

Learning to Recognize Plankton*

Tong Luo, Kurt Kramer
 Dmitry Goldgof, Lawrence O. Hall
 Dept. of Computer Science & Engineering
 University of South Florida
 Tampa, FL 33620
 {tluo2, kkramer, goldgof, hall}@csee.usf.edu

Scott Samson, Andrew Remsen
 Thomas Hopkins
 Dept. of Marine Science
 University of South Florida
 St. Petersburg, FL 33701
 samson@seas.marine.usf.edu

Abstract – We present a system to recognize underwater plankton images from the Shadow Image Particle Profiling Evaluation Recorder. As some images do not have clear contours, we develop several features that do not heavily depend on the contour information. A soft margin support vector machine (SVM) was used as the classifier. We developed a new way to assign probability after multi-class SVM classification. Our approach achieved approximately 90% accuracy on a collection of images with minimal noise. On another image set containing manually unidentifiable particles, it also provides promising results. Also, our approach is more accurate on the two data sets than a $C4.5$ decision tree and a cascade correlation neural network at the 95% confidence level.

Keywords: plankton recognition; support vector machine; learning; feature selection; probabilistic output.

1 Introduction

Recently, the Shadow Image Particle Profiling Evaluation Recorder (SIPPER) was developed to continuously sample plankton and suspended particles in the ocean [17]. The SIPPER uses high-speed digital line-scan cameras to record images of plankton and other particles, thus avoiding the extensive post-processing necessary with analog video particle images. The large sampling aperture of the sensor combined with its high imaging resolution (50 μm per pixel), means that it is capable of collecting tens of thousands of plankton images an hour. This soon would overwhelm a scientist attempting to manually classify the images into recognizable plankton groups. Therefore, an automated plankton recognition system is necessary to solve the problem or at the very least to help with the classification.

Tang [19] developed a plankton recognition system to classify plankton images from video cameras. The moment invariants and Fourier descriptor features from contour images were extracted. Also, granulometric features from the gray-level images were computed. Finally, a learning vector quantization neural network was

used to classify examples. Tang [19] achieved 92% classification accuracy on a medium-size data set.

The project ADIAC (Automatic Diatom Identification and Classification) has been ongoing in Europe since 1998. Different feature sets and classifiers have been experimented with to recognize separate species of diatom taken from photo-microscopes. Loke [12] and Ciobanu [3] studied some new contour features. Santos [18] extended the contour features to multi-scale Gabor features together with texture features. Wilkinson [22] applied morphological operators to help extract both contour and texture information. Fischer [9] summarized these features and used ensembles of decision trees to classify the combined feature set. Greater than 90% overall accuracy was achieved on the diatom images.

However, images from previous work are of relatively good quality or at least with clear contours. Therefore, complicated contour features and texture information can be extracted easily. The SIPPER images, on the other hand, present several difficulties:

1. Many SIPPER images do not have clear contours. Some are partially occluded. Therefore, we cannot depend mainly on contour information to recognize the plankton.
2. The SIPPER image gallery includes many unidentifiable particles as well as different types of plankton.
3. The SIPPER images in our experiments are binary, thus lacking enough texture information.

Not depending heavily on contour information, several special features are developed in our system, and a support vector machine (SVM) [20] is used to classify the feature vectors. To reduce the computation time and improve the classification accuracy, we applied the wrapper approach [11] with backward elimination to select a subset of the features. We also developed a new way to compute probabilistic outputs from a multi-class support vector machine.

This paper is organized as follows. Section 2 introduces the binary SIPPER images used in our experiments. In Section 3, we discuss the preprocessing of

the images and the extraction of the features. Section 4 describes the support vector machine and the way we assign the probability in a multi-class support vector machine. We applied wrappers [11] with backward elimination to select the best feature subset in Section 5 and experimental results for our system are detailed in Section 6. Finally we summarize our work and propose some ideas for future work in Section 7.

2 Image gallery

Domain experts build the training data as follows. The SIPPER images are first converted to binary images by choosing a proper threshold. A morphological closing operation is used to analyze the connectivity and then segment the binary images. Next, an expert manually classifies the images into recognizable plankton groups. The rest of the unrecognizable images are put into the unidentifiable-particle group. Figure 1 contains typical examples from SIPPER images.

