Multiple Competitive Learning Network Fusion for Object Classification

Xiaoou Tang

***Abstract*—This paper introduces a multiple competitive learning neural network fusion method for pattern recognition. By defining a confidence level measure for the learning vector quantization network classifier, we develop both a serial and a parallel network fusion algorithm to combine the discriminatory ability of different individually trained networks. We use two distinct feature vectors, gray-scale morphological granulometry and Fourier boundary descriptor, to demonstrate the efficacy of the classifier. The algorithms are applied on the classification of more than 8000 underwater plankton images. The classification accuracy for training data and for testing data are over 92% and 85%, respectively.**

***Index Terms*— Classifier, learning vector quantization neural network, network fusion, object recognition.**

I. Introduction

THREE-DIMENSIONAL object classification through two-dimensional images has long been an important topic in pattern recognition. Applications range from industrial inspection and scene analysis to medical image processing. Research has been centered around the two basic components of a general object recognition system: extraction of a set of features for image representation [1]–[7] and classification of the extracted feature vector [8]–[13].

This paper focuses mainly on the classifier design aspect of an image classification system. Specifically, we develop a network fusion approach for the learning vector quantization (LVQ) network classifier. Using a confidence level (CL) measure to judge whether a pattern is classified correctly by a LVQ network to the degree of acceptable confidence, we can decide whether to pass it through more networks trained by different feature vectors. The idea is that a sample pattern that falls near a class boundary of one network, may fall into a class center of another network trained by a different set of features. Both a serial and a parallel network fusion approaches have been developed to combine the discriminatory power of individual networks. To demonstrate the effectiveness of the designed classifier, we select two feature vectors, Fourier descriptors (FD) and gray-scale granulometries, which have been studied extensively in previous pattern recognition research [1]–[7], [14]–[19], and test the algorithms in a set of interesting plankton object identification experiments.

Marine plankton distribution study is fundamental to biological oceanographic research. To conduct high-resolution mapping of plankton taxonomic abundance, Davis *et al.* designed an underwater video microscope system called video plankton recorder (VPR) [20]–[22]. When towed through the water, the VPR continuously captures magnified plankton images (Figs. 1 and 2), providing information on the plankton distribution on scales from microns to hundreds of kilometers. The overwhelmingly large amount of image data make manual processing laborious and time consuming, necessitating an automatic approach.

In this paper, we test the multiple LVQ networks on the classification of two data sets obtained by two VPR cameras of different magnifications. The data comprise nine classes of more than 8000 images. We achieve over 92% classification accuracy for training data and over 85% accuracy for testing data, effectively demonstrating the efficacy of the algorithms.

In the next section, we review Fourier boundary descriptors, granulometric features, and the LVQ classifier. Section III introduces a neuron classification confidence level measure and two multiple network combination schemes. In Section IV, we present the classification results of the plankton images using the neural network classifiers. Finally, Section V summarizes the conclusions and future work.

II. Basic Algorithm Review

This section gives a background review of the two feature vectors (the Fourier descriptor and the gray-scale granulometry) and the LVQ neural network classifier.

A. Fourier Descriptor

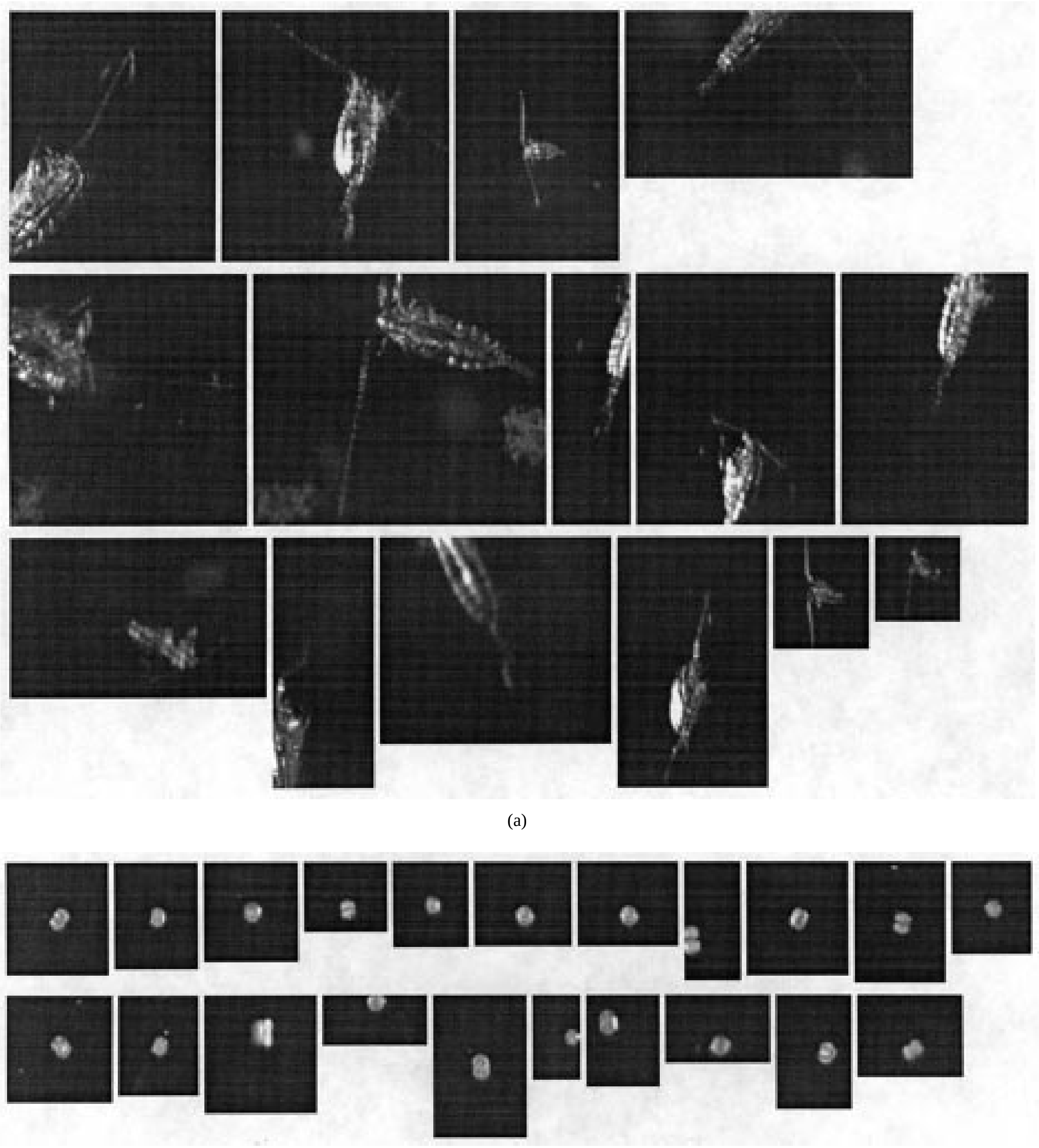
Fourier descriptors have been utilized as pattern features in a number of applications [1]–[7]. Kauppinen *et al.* [7] gave a detailed experimental comparison of different models, including the curvature function, centroidal radius, and complex contour coordinates. In our experiments, we use the radius Fourier descriptor which has been shown to be one of the best among the FD's tested in [7].

For a closed boundary defined by a closed sequence of successive boundary pixel coordinates, the centroidal radius function is the distance from the boundary points to the centroid of the object. Then the Fourier descriptor is defined as a normalized discrete Fourier transformation (DFT) of the radius

Manuscript received June 1, 1996; revised November 8, 1997. This work is supported by the Office of Naval Research under Grant N00014-93-1-0602. This is contribution number 9214 of the Woods Hole Oceanographic Institution.

The author is with the Deep Submergence Laboratory, Woods Hole Oceanographic Institution, Woods Hole, MA 02543 USA (e-mail: xtang@whoi.edu).

