Automatic In Situ Identification of Plankton

Matthew B. Blaschko, Gary Holness, Marwan A. Mattar, Dimitri Lisin,
Paul E. Utgoff, Allen R. Hanson, Howard Schultz, Edward M. Riseman,
Department of Computer Science, University of Massachusetts Amherst, MA 01003
{*blaschko,gholness,mmattar,dima,utgoff,hanson,schultz,riseman*}*@cs.umass.edu*
Michael E. Sieracki, William M. Balch and Ben Tupper
Bigelow Laboratory for Ocean Sciences, West Boothbay Harbor, ME 04575
{*msieracki,bbalch,btupper*}*@bigelow.org*

Abstract

Earth's oceans are a soup of living micro-organisms known as plankton. As the foundation of the food chain for marine life, plankton are also an integral component of the global carbon cycle which regulates the planet's temperature. In this paper, we present a technique for automatic identification of plankton using a variety of features and classification methods including ensembles. The images were obtained in situ by an instrument known as the Flow Cytometer And Microscope (FlowCAM), that detects particles from a stream of water siphoned directly from the ocean. The images are of necessity of limited resolution, making their identification a rather difficult challenge. We expect that upon completion, our system will become a useful tool for marine biologists to assess the health of the world's oceans.

1. Introduction

The Earth's oceans serve as major sources and sinks of bio-active elements that naturally cycle through the biosphere. The ocean water contains living (plankton) and non-living (detrital) particles. The importance of plankton for the global ecosystem cannot be overestimated. Microscopic algae, phytoplankton, are sometimes called the grasses of the sea. Just like land plants, they consume carbon dioxide and produce oxygen through photosynthesis. Phytoplankton are an integral component of the global carbon cycle, which is responsible for regulating the temperature of the planet [11]. They are also the first link in the food chain for all marine creatures. Their primary consumers are the zooplankton, who in turn become food for larger animals.

Studying plankton is important to ecological research. For example, understanding the carbon cycle is necessary to be able to predict global climate changes. On a less global scale, studying plankton can allow marine biologists to create early warning systems for detecting harmful algal blooms in coastal waters. Applications in other fields could include ship ballast water treatment, drinking water treatment, public health, bio-terrorism defense, and industrial chemical processing.

An instrument for monitoring the abundance of phytoplankton and small zooplankton, the Flow Cytometer And Microscope (FlowCAM) [28], detects and takes images of micro-organisms from a stream of water siphoned directly from the ocean. It has a rudimentary image segmentation capability, used to crop out individual organisms. The instrument is used by marine biologists to estimate the population sizes of different plankton species. In particular, scientists are interested in potentially harmful organisms.

The FlowCAM is capable of generating an enormous amount of image data. Currently, identifying the organisms appearing in all these images is a daunting task, performed manually by the marine biologists. This task is further complicated by limited image resolution. In this paper, we present early classification results of a system for automatic identification of plankton in the FlowCAM images.

2. Previous Work

Research to automate the task of labeling plankton specimens has been going on for at least 20 years [20]. However, many systems have been shown to work only under controlled laboratory conditions with cultured populations. The specimens are carefully prepared and high-resolution images are used for the classification process. For example, the ADIAC project [11] has been successful in developing a system for the classification of diatoms from images taken at a magnification of 100x in laboratory microscopes. They use a wide variety of shape and texture features along with a range of classification techniques. Some of the features consist of morphological characteristics specific to diatoms identified by experts.

Work has also been done attempting to classify field-collected specimens. For example, Culverhouse *et. al* [7] have developed a system known as DiCANN for classification of dinoflagellates, which has been tested on images from underwater video-cameras and by laboratory-based instruments. DiCANN uses several types of texture and edge-based image features as inputs to a neural network.

What sets the work described in this paper apart from others, are the issues arising from the image acquisition process. The FlowCAM is a very efficient imaging instrument, which can produce an enormous number of images very quickly with little or no human intervention. However, the images are of lower quality for two reasons. First, the organisms flow in front of the camera in a stream of water, rather than being on a stationary slide, resulting in blurred and out of focus images. Second, the FlowCAM uses magnification of only 4x to 10x, because higher magnification would greatly reduce the field of view, and many organisms will simply be missed.