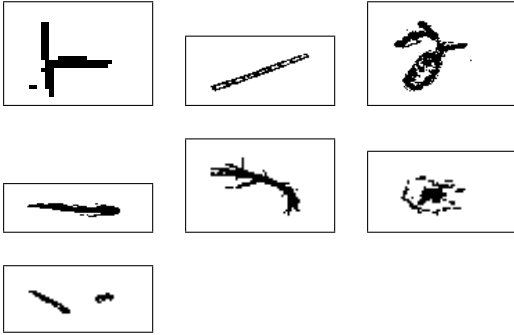


Figure 1: Figures from left to right, from top to bottom are typical examples of copepod, diatom, doliolid, larvacean, trichodesmium, protocista and an manually unidentifiable particle.

3 Feature computation

The SIPPER images have a lot of noise around or on their bodies and some do not even have a clear contour, thus making contour features (Fourier descriptor [23] etc.) unstable and inaccurate. To solve this problem, we first preprocess the images to suppress the noise. We only use invariant moments and granulometric features, which are relatively stable with respect to noise and do not depend heavily on the contour image. To capture the specific information from our SIPPER image set, domain knowledge is used to extract some specific features such as size, convex ratio, transparency ratio, etc.

3.1 Noise suppression

We applied connected component analysis to eliminate the noise pixels far from the plankton bodies. In addition, a morphological closing operation is used to

separate the holes inside the plankton body from the background [15].

3.2 Moment invariants

Moment features are widely used as general features in shape recognition. The standard central moments are computed as follows:

(\bar{x}, \bar{y}) is the center of the foreground pixels in the image. The $(p + q)$ -order central moments are computed with every foreground pixel at (x, y) :

$$\mu(p, q) = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (1)$$

Then central moments are normalized by size:

$$\eta(p, q) = \frac{\mu(p, q)}{\mu(0, 0)^{\frac{p+q}{2}}} \quad (2)$$

Hu [10] introduced a way to compute the seven lower order moment invariants based on several nonlinear combinations of the central moments. Using the normalized central moments, we get the scale, rotation and translation invariant features. We compute the same 7 moment invariants on the whole object and the contour image after a morphological closing operation, respectively.

3.3 Granulometric features

Since the Hu moments only contain low order information from the image, we extracted several granulometric features [13] to capture the high order information. Granulometric features are computed by doing a series of morphological openings with different sizes of structure elements. Then we record the differences in size between the plankton with and without openings. Granulometric features are relatively robust to noise and have the inherent information of shape distribution. Tang [19] found that granulometric features were the most important features in his experiment.

We applied 3×3 , 5×5 , 7×7 and 9×9 squares as structure elements and did a series of morphological openings. Then differences in size were normalized by the original plankton size to obtain the granulometric features. Also, we applied 3×3 , 5×5 and 7×7 squares as structure elements, and did a series of morphological closings. The differences in size were normalized in the same way. We did not apply 9×9 squares to the closing because the SIPPER images are so small that most of them are diminished after the closing with 7×7 square as the structure element.

3.4 Domain specific features

Moment invariants and granulometries are general features, which can only capture some global information. They are far from enough to classify SIPPER images. Given advice from domain experts, we developed some domain specific features to help classification.

The domain specific features include size, convex ratio, transparency ratio, ratio between the two eigenvalues, and ratio between the plankton’s head and tail.

- **Size:** Size is the area of the plankton body, that is, the number of foreground pixels in the plankton image.
- **Convex ratio:** We implemented a fast algorithm [1] to get the convex hull of the plankton image. The convex ratio is the ratio between the plankton image size and the area of the convex hull. This feature contains information about the plankton boundary irregularity.
- **Transparency ratio:** This is the ratio between the area of the plankton image and the area of the plankton after filling all inside holes. The transparency ratio helps in recognizing the transparent plankton.
- **Ratio between the two eigenvalues:** Since some plankton are linear we first compute the two eigenvalues of the image. Then the ratio between them is computed.
- **Ratio between the head and the tail:** Some plankton such as larvaceans have a large head relative to their tail. We compute the ratio between the head and tail to differentiate them. To do this we first rotate the image to make the axis with the bigger eigenvalue parallel to the x-axis. Assuming the smallest and largest x values are 0 and T respectively, we accumulate the number of foreground pixels along the x-axis from 0 to $\frac{1}{4}T$ and from $\frac{3}{4}T$ to T respectively. Then we take the ratio between them as the ratio between the head and the tail.