Publisher Item Identifier S 1083-4419(98)04984-X.



(a)

(b)

Fig. 1. Sample plankton images from camera #1 (a) Calanus and (b) Diat-centr.

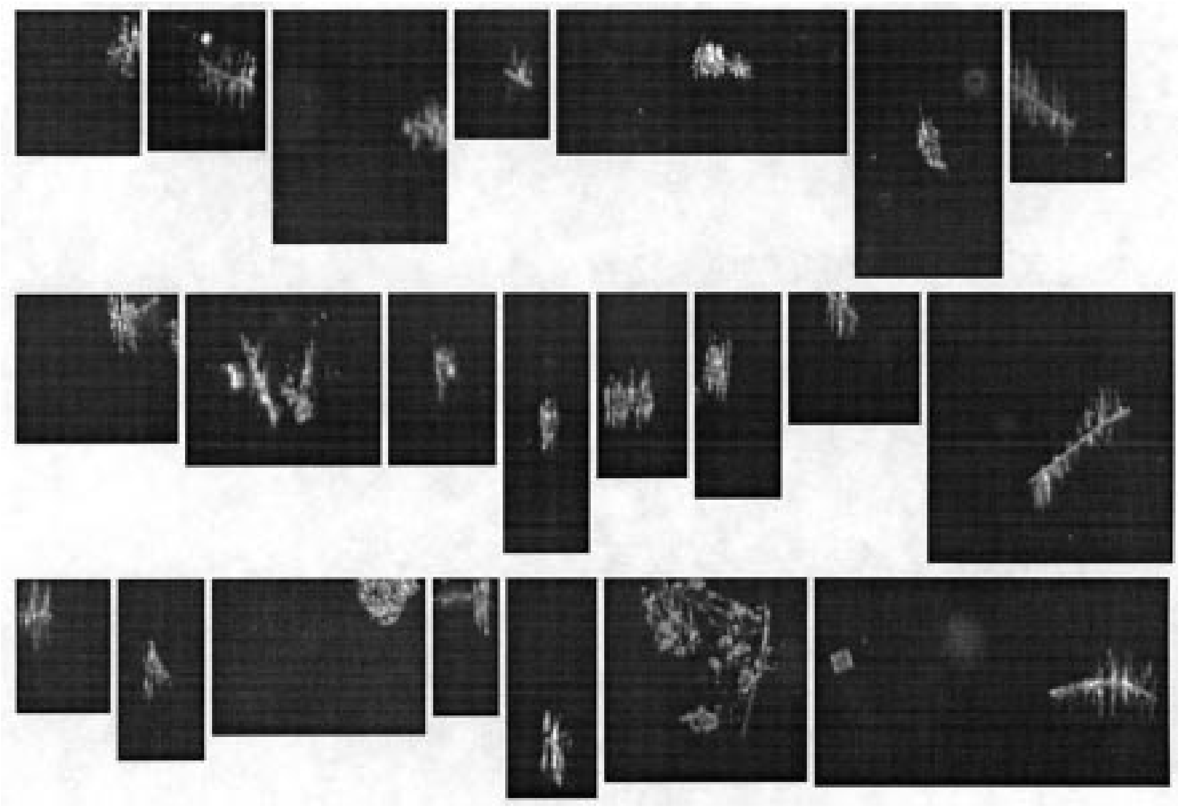
boundary function. Since the FD features are computed around the centroid of the object, they are translation invariant. The rotation and scale invariance of the feature vector can be derived from the shift invariant and linear properties of the DFT magnitudes. Only half of the spectrum is used because of the symmetry property of the Fourier transform of real functions.

The sampling number we use for the radius boundary function is 256 points. This number is much higher than many previous studies, because plankton objects have noisier, more irregular boundaries requiring a higher sampling rate to capture high-frequency information.

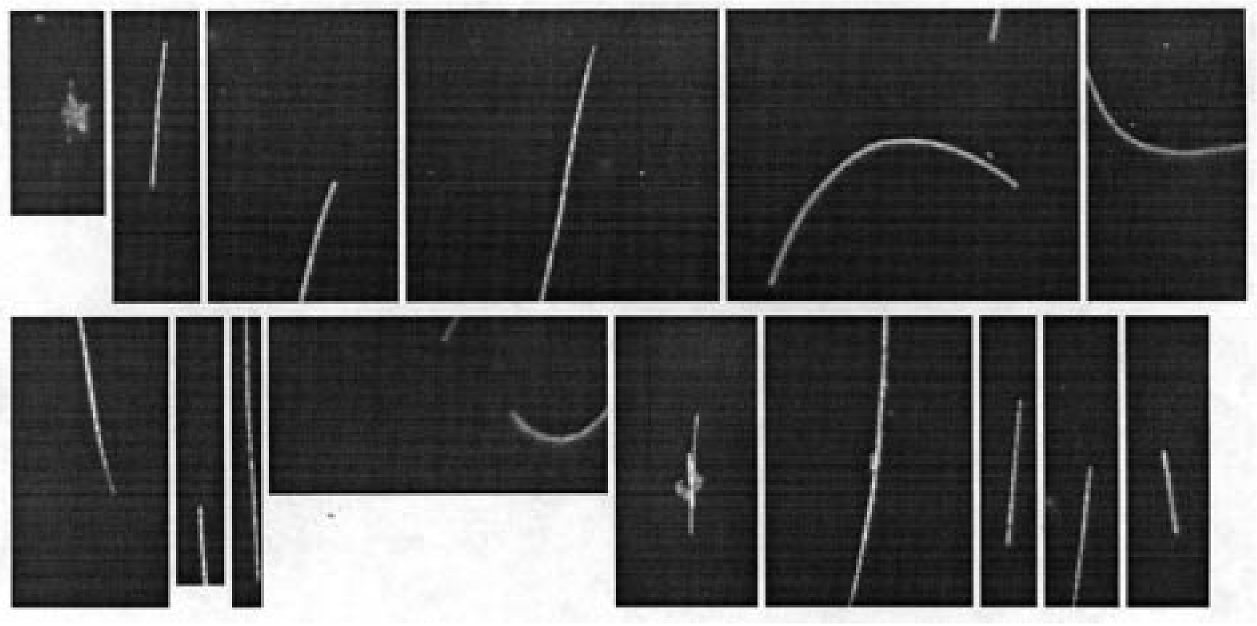
B. Gray-Scale Granulometries

Granulometries were introduced first by Matheron as tools to extract size distributions from binary images [14]. By performing a series of morphological openings of increasing kernel size, we can obtain the granulometry function which maps each kernel size to the number of image pixels removed during the opening operation with the corresponding kernel [16]. In mathematical morphological terms, the granulometric pattern function is defined as follows:

$$A(k) = \mathrm{sum}\{I \circ S_k\} \qquad (1)$$



(c)



(d)

Fig. 1. (*Continued.*) Sample plankton images from camera #1 (c) Diat-chaft and (d) Diatom-rod.