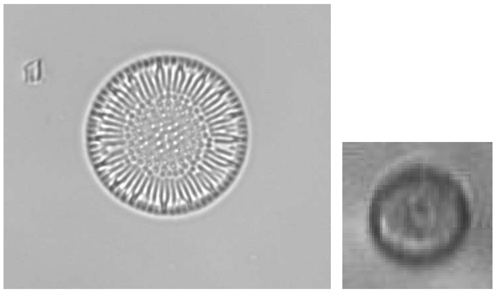

Figure 1. Left: ADIAC image. Right: FlowCAM image

To illustrate the quality of the FlowCAM images, Figure 1 shows one of them next to an image used in the ADIAC Project[11]. We have also compared the quality of FlowCAM images to those used in ADIAC more objectively, by computing the average power spectrum over 100 images from each type. Figure 2 shows the average spectrum for the ADIAC and the FlowCAM images. Clearly we can see that the FlowCAM images contain very little high-frequency information compared to ADIAC, which indicates a general lack of detail.

It is also important to note that our specimens are not restricted to a particular taxa, such as the diatoms or the dinoflagellates. The FlowCAM images will contain anything that comes through the tube, limited only by the tube's diameter. In order to make our system flexible enough to accommodate such diversity, so far we have only used very generic shape and texture features. While we have been able to achieve accuracy rates comparable to the consistency rates of human experts [6], it may be possible to improve performance further, by using expert-selected features specific to particular taxa of organisms.

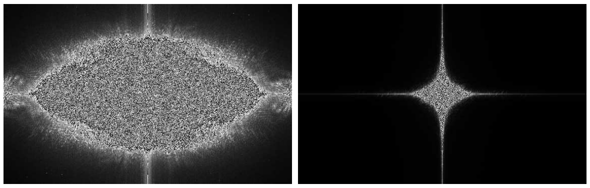

Figure 2. Left: average power spectrum of ADIAC images. Right: average power spectrum of FlowCAM images.

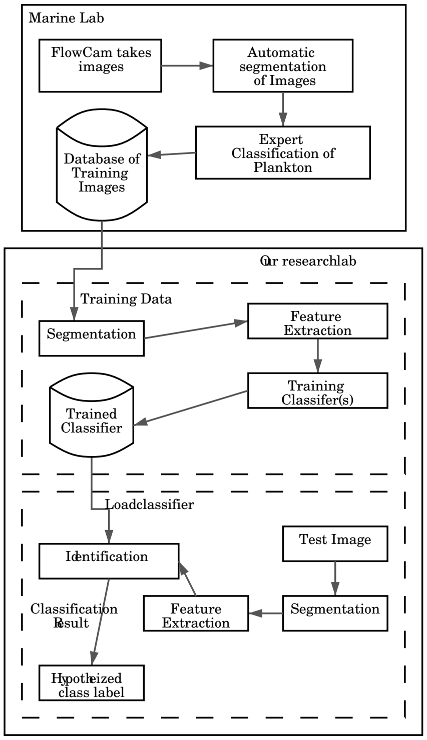

Figure 3. System Layout

3. Approach

Our data set consists of FlowCAM generated images that have been expertly classified by marine scientists. The classifications correspond to broad groupings that reflect the general distinctions useful to marine scientists, and that often (but not always) corresponds to biological and visual similarity. As the marine scientists classify the images, they also provide a confidence level that indicates their degree of

Class Label	number of images
Unknown	39
Centric diatoms	26
Pennate diatoms	124
Dinoflagellates	29
Ciliates	179
Unidentified cell	32
Non-cell	113
Mesodinium	71
Laboea	30
Skeletonema	169
Thallasiosira	86
Thallasionema cf.	23
Pseudo-nitzschia	61

Table 1. Data Set

certainty that the given label is correct, as well as an image quality level. We report results here for experiments on a 13 class problem outlined in Table 1. This data set accounts for every class with at least 20 images with high confidence expert classification.

Figure 3 outlines the layout of our system. After receiving plankton images from the Marine Lab, we perform segmentation techniques outlined in Section 4. Then we extract the features (Section 5), from these segmented images and train classifiers (Section 6). At this point, the trained classifier is saved locally and ready for test images. A test image follows the same processing steps, its features are sent to the trained classifier, and a class hypothesis is returned.

