INTELLIGENT IMAGE ANALYSIS USING ADAPTIVE RESOURCE-ALLOCATING NETWORK

Kyoung-Mi Lee
Department of Computer Science
The University of Iowa
Iowa City, IA 52242
E-mail: klee1@cs.uiowa.edu
Web: www.cs.uiowa.edu/~klee1

Abstract. This paper presents a unified image analysis approach for object detection, segmentation, and classification using an adaptive resource-allocating network (ARAN), which is based on using unsupervised learning to cluster shapes and supervised learning to classify objects. The proposed neural network is an incrementally grown network by creating a new cluster whenever an unusual shape is presented and by adjusting the clusters. As each hidden node represents each cluster, using centers and widths of the hidden nodes as templates provides faster and more accurate object detection and segmentation, as system performance improves with continued use. The effectiveness of the resulting system is demonstrated on a medical diagnosis task using cytological images.

INTRODUCTION

Three central tasks in image analysis are the detection, segmentation, and classification of objects from given images. These three problems have been addressed with neural networks as a means for gathering knowledge used in the imaging process and thus for producing more robust image analysis systems. While specially they are useful tools for classification because they realize a nonlinear mapping between input and output spaces, most neural network approaches in detection and segmentation were focused on examining pixel intensities within a spatially well-defined boundary. A current trend in automatic image analysis, however, is the use of model-based methods [8]. While a shape can be represented by various models, regardless of the shape model used, it is usually desirable for a model to describe an expected shape, and to contain information about shape variation within the training set [4].

Furthermore, most neural networks applied in image analysis have fixed size. In general, this approach is useful only when the network architecture is chosen correctly. Too small a network cannot learn the problem well, but too large a size will lead to overfitting and poor generalization performance. Thus it is highly desirable that algorithms can find an appropriate network architecture automatically. Recently, various researchers have investigated by altering the network architecture as learning proceeds. One involves using a larger than needed network and training it until an acceptable solution is found [12]. The other approach starts with a small network and then adds additional hidden nodes and weights until satisfactory solution is found [6].

In this paper, we propose an adaptive on-line neural network which is based on radial basis function (RBF) whose centers are interpretable as "prototypes". Section 2 reviews the background algorithms and Section 3 describes the shape model for a single shape and a clustered shape. In Section 4, we propose the neural network for on-line modeling. Section 5 presents the object-based image analysis system using the proposed network. Experimental results and conclusions are presented in Section 6 and 7, respectively.

BACKGROUND

The three-layer RBF network has a feedforward architecture with an input layer, $\mathbf{x}$, a hidden layer, $\mathbf{y}$, and an output layer, $\mathbf{z}$. Those are sets of I, J, and K nodes, respectively. Each hidden node represents a single RBF and computes a kernel function of $\mathbf{x}$ which is usually the following Gaussian function:

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

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are the center and the width of the jth hidden node, respectively, and $\|\cdot\|$ is a distance measure that is generally taken to be the Euclidean norm as

$$\|\mathbf{x} - \boldsymbol{\mu}_j\| = \sqrt{\sum_i (x_i - \mu_{ji})^2}, \quad i = 1, 2, \cdots, I, \tag{2}$$

where x_i and μ_{ji} are the ith features of $\mathbf{x}$ and $\boldsymbol{\mu}_j$, respectively. Each output layer node is given by the sigmoid function of the weighted sums of the outputs from the hidden nodes, $\sum_j \omega_{kj}\phi_j(\mathbf{x})$, where ω_{kj} is a weight between the jth hidden node and the kth output node.

In 1991, Platt proposed a sequential learning technique for RBF networks [11]. The resulting architecture was called the resource allocating network (RAN) and found to be suitable for on-line modeling of non-stationary process. Initially, the network contains no hidden nodes. On incoming training examples, based on two criteria, RAN is either grown or the existing network parameters (the centers and weights) are adjusted using a least mean square gradient descent. The first criterion is the prediction error criterion which states that the error in predicting the output by the existing network architecture should be significant. The other criterion is the novelty criterion which states that the distance between the observation and the winning center should be greater than threshold. If both the criteria are met then the data is memorized and a new hidden node is added to the network.

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. The essential components of the proposed system uses a Point Distribution Model (PDM) [3], a statistical shape model. An object description is simply a labeled set of I points $(x_i)_{i=1 \cdots I}$. These must be positioned consistently on each example in the training set.

