of individual cells.

Keywords

image analysis, on-line learning, adaptive resource-allocating network

I. Introduction

Object detection, segmentation, and classification are the key building blocks of a computer vision system for image analysis. The goal of detection and segmentation is to locate and extract meaningful objects from the image. In cytological and histological images, this detection and segmentation play important roles in breast cancer classification between malignant and benign.

These three problems have been addressed with artificial neural networks (ANNs) as a means for gathering knowledge used in the imaging process and thus for producing more robust image analysis systems. For detection, most neural network approaches were focused on examining pixel intensities of image blocks. For example, both supervised and unsupervised learning in neural networks [1] have been used to determine which image blocks contain a cell nuclei. By testing the orientation of the gray value gradient, a neural network can segment detected nuclei body from background [2]. While most neural network approaches in image analysis were focused

on examining pixel intensities, a current trend in automatic image analysis is the use of model-based methods. It is usually desirable for a model to describe an expected shape and to contain information about shape variation within the training set [3]. In this paper, we use shape features to represent objects, nuclei, and supervised learning based on those features for classification. The internal representation formed by the hidden layer of the proposed network can be used as prototypes to aid nuclei detection and segmentation.

Furthermore, those ANNs are useful only when the network architecture is chosen correctly and when the goal is one or two of the three problems. In this paper, we propose an adaptive on-line neural network which is incrementally grown and based on a shape-based approach. Section II describes the shape model for a single shape and a clustered shape and Section III reviews the RBF network and sequential learning. The neural network for on-line modeling is proposed in Section IV. Section V presents the object-based image analysis system using the proposed network. Experimental results and conclusions are presented in Section VI and VII, respectively.

II. Modeling Shapes

In order to utilize or to extract the shape information of objects in an image, a suitable method for representing shapes is needed. In order to use the shapes as input data of a learning system, points on the shape must be positioned consistently on each of the training examples. In this paper, we adopt the centroid-radii model by a set of points in polar coordinates [4], [5]. Assume that OA is an arbitrary radius of the shape. The algorithm starts from OA and, moving clockwise, divides the circle into I equal arcs to place points around the boundary (Fig. 1). So the shape can be represented as $\mathbf{x} = (x_i)_{i=1\cdots I}$, where x_i is the ith radius at the angle $\left(\frac{360}{I}\right) i$. The model is invariant to translation and can be reflected by reversing the order of the points.

Since each representation can be rotated and scaled, only one model is needed for similar shapes with different orientation and scale.

A given set of N shapes, $\mathbf{x}^{1\cdots N}$, is partitioned into clusters. Each of these clusters can be represented by the center of the cluster, defined to be a clustered shape, $(\boldsymbol{\mu}_j)_{j=1\cdots J}$, where J is the number of clusters. To measure the spread of a set of data around the center of the data in the cluster, the standard deviation ($\boldsymbol{\sigma}$) is stored for each component of the clustered shape [6].

III. RBF network and Sequential Learning

The three-layer RBF network has a feedforward architecture with an input layer, $\mathbf{x}$, a hidden layer, $\boldsymbol{\mu}$, and an output layer, $\mathbf{z}$. The layers contain sets of I, J, and K nodes, respectively. Each hidden node represents a single RBF and computes a Gaussian function of $\mathbf{x}$:

$$\phi_j(\mathbf{x}) = \phi(\|\mathbf{x} - \boldsymbol{\mu}_j\|) = \exp\left(-\sum \frac{(x_i - \mu_{ji})^2}{\sigma_{ji}^2}\right), \quad j = 1, 2, \cdots, J, \tag{1}$$

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are the center and the width of the jth hidden node, respectively. Each output layer node is given by the sigmoid function ($\mathbf{s}$) of the weighted sums (z_k) of the outputs from the hidden nodes,

$$\mathbf{s}(z_k) = \frac{1}{1 + e^{-z_k}}, \qquad z_k = \sum_j \omega_{kj}\phi_j(\mathbf{x}), \quad k = 1, 2, \cdots, K, \tag{2}$$

where ω_{kj} is the weight between the jth hidden node and the kth output node.

In 1991, Platt proposed a sequential learning technique for RBF networks [7]. The resulting architecture was called the resource allocating network (RAN) and found to be suitable for on-line modeling of non-stationary processes. Initially, the network contains no hidden nodes. On incoming training examples, the RAN is either grown or the existing network parameters (the centers and weights) are adjusted using a least mean square gradient descent, based on two

criteria: prediction error $(z_k - t_k)$, where t_k is desired output, and the novelty criterion which is based on the distance between the observation and the winning center. If both the criteria are greater than thresholds, then a new hidden node is added to the network.

IV. An Adaptive Resource-Allocating Network

In this section, we propose a neural network based on RAN for on-line modeling with incremental growth (see Fig. 2). The hidden layer of the network clusters objects based on shapes and assigns a node to each cluster. To aid the classification of heterogeneous shape-based clusters, additional features $\mathbf{f}$ are directly connected to the output layer. So the output node (Eq. (2)) is given by the sigmoid function of the combined sum:

$$\mathbf{s}(z_k) = \mathbf{s}(\textstyle\sum_m \omega_{km} f_m + \sum_j \omega_{kj}\phi_j + \omega_{k0}), \quad m = 1, 2, \cdots, M,$$

where M is the number of nuclear features and ω_{km} is a weight between the kth output and the mth nuclear feature, and ω_{k0} is a bias.

A. Initializing the hidden layer

The network can initialize the hidden layer in one of two ways, depending on whether *a priori* knowledge about the shapes is available. In the breast cancer application, the objects in question are human cell nuclei, which are more or less elliptical in shape. Therefore we can define initial templates, $\mathbf{T}_j$, as a set of ellipses with different shapes (Fig. 3).

$$\boldsymbol{\mu}_j = \mathbf{T}_j, \quad \boldsymbol{\sigma}_j = \mathbf{0}, \qquad j = 1 \cdots J_0, \tag{3}$$

where J_0 is the number of initial templates. If we assume no knowledge of the shapes, then the network includes no hidden nodes, as in the original RAN.

B. On-line learning rate

Selection of a value of the learning rate parameter, η, has a significant effect on the network performance since it is related to the rate of convergence. It was discovered that the appropriate manipulation of η during the training process can lead to very good results. Eaton and Olivier stated that the range of η depends on the number and types of input and suggested a calculation of the learning rate for backpropagation using batched updates [8]:

$$\eta = \frac{\beta}{\sqrt{N_1^2 + N_2^2 + \cdots + N_J^2}}$$

where β is a constant which is chosen empirically and N_j is the number of data in the jth cluster. In contrast to Eaton and Olivier's assignment before training, since the proposed network updates clusters in on-line learning, we can calculate the adaptive learning rate based on the size of the training data during sequential training.