4. Segmentation

Active contours or deformable models, also known as snakes [21], are energy minimizing splines that can move under the influence of a potential field computed over the intensity surface of an image. The potential field exerts a force such that the snake comes to rest at the boundaries of desired features. Snakes are a useful segmentation methodology, because they yield smooth closed contours. Two major issues concerning snakes are limited capture range and inability to converge into concavities. Gradient Vector Flow(GVF) fields [34] are dense fields computed by minimizing an energy function over the image edge map. These two problems are addressed by using GVFs as an external energy function for snakes.

We have also found that a simple global bimodal segmentation is effective in many cases for separating the plankton from the background, which tends to be significantly brighter than the object. We use expectation maximization (EM) to fit a mixture of two Gaussians to the histogram of grey values for a given image [8]. The Bayesian decision boundary defines the cut point between foreground and background. After that, morphological hole filling [29] is used to capture the stray bright pixels inside the object.

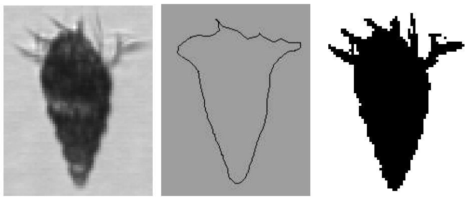

Figure 4. A sample image, its snake-based segmentation, and its intensity-based segmentation

Both segmentation methods are used in this paper as a preprocessing step for extracting different features. The intensity-based segmentation is better at capturing fine details, while the snake-based segmentation is guaranteed to produce a smooth closed contour. Figure 4 shows an example of both methods.

5. Features

Features were grouped into five types: simple shape, moments, contour representations, differential and texture features. This grouping reflects different ways, in which the image is represented. The various feature types are explained briefly below and summarized in Table 2.

5.1. Simple Shape Descriptors

Binary images, obtained from the intensity based segmentation, readily provide several simple geometric properties such as perimeter and area that are used as building blocks for other meaningful shape descriptors. We implemented six such descriptors: perimeter P, area A, ratio of the major to minor axes of rotation, compactness, rectangularity and eccentricity. We also implemented three different convexity measures defined below.

Let S represent a set of contour points obtained from the GVF snakes, $CH(S)$ is defined as its convex hull, $Per_1(S,\theta)$ is the perimeter (in the sense of the l_1 metric) when S is rotated by θ deg and $Per_2(R(S,\theta))$ is the Euclidean perimeter of the bounding box when S is rotated by θ deg. The first one $C_1(S) = \frac{Area(S)}{Area(CH(S))}$ is a standard one [30] while the other two were introduced recently [35]: $C_2(S) = \frac{Per_2(CH(S))}{Per_2(S)}$,

$$C_3(S) = \min_{\theta \in [0,2\pi]} \frac{Per_2(R(S,\theta))}{Per_1(S,\theta)}$$

5.2. Moment-based Features

Another way of describing shape is based on moments. They are extensively used for shape representation, pattern recognition and image reconstruction, which makes them a very useful feature set to include. We have experimented with moment invariants and complex Zernike and pseudo-Zernike moments [17].

Geometric moments for a binary image of size N x M are defined as, $m_{pq} = \sum_{x=0}^{n} \sum_{y=0}^{m} x^p y^q f_{xy}, \quad p,q = 0,1,2,3,...$ where f_{xy} is a binary function equal to 1 if the pixel (x,y) is inside the contour and 0 if it is outside. Translation invariance is achieved by using central moments. Scale invariance is obtained by normalizing the central moments, as proposed by Hu [17]. Rotation invariance is obtained by combining the normalized central moments using the theory of algebraic invariance. We used the first seven moment invariants introduced by Hu [17].

The Zernike and pseudo-Zernike moments are a set of complex, orthogonal polynomials defined over the interior of the unit circle $x^2 + y^2 = 1$. They have been shown empirically to be the best moments in terms of information redundancy, noise sensitivity and feature representation capabilities [31, 4] because of their ability to capture the fine detail in an image in their higher order moments. However, given our low-resolution images and lack of fine detail we only used moments up to order five with all positive repetitions, which yields a total of thirty-two features for both Zernike and pseudo-Zernike moments.