4 Support vector machines and probability model

Support vector machines (SVMs) [20] are receiving increasing attention these days and have achieved very good accuracy in pattern recognition, text classification, etc. [4]. In this section we describe SVMs and introduce a way to assign a probability value after multi-class SVM classification.

4.1 Support vector machines

In binary classification, SVMs try to find a hyperplane to separate the data into two classes. In the case in which all the data are well separated, the margin is defined as two times the distance between the hyperplane and the closest example. SVMs search for the hyperplane with the largest margin, which provides good generalization ability based on Vapnik’s VC dimension theory [20]. To increase the classification ability, SVMs first map the data into a higher dimension feature space

with $\phi(x)$, then use a hyperplane in that feature space to separate the data. In the feature mapping stage, the kernel $k(x, y) = \langle \phi(x) \cdot \phi(y) \rangle$ is used to avoid explicit inner product calculation in the high-dimension feature space. C-SVM, a typical example of soft SVMs, is described as follows. The slack variable ξ_i is used to handle non-separable cases.

Training set: there are m examples: x_1, x_2, \dots, x_m with class label $y_i \in \{-1, 1\}$.

C-SVM:

$$\text{minimize } \frac{1}{2} \langle w, w \rangle + \frac{C}{m} \sum_{i=1}^m \xi_i \quad (3)$$

$$\text{subject to: } y_i(\langle w, \phi(x_i) \rangle + b) \geq 1 - \xi_i \quad (4)$$

where w is normal to the hyperplane, C is a scalar value that controls the trade off between the empirical risk and the margin length, ξ_i is the slack variable and $C, \xi_i > 0$.

The decision function is $f(x) = \sum_i \alpha_i k(x_i, x) + b$, where α_i and b are computed from Eq. (3) and (4).

The Karush-Kuhn-Tucker condition of the optimal solution to Eq. (3) and (4) is:

$$\alpha_i(\langle w, \phi(x_i) \rangle + b - 1 + \xi_i) = 0 \quad (5)$$

The α_i is nonzero only when Eq. (6) is satisfied. In this case the x_i contributes to the decision function and is called a support vector (SV).

$$y_i(\langle w, \phi(x_i) \rangle + b) = 1 - \xi_i \quad (6)$$

Therefore, we get a sparse solution of the decision function, where only SVs contribute.

There are two main approaches to extending SVMs to multi-class classification:

1. **One-vs-all:** A set of binary SVMs are trained to separate one class from the rest. The drawback is that we are handling unbalanced data when building binary SVMs. Moreover, each binary SVM is built on a totally different training set. There might be cases in which some binary SVMs conflict with each other for some examples. It is difficult to assign the class only by the real-valued outputs from every binary SVM.
2. **One-vs-one:** All possible groups of 2 classes are used to build binary SVMs. In the N class case, we will build $\frac{N(N-1)}{2}$ binary SVMs. When a new example comes, all the binary SVMs vote to classify it. When N is small, each binary SVM only learns on a fraction of the data which can be time efficient. However, the training phase is very long if we have very big N . We use the one-vs-one approach since there are only 5 or 6 classes in our problem.

4.2 Assigning probability values in support vector machines

A probability associated with a classifier is often very useful and it gives some confidence about the classification result. For instance, the classifier could reject the example and leave it to a human to classify it when the confidence is very low. Platt [16] introduced the sigmoid function as the probability model to fit $P(y = 1|f)$ directly. The parametric model is shown in Eq. (7).

$$P(y = 1|f) = \frac{1}{1 + \exp(Af + B)} \quad (7)$$

where A and B are scalar values. f is the decision function of the binary SVM.

The A and B are fit with maximum likelihood estimation from the training set. Platt tested the model with 3 data sets including the UCI Adult and two other web classification data sets. The sigmoid-model SVM had good classification accuracy and probability quality in his experiments.