C. Training the network

Training data are supplied to the neural network in the form of pairs ($\mathbf{x}^n$,$\mathbf{t}^n$) of input and target vectors. If a new input $\mathbf{x}^n$ is not similar to any clusters and the prediction error is significantly large, a new cluster is created by Platt's method:

$$N_{J+1} = 1, \quad \boldsymbol{\mu}_{J+1} = \mathbf{x}^n, \quad \boldsymbol{\sigma}_{J+1} = \kappa(\mathbf{x}^n - \boldsymbol{\mu}_{nearest})^2, \quad J = J + 1,$$

where κ is a constant for an overlap factor and $\boldsymbol{\mu}_{nearest}$ is the existing nearest hidden node from $\mathbf{x}^n$. This makes new shapes more likely to match the newly-created hidden node. Thus, after training on a set of shapes, each hidden node represents a cluster of shapes in the shape space.

If the input shape $\mathbf{x}^n$ is most similar to the cluster $\boldsymbol{\mu}_j$ or the prediction error is small, the winning cluster $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are updated by adding $\mathbf{x}^n$ with stepwise gradient descent.

$$\begin{aligned} N_j &= N_j + 1, \\ \Delta\omega_{kj} &= 2\alpha_j\eta\phi_j(\mathbf{x}^n)(z_k^n - t_k^n)z_k^n(1 - z_k^n), \\ \Delta\mu_{ji} &= 2\alpha_j\eta\frac{(x_i^n - \mu_{ji})}{\sigma_{ji}^2}\phi_j(\mathbf{x}^n)\omega_{kj}(z_k^n - t_k^n), \\ \Delta\sigma_{ji} &= 2\alpha_j\eta\frac{(x_i^n - \mu_{ji})^2}{\sigma_{ji}^3}\phi_j(\mathbf{x}^n)\omega_{kj}(z_k^n - t_k^n), \end{aligned} \tag{4}$$

where α_j is the parameter indicating the similarity between $\boldsymbol{\mu}_j$ and $\mathbf{x}^n$ and is set to 1 if $\boldsymbol{\mu}_j$ is the winning cluster. Note that this update is performed only once for each new shape, in contrast to a standard backpropagation, which uses each point repeatedly during training.

In the adaptation process, hard competitive learning (winner-take-all) comprises methods where each input only determines the adaptation of one node, the winner. Due to the smaller steps in the adaptation process, hard competitive learning will have more possibilities to get stuck into well-separated local optima. One way to cope with this problem is to change to the soft competitive learning (winner-take-most) which adapts not only the winner but also some other nodes. So, soft competitive learning will lead quickly to a nearby optimum. The proposed network updates clusters depending on their similarity with $\mathbf{x}^n$:

$$\alpha_j = \frac{\|\mathbf{x}^n - \boldsymbol{\mu}_j\| - \|\mathbf{x}^n - \boldsymbol{\mu}_{nearest}\|}{\|\mathbf{x}^n - \boldsymbol{\mu}_{farthest}\| - \|\mathbf{x}^n - \boldsymbol{\mu}_{nearest}\|},$$

where $\boldsymbol{\mu}_{farthest}$ is the farthest hidden node from $\mathbf{x}^n$. Since nuclear features are connected directly to the output node, their relationships to the outputs are the same as the hidden layer.

V. Description of the Image Analysis System

Fig. 4 shows a flowchart of the proposed object-based image analysis system for object detection, segmentation and classification. The proposed network uses hidden nodes as templates

for detection (using a template-matching algorithm) and segmentation (via initialization of the snakes). While these three steps are performed repeatedly, the system learns and gathers knowledge of shapes that the user would like to find. The following two sections describe the image analysis techniques that we used for detection and segmentation.

A. Object detection: GHT

The proposed system uses the generalized Hough transform (GHT) [9], a standard template matching algorithm, to detect the boundaries of objects by making use of gradient direction information at edge pixels in an image. GHT requires template shapes to be predefined. Further, when the scale and orientation of an input shape are variant and unknown in advance, brute force is usually employed to enumerate all possible scales and rotations of the input shape in the GHT process.

In this work, the system uses an iterative version of GHT (IGHT) which incorporates a clustered shape [6]. An important aspect of the this method is the use of intelligent search to guide the order in which template matches are attempted. As the images are processed, the program records the number of times each template has matched an object. This allows us to search first for the object shapes that were most common in previous images. By searching first for the objects that we are most likely to find, we significantly reduce the expected time required, since not all templates will appear in every image.

B. Object segmentation: Snake

To segment the exact boundaries of objects, the proposed system uses an energy-minimizing contour, called a "snake" [10], which is guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. While snake provides a powerful interactive tool for image segmentation, it is vulnerable to its initial position.

In the breast cancer nuclei application, the model is a closed curve that is attracted to strong edges in the image, and forms an arc in the absence of such edge information. The snakes are initialized using the results of the IGHT that searches for learned shapes of various sizes. Since the templates for IGHT are created for unanticipated shapes, they may give better initialization, resulting improved quality of segmentation. The user can then edit the resulting outline, remove an incorrect boundary, or draw a boundary on an undetected object by hand. From the perspective of the learning method, this process creates a collection of positive examples of the shapes that this user would like to find and outline.

VI. Experimental Results

A. Data images and experimental setup

The algorithm was tested on 640×480 gray-scaled cytological images from fine-needle biopsies of breast masses. These images are classified as benign or malignant on a per-sample basis (K=2). Since no classification is available for individual cell nuclei, we assume all nuclei in benign images to be benign and all nuclei in malignant images to be malignant. This assumption is reasonable but not entirely accurate, resulting in an undetermined amount of classification noise.

The system was trained on a sequence of 90 training images (N=3423 nuclei), alternating between benign and malignant. After each 18 training images (at 18, 36, 54, 72, and 90), ten test images (5 benign and 5 malignant) were automatically segmented and classified using all trained templates. Fig. 5 shows one of the test images.

During the IGHT, the system computes a global accumulator of the same size as the image which contains, in each cell, a detection value representing a match score for the best-matching template. Then the system runs snakes in order of the detection value and uses 24 radii (I) for each nucleus as the input of the neural network. Then the system computes 5 nuclear features

(M): size, radius, perimeter, smoothness, and concavity of each nucleus [11]. Size means the amount of scaling based on the maximal radius. Radius is the average length of the 24 radial line segments from the centroid of the nucleus and each boundary point. Perimeter is calculated by summing distances between consecutive points on the boundary. Smoothness is the total difference between the length of a radius and the mean length of two neighbor radii surrounding it. Concavity means the severity of any concavities on the boundary. So, the network for cytological image analysis initially consists of 29 input nodes ($I + M$), 5 hidden nodes (J_0), and 2 output nodes (K). The parameter values used for this experiment are: β=1.2 and κ=0.5.