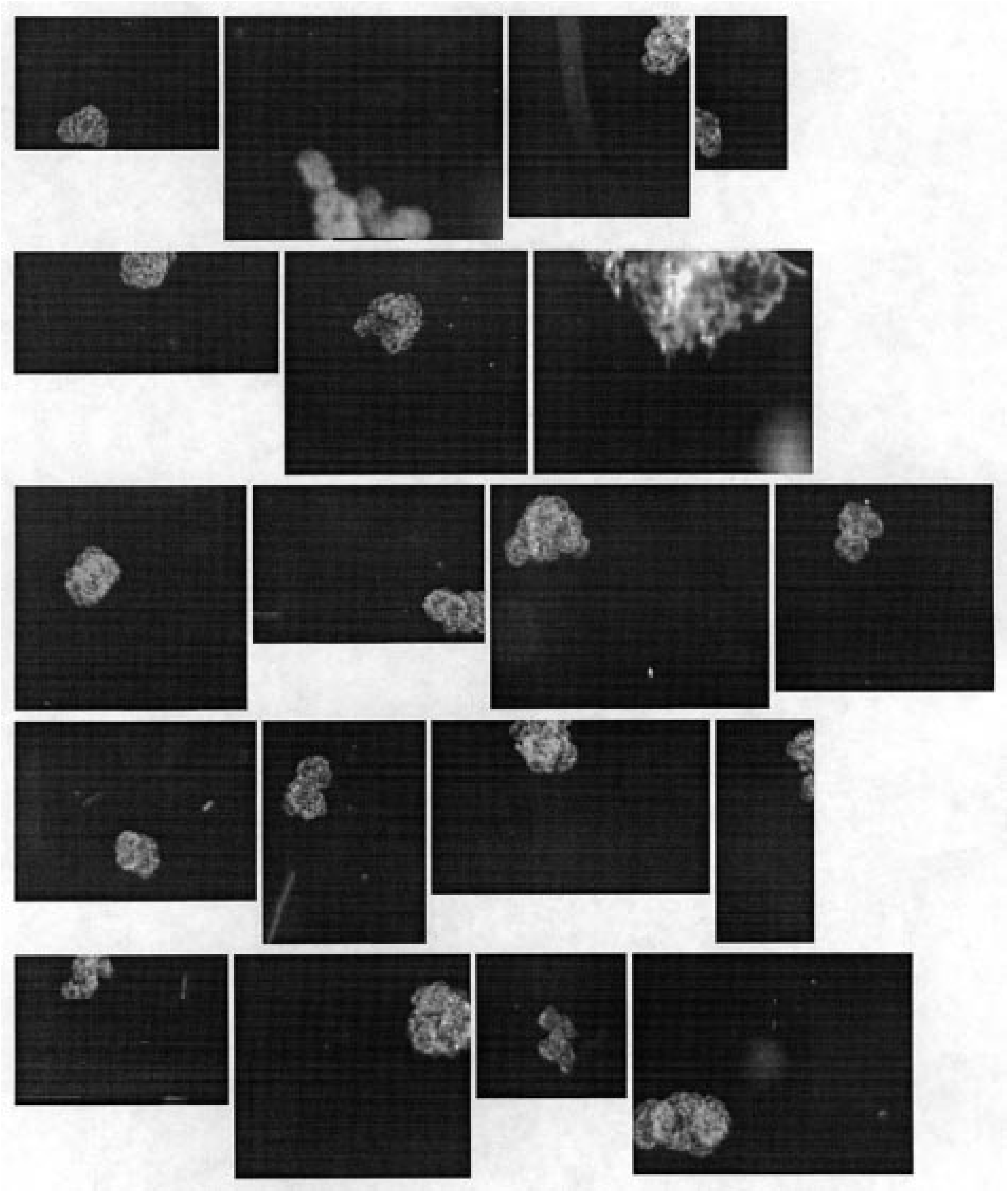
where "$\circ$" is the binary morphological opening, I represents the original image, and S_k, $k = 1, 2, \cdots$, is a sequence of opening kernels of increasing size. $A(k)$ is the sum of the number of image pixels in $I \circ S_k$, which is often called the size distribution. A normalized size distribution is

$$N(k) = 1 - \frac{A(k)}{\mathrm{sum}(I)}. \tag{2}$$

The granulometric pattern function is then defined as

$$G(k) = N(k+1) - N(k) \tag{3}$$

which can be viewed as the discrete density function of image structure sizes. The concept of granulometries can easily be extended to gray-scale images by replacing the binary opening with a gray-scale morphological opening [18]. We

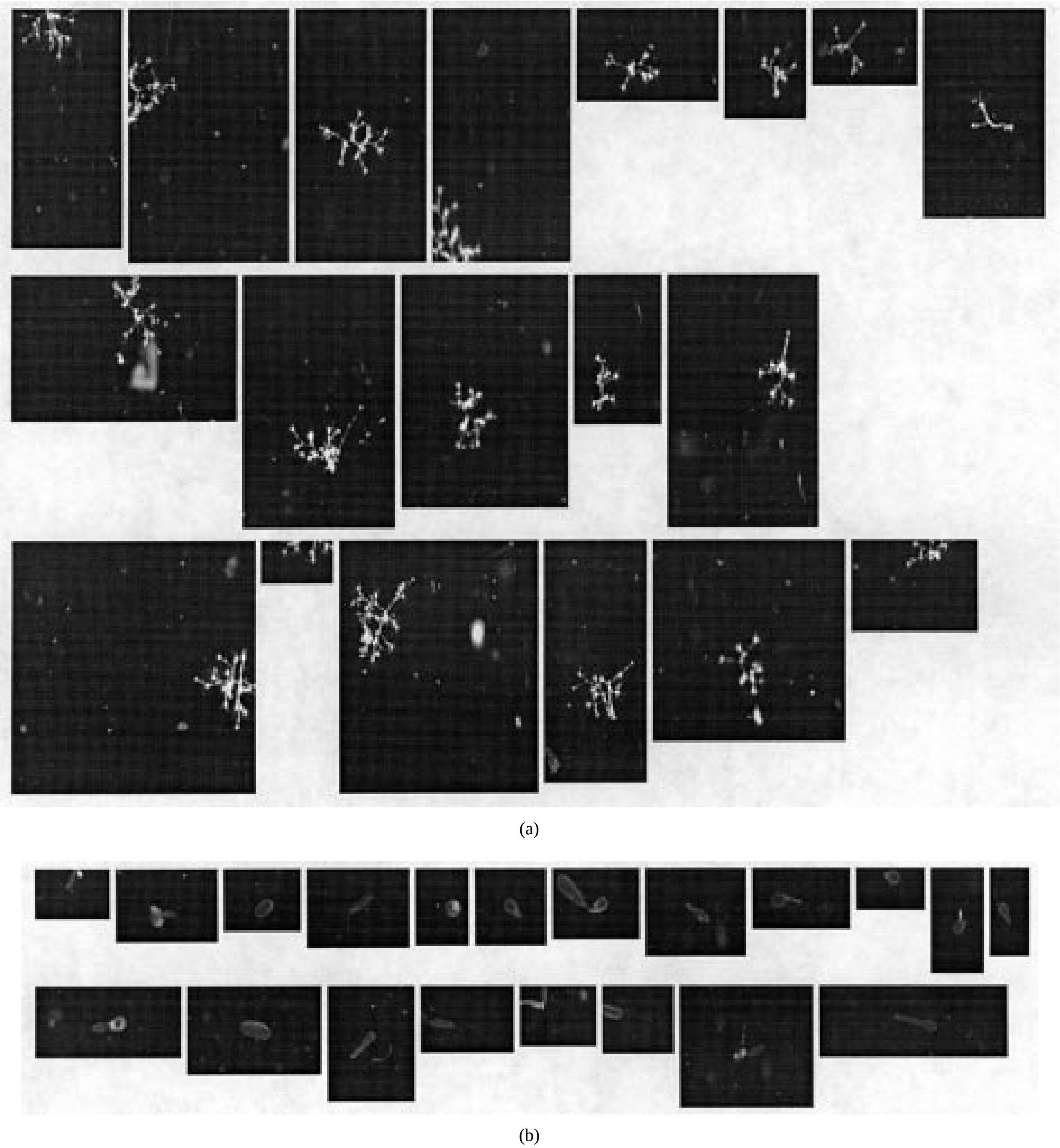


(e)

Fig. 1. (*Continued.*) Sample plankton images from camera #1 (Diat-chsoc).

can understand the gray-scale granulometries from a one-dimensional signal shown in Fig. 3. Intuitively, a gray-scale opening can be viewed as sliding the structuring kernel beneath the signal and at each position recording the top points on the structure element. After the first opening in Fig. 3(a), the resulting curve is removed of structures smaller than the size of the first opening kernel (all the dark dots in the original signal), as shown in Fig. 3(b). The same process reiterates with increasing opening kernel sizes. Fig. 3(c) shows the granulometric curve representing the volume removed at each opening kernel size.