We follow his sigmoid model and extend it to the multi-class case. In the one-vs-one multi-class SVM model, since it is time consuming to do the parameter fitting for all $\frac{N(N-1)}{2}$ binary SVMs, we developed a practical method to compute the probability value while avoiding parameter fitting.

1. We assume $P(y = 1|f = 0) = P(y = -1|f = 0) = 0.5$. It means that a point right on the decision boundary will have 0.5 probability of belonging to each class. We get rid of parameter B in this way.
2. Since each binary SVM has a different margin, a crucial criterion in assigning the probability, it is not fair to assign a probability without considering the margin. Therefore, the decision function $f(x)$ is normalized by its margin in each binary SVM. The probability model of SVMs is shown as following.

$$P_{ij}(y = 1|f) = \frac{1}{1 + \exp(\frac{-Af}{\|w\|})} \quad (8)$$

$$P_{ij}(y = -1|f) = 1 - P_{ij}(y = 1|f) = P_{ji}(y = 1|f) \quad (9)$$

P_{ij} : binary SVM on class i vs. class j , class i is +1 and class j is -1

3. After we get the probability value for each binary SVM, the final probability for class i is computed as follows:

$$P(i) = \prod_i^{i \neq j} P_{ij}(y = 1|f) \quad (10)$$

Normalize $P(i)$ to make $\sum_i P(i) = 1$

4. output $k = \arg \max_i P(i)$ as the prediction.

A is determined through numeric search based on the cost function $\sum_i \log P(k)$ from 10-fold cross validation. After we finish learning a SVM model and set up a rejection threshold t , we reject an example and leave it to be classified by a person if $\max_i P(k) < t$.

5 Feature selection

Feature selection helps reduce the feature computation time and increase the accuracy. There are mainly two ways to do feature selection [6]. The filtering approach attempts to select a subset of features without applying learning algorithms. It is fast, but seems unlikely to result in the best accuracy. The wrapper approach [11] selects a feature subset by applying the learning algorithm. It has the potential to give us very good accuracy but is computationally expensive. A feature selection method specifically for SVMs has been proposed recently. Weston [21] tried to minimize the generalization bound by minimizing the radius of the sphere including all the training examples. The drawback of this approach is that the generalization bound is too loose.

In our system, we applied the wrapper approach with backward elimination. Backward elimination means starting with all the features and keep eliminating features. The average accuracy from a five-fold cross validation is used as an evaluation function. If there are p continuous nodes whose average accuracy does not increase we stop. We performed best first search (BFS) to explore the feature subset space in the beginning of the algorithm. However, BFS tends to stop with many features. In order to explore feature subsets with less features, greedy beam search (GBS) was employed after the stop of BFS. GBS operates by only expanding the best q (beam width) leaf-nodes without any backtracking. It can quickly reduce the number of features to 1.

To reduce the effect of overfitting, we take 20 percent of the data as a hold-out data set, and run the feature selection on the remaining data while testing the selected feature subsets on the hold-out data.

6 Experiments

Several experiments have been done to test our system. The Libsvm [2] support vector machine software is modified and used in our experiments. Libsvm uses decomposition in its optimization and a one-vs-one approach to do multi-class classification. We modified libsvm to produce a probabilistic output. In all experiments the gaussian radial basis function (RBF) is used as the kernel.

The gaussian RBF kernel: $k(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$ where σ is a scalar value.

6.1 Initial experiments

The first training set has total 1285 SIPPER images ($50\mu\text{m}$ resolution). There are 64 diatoms, 100 protocista, 321 doliolids, 366 larvaceans, and 434 Trichodesmium. We used C -SVM module with parameters $C = 200$ and $\sigma = 0.03$. To evaluate the accuracy of SVMs, we also compared it with a cascade correlation neural network [8] and a C4.5 decision tree with the default pruning settings [14]. Figure 2 shows the average accuracy of the three learning algorithms from 10-fold cross validation. A paired-t test is used to compare the results at the 95% confidence interval. The SVM is more accurate than the other two learning algorithms at the 95% confidence level.