B. Number of hidden nodes

Fig. 6 shows the number of hidden nodes/templates that had to be searched on a logarithmic scale. The original IGHT algorithm was performed with a constant, predefined set of elliptical templates. These vary significantly in size and shape in an attempt to capture the basic shape of all nuclei that might be encountered. The neural system builds a set of templates as it is used. After an initial ramping up period, the number of templates is very stable at a level more than an order of magnitude smaller than the original set. The curves labelled "Hard competitive" and "Soft competitive" show the number of hidden nodes using hard competitive learning and soft competitive learning, respectively. These reflect the number of templates searched in order of their frequency in previous images. The soft competitive learning used in the neural network results in a 95.84% reduction in the number of necessary templates from IGHT and a 52.94% reduction from hard competitive learning.

C. Cluster separation

Fig. 7 shows the set of templates constructed after training 90 images. The number next to each template name is the number (N_j) of trained shapes. To estimate how compact and

well-separated the clusters are, we calculated two distances [12]: intra-cluster distance(Table I), which is the averaged distance between a template j and objects in the corresponding cluster, $\mathbf{x}_{\in j}$, and inter-cluster distance (Table II), which is the distance between templates a and b,

$$\text{intra-cluster distance}_j = \frac{\sum \|\boldsymbol{\mu}_j - \mathbf{x}_{\in j}\|_2}{N_j} \quad \text{and} \quad \text{inter-cluster distance}_{a,b} = \|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|_2,$$

where $\|v_1 - v_2\|_2$ is the Euclidean distance between two vectors v_1 and v_2 and N_j is the number of objects in the cluster $\mathbf{x}_{\in j}$. Most templates are well clustered, having a larger inter-cluster distance from any other template than the intra-cluster distance. There are two exceptions: one is between T2 and T3, and the other between T4 and T5. Since the inter-cluster distance (20) between T2 and T3 is smaller than the intra-distance T3 (20.36), T3 is clustered so loosely that some objects in T3 are difficult to separate from T2. A similar case happens in between T4 and T5. We note that the network is initialized with some close templates, however, it never creates overlapping templates.

D. Segmentation

To evaluate the segmentation performance of the algorithm, we used two measures, segmentation sensitivity and segmentation accuracy. Segmentation sensitivity is the likelihood that a nucleus will be detected by a expert. Segmentation was performed until approximately 80% of the nuclei in the image were correctly outlined, up to a maximum of 50 nuclei, since only a few dozen cells are typically required for successful diagnosis. Segmentation accuracy is the likelihood that a segmented object is an actual nucleus. An expert then judges correctly outlines nuclei as well-segmented nuclei. The well-segmented nuclei are classified as described in Sec VI-E and used for training the system. Fig. 8 illustrates the average accuracy of the 10 test images using templates generated by the neural network. The new system has better segmentation accuracy than the original IGHT algorithm with predefined elliptical templates after training or 36 images,

and achieves this performance in an order of magnitude less time per image (see Fig. 6). Fig. 8 shows that as the system trains with more images, both prediction accuracy and segmentation sensitivity increase, although gains are modest after around 50 images.

E. Classification

Table III shows the classification counts for well-segmented nuclei with soft-competitive learning. After only the first training set (18 images), nuclei were correctly classified at 65.02%. As the number of training sets reached five, the average classification rate achieved 94.13%. For well-segmented cells in the 10 test images, we achieved 94.61% accuracy on the benign cells and 93.51% on the malignant cells after training 90 images. We also tested on hard-competitive learning and observed nearly identical. So, the proposed network uses soft-competitive learning for fast detection with fewer templates (Fig. 6), instead of for better classification. Table III also shows the neural network without 5 nuclear features achieved 58.92 % accuracy in average, and that with the only size feature 86.15 % accuracy in average, respectively. For cross validation, we repeated this classification procedure 9 times more, with each image serving as a test image once, and achieved 94.13% averaged result of 10 test sets. The values in Table IV are the averaged values and the standard errors of classification rate of 10 test sets. As training continues, the standard errors of 10 test sets at each training set are decreasing from 2.36 to 0.01.

Another measurement of classification performance is the ROC curves with classification sensitivity and specificity. Classification sensitivity is the probability that a nucleus will be correctly classified as malignant. Specificity is probability that a nucleus without the disease will be correctly identified as benign. After training, the system provides classification outputs of each nucleus through the neural network. At each classification level, the system counts true positive cases and computes the rates on the total number of nuclei. The area under the curve (AUC) is

the average sensitivity over all possible specificities for evaluating diagnostic procedures. Fig. 9 shows that the AUC of the proposed neural network (0.9315) is larger than that (0.7801) of the instance-based learning scheme, which classify new objects based on the majority class of the matching template [5], [6].

Table V shows an experimental result on classification rate depending on the learning rate. We tested the same 426 well-segmented nuclei after training the network on 90 images using different values of η. Using $\eta = 0.5$, the algorithm oscillated, moved very slowly towards the real descent direction, and caused the lowest classification rate. The value 0.05 of η is too small to allow a fast learning process. Since the proposed adaptation is based on the characteristics of the training data, it is able to quickly adapt the learning rate to a particular situation and find a better local optimum. In terms of reducing the error as much as possible, the automatic adaptation of η outperforms several reasonable constant values for η.

In addition to classification of individual nuclei, we tested the classification of images. In our application, the diagnosis of breast cancer is based on an image corresponding to each patient. We calculate the malignancy of each image as

$$\text{malignancy} = \frac{\text{number of malignant cells in an image}}{\text{number of well segmented cells in an image}}.$$

Fig. 10 shows classification accuracy on the 10 test images after every 18 training images. The 5 benign images have good results from the beginning of training. Since the characteristics of malignant nuclei (i.e. size and shape) are various, some of the malignant images were more difficult to classify well early in the training. However, after training on 54 images, the network started to find patterns to consistently differentiate malignant nuclei from benign nuclei. We tested additional 100 images after training on 90 images and achieved 94% and 96% accuracy at malignancy thresholds of 0.5 and 0.42, respectively.

VII. Conclusion

We proposed an incrementally grown network, specially designed for image analysis tasks. This paper demonstrates how the proposed neural network is used to aid the detection, segmentation and classification of the nuclei being outlined. Using centers and widths of the hidden nodes as templates for detection and segmentation provides a guide for an intelligent search through the space of possible nuclei. This approach improves object detection, by gathering a collection of desired objects specific to the application; it improves segmentation by generating higher quality initial outlines with different classes of objects. The centers and widths are updated proportionally using soft competitive learning, creating a set of tight, well-separated clusters. The system supports the visual display of hidden nodes as templates. We demonstrated that our learning rate in better classification.