Zernike and pseudo-Zernike moments are only rotationally invariant. Thus, translation invariance is achieved by mapping the centroid of the shape to the center of the unit circle before calculating the moments. Scale invariance is achieved by dividing the moments obtained from the above calculation by the area of the shape, i.e. m_{00}.

5.3. Contour Representations

There are various ways to describe the contours of objects. One example is the *r*-θ boundary descriptors computed starting from a consistently chosen anchor point. An *r*-θ feature set is the distribution of magnitude and orientation for vectors tracing the contour. An *r*-θ centroid distance is the distribution of vector orientations and distances from object centroid to each point on the discretized contour. Another representation, an *r*-θ Hough descriptor, can be computed by flattening the r-table output by a generalized Hough transform [1]. In-plane rotation invariance is achieved by measuring orientation as a relative angle. All three types of the *r*-θ representations are computed from the boundary resulting from applying the snake-based segmentation.

Alternatively, we can encode the sequence of (x,y) points around the contour as a sequence of complex numbers, $x + i * y$, and compute the discrete Fourier transform [19]. We convert the imaginary descriptors to polar co-ordinates, (r,θ) and treat r and θ as individual features. Rotational invariance is achieved by first subtracting the phase of the first descriptor from all other descriptors. Similarly, scale invariance is achieved by dividing each descriptor by the magnitude of the first descriptor. Subsequent (r,θ) pairs are used.

5.4. Texture Features

The simplest description of texture can be derived by simply computing the mean and variance of the grey values in an object. Commonly used texture measures, that are more sophisticated, have been derived from co-occurrence matrices. The entry (i, j) of the co-occurrence matrix $P_{d_x d_y}(i, j)$ is the number of occurrences of the pair of grey levels i and j at horizontal step d_x and vertical step d_y pixels. In our work, we calculate the energy, inertia, entropy, and homogeneity [15] of the co-occurrence matrix as texture features. We computed the matrix for the image region containing the cell body.

Local binary patterns are grey-scale and rotation invariant texture operators [24]. Local binary patterns are computed from P $(P > 1)$ image pixels inside the cell body, where each sampled image point p is calculated at a radius R $(R > 0)$ according to the formula $(-R\sin(2\pi p/P), R\cos(2\pi p/P))$. Each pixel in the image then lies at the center of a circle of grey value samples. Grey-scale and rotation invariant classifications are calculated at every point and accumulated into a histogram of counts for each local binary pattern. The features are taken from the accumulated histogram, normalized by the object size.

5.5. Differential Features

Global differential image descriptors have been shown to be effective for object recognition [26]. A function of the image called the shape index is computed: $S(p,\sigma) = \arctan[\frac{\kappa+\mu}{\kappa-\mu}](p,\sigma)$, where κ is the isophote curvature of the intensity surface, and μ is the flowline curvature. The curvatures are computed via combinations of image derivatives, which are computed using the Gaussian derivative filters.

We calculate the shape index at every pixel in the image at a range of scales, and aggregate the values into a histogram by quantizing the shape index. Before we do this, however, we ignore areas of low curvature by excluding points where the isophote (flowline) is below the mean isophote (flowline). Currently histograms are calculated with $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}\}$.

6. Classification

The feature vector computed from each image in the data set can be represented as a point a multi-dimensional space. It is the goal of classification to estimate a function that partitions the space such that points in the same region are labeled as belonging to the same class. In supervised learning, a model for this function is learned from a pre-labeled set of instances. Two supervised learning approaches used in our experiments are single classifiers and ensembles.

6.1. Single Classifiers

In supervised learning, we are given a set of inputs $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i =< x_{i,1}, \ldots, x_{i,d} >$ are the instances, $x_{i,j}$ are features, and $y_i \in \{1, \ldots, k\}$ are class labels drawn from a finite set. A classifier, C, implements a model for a hypothesis, $y = f(x_i)$ mapping instances to class labels, estimated from data. Given a combination of feature set, induction algorithm, and parameter settings, a model is built from training instances.

