2005

Combining Local and Global Image Features for Object Class Recognition

Dimitri A Lisin, *University of Massachusetts - Amherst*
Marwan A Mattar, *University of Massachusetts - Amherst*
Matthew B Blaschko, *University of Massachusetts - Amherst*
Mark C Benfield, *University of Massachusetts - Amherst*
Erik G Learned-Miller, *University of Massachusetts - Amherst*

Combining Local and Global Image Features for Object Class Recognition

Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko,[*]
Mark C. Benfield†, Erik G. Learned-Miller

Computer Vision Laboratory
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003 USA

†Dept. of Oceanography &
Coastal Sciences/Fisheries Inst.
Louisiana State University
Baton Rouge, LA 70803 USA

{dima,mmattar,blaschko,elm}@cs.umass.edu, mbenfie@lsu.edu

Abstract

Object recognition is a central problem in computer vision research. Most object recognition systems have taken one of two approaches, using either global or local features exclusively. This may be in part due to the difficulty of combining a single global feature vector with a set of local features in a suitable manner.

In this paper, we show that combining local and global features is beneficial in an application where rough segmentations of objects are available. We present a method for classification with local features using non-parametric density estimation. Subsequently, we present two methods for combining local and global features. The first uses a "stacking" ensemble technique, and the second uses a hierarchical classification system. Results show the superior performance of these combined methods over the component classifiers, with a reduction of over 20% in the error rate on a challenging marine science application.

1 Introduction

Most object recognition systems tend to use either global image features, which describe an image as a whole, or local features, which represent image patches. Global features have the ability to generalize an entire object with a single vector. Consequently, their use in standard classification techniques is straightforward. Local features, on the other hand, are computed at multiple points in the image and are consequently more robust to occlusion and clutter. However, they may require specialized classification algorithms to handle cases in which there are a variable number of feature vectors per image.

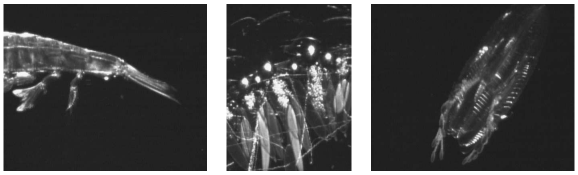
One contribution of this paper is a novel method for object recognition with local features. We propose to model classes of images as a probability distribution over local features. The probability density functions are estimated non-parametrically, and are then used to build a maximum likelihood classifier. We will refer to this classifier as Non-Parametric Density (NPD). This method is shown to perform better than several other local feature classifiers. It also has the advantage of being able to output a posterior distribution over labels, rather than a single class label.

Despite the robustness advantages of local features, global features are still useful in applications where a rough segmentation of the object of interest is available. Automatic detectors exist for several broad classes of objects, such as faces [22] or signs [20]. For such applications global features provide information that is useful for class discrimination.

Due to the fundamental difference in how local and global features are computed, we expect that the two representations would provide different kinds of information. Most local features represent texture in an image patch. For example, SIFT features use histograms of gradient orientations [11]. Global features include contour representations, shape descriptors, and texture features. Global texture features and local features provide different information about the image because the support over which texture is computed varies. We expect classifiers that use global features will commit errors that differ from those of classifiers based on local features. This is supported by the confusion matrices in Tables 1 and 2, which will be discussed further below.

We present two techniques to exploit this partial independence of error to improve classification accuracy. The first method uses stacking [16] to combine the output of separate classifiers for local and global features. The approach uses the fact that the NPD classifier described above outputs posterior distributions over class labels. The sec-

[*] Current affiliation: Intelligente Systeme, Fachbereich Informatik, TU Darmstadt, Germany



(a) Euphausiid (b) Siphonophore (c) Ctenophore

Figure 1: A few example images from the VPR data set.

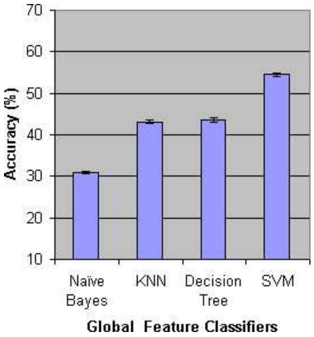


Figure 2: Global Feature Classifiers

ond method forms a two-tier hierarchy of classifiers, where the first stage uses a global feature classifier and the second stage uses a local feature classifier. We group the classes that are confused in the global feature space and rely on the local classifier to sort the resulting superclass. Both techniques significantly improved classification accuracy over any single component classifier.