Further directions of this work shall include a pruning technique and different learning rates for different nodes. We are also interested in comparing these results with the previously implemented Xcyt system, currently used in clinical practice [13]. We also planning to use the system in a large-scale image analysis task, detecting and classifying cells from heterogeneous tissues.

Acknowledgments

This work was partially supported by NIH grant CA64339-04 and NSF grant IIS-99-96044.

References

[1] F. Schnorrenberg, N. Tsapatsoulis, C. Pattichis, C. Schizas, S. Kollias, M. Vassiliou, and K. Kyriacou, "Modular neural network system for the analysis of nuclei in histopathological sections," *IEEE Engineering in Medicine and Biology*, vol. 19, no. 1, pp. 48–63, 2000.

[2] T. W. Nattkemper, H. Wersing, W. Schubert, and H. Ritter, "A neural network architecture for automatic segmentation of fluorescence micrographs," in *Proceedings of European Symposium on Artificial Neural Networks*, 2000, pp. 177–182.

[3] N. Duta, A. K.Jain, and Marie-Pierre Dubuisson-Jolly, "Learning 2D shape models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, vol. 2, pp. 8–14.

[4] C. C. Chang, S. M. Hwang, and D. J. Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid," *Pattern Recognition*, vol. 24, no. 11, pp. 1053–1063, 1991.

[5] K.-M. Lee and W. N. Street, "Automatic segmentation and classification using on-line shape learning," in *Proceedings of the 5th IEEE Workshop on the Application of Computer Vision*, 2000, pp. 64–70.

[6] K.-M. Lee and W. N. Street, "Model-based detection, segmentation and classification for image analysis using on-line shape learning," *Machine Vision and Applications, in press.*

[7] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, 1991.

[8] H. A. C. Eaton and T. L. Olivier, "Learning coefficient dependence on training set size," *Neural Networks*, vol. 5, no. 2, pp. 283–288, 1992.

[9] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[11] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, "Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates," *Cancer Letters*, vol. 77, pp. 163–171, 1994.

[12] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.

[13] W. N. Street, "Xcyt: A system for remote cytological diagnosis and prognosis of breast cancer," in *Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis*, L. C. Jain, Ed., pp. 297–322. World Scientific Publishing, 2000.