Induction algorithms employed in our experiments include decision trees (DTs), naive Bayes (NB), ridge regression (RRs), k-nearest neighbor (KNN), and support vector machines(SVMs) [12]. These algorithms produce classifiers with different desirable properties. DTs are well suited to problems for which training data may contain errors. Ridge regression adds a penalty term (ridge estimator) to a linear regression model to control overfitting. In KNN, the K parameter acts as a smoothing term for noisy data. SVMs tend to perform well in problems of high dimension. By solving a maximum margin optimization problem, SVMs find the set of instances (the support vectors) suitable for predicting class membership by weighted combination [5]. Because it makes an independence assumption between features, Naive Bayes serves as a good baseline. The feature set description appears in detail in the results (Section 7).

6.2. Ensemble Classifier

An ensemble is a set or committee of classifiers whose individual hypotheses are combined into a single class label hypothesis. Ensembles have been shown to have better accuracy [9] than a single classifier, if the component classifiers in the ensemble are accurate and diverse [9]. Accuracy means that the error rate for any C_i in the ensemble is better than random. Diversity means that for any two C_i, C_j in the ensemble, errors are committed independently.

Approaches to achieving independent errors can focus on manipulating feature subsets, the training instances, induction algorithms, and class labels [10]. Instability means that small changes in the training instances elicit changes in a classifier's hypothesis. With the additional assumption that the C_i are unstable, the Bagging or ”Bootstrap Aggregating” [3] method achieves independence by uniform sampling from the training data to produce a bootstrap replicate. A bootstrap replicate is used to train each of the C_i in the ensemble. A weighting scheme is used to combine hypotheses of the C_i.

By associating weights with each training instance, a distribution, D_t is defined. The boosting method [13] achieves independence by using this distribution to focus on particular training instances for sequentially trained classifiers $C_1, \ldots, C_N$. Weights associated with the instances are modified as a function of a trained classifier's error. Difficult instances on which a classifier commits errors have their weights increased or “boosted.” This results in a new distribution D_{t+1} in which more probability mass is associated with the difficult instances. Using D_{t+1}, this process continues on the next classifier until all of the classifiers in the ensemble are trained. The hypotheses are combined by weighted average based on each classifier's error rate. Boosting and bagging were used as ensemble methods.

Category	Feature Set
Simple Shape	Simple Shape (SS) [9]
Moments	Moment Invariants (MI) [7] Zernike / P-Zernike (Zke / P-Zke) [32]
Contour	R-θ Hough [128] Fourier [10]
Differential	Shape Index (SI) [120]
Texture	Size, Mean, Variance (SMV) [3] Co-occurrence (CO) [140] Local Binary Patterns (LBP) [54]

Table 2. Feature Sets. The number in the square brackets reflects the number of features used from each feature set.

7. Experiments and Results

The goal of our experiments was to explore classification performance given various feature selections, classifier induction algorithms, and instance manipulations for single classifiers and ensembles. Feature selection was performed by choosing individual feature types as input to single classifier induction algorithms. Additional feature selection was performed by combining feature types across categories input into a single classifier induction algorithm. Training instances were varied using boosting (AdaBoost.M1) and bagging ensembles of classifiers using the same induction algorithm and feature set.

Next, the induction algorithms in an ensemble were varied in our bagging multi-classifier ensemble implementation. A multi-classifier [27] is an ensemble where the en-