The centroid-radii model is adopted to represent the shape by a set of points in polar coordinates [2, 8]. Assume that OA is an arbitrary radius of the shape. The algorithm starts from OA and moving clockwise and the circle is divided into I equal arcs to place points around the boundary. So the shape can be represented as a vector $\mathbf{x} = (x_1, x_2, \cdots, x_I) = (r_{\theta_1}, r_{\theta_2}, \cdots, r_{\theta_I})$, where r_{θ_i} is the ith radius from the centroid to the boundary of the shape and $\theta_i = \left(\frac{360}{I}\right) i$. This model is invariant to translation through the use of an object-centered coordinate system and to reflection through the use of the reflected shape. Since each representation can be rotated and store scaling factors, only one shape is needed. However, the lack of the internal information limits to topologically simple shapes with no holes.

A given set of N shapes, $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^N\}$ will be partitioned into clusters such that shapes within the same cluster have a high degree of similarity, while shapes belonging to different clusters have a high degree of dissimilarity. Each of these clusters can be represented by a center of cluster, defined to 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 added into the clustered shape.

ADAPTIVE RESOURCE-ALLOCATING NETWORK

In this section, we propose an adaptive neural network for object-based image analysis (see Figure 1). The proposed network is based on RAN for on-line modeling with incremental growth. The hidden layer of the proposed network first clusters objects based on shapes and then assigns a node to each cluster. To find an activation node in the hidden layer, a given shape is compared to centers of nodes, using the distance measure. In the proposed network, we want to get a clustered shape in multidimensional space. That is, instead of single dimensional width, we need to store the width of each dimension of the clustered shape. Thus Eq. (1) can be modified as following:

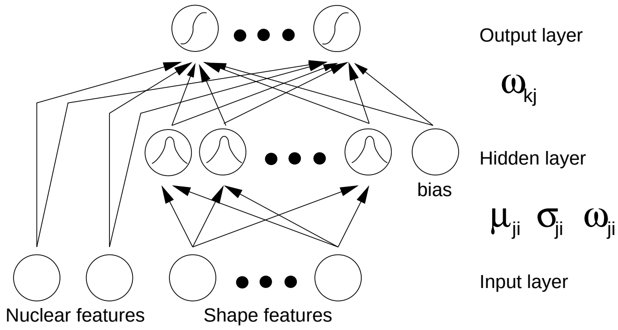


Figure 1: Architecture of the proposed neural network

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

To help classification of heterogeneous shape-based clusters, additional features $\mathbf{m}$ are directly connected to the output layer. In the breast cancer application, six nuclear features as size, radius, perimeter, smoothness, and concavity, are used. So the output layer node is given by the sigmoid function of the combined sum as:

$$z_k = \sum \omega_{km} m + \sum_j \omega_{kj}\phi_j + \omega_{k0}, \quad k = 1, 2, \cdots, K,$$

where m is each nuclear feature, ω_{km} is a weight between the kth output and the mth nuclear feature, and ω_{k0} is a bias.

Initialization of 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 initialize a set of elliptical templates with different shapes. Since the algorithm is invariant to size, orientation and reflection, only one template with each shape is created. A center $\boldsymbol{\mu}_j$ is set to the initial template and an initial width $\boldsymbol{\sigma}_j$ is set by $\mathbf{0}$ (see Eq. (4)). J is set by the number of initial templates. If we assume no knowledge of the shapes, then the network includes no hidden nodes, as the original RAN.

Adaptive 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 be lead to very good results and, hence a large number of different methods for its adaptation have been proposed in the literature [10].

Eaton and Olivier stated that the range of η that will produce rapid training depends on the number and types of input [5]. In addition, they observed that the length of the error gradient increases with the size of the training set. Since the proposed network has clusters at the hidden layer, these conclusions lead to calculate the learning rate value:

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

where α is a constant which is chosen empirically and N_j is the number of data clustered in the jth hidden node. In contrast of Eaton and Olivier's network, since the proposed network is based on on-line learning, we can calculate the adaptive learning rate based on the characteristics of the training data during a sequential training.

Overfitting avoidance

In general, we do not know how many hidden nodes, or equivalently, how many weights, we will need to produce a reasonable approximation to the data. If a network that is too complex and fits the noise, not just the data, it leads to overfitting which can easily lead to predictions that are far beyond the range of the training data and can also produce wild predictions in multilayer perceptrons even with noise-free data.