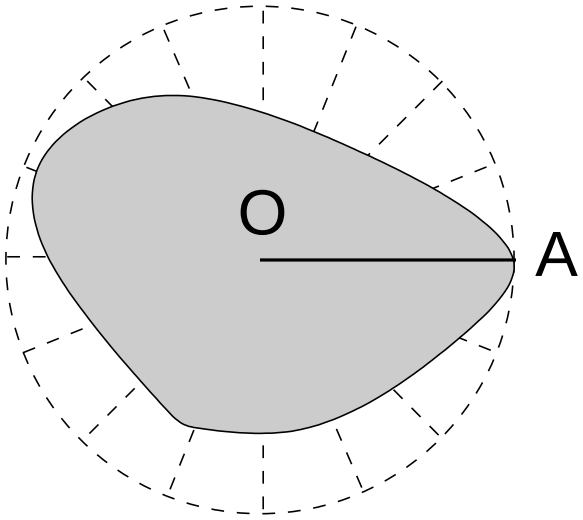

Fig. 1. Shape representation using the centroid-radii model

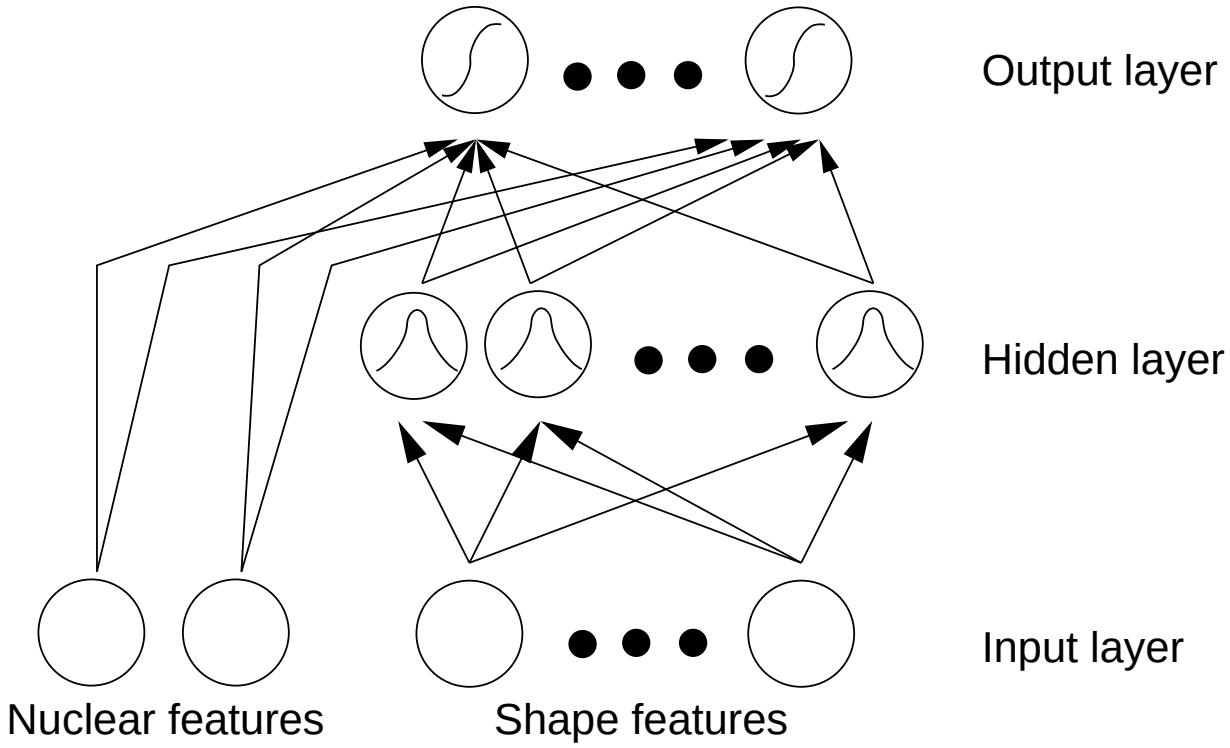

Fig. 2. Architecture of the proposed neural network

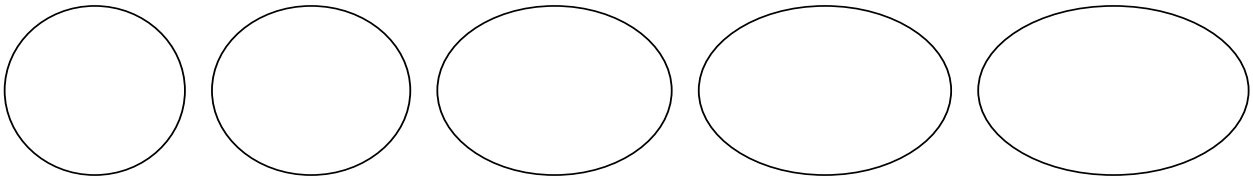

Fig. 3. Examples of initial templates for cell nuclei

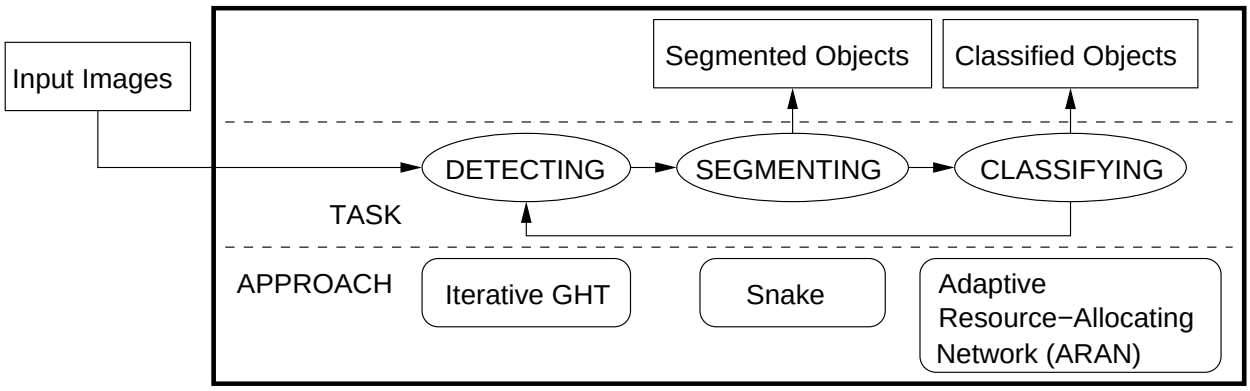

Fig. 4. Flowchart of the automatic object-based image analysis system

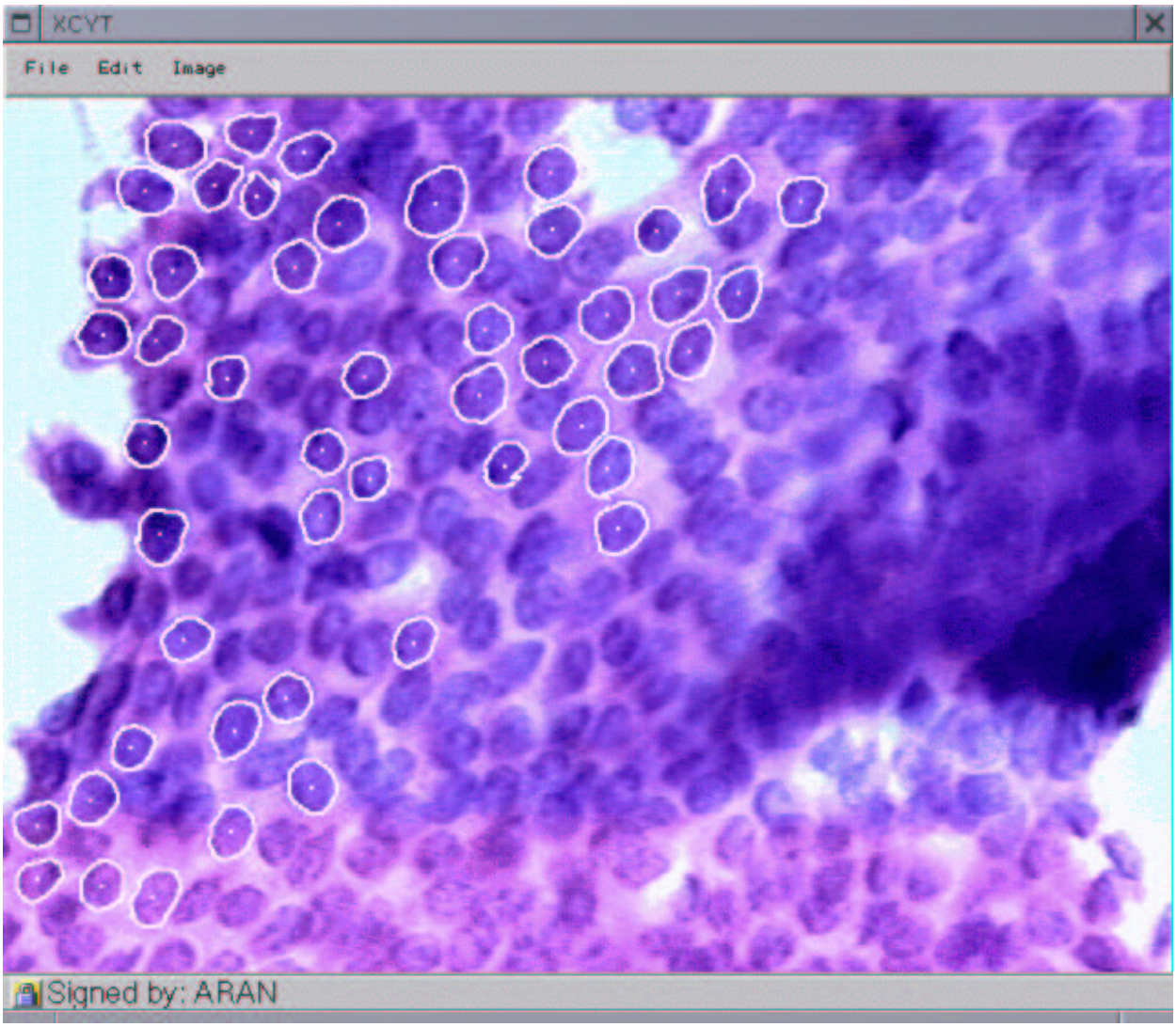

Fig. 5. Test image reduced by 50% where white boundaries are results of IGHT and snake and are input to the neural network.