In addition, different types of gray-scale granulometric curves can be computed, depending on the underlying family of openings or closings used. For example, curves based on



(a)

(b)

Fig. 2. Sample plankton images from camera #2 (a) Hydroids and (b) Phaeoblimp.

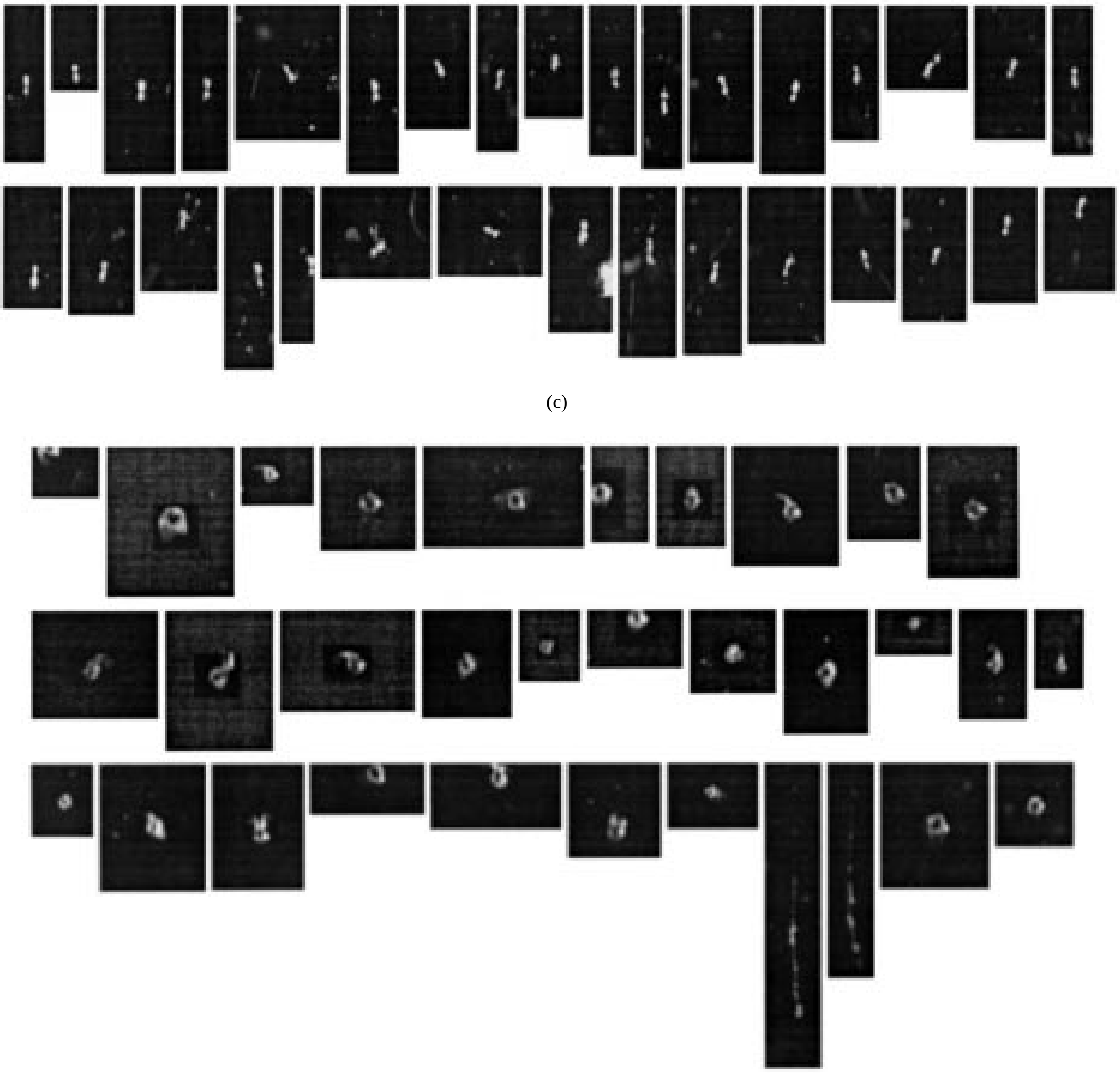
openings with line structures capture information on bright linear image features, whereas curves based on closings with disk-shaped elements capture information on dark, blobby image parts. To capture information on both linear and blobby image features, whether dark or bright, we use all the four types of gray-scale granulometries [19]. A fast gray-scale granulometry algorithm developed by Vincent [17], [18] is used to compute the granulometric curves in this work.

Very large feature vectors usually contain redundant information and require more computational power to classify. The Karhunen–Loeve transform (KLT) is adopted to reduce feature vector length for our system. Its decorrelation ability serves to decorrelate neighborhood features, and its energy packing property helps to compact useful information into a few dominant features [19].

C. Learning Vector Quantization Classifier

The learning vector quantization classifier [8]–[10] is selected for feature classification because of its ability to map complex and diverse feature distribution without making statistical assumptions.

The objective of a vector quantization process is to find a compact representation of a set of feature vectors $x \in R^n$ by M codebook vectors, also called prototypes, $\omega_i \in R^n$, whose



(c)

(d)

Fig. 2. (*Continued.*) Sample plankton images from camera #2 (c) Pseudowegg and (d) Pteropod.

local point density approximates the probability density function $p(x)$. Kohonen developed a supervised neural network learning algorithm, called the learning vector quantization, to allocate these prototypes [8]–[10].

A LVQ network consists of two layers, a hidden competitive layer and a linear output layer, as shown in Fig. 4. The prototype vectors are the connection weights between the input neurons and the hidden neurons. When a training vector x is fed into the network, Euclidean distances are computed between x and each neuron weight vector ω_i. The neuron whose weight vector has the smallest distance to the input vector is the winner neuron. The second layer transforms the hidden competitive layer's neuron class into the final output class. The competitive layer neurons are connected to the output classes according to *a priori* probabilities of the classes. If any hidden neuron in a particular class wins, the corresponding neuron class in the linear output layer becomes the class label of the input vector.

The training process is governed by a set of learning rules. Let the input training vector x belong to class C_t, and its closest codevector ω_r be labeled as class C_s. Then ω_i is updated by the learning rules [9]

$$\begin{aligned} \Delta\omega_r &= \alpha(x - \omega_r) \quad \text{if } C_s = C_t \\ \Delta\omega_r &= -\alpha(x - \omega_r) \quad \text{if } C_s \neq C_t \\ \Delta\omega_i &= 0 \quad \text{for } i \neq r \end{aligned} \tag{4}$$

where α is the learning rate. Only the closest of the vectors ω_i is updated, with the updating direction determined by the correctness of the classification. Effectively, these code-vectors are pulled toward regions where training samples of the same class locate.

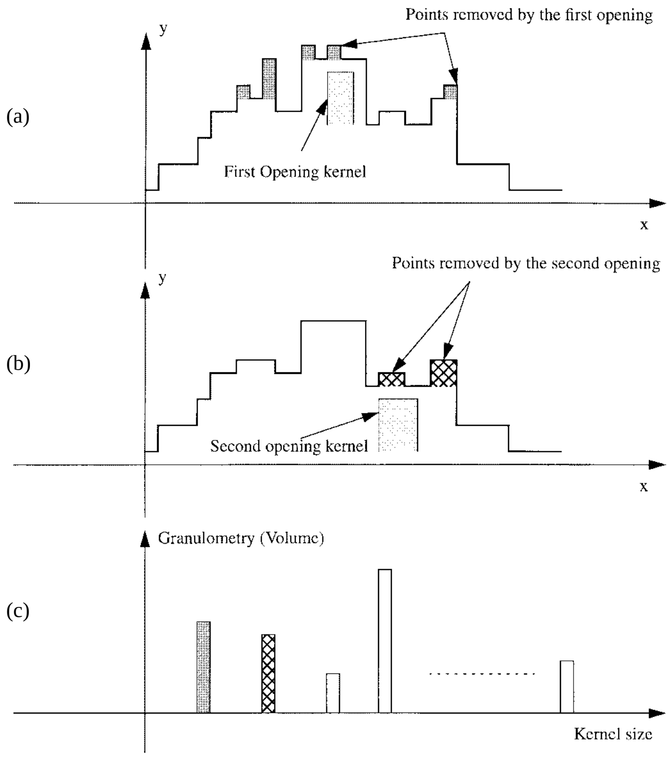


Fig. 3. Illustration of gray-scale granulometry.