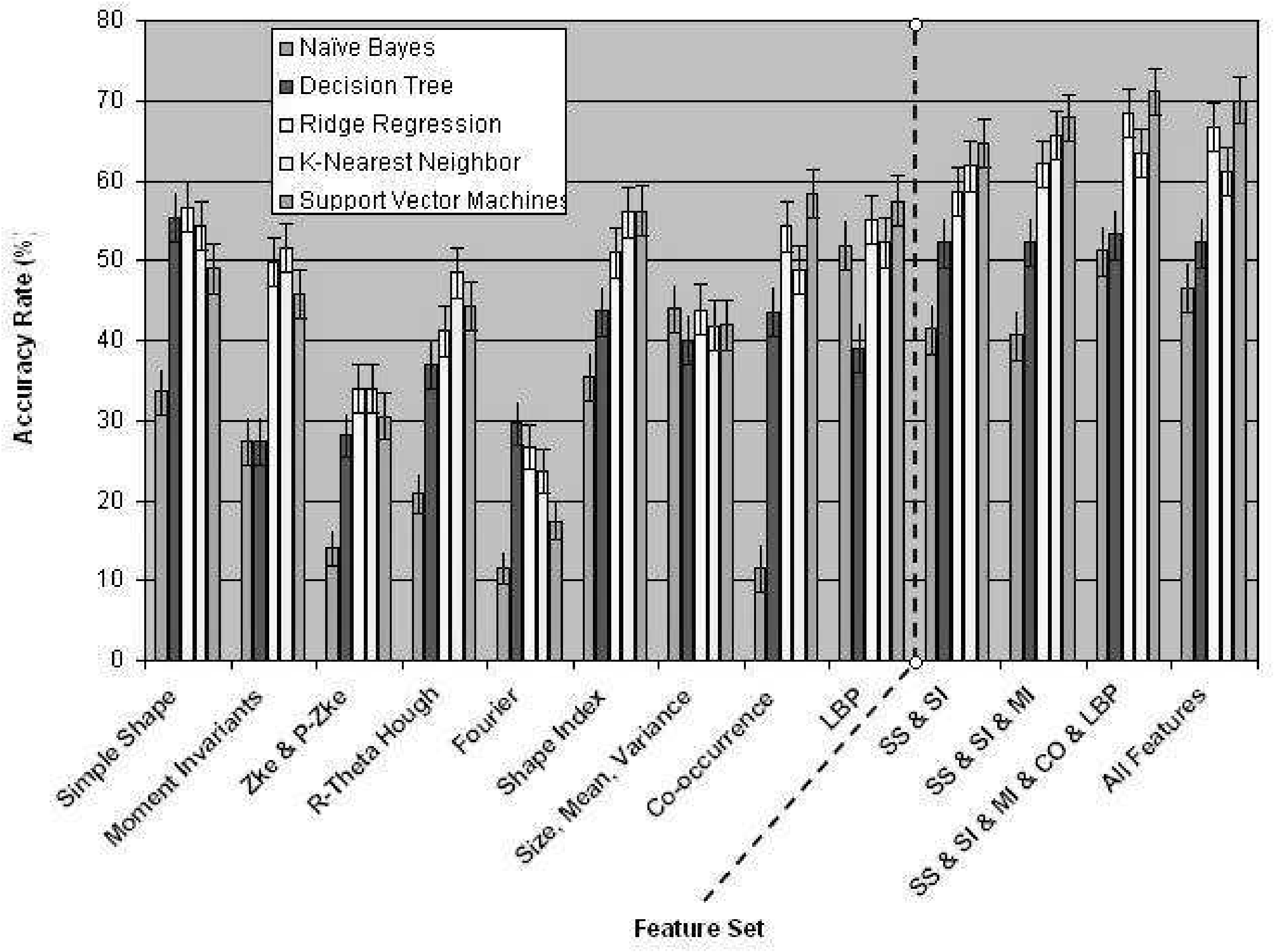

Figure 5. Classification results for various feature selections

Ensemble method	Induction Algorithms	feature set	Accuracy
Boosting	RR $\lambda = 0.5$	mixed-SI,MI,CO,LBP,SS	67.31 ± 2.9338
Boosting	SVM (poly-1)	mixed-SI,MI,CO,LBP,SS	68.02 ± 2.9170
Bagging	SVM (poly-1)	mixed-SI,MI,CO,LBP,SS	72.61 ± 2.7893
Bagging MC	SVM (poly-3)	single-CO,MI,$r-\theta$ Hough,SS	52.93 ± 7.9107

Table 3. Ensemble Variations

semble members use different induction algorithms. In our implementation the each ensemble member can also use a different feature set. Finally, the induction algorithm, feature set, and training instances for each ensemble member were varied in our bagging multi-classifier (BMC).

These experiments were performed using the Weka toolkit [33]. The data set for which results are reported consists of 982 FlowCam images from 13 classes. Results were validated using 10-fold cross validation (10-CV) on the randomized data set. The same folds were used in every case. Results for single classifiers are reported in Figure 5. Along the horizontal axis are the feature sets. The dotted line separates the single feature types on the left from the mixed feature types on the right. The five color bars represent 10-CV percent accuracy for the Naive Bayes, Decision Trees (C4.5), Ridge Regression, K-Nearest Neighbors, and SVM classifier induction algorithms.