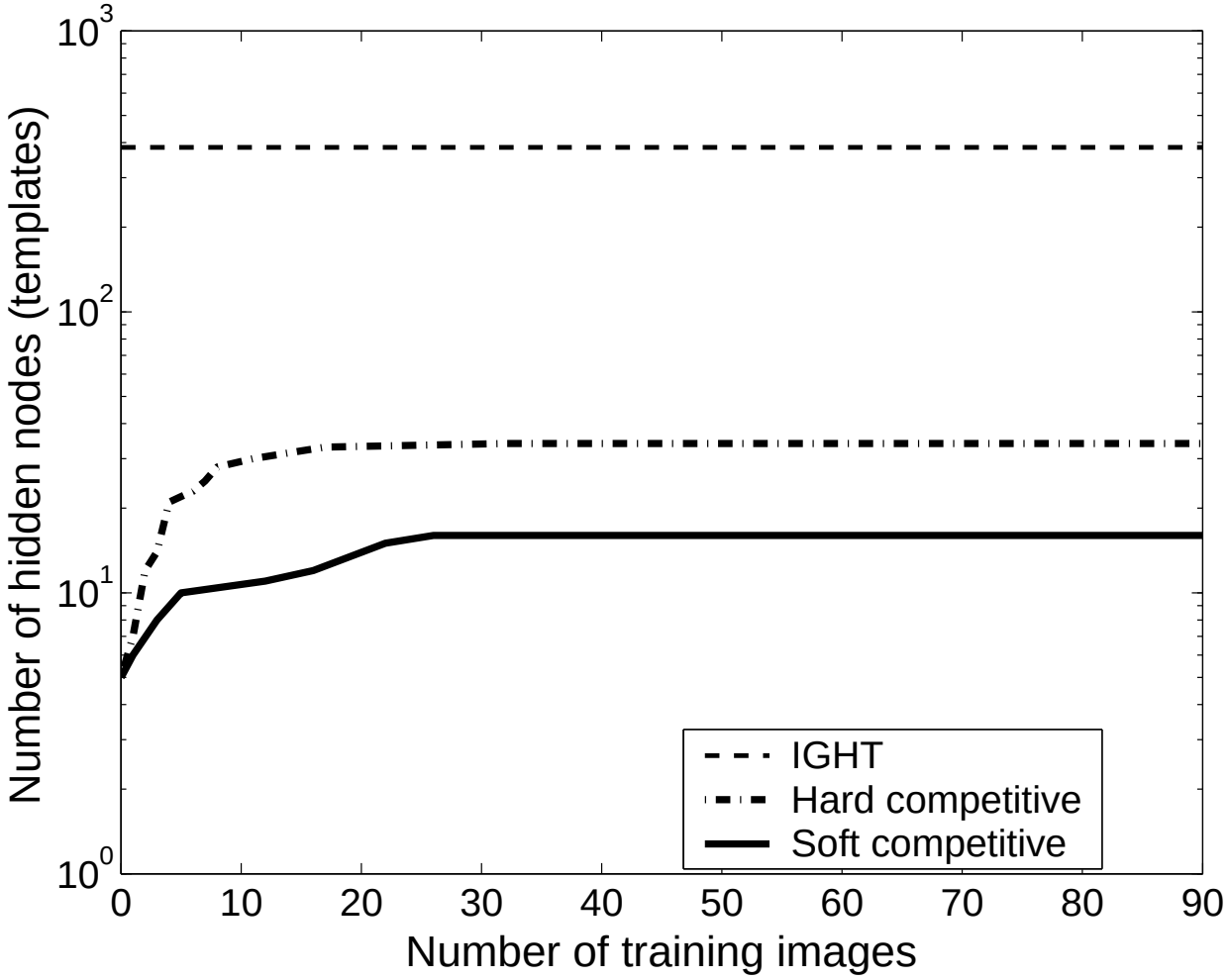

Fig. 6. Number of hidden nodes (templates)

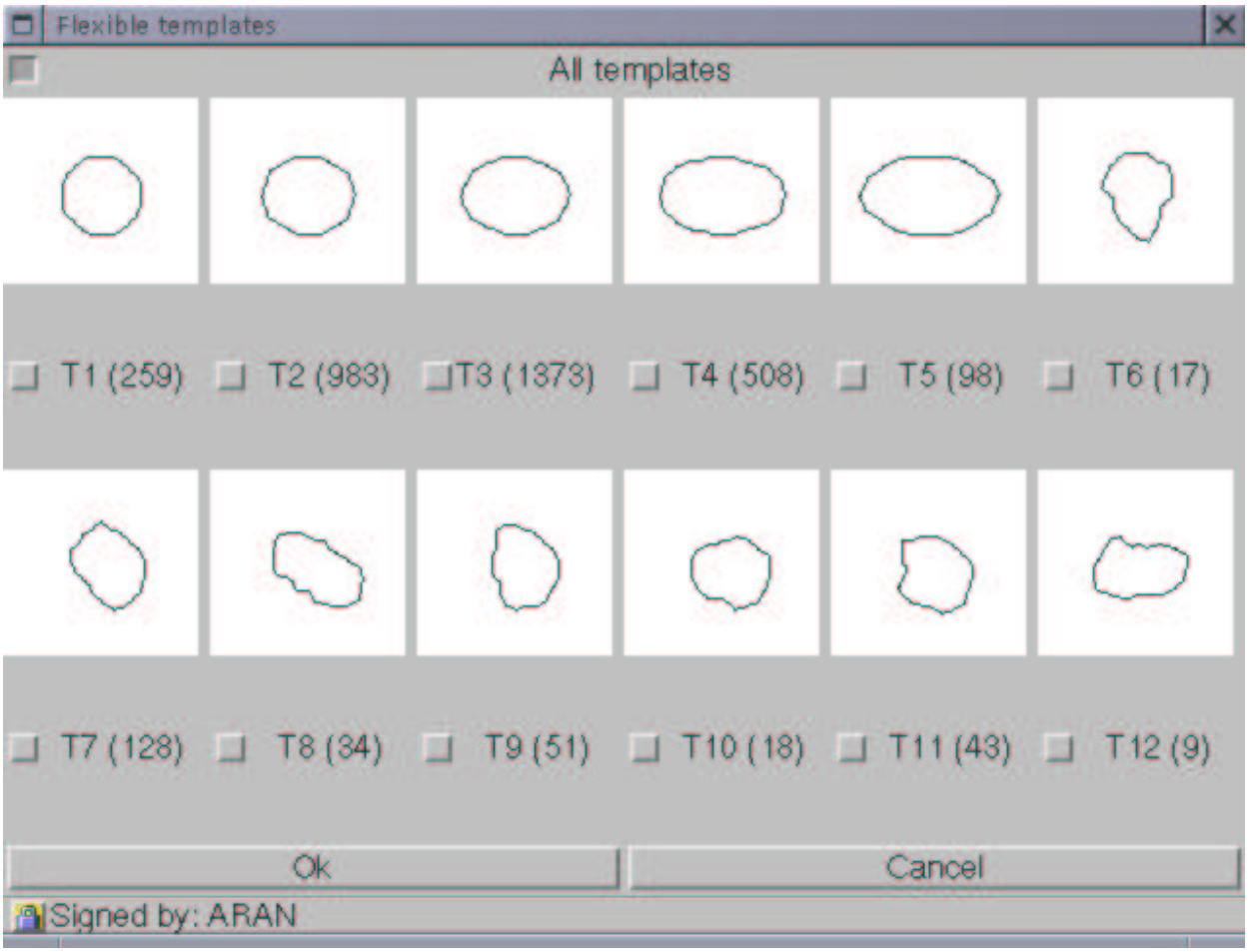

Fig. 7. Set of cell templates after training 90 images where the first five templates are initial template in Fig. 3 and the following 7 templates are created during training