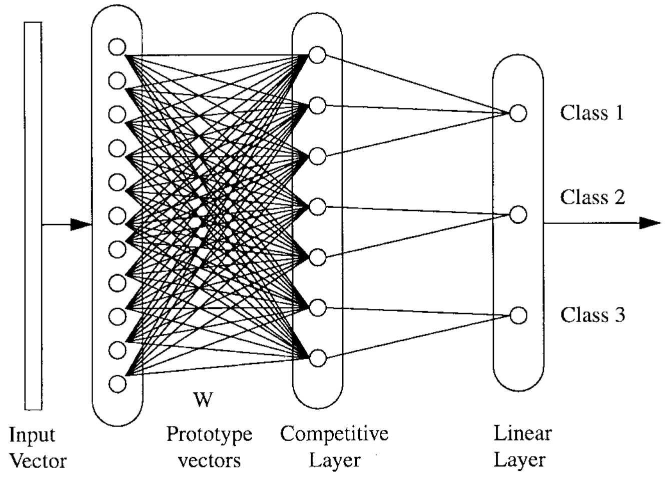


Fig. 4. Learning vector quantization neural network.

III. Multiple LVQ Network Fusion

This section describes the network fusion approaches. First we introduce the concept of the neuron confidence level measure. Then, using this confidence level measure, we develop two network fusion approaches: the serial fusion approach and the parallel fusion approach.

A. Neuron Confidence Level

As the neurons are trained to map the topology of the training feature vector distribution, some neurons are driven toward the center of each class while others are placed at the boundary of the class. When a test sample is nearest to a neuron in a class center, it is more likely to belong to that class; when the associated neuron locates around the classification boundary, misclassification will likely occur. We apparently have different levels of confidence on a correct classification using these two types of neurons. To determine this level of confidence for all the neurons, we define the following measure for each neuron:

$$P_{ij} = \frac{N_{ij}}{N} \tag{5}$$

$$CL = P_{ii} \tag{6}$$

where N_{ij} is the number of class j training samples among the first N nearest neighbors of a class i neuron, P_{ij} is the normalized percentage, and CL is the defined confidence level of the neuron, with a value ranging from 0 to 1. The CL measures how many of the neuron's neighborhood training samples are in the same class as the neuron itself. A neuron in the class center surrounded mostly by the same class training samples will have a high CL value, while a neuron close to mixed classes of training features at the class boundary will have a lower confidence level. When a testing sample is submitted to the network, the classification output has not only the class label that associated with its nearest neuron, but also the CL of that neuron.

One attractive feature of this confidence level definition is that we only need to compute it for the relatively small number of neurons, instead of an infinite number of training and testing samples. Moreover, the computation is carried out during the training stage, therefore no additional computation is introduced for the classification stage where computation complexity affects the real time performance the most.

There are two direct applications using such a CL measure. First, this measure provides a way of trading off between the degree of human operator intervention and the classification accuracy. For an example, when the acceptable confidence level is set high, we have greater confidence on the processed data, but end up with more rejected samples for later re-classification by hand. Another more important benefit of the confidence level measure is that it makes possible the network fusion approaches described next.

B. Serial Network Fusion

With the CL measure, we can judge whether a pattern is classified correctly by a network to the degree of acceptable confidence, then decide whether to pass it through another network trained by a different set of features. Fig. 5 shows such a serial network fusion scheme:

1) Classify the input sample using the first network. Compare the confidence level of the classification with a preset threshold T. If the CL value is above the threshold, the sample is considered correctly classified, and we proceed to classify the next sample. Otherwise, submit the sample to the next classifier.
2) Repeat step 1 for the second network classifier, and continue through all the networks if necessary.
3) If the pattern falls through all the classifiers, lower all the confidence thresholds and go back to step 1. Note that for the second pass, the classification labels and CL's

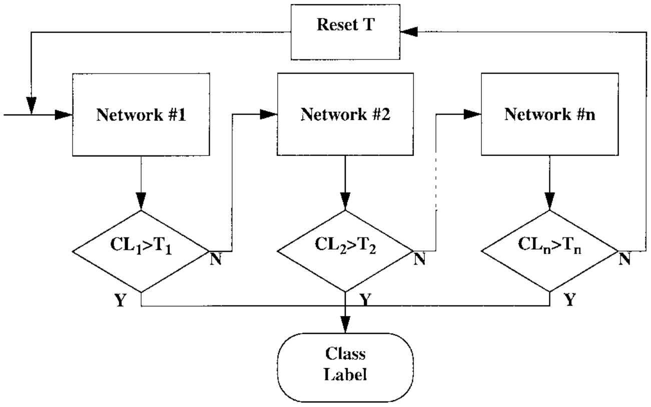


Fig. 5. Serial network fusion method.

are already computed, thus, when passing the sample through the networks, we only need to compare the confidence level with the new thresholds.

4) To end the loop, we may need to either reject the sample as unclassifiable (i.e., it requires human interpretation), or set the threshold T of a network in Fig. 5 to zero, to accept the classified label regardless of confidence level.

One advantage of this approach over the parallel fusion network described next is that most samples will only need to pass through the first few networks instead of needing to be processed by all the networks.

C. Parallel Network Fusion

A more straightforward parallel network fusion method is shown in Fig. 6, where we classify a sample through several networks in parallel. By comparing the confidence levels of the output neurons, we choose the class label with the highest confidence level

$$C = C_p \quad \text{when} \quad L_p = max\{L_k, k = 1 \text{ to } n\} \tag{7}$$

where C is final class label, C_p is the class label of the pth network which has the highest confidence level L_p among all the n networks. When a network consistently produces better results than the others, we can modify its output CL by multiplying a bias weight larger than 1 to favor its chance of winning the confidence level competition. In fact, if we bias the confidence level of all the networks to manipulate the output, then rule (7) becomes

$$C = C_p \quad \text{when} \quad B_pL_p = \max\{B_kL_k, k = 1 \text{ to } n\} \tag{8}$$

where B_k represents the bias weight for the kth network. We do not have an analytical way to choose the bias weights yet. Instead, they are derived empirically from the experiments.

IV. Experiments

We conducted a series of experiments to study the network fusion methods. The two image data sets used in the experiments are described in Part A of this section. Using the network parameters obtained through experiments described in Part B, we study the confidence level measure in Part C. Finally, the performances of the two network fusion methods are compared with that of the individual networks in Part D.

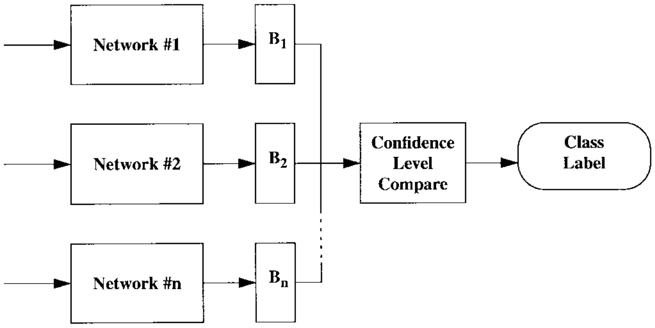


Fig. 6. Parallel network fusion method.