One way to avoid overfitting is to find the approximated solution of centers and widths while reducing the number of degrees of freedom for the neural network. Tsujii *et al.* proposed to find the approximated solution of centers and widths [14] by adding an inverse $\boldsymbol{\sigma}_j^2$ penalty term to the minimization criterion. In this paper, we have incorporated an inverse σ_j penalty term to the distance metric to give more power to the network to regularize the class distribution. Thus, Eq. (2) can be computed as:

$$\|\mathbf{x} - \boldsymbol{\mu}_j\|_\omega = \sqrt{\sum_i |\omega_{ji}(x_i - \mu_{ji})|^2}, \quad i = 1 \cdots I, \tag{3}$$

where

$$\omega_{ji} = \begin{cases} \dfrac{1}{\sigma_{ji}}, & \text{if } \sigma_{ji} \neq 0 \\ 1, & \text{otherwise.} \end{cases}$$

By adding an inverse $\boldsymbol{\sigma}_j$ penalty term to the distance metric, the training should be more accelerated than the conventional method. To avoid simple orientation differences, the distance is performed at different angles and the final distance is defined to be the minimum of these distances.

Training the network

Training data are supplied to the neural network in the form of pair ($\mathbf{x}^n$,$\mathbf{t}^n$) of input and target vectors, where n means the nth pair. 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 with $\mathbf{x}^n$ by allocating a new hidden node and the number of clusters is increased.

$$N_{J+1} = 1, \quad \boldsymbol{\mu}_{J+1} = \mathbf{x}^n, \quad \boldsymbol{\sigma}_{J+1} = \mathbf{0}, \qquad (4)$$
$$J = J + 1.$$

Thus, after training on a collection of shapes, each hidden node represents a cluster of shapes that are nearby one another 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 $\boldsymbol{\mu}_j$ as well $\boldsymbol{\sigma}_j$ are updated by adding $\mathbf{x}^n$. When learning online, the immediate environment of a single weight within a multi-layer network is highly non-stationary, due to the simultaneous adaptation of other weights, if not due to the learning task itself. Gradient descent is probably the simplest approach for attempting to update as:

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

where α_j is the parameter indicating the similarity between $\boldsymbol{\mu}_j$ and $\mathbf{x}^n$ and is set 1 if $\boldsymbol{\mu}_j$ is the winning cluster.

Due to the smaller steps that are taken for the adaptation process, competitive learning will lead quickly to a nearby optimum, while the winner-take-all function will have more possibilities to get stuck into well-separated local optima [13]. So the proposed network updates in addition to the winner also some other clusters depending on their similarity with $\mathbf{x}^n$:

$$\alpha_j = \frac{\|\mathbf{x}^n - \boldsymbol{\mu}_j\| - \min\limits_{1 \le l \le J}(\|\mathbf{x}^n - \boldsymbol{\mu}_l\|)}{\max\limits_{1 \le l \le J}(\|\mathbf{x}^n - \boldsymbol{\mu}_l\|) - \min\limits_{1 \le l \le J}(\|\mathbf{x}^n - \boldsymbol{\mu}_l\|)}.$$

Since nuclear features are connected directly to the output node, it's relationship is the same to the two-layer network. So bias and weights with nuclear features are updated, if $z_k^n \neq t_k^n$, as:

$$\begin{aligned}
\Delta\omega_{k0} &= \eta|z_k^n - t_k^n|, \\
\Delta\omega_{km} &= \eta|z_k^n - t_k^n| m.
\end{aligned}$$

DESCRIPTION OF THE IMAGE ANALYSIS SYSTEM

Figure 2 shows a flowchart of the proposed object-based image analysis system for object detection, segmentation and classification. While these three parts are performed repeatedly, the system learns and gathers knowledge of shapes the user wants to find.