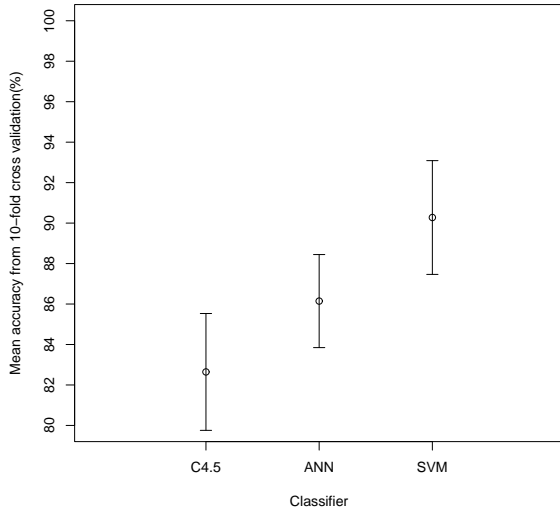


Figure 2: The mean accuracy and the range of standard deviation of the C4.5, the cascade correlation neural network (ANN) and the SVM from 10-fold cross validation on the 1285 SIPPER image set.

Table 1 shows the confusion matrix of the SVM from a 10-fold cross validation experiment. The overall average accuracy is 90.04%. While we have greater than 84% accuracy on most plankton, we only achieve 79% accuracy on the diatom class. The reason is that we only have 64 diatom samples in our training set and the SVM favors the class with more samples. For instance, given there is an overlap in the feature space between two classes: one with many examples, one with few examples. It is more likely that most examples within that overlap come from the class with more examples. To minimize the Eq. 3, the decision boundary is pushed away from the class with more examples and thus favoring that class.

Table 1: Confusion matrix from a 10-fold cross validation on 1285 SIPPER images with all 29 features. P, Di, Do, L and T represent Protocista, Diatom, Doliolid, Larvacean and Trichodesmium respectively.

Classified	as P	as Di	as Do	as L	as T
P	84.4%	1.6%	9.4%	4.7%	0.0%
Di	2.0%	79.0%	11.0%	6.0%	2.0%
Do	0.8%	0.3%	92.8%	3.1%	0.0%
L	0.8%	0.3%	4.4%	88.0%	6.6%
T	0.0%	0.5%	0.2%	6.2%	93.1%

6.2 Experiments with unidentifiable particles

Encouraged by the initial experiment, we chose plankton images from some other collections of sample images. We picked the five most abundant types of plankton, which account for 95% of the plankton samples from the particular area of acquisition in the Gulf of Mexico. They are copepods, doliolids, larvaceans, protocista and Trichodesmium. The image quality in this training set is not as good as in the initial experiment. Some information, unknown to us, was used by ocean experts to label the images. Also, we are forced to handle unidentifiable particles in this experiment.

There are a total of 6000 images: 1000 images of each plankton class and 1000 unidentifiable particles. We use C -SVM with $C = 200$ and $\sigma = 0.032$. Figure 3 shows the average accuracy of three learning algorithms from 10-fold cross validation. A paired-t test is used to compare the results at the 95% confidence interval. The SVM is more accurate than the other two learning algorithms at the 95% confidence level.

Table 2 shows the confusion matrix from a 10-fold cross validation. The overall average accuracy is 75.12%. The average accuracy from the five types of plankton is 78.56%.

There are a significant number of larvaceans confused with Trichodesmium. This observation disagrees with the first experiment where we had high classification accuracy for both types of plankton. The justification is that some larvacean and Trichodesmium are linear objects. Domain experts know that there are some ocean areas where larvacean or Trichodesmium are less common. They labeled the linear objects as larvacean or Trichodesmium because they know the other plankton are less commonly found in the particular ocean areas examined. Therefore, there are many linear particles without significant features to differentiate between the two types of plankton in this training set, thus dropping the classification accuracy on larvaceans and Trichodesmium.