A. Image Data

The two image data sets used for the classification experiments were obtained using two cameras of different magnification settings. The first set of data is composed of five classes of 7135 images captured by camera #1 (10-μm resolution, see examples in Fig. 1). These include 491 Calanus, 2145 Diat-centr, 913 Diat-chaet, 1609 Diatom-rod, and 1977 Diat-chsoc images. The second data set contains four classes of 1124 images taken by a lower resolution camera #2 (55-μm resolution, see examples in Fig. 2). These include 375 Hydroids, 214 Phaeoblimp, 143 Pseudowegg, and 392 Pteropod. Half of the images are used as training data and half as testing data.

The images were hand sorted by human operators. Because of the large number of images, misidentifications do happen [see the first image in Fig. 1(d) and the last three to four images in Fig. 2(d)]. We have not attempted a second pass to correct the misidentifications, which are estimated to be within 1–2% of all the images. The images were also obtained from several different cruises under fairly different physical environments. This further adds difficulty to the classification because of the increased pattern diversification within each class.

B. Experimental Testing of System Parameters

Before studying the network fusion methods, we need to specify the LVQ network configurations for each individual classifier. Since there is no analytical formula to determine the LVQ network structure, such as the number of neurons in the input layer and in the hidden layer, we conduct a series of experiments using different network parameter settings to find an empirical solution. The purpose of these experiments is not to find the optimal parameters, but only to identify a reasonable set of parameters that do not severely bias the outcome of the comparison between the individual network performance and the combined network performance. Four individual networks were studied for the experiments: camera #1 data set granulometric feature trained network, camera #1 data set FD feature trained network, camera #2 data set granulometric feature trained network, and camera #2 data set FD feature trained network.

It is impractical to test all possible combinations of different parameters. We first fix the number of neurons in the input

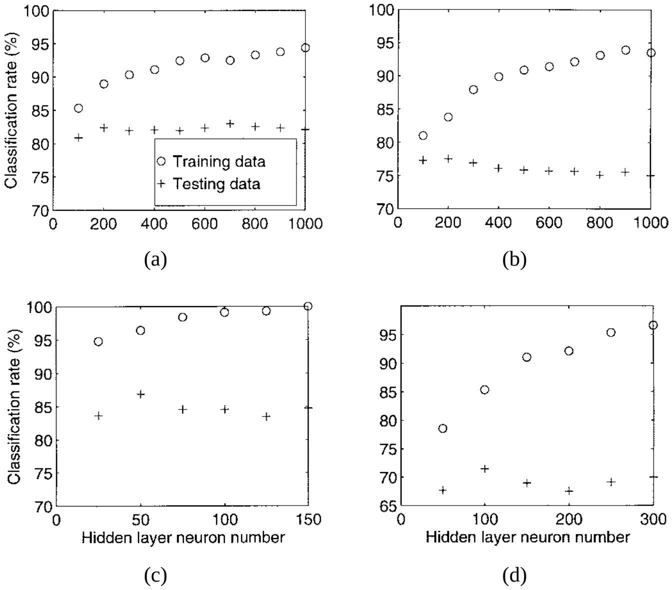


Fig. 7. Plots of classification rates (%) versus competitive layer neuron numbers for the four networks. (a) Granulometry, camera #1; (b) FD, camera #1; (c) granulometry, camera #2; and (d) FD, camera #2.

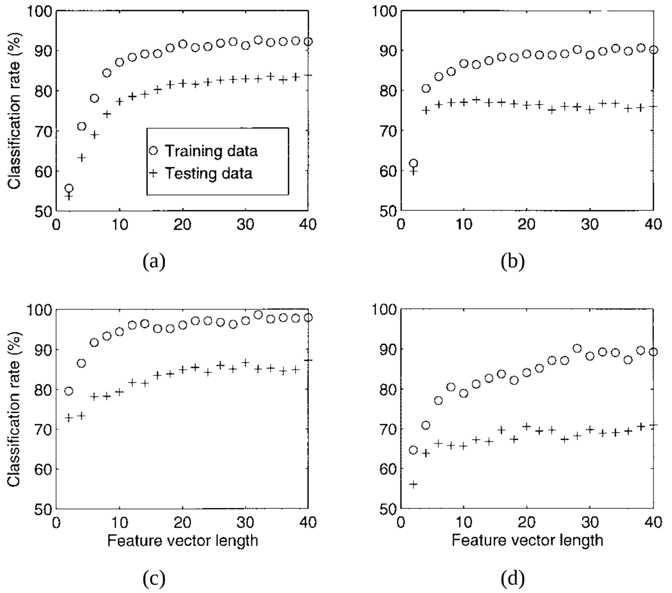


Fig. 8. Plots of classification rates (%) versus feature vector lengths for the four networks. (a) Granulometry, camera #1; (b) FD, camera #1; (c) granulometry, camera #2; and (d) FD, camera #2.

layer, i.e., the feature vector length, to study the effect of varying the hidden layer neuron number. Using KLT, we pick the first 25 principal component features to form the input feature vector. Fig. 7 shows the changing classification rates of the four networks with the number of hidden layer neurons. A general trend is that as the neuron number increases, the training data classification accuracy increases continuously, while the testing data accuracy levels out after an initial increase.

It is not difficult to understand why the training data classification accuracy improves with increasing neuron number, because more neurons give finer mapping of the training vector space. In fact, if the number of neuron increases to the same number as the training samples, the network becomes the nearest neighbor classifier. With each neuron matching one training sample, a perfect classification can be achieved for the training data. But the same does not hold true for the testing data. The diversity of the data shown in Figs. 1 and 2 means that the feature distribution pattern of testing data cannot match the training data perfectly. Finer mapping of the training data does not always mean better mapping of the testing data.

For example, let's look at the scenario when a few training samples of class A are surrounded by a large number of samples from class B. Initially, when there are only a small number of neurons to map the training feature space, these few class A samples may not have enough strength to attract a class A neuron. Therefore, they may simply be assigned to class B neurons that are used to map the surrounding class B samples. However, after we increase the total neuron number, the training feature space is mapped more finely by the higher number of neurons. These few class A samples will have a better chance to attract a class A neuron to represent their small feature space, so they can be correctly classified. However, when this trained class A neuron is applied to the testing data, there maybe no class A testing samples located around this neuron. Thus this added neuron may only cause misclassifications for the surrounding class B testing samples.

Based on the results in Fig. 7, we select 400 and 100 for the hidden layer neuron numbers for camera #1 and #2 data, respectively, to test the effect of changing feature vector length. Fig. 8 shows how the classification rates change against the feature vector length. We can see that the classification accuracies flatten out after around ten to 20 features for the four networks. This is because features selected later have less discrimination power due to the KLT energy compacting property. These later features eventually start to bring in more noise than valuable information.

C. Confidence Level Study

Based on the experiments in B, we select the following network structure: feature vector length of 25, hidden layer neuron number of 400 and 100 for camera #1 and #2 data, respectively. With these parameters we study the properties of the neuron confidence levels in this subsection.

For CL computation, the number of the nearest training sample number N in (5) is set to 40 and ten for camera #1 and #2 networks, respectively. In Fig. 9, we show the distribution of the neuron confidence levels for the four networks. A good network should have a high concentration of neurons with CL near 1. Fig. 9 shows that granulometric feature trained network has more high CL neurons than the FD feature trained network, which translates to better classification accuracy in the later experiments.