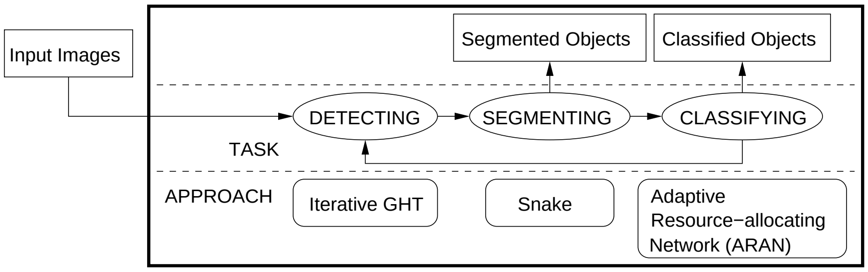


Figure 2: Flowchart of the automatic object-based image analysis system

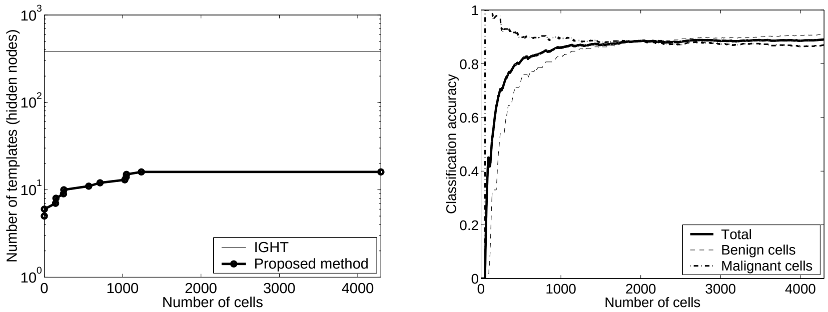
For object detection, the generalized Hough transform (GHT) algorithm [1] is the most popular template matching method of detecting or locating shapes by making use of gradient direction information at edge pixels. In this paper, the system uses an iterative version of GHT (IGHT) [7] which incorporates a clustered shape $\boldsymbol{\mu}$, defined as a flexible template [9]. For each new image, IGHT is performed using the existing set of templates. The appropriate template is overlaid onto the image at the appropriate rotation angle. This process is repeated until a user-defined number of objects has been located, or until the match scores degrade below an adjustable threshold. Then, the snakes to segment objects are initialized using the results of IGHT that searches for ellipses of various sizes. The segmented objects are presented into the proposed network for classification.

An important aspect of the proposed 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.

EXPERIMENTAL RESULTS

The algorithm was tested on cytological images from fine-needle biopsies of breast masses, the images gray-scaled with 640×480 pixel spatial resolution. These images are classified as benign or malignant on a per-sample basis; no classification is available for individual cell nuclei. We used 24 radii for cell shapes and 6 nuclear features for non-shape features, as well 5 initial elliptical templates. So, the network for cell classification initially consists of 30 input nodes, 5 hidden nodes, and 2 output nodes.

The system was trained on a sequence of 90 training images, (4295 cells), alternating between benign (2250 cells) and malignant (2045 cells). Figure 3 (a) shows the number of 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 encoun-



(a) (b)

Figure 3: Results during training

tered. The neural system builds a set of templates as it is used. After an initial ramping up period, these numbers set very stable at a level more than an order of magnitude below the original set of templates. The curve labelled "Proposed method" shows the result of searching the templates in order of their frequency in previous images. After sorting the set of templates, we applied 5 templates at once. Whenever the result degrades, next 5 templates are used. The search is stopped when a result equivalent in segmentation accuracy to searching all the learned templates is reached. This results in a 95.84% reduction in the number of necessary templates. In addition, the user can reduce search time even further, by choosing specific templates or by choosing a preferred size.

Classification accuracy was also measured as the training preceded. Each incoming object is classified in the previously trained neural network. Figure 3 (b) shows that as the system is trained on more and more objects, its ability to correctly classify the nuclei increases. After a few images, the classification rate stabilizes at 89.01 on average: 90.84% for the benign cells and 86.99% for the malignant cells.