6.3 Feature selection

Feature selection was tested on the bigger training set as described in Section 6.2. Although the SVM seems superior to the other two learning algorithm, there is

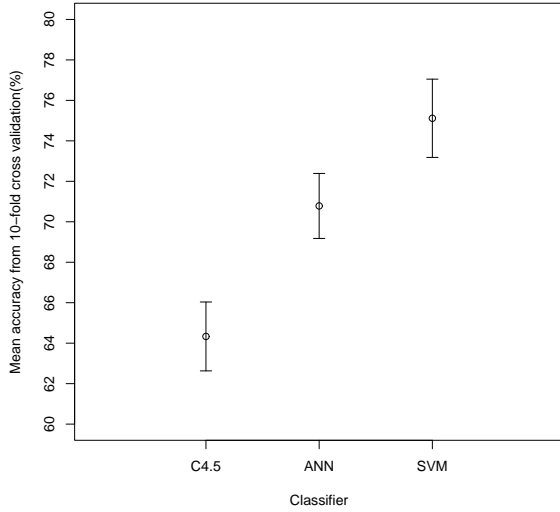


Figure 3: The mean accuracy and the range of the standard deviation of the C4.5, the cascade correlation neural network (ANN) and the SVM from 10-fold cross validation on the 6000 SIPPER image set.

Table 2: Confusion matrix from a 10-fold cross validation on 6000 SIPPER images with all 29 features. C, D, L, P, T, and U represent Copoped, Doliolid, Larvacean, Protoctista, Trichodesmium and Unidentifiable particles respectively.

Classified	as C	as D	as L	as P	as T	as U
C	84.2%	0.6%	3.1%	1.0%	5.5%	5.6%
D	0.2%	82.9%	2.4%	8.7%	0.4%	5.4%
L	3.2%	1.9%	68.8%	1.4%	11.1%	13.6%
P	1.7%	5.3%	1.1%	84.4%	3.1%	4.4%
T	3.3%	0.6%	9.4%	1.8%	72.5%	12.4%
U	4.3%	3.1%	15.8%	5.4%	13.5%	57.9%

no guarantee that it is still true after feature reduction. Therefore, we experimented with feature selection (wrapper approach) on the SVM and its direct competitor: the cascade correlation neural net. We did not use the decision tree in the comparison because it is far less accurate than the SVM on this data set, thus unlikely to be the best. The data set was divided into two parts: 80% as training and 20% as validation. We set stopping criterion p to 150 and the beam width q to 5 in our experiment.

Figures 4 and 5 show the experimental results for the average accuracy from the 5-fold cross validation on the training data and the test accuracy on the validation data respectively. The SVM provides better accuracy than the neural net on both the training set and the validation set. To choose the least number of features for the SVM, McNemar’s test [7] is applied on the valida-

Table 3: Confusion matrix from a 10-fold cross validation on 6000 SIPPER images with the best 15-feature subset. C, D, L, P, T, and U represent Copoped, Doliolid, Larvacean, Protoctista, Trichodesmium and Unidentifiable particles respectively.

Classified	as C	as D	as L	as P	as T	as U
C	84.5%	0.9%	3.1%	0.5%	5.6%	5.4%
D	0.7%	85.2%	1.1%	9.3%	0.4%	3.3%
L	4.3%	2.1%	67.2%	1.1%	12.5%	12.8%
P	1.8%	5.0%	0.7%	85.8%	3.0%	3.7%
T	4.5%	0.4%	10.0%	1.5%	72.5%	11.0%
U	5.1%	2.3%	15.6%	5.4%	13.4%	58.2%

tion set to compute the 95% confidence interval. When the number of features are less than 15, the accuracy is outside the confidence interval. Therefore, we choose the 15-feature subset as the optimal feature subset and it provides slightly better accuracy than using all the features on the validation data set.

To test the overall effect of feature selection, we applied 10-fold cross validation on whole 6000 image set. The confusion matrix is shown as Table 3. The overall average accuracy is 75.57%. The average accuracy from the five types of plankton is 79.04%. Both indicate that the best 15-feature subset performs slightly better than all 29 features.