Using these networks to classify image samples, we obtain not only the class label but also a confidence level for the classified image. Plots of the classification accuracy of image samples with different confidence levels are shown in Fig. 10. Fig. 10(a) and (b) show the percentage of classified images with CL's higher than a threshold. Toward the end of each of the plots in Fig. 10(a) and (b), as the acceptable CL threshold

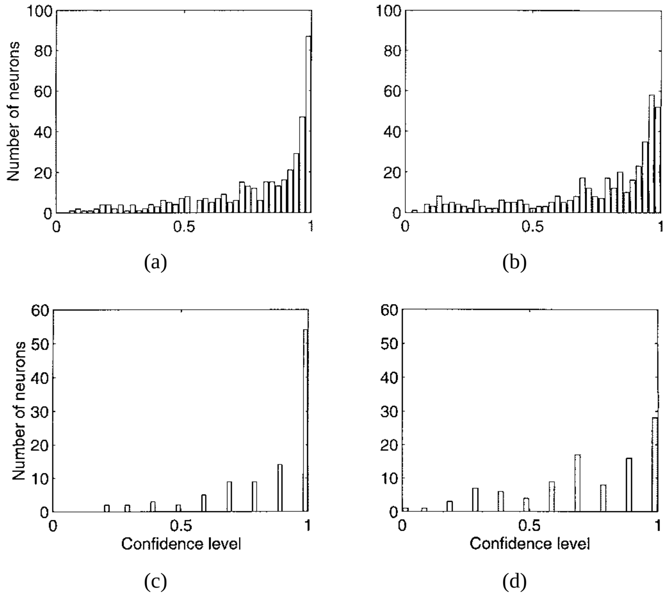


Fig. 9. Plots of the number of neurons for each confidence level for the four networks. (a) Granulometry, camera #1; (b) FD, camera #1; (c) granulometry, camera #2; and (d) FD, camera #2.

is set higher, less data are accepted as properly classified, and more data are rejected. However, the classification accuracy of these classified and accepted data increases, especially for the testing data, as shown in Fig. 10(c) and (d). For example, the last point in Fig. 10(b) for camera #1 shows that only around 20% of the processed images have a confidence level of 1, but the classification accuracy of these data reaches nearly 100%. Therefore, when the experiment requires high classification accuracy, we can set the threshold high, then hand classify the rest of the data. If a lower accuracy is good enough for the statistical plankton distribution mapping, we can set a much lower CL threshold, so that more data can be automatically classified. This CL measure gives the biologists more control over the analysis of the experimental data.

D. Network Fusion Study

Using the above CL measure, we experiment with the network fusion methods and compare them with the individual networks. As shown in Fig. 11, the classification accuracies for camera #1 data using the granulometry network are 91.1 and 82.0% for the training data and the testing data, respectively. Using the FD feature trained network, we have 89.8 and 76.1% accuracies for the training data and the testing data, respectively. Better results are also achieved using granulometric feature trained network than FD feature trained network for camera #2 data. Therefore for the serial network shown in Fig. 5, we use the granulometry network as the first net and the FD network as the second net. For simplicity, we set the CL thresholds for both networks to be equal to the same value T. We then use the granulometry net again with T set to 0 to classify any samples that fall through both nets, i.e., samples with confidence level lower than T in both nets. There could be many other different configurations for the serial network. Here we only use a simple structure to study the feasibility of the serial network fusion approach.

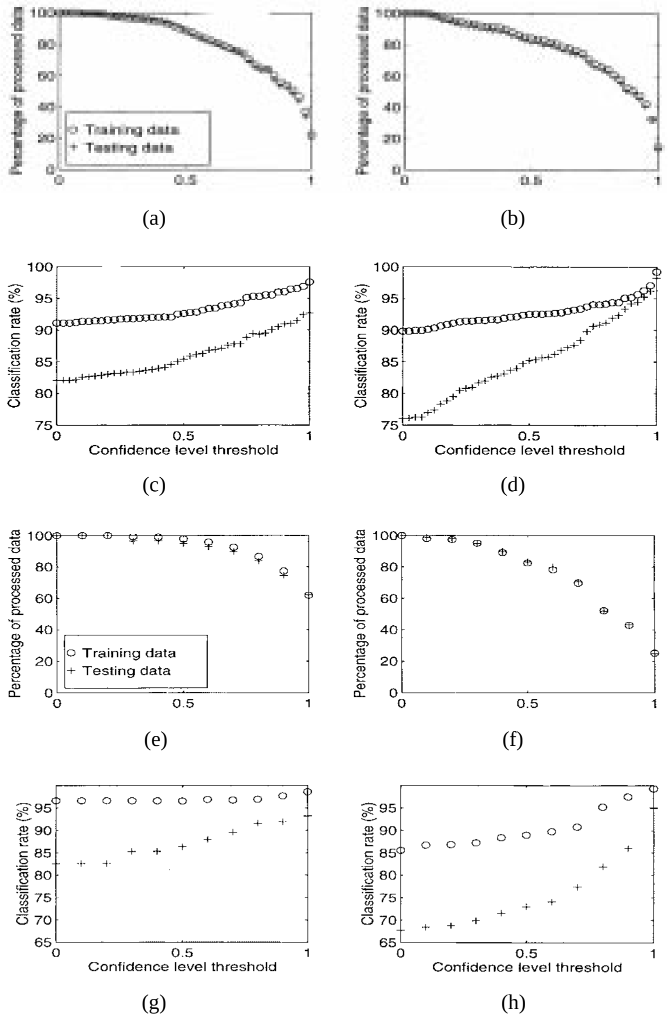
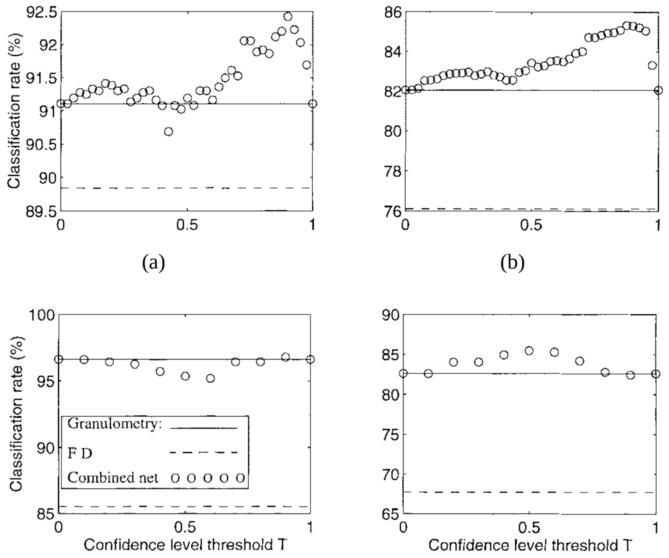


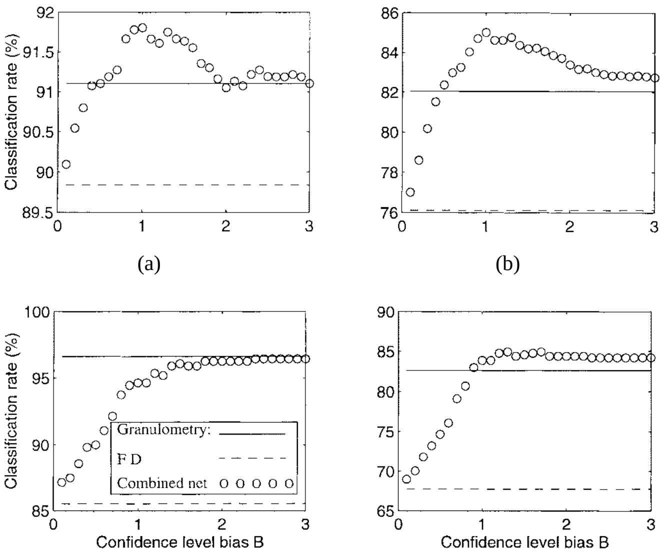
Fig. 10. Classification accuracy for different confidence level thresholds for the two camera data sets. A point in (a), (b), (e), and (f) marks the percentage of classified samples with confidence levels higher than the threshold. A point in (c), (d), (g) and (h) represents the classification rate (%) of samples with confidence levels higher than the threshold. (a) Granulometry, camera #1; (b) FD, camera #1; (c) granulometry, camera 1; (d) FD, camera #1; (e) granulometry, camera #2; (f) FD, camera #2; (g) granulometry, camera 2; and (h) FD, camera #2.