In general, a classifier induced from a mixed feature set outperformed its counterpart induced from a single feature set. Referring to Figure 5, SVMs using mixed data set SS/SI/MI/CO/LBP with an average accuracy of $71.08 \pm 2.8350\%$ outperformed SVMs using SS ($48.99 \pm 3.1266\%$), SI ($56.16 \pm 3.1014\%$), MI ($45.82 \pm 3.1163\%$), CO ($58.35 \pm 3.0833\%$), and LBP ($57.43 \pm 3.0925\%$). Each feature type contributes different information about an image. Thus, classifiers induced from mixed types have improved performance. Moreover, since the information is different, classification error for each instance is more likely to differ.

This leads to the question of whether a multi-classifier ensemble outperforms a single classifier and traditional ensemble. Moreover, does performance change if we hold the feature set fixed across members of the multi-classifier ensemble? How does performance change if we vary the feature sets across the multi-classifier ensemble ? If mixed feature types give rise to different error distributions, what does this say about error independence between members of a multi-classifier?

A number of experiments were performed to address these questions. In the first experiment, the induction algorithm was held fixed across members of a multi-classifier ensemble and their feature sets were varied across feature types. In the second experiment, both the induction algorithm and feature sets were varied. In the third experiment, the induction algorithm was varied and each ensemble member input mixed feature types. These results are reported in Table 3.

For all multi-classifier ensemble experiments we measured independence by pairwise correlation and χ^2 test for independence. In some cases, the ensemble methods were observed to perform better than the best single SVM classifier with mixed feature set. In the case of a bagging SVMs with single mixed feature set SI/MI/CO/LBP/SS, we saw a modest improvement in performance. This improvement was not statistically significant. One reason for this is explained by correlation and χ^2 statistics for the MC bagging classifier with single CO, MI, $r - \theta$ Hough, and SS features. Correlated error among the ensemble members was observed. Moreover, the χ^2 statistics told us that Independence in classification error among pairs of ensemble members was highly unlikely.

8. Conclusions and Future Work

Feature types described in this experiment can be termed "global" as they describe the entire image, yielding compact representations. Such representations, however are sensitive to clutter and occlusion. In many cases an image produced by the FlowCAM contains a single object, but sometimes portions of nearby organisms are present and clutter the image. Also, global features may not be the best way to represent images containing chains of cells.

A different paradigm in object recognition is to use local features, which are descriptors of local image neighborhoods computed at *interest points*. A variety of interest points detectors have been proposed including Harris corners [16], local maxima of the Difference-of-Gaussians function [23], outlier-based salient points, etc.

Generally, interest points are computed at multiple scales and are expected to be repeatable, i. e. detectable in different views of the same object, and to somehow capture the essence of the object's appearance. A descriptor is computed at each interest point to represent the image patch around it. The descriptors, are vectors of values that can be matched using some distance metric. Examples include Gaussian derivative responses [25], differential invariants [32], and histograms of gradient orientations in the neighborhood of an interest point [23]. The Hausdorff distance [18] between two point feature sets can be computed not between the feature locations in the images, but between the points in the feature space. Barla, *et. al* [2] propose a Hausdorff-Based Mercer's kernel to be used in a SVM-based approach to detect unusual activity in videos.

The ability to generate hundreds of feature values per image necessitates methods for reducing the dimensionality of the feature space. In our experiments, we selected from among 780 features. Statistical Learning Theory tells that sample complexity increases logarithmically with the size of the hypothesis space. The more dimensions, the larger the hypothesis space and, thus, the more samples are needed. By selecting a subset of the features for training a classifier, the number of instances needed can be greatly reduced. Moreover, if two features are strongly corrolated, eliminating one of them causes no loss of information.

An overview of feature selection is given in [14]. Major classes of supervised feature selection include wrapper techniques, forward and backword selection. An information theoretic algorithm based on the Markov Blanket was proposed by [22]. Feature selection in current work was limited mainly to hand picking variables based on experimentation and our intuition. Future research will include more principled methods for performing feature selection.

In our experiments, mixed feature types were shown to outperform single feature types. In most of our experiments SVMs performed relatively well with respect to other classification induction algorithms. SVMs perform well in feature spaces of high dimension. Since our feature space had 780 dimensions, this explains SVMs' performance. We also found that none of our attempted ensemble techniques improved classification accuracy with statistical significance. Upon computing metrics for independene, we found that ensemble members did not commit independent errors. Future research includes inventing methods for inducing error independence in ensembles. This will require work in the feature domain as well as methods for combining base level classifiers into an ensemble. At this early stage in our research, we have achieved accuracy rates consistent with human performance [6]. With error independence we are certain, ensembles will lead to a significant increase in performance in this important problem domain.