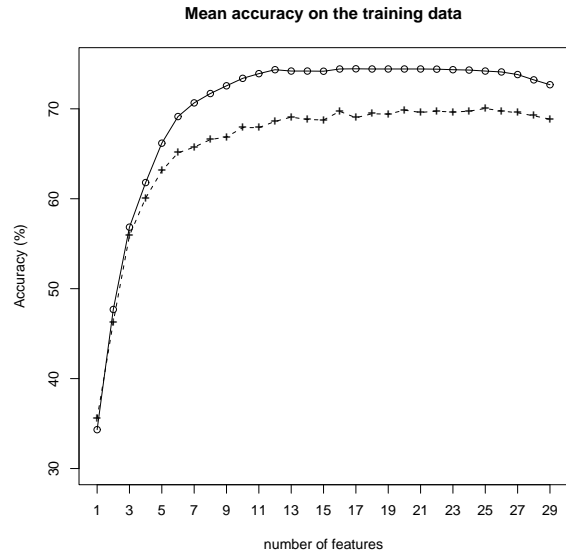


Figure 4: Feature selection on the training set: The solid line represents accuracy of the SVM and the dashed line represents the accuracy of the neural net.

6.4 Probability assignment experiments

We used the same training set as in the last experiment with the 15-feature subset. To determine A in

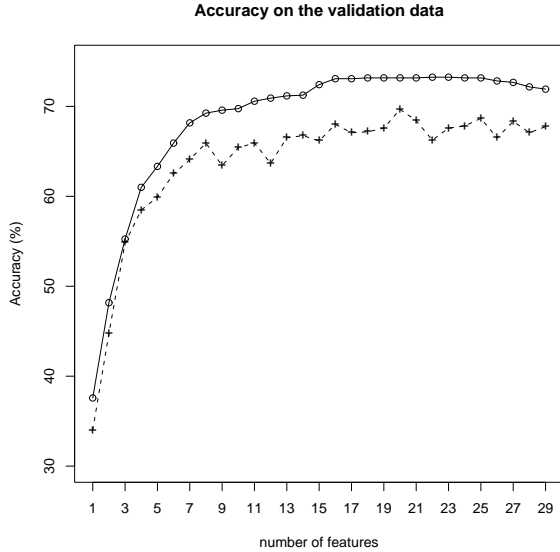


Figure 5: Selected feature subsets on the validation set: The solid line represents accuracy of the SVM and the dashed line represents the accuracy of the neural net.

Eq. (7), we varied the A value in 10-fold cross validation experiments and picked the one with the lowest cost $\sum_i \log P(k)$. The best value for A in our experiment was 1500. We drew a rejection curve by varying the rejection threshold. Figure 6 shows that the accuracy goes up as the rejection ratio increases, which is reasonable.

7 Conclusions and future work

This paper presents a plankton recognition system for binary SIPPER images. General features as well as domain specific features are extracted and a support vector machine is used to classify examples. We also developed a new way to assign a probability value after the multi-class SVM classification. We tested our system on two different data sets. The recognition rate exceeded 90% in one experiment and was greater than 75% on the more challenging data set with unidentifiable particles. SVM is more accurate than the C4.5 decision tree and the cascade correlation neural network at the 95% confidence level on the two data sets. The wrapper approach with backward elimination successfully reduced the number of features from 29 to 15 with slightly better accuracy than using all features.

The system does not do well at recognizing unidentifiable particles. It is hard to develop specific features to describe the features of the unidentifiable particles because they vary so much. More powerful descriptive and robust general features seem needed in our future work. Recently, an advanced SIPPER system had been developed to produce grayscale SIPPER images at 25 μm resolution. We are in the process of developing meth-

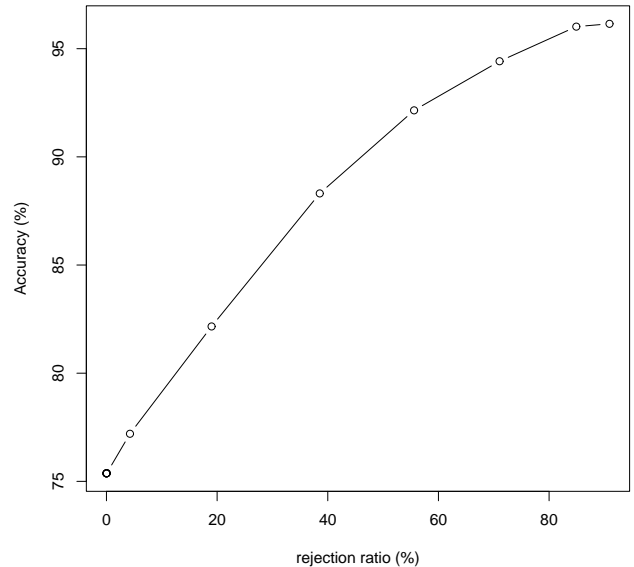


Figure 6: Rejection curve-Overall accuracy vs. rejection rate.

ods and features for higher resolution (25 μm resolution) grayscale SIPPER images.