Fig. 11 shows how the classification rates change as we vary the CL threshold T. At the beginning of each of the plots (i.e., when $T = 0$), the serial network is equivalent to the first granulometry network, which simply classifies all input samples. At the end of the plots (i.e., when $T = 1$), only the last granulometry network is effective, because the samples fall through the first granulometry network and the FD network, which cannot produce classified samples with CL higher than 1. The FD network helps the granulometry network between the two end points of the plots. Nearly all the combined network classification results for testing data are better than both individual networks, with the best results reaching 85.3% and 85.5% for camera #1 and #2, respectively. For the training data, the classification rates drop slightly below the results of the granulometry networks in the middle part of the plots for both cameras, probably because the training samples are more closely tied in with their corresponding neurons, thus reject any outside corrections.



(a) (b) (c) (d)

Fig. 11. Comparison of the classification results of the individual networks with the serial fusion network. The "o" markers represent the results of the serial fusion network with different confidence level threshold T. The classification rate of the granulometric feature trained network is marked by a solid line, and the classification rate of the FD feature trained network is marked by a dashed line. (a) Training data, camera #1; (b) testing data, camera #1; (c) training data, camera #2; and (d) testing data, camera #2.



(a) (b) (c) (d)

Fig. 12. Comparison of the classification results of the individual networks with the parallel fusion network. The "o" markers represent the results of the parallel fusion network with different confidence level bias B. The classification rate of the granulometric feature trained network is marked by a solid line, and the classification rate of the FD feature trained network is marked by a dashed line. (a) Training data, camera #1; (b) testing data, camera #1; (c) training data, camera #2; and (d) testing data, camera #2.

For the parallel network fusion, the results are also better than the individual networks. We again use the granulometry network and the FD network to form the parallel network in Fig. 6. Since there are only two networks, we only need to multiply the confidence level of the granulometry network output by a bias B. Fig. 12 shows the effect of this bias on the final classification accuracy of the parallel fusion network. At the beginning of the plots, the results are close to the single FD network, because B is small, meaning we have intentionally ignored results from the granulometry network. At the end of the plots, since the larger bias B favors the granulometry network heavily, the result is close to a single granulometry network. Good results for the testing data are obtained in the region surrounding a bias value of 1, with the best results reaching 85.0% and 84.9% for cameras #1 and #2, respectively.

The results demonstrate that combining different networks does help the pattern classification process. This research is only a first step illustration of how the system works. As we develop more feature vectors of different discrimination characteristics, the network fusion approaches are expected to demonstrate greater discriminatory power.

V. Conclusions

We have developed a working multiple network system to classify plankton images. Given the difficulty of the data, we have achieved good results with two fairly large data sets. The newly proposed confidence level measure also gives marine biologists more control over the plankton classification experiments. With the VPR system, including the data acquisition unit and the pattern classification system, real-time automatic sorting of plankton images into taxonomic categories becomes possible.

To maintain a high degree of classification accuracy over a larger number of plankton species, we need to develop more distinct pattern features. A hierarchical classification system may also be necessary to classify major species and subspecies in several steps. The algorithm developed in this work is an encouraging step toward such a hierarchical network. We also intend to investigate several issues regarding the network fusion methods that are not covered by this study. For example, we will study: 1) how to analytically select the network parameters like the nearest neighbor number N, the CL threshold T, and the bias B; 2) other ways to define the confidence level; 3) how to take advantage of the measures like P_{ij} in (5) when $i \neq j$; and 4) using the confidence level as a training progress measure at the network training stage. We believe these studies will further benefit the LVQ neural network research.

Acknowledgment

The author would like to thank C. S. Davis, S. M. Gallager, W. K. Stewart, and C. J. Ashjian for the original plankton images and many valuable and interesting discussions, L. Vincent for the fast granulometry programs, and T. Snow and H. Huang for their careful reading of the manuscript.

References

[1] C. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.*, vol. C-21, pp. 269–281, Mar. 1972.

[2] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 170–179, Mar. 1977.

[3] T. Pavlidis, “A review of algorithms for shape analysis,” *Comput. Graph. Image Processing*, vol. 7, pp. 243–258, 1978.

[4] R. Chellappa and R. Bagdazian, “Fourier coding of image boundaries,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 102–105, 1984.

[5] A. P. Reeves, R. J. Prokop, S. E. Andrews, and F. P. Kuhl, “Three-dimensional shape analysis using moments and Fourier descriptors,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 937–943, Nov. 1988.

[6] T. Reti and I. Czinege, “Shape characterization of particles via generalized Fourier analysis,” *J. Micros.*, vol. 156, pt. 1, pp. 15–32, Oct. 1989.

[7] H. Kauppinen, T. Seppanen, and M. Pietikainen, “An experimental comparison of autoregressive and Fourier-Based descriptors in 2-D shape classification,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 201–207, Feb. 1995.

[8] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.

[9] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed. Berlin: Springer-Verlag, 1987.

[10] ——, “The self-organizing map,” *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.

[11] C. W. Therrien, *Decision Estimation and Classification, An Introduction to Pattern Recognition and Related Topics*. New York: Wiley, 1989.

[12] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[13] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982, pp. 354–357.

[14] G. Matheron, *Random Sets and Integral Geometry*. New York: Wiley, 1975.

[15] J. Serra, *Image Analysis and Mathematical Morphology*. London, U.K.: Academic, 1982.

[16] P. Maragos, “Pattern spectrum and multiscale shape representation,” *IEEE Trans. Pattern Anal. Machine lntell.*, vol. 11, pp. 701–716, July 1989.

[17] L. Vincent, “Fast opening functions and morphological granulometries,” in *Proc. Image Algebra Morphological Image Processing*, San Diego, CA, July 1994, vol. SPIE–2300, pp. 253–267.

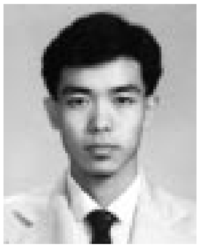
[18] ——, “Fast grayscale granulometry algorithms,” *EURASIP Workshop ISMM’94, Mathematical Morphology Applications Image Processing*, Sept. 1994, pp. 265–272.

[19] X. Tang, “Transform texture classification,” Ph.D. dissertation, Mass. Inst. Technol., MIT/Woods Hole Oceanographic Inst. Joint Program, June 1996.

[20] C. S. Davis, S. M. Gallager, N. S. Berman, L. R. Haury, and J. R. Strickler, “The video plankton recorder (VPR): Design and initial results,” *Arch. Hydrobiol. Beith.*, vol. 36, pp. 67–81, 1992.

[21] C. S. Davis, S. M. Gallager and A. R. Solow, “Microaggregations of oceanic plankton observed by towed video microscopy,” *Science*, vol. 257, pp. 230–232, July 10, 1992.

[22] C. S. Davis, S. M. Gallager, M. Marra, and W. K. Stewart, “Rapid visualization of plankton abundance and taxonomic composition using the video plankton recorder,” *Deep-Sea Res. II*, vol. 43, nos. 7–8, pp. 1947–1970, 1996.



Xiaoou Tang received the B.S. degree in 1990 from the University of Science and Technology of China, Hefei, the M.S. degree in 1991 from the University of Rochester, Rochester, NY, and the Ph.D. degree in 1996 from the MIT/Woods Hole Oceanographic Institution Joint Program, Massachusetts Institute of Technology, Cambridge.

He is currently a Postdoctoral Investigator at the Deep Submergence Laboratory of Woods Hole Oceanographic Institution, Woods Hole, MA. His research interests include image processing, pattern recognition, underwater robotics, and sonar systems.