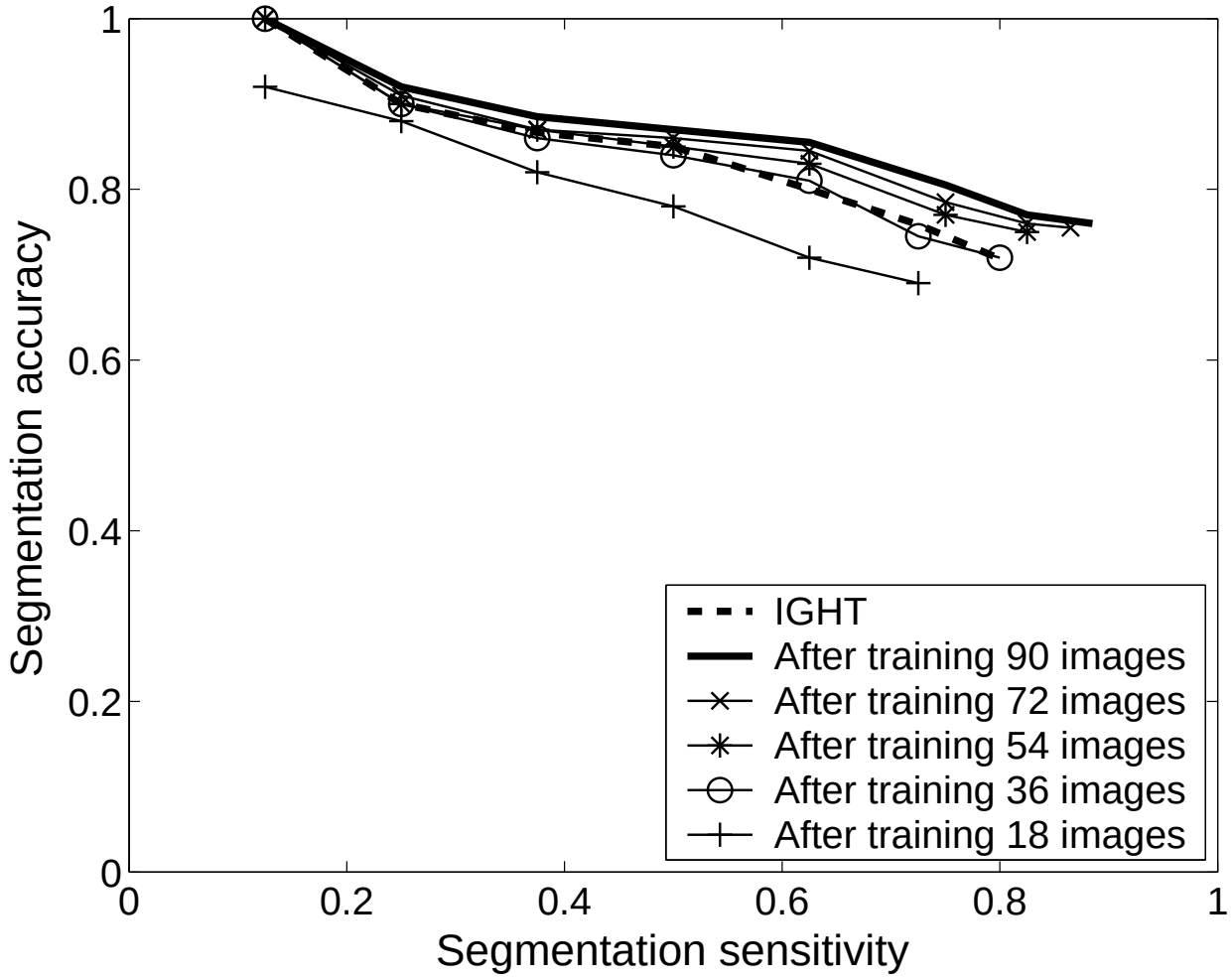

Fig. 8. Segmentation accuracy on test images. For each image, as more nuclei are outlined, the segmentation accuracy is measured at various sensitivity levels. Curves are computed every 18 training images, and averaged over the ten test images.