8 Acknowledgments

The ratio between two eigenvalue and the ratio between head and tail features were suggested by Xiaou Tang (personal communication). The research is partially supported by the United States Navy, Office of Naval Research, under grant number N00014-02-1-0266 and the NSF under grant EIA-0130768.

References

- [1] Mark De Berg(Editor), Marc Van Kreveld, Mark Overmars, O. Schwarzkopf, Mark de Berg, *Computational geometry: algorithms and applications*, Springer, 2001.
- [2] Chih-Chung Chang and Chih-Jen Lin, "LIB-SVM: a library for support vector machines", <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>, 2002.
- [3] A. Ciobanu and H. D. Buf, "Identification by contour profiling and legendre polynomials", *Automatic diatom identification*, pp. 167–186, World Scientific, 2002.
- [4] N. Cristianini, J. Shawe-Taylor, *Introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, 2000.

- [5] L. F. Costa, R. M. Cesar Jr., *Shape analysis and classification*, CRC press LLC, 2001.
- [6] Thomas G. Dietterich, "Machine learning research: four current directions", *AI Magazine*, Vol 18, No. 4, pp. 97–136, 1997.
- [7] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms", *Neural Computation*, Vol 10, No. 7, pp. 1895–1924, 1998.
- [8] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture", *Advances in neural information processing systems*, Vol 2, pp. 524–532, 1991.
- [9] S. Fischer and H. Bunke, "Identification using classical and new features in combination with decision tree ensembles", *Automatic diatom identification*, pp. 109–140, World Scientific, 2002.
- [10] M. K. Hu, "Visual pattern recognition by moment invariants", *IRE Trans. Information theory*, Vol IT, No. 8, pp. 179–187, 1962.
- [11] J. Kohavi, "Wrappers for feature subset selection", *Artificial intelligence, special issue on relevance*, Vol 97, No. 1-2, pp. 273–324, 1997.
- [12] R. E. Loke and H. d. Buf, "Identification by curvature of convex and concave segment," *Automatic diatom identification*, pp. 141–166, World Scientific, 2002.
- [13] G. Matheron, *Random sets and integral geometry*, John Wiley and Sons: New York, 1975.
- [14] J. R. Quinlan, *C4.5: Programs from empirical learning*, Morgan Kaufmann, San Francisco, 1993.
- [15] I. Pitas, *digital image processing algorithms and applications*, John Wiley and Sons, Inc., 2000.
- [16] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", *Advances in Large Margin Classifiers*, pp. 61–74, 2000.
- [17] S. Samson, T. Hopkins, A. Remsen, L. Langebrake, T. Sutton, and J. Patten, "A system for high resolution zooplankton imaging", *IEEE journal of ocean engineering*, Vol 26, No. 4, pp. 671–676, 2001.
- [18] L. M. Santos and H. D. Buf, "Identification by gabor features," *Automatic diatom identification*, pp. 187–220, World Scientific, 2002.
- [19] X. Tang, W. K. Stewart, L. Vincent, H. Huang, M. Marra, S. M. Gallager and C. S. Davis, "Automatic plankton image recognition," *Artificial intelligence review*, Vol 12, No. 1-3, pp. 177–199, 1998.
- [20] Vladimir N. Vapnik, *The nature of statistical learning theory*, Springer, 2000.
- [21] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, "Feature selection for SVMs", *Neural information processing systems*, Vol 13, 2001.
- [22] M. H. F. Wilkinson, A. C. Jalba, E. R. Urbach, and J. B. T. M. Roerdink, "Identification by mathematical morphology," *Automatic diatom identification*, pp. 221–244, World Scientific, 2002.
- [23] C. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curve", *IEEE transaction on computers*, Vol C, No. 21, pp. 269–281, 1972.