Acknowledgements

We thank Bigelow Laboratory for Ocean Sciences for their contribution to this research. This research was supported by the National Science Foundation under grant ATM-0325167. The fourth author is supported by a fellowship from Eastman Kodak Company Research Labs.

References

[1] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, 1982.

[2] A. Barla, E. Franceschi, F. Odone, and A. Verri. Image kernels. In *Pattern Recognition with Support Vector Machines : First Int. Workshop*, 2002.

[3] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[4] C. Chong, P. Raveendran, and R. Mukundan. A comparative analysis of algorithms for fast computation of zernike moments. *Pattern Recognition*, 36:731–742, 2003.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-base learning methods*. Cambridge University Press, 2000.

[6] P. Culverhouse, R. Williams, B. Reguera, V. Herry, and S. Gonzalez-Gil. Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. In *Marine Ecology Progress Series*, volume 247, pages 17–25, 2003.

[7] P. Culverhouse, R. Williams, B. Reguera, V. Herry, and S. Gonzalez-Gil. Expert and Machine Discrimination of Marine Flora: a comparison of recognition accuaracy of field-collected phytoplankton. In *IEEE International Conference on Visual Information Engineering*, 2003.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[9] T. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[10] T. Dietterich. *The Handbook of Brain Theory and Neural Networks, 2nd Ed*, chapter Ensemble Learning, pages 405–408. 2002.

[11] H. du Buf and M. M. Bayer. *Automatic Diatom Identification*. World Scientific Publishing Company, 2002.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2nd edition edition, 2000.

[13] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Int. Conf. Machine Learning*, pages 148–156, 1996.

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(4):786–804, 1979.

[16] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conf.*, 1988.

[17] M. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8:179–187, 1962.

[18] D. Huttenlocher, D. Klanderman, and A. Rucklige. Comparing images using the Hausdorff distance. *IEEE PAMI*, 15(9):850–863, 1993.

[19] B. Jähne. *Digital Image Processing*. Springer-Verlag, 5th edition edition, 2002.

[20] H. Jeffries, M. Berman, A. Poularikas, C. Katsinis, I. Melas, K. Sherman, and L. Bivins. Automated sizing, counting and identification of zooplankton by pattern recognition. *Marine Biology*, 78:29–334, 1984.

[21] M. Kass, A. Witkin, and D. Terzopoulos. Active contour models. *International Journal Computer Vision*, 1(4), 1988.

[22] D. Koller and M. Sahami. Toward optimal feature selection. In *13th Int. Conf. on Machine Learning*, 1996.

[23] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Int. Conf. on Computer Vision*, 1999.

[24] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 24(7):971–987, 2002.

[25] J. Piater. *Visual Feature Learning*. PhD thesis, Dept. of Comp. Sci., U. of Mass. Amherst, 2001.

[26] S. Ravela. *On Multi-Scale Differential Features and their Representations for Image Retrieval and Recognition*. PhD thesis, University of Massachusetts Amherst, 2004.

[27] F. Roli, G. Giacinto, and G. Vernazza. Methods for designing multiple classifier systems. In *Second International Workshop on Multiple Classifier Systems*, 2001.

[28] C. Sieracki, M. Sieracki, and C. Yentsch. An imaging-in-flow system for automated analysis of marine microplankton. *Mar. Ecol. Progr. Ser.*, 168:285–296, 1998.

[29] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999.

[30] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and, Machine Vision*. Chapman and Hall, 1993.

[31] C. Teh and R. Chin. On image analysis by the method of moments. *IEEE PAMI*, 10(4):496–513, 1988.

[32] K. Walker, T. Cootes, and C. Taylor. Locating salient facial features using image invariants. In *International Workshop on Automatic Face and Gesture Recognition*, 1998.

[33] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.

[34] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. on Image Processing*, (3):359–369, 1998.

[35] J. Zunic and P. Rosin. A new convexity measure for polygons. *IEEE PAMI*, 26(7):923–934, 2004.