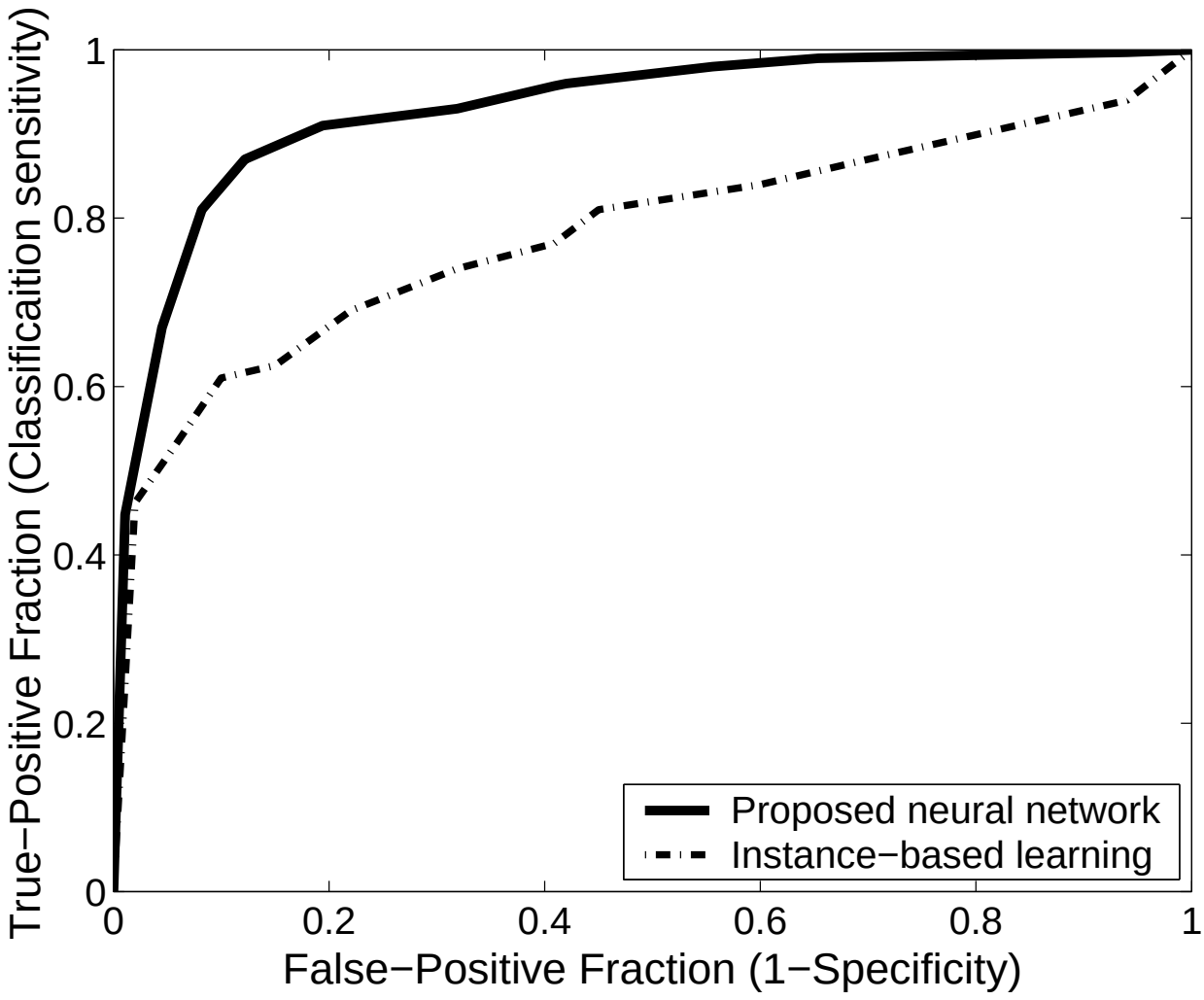

Fig. 9. Classification performance compared of the proposed neural network and instance-based learning

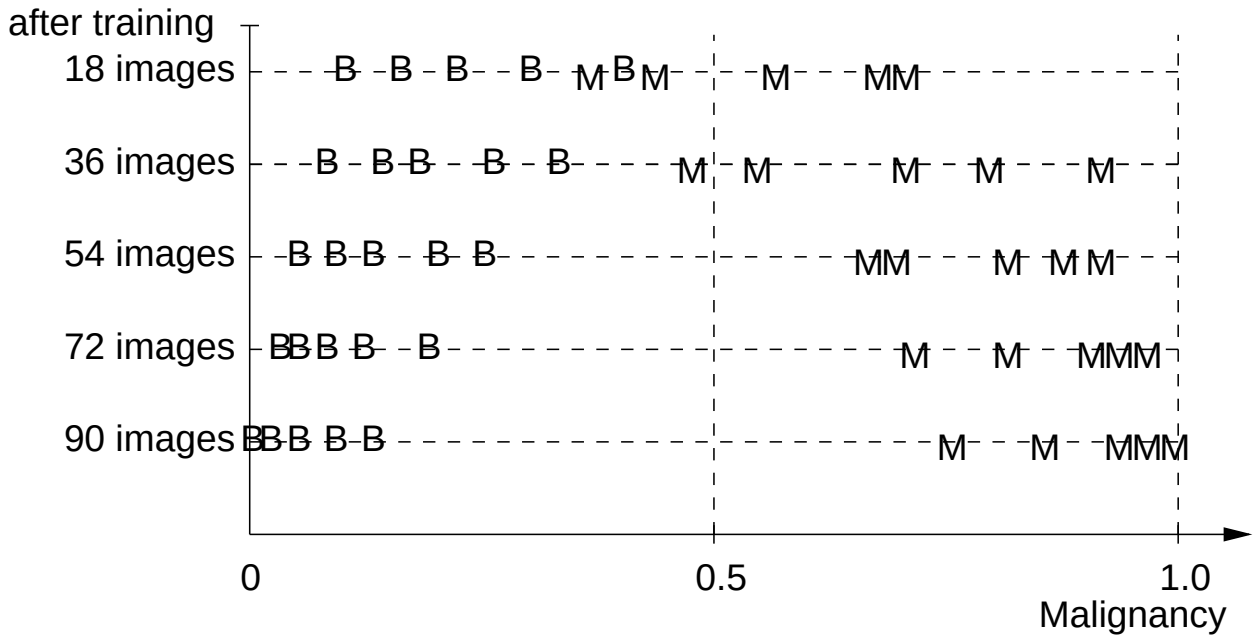

Fig. 10. Classification of images where B means a benign test image and M a malignant test image

TABLE I. Intra-cluster distance between a template and shapes in its corresponding cluster

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
6.38	16.85	20.36	16.00	23.52	26.26	20.95	33.79	23.14	31.65	27.09	37.52

TABLE II. Inter-cluster distance between templates

	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
T1	17.66	34.60	55.76	70.07	37.98	42.68	60.8	43.57	49.97	41.57	49.06
T2	0	20.00	39.67	54.27	53.12	49.99	65.2	52.27	54.75	49.59	48.12
T3		0	22.60	35.92	68.80	65.16	72.08	66.95	65.00	65.26	56.46
T4			0	21.21	90.37	86.62	87.67	88.57	85.13	85.34	77.82
T5				0	104.44	99.22	98.65	101.20	95.96	96.00	91.52
T6					0	37.67	67.62	35.14	60.00	51.18	77.37
T7						0	45.00	29.00	37.25	41.04	67.96
T8							0	56.39	45.56	51.53	59.38
T9								0	44.01	38.44	74.40
T10									0	33.47	43.97
T11										0	56.33

TABLE III. Confusion matrix and classification rate (%) on 10 test images with soft-competitive learning

Class		Trained with images				
True	Predicted	18	36	54	72	90
Benign	Benign	161	176	201	219	228
Benign	Malignant	58	52	35	18	13
Malignant	Benign	72	51	31	16	12
Malignant	Malignant	84	121	147	164	173
Well segmented nuclei		375	400	414	417	426
Classification rate (%)	with nuclear features	65.33	74.25	84.06	91.85	94.13
	with only the size feature	64.27	72.25	80.43	84.89	86.15
	without nuclear features	56.53	57.00	57.49	58.03	58.92

TABLE IV. Cross-validation on 10 sets of 10 test images with soft-competitive learning

Cross validation	Trained with images				
	18	36	54	72	90
Average	65.02	74.25	84.01	91.96	94.13
Standard error	2.36	1.07	0.35	0.12	0.01

TABLE V. Learning rate (η) vs. classification rate (%)

η	Fixed			Adaptive
	0.5	0.1	0.05	
Classification rate	67.37	89.2	73	94.13