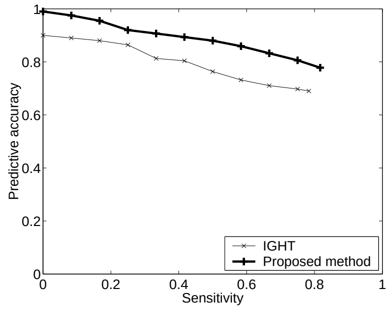


Figure 4: Segmentation accuracy

We tested 10 images, 5 benign and 5 malignant, by showing a pair of one benign image and one malignant image after training every 18 images. Segmentation performance was measured as follows. Test images were set aside and used for segmenting and classifying automatically with all trained templates after training images. We used two measures to evaluate the performance of the algorithm: sensitivity and predictive accuracy. Since it is typically required to segment a few dozen cells for diagnosis and prognosis, we use the predictive accuracy of the segmentation method on the first 50 outlined nuclei (in the case of the benign images) or at 80% sensitivity (in the case of the malignant images). For 472 more cells, 250 benign cells and 222 malignant cells, with all trained templates, we achieved 80.72% as an average segmentation correctness: 84.4% in the benign cells and 76.58% in the malignant cells. Figure 4 shows the average accuracy of 10 test images. The new system has better accuracy to the original IGHT algorithm, and achieves this performance in an order of magnitude less time per image.

Table 1: Confusion matrix for cells on 10 test images

Class		Test				
True	Predicted	1	2	3	4	5
Benign	Benign	31	34	38	40	41
Benign	Malignant	10	7	6	3	1
Malignant	Benign	12	9	5	2	1
Malignant	Malignant	14	25	27	39	36
Classification rate (%)		67.16	78.67	85.53	94.05	97.47

Table 1 shows the classification counts for well-segmented cells. With only the one training set (18 images), cells were classified as 67.16% in average. As the number of training sets reached five, average classification rate achieved 97.47%. For 381 well-segmented cells in the segmentation test of 10 images, we achieved 85.3% as an average classification correctness: 87.2% in the 211 benign cells and 82.94% in the 170 malignant cells.

CONCLUSION

This paper proposes a neural network approach for the intelligent image analysis. The following summarizes the contribution in this paper. The proposed network uses

- a combined structure, with a three-layer for shape features and a two-layer for nuclear features, to build a model-based image analysis system,
- weighted distance metrics along each dimension to avoid overfitting,
- an adaptive learning rate for on-line learning, and
- a competitive learning for efficient optimization.

This paper demonstrates how the proposed neural network is used to aid the detection, segmentation and classification of objects being outlined. Using centers and widths of the network provides a guide for an intelligent search through the space of possible objects. Further directions of this work shall include a prunning technique and different learning rates for different nodes.

ACKNOWLEDGMENTS

The author would like to thank W. Nick Street who not only provided useful comments on this work, but also supported by NIH grant CA64339-04 and NSF grant IIS-99-96044.

REFERENCES

[1] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," **Pattern Recognition**, vol. 13, no. 2, pp. 111–122, 1981.
[2] 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.
[3] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham, "Active shape models-their training and application," **Computer Vision and Image Understanding**, vol. 61, no. 1, pp. 38–59, 1995.
[4] N. Duta, A. K.Jain and M.-P. Dubuisson-Jolly, "Learning 2D shape models," in **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, 1999, vol. 2, pp. 8–14.
[5] 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.
[6] T. Y. Kwok and D. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems [review]," **IEEE Transactions on Neural Networks**, vol. 8, no. 3, pp. 630–645, 1997.
[7] K.-M. Lee and W. N. Street, "A fast and robust approach for automated segmentation of breast cancer nuclei," in **Proceedings of the IASTED International Conference on Computer Graphics and Imaging**, 1999, pp. 42–47.
[8] 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.
[9] K.-M. Lee and W. N. Street, "A new approach of generalized Hough transform with flexible templates," in **Proceedings of the 2000 International Conference on Artificial Intelligence**, 2000, vol. III, pp. 1133–1139.
[10] M. Moreira and E. Fiesler, "Neural Networks with Adaptive Learning Rate and Momentum Terms," Techn. Report 95-04, **IDIAP, Martigny, Switzerland**, 1995.
[11] J. Platt, "A resource-allocating network for function interpolation," **Neural Computation**, vol. 3, no. 2, pp. 213–225, 1991.
[12] R. Reed, "Pruning algorithms - a survey," **IEEE Transactions on Neural Networks**, vol. 4, no. 5, pp. 740–747, 1993.
[13] P. Scheunders, "A comparison of clustering algorithms applied to color image quantization," **Pattern Recognition Letters**, vol. 18, pp. 1379–1384, 1997.
[14] O. Tsujii, M. T. Freedman and S. K. Mun, "Classification of microcalcifications in digital mammograms using trend-oriented radial basis function neural network," **Pattern Recognition**, vol. 32, no. 5, pp. 891–903, 1999.