The primary application of these techniques is to marine science data collected by a tool called the Video Plankton Recorder (VPR) [2]. The Video Plankton Recorder captures images of multicellular organisms that have organs and appendages with distinct visual appearances (Figure 1). The data set consists of 1826 gray-scale images that belong to one of 14 classes, which have been identified by experts. The data set is challenging from a classification viewpoint for several reasons. Organisms are photographed from arbitrary three-dimensional views. The size of the organisms relative to the field of view of the camera results in many images in which an organism is only partially visible. The highest accuracy that we were able to achieve with techniques that use either local or global features alone is approximately 54%, while combining the two types of features boosted it to 65.5%. For comparison, Davis et al. [3] report 60-70% accuracy on a similar dataset also acquired by VPR, but only containing 7 classes. It is consequently a challenging and attractive data source for testing our methods.

2 Classification with Global Features

Many object recognition systems use global features that describe an entire image. Most shape and texture descriptors fall into this category. Such features are attractive because they produce very compact representations of images, where each image corresponds to a point in a high-dimensional feature space. As a result, any standard classifier can be used.

On the other hand global features are sensitive to clutter and occlusion. As a result it is either assumed that an image only contains a single object, or that a good segmentation of the object from the background is available. In our case, an image often does contain a single object, but sometimes several organisms or particles are present.

We have found that a simple global bimodal segmentation is usually effective for separating the plankton from the background, which tends to be significantly darker than the object. We use expectation maximization (EM) to fit a mixture of two Gaussians to the histogram of gray values for a given image [4]. The Bayesian decision boundary defines the cut point between foreground and background. After that, morphological hole filling [18] is used to capture the stray dark pixels inside the object.

From the segmentation of each image we have computed three simple shape descriptors: area, perimeter, and compactness (perimeter squared over area). We have also used two kinds of global texture features: local binary patterns (LBP), which are gray-scale and rotation invariant texture operators [13], and shape index which is computed using the isophote and the flowline curvatures of the intensity surface [14]. These features comprise an effective subset of the features explored for plankton categorization in [1]. Classification results for several commonly used classifiers are shown in Figure 2.

3 Classification with Local Features

A different paradigm is to use local features, which are descriptors of local image neighborhoods computed at multiple interest points. In this section, we describe typical ways in which local features are used. One of the key issues in dealing with local features is that there may be differing numbers of feature points in each image, making comparing images more complicated. We present the Hausdorff Average, a standard technique for comparing point sets of different sizes, and apply it to comparing images represented with local features. Subsequently we offer a probabilistic method, which evaluates the average log likelihood of feature points under a non-parametric density estimate for the class, to evaluate the likelihood of the class for a particular image. Our proposed method outperforms the Hausdorff Average method and is an important component of our combined local-plus-global method.

Typically, interest points are detected at multiple scales and are expected to be repeatable across different views of an object. The interest points are also expected to capture the essence of the object's appearance. The feature descriptor describes the image patch around an interest point.

The usual paradigm of using local features is to match them across images, which requires a distance metric for comparing feature descriptors. This distance metric is used to devise a heuristic procedure for determining when a pair of features is considered a match, e. g. by using a distance threshold. The matching procedure may also utilize other constraints, such as the geometric relationships among the interest points, if the object is known to be rigid.

One advantage of using local features is that they may be used to recognize the object despite significant clutter and occlusion. They also do not require a segmentation of the object from the background, unlike many texture features, or representations of the object's boundary (shape features).

In this paper we have used the SIFT (Scale Invariant Feature Transform) features proposed by Lowe [11], which use local maxima of the difference-of-Gaussians function as interest points and histograms of gradient orientations computed around the points as the descriptors.

3.1 Feature Matching

Usually, local features from a pair of images are matched to produce a list of reliable point correspondences. The correspondences can then be used to perform image classification. In previous work by Lowe [11], image matching was performed by counting the number of vectors in the testing image that ”matched” to vectors in the training image. Two vectors match if their Euclidean distance falls below a threshold. We decided to use the number of matches between two images as our similarity measure.

Let $m(A, B)$ be the number of matches obtained by matching features from image A to features from image B. Note that in general $m(A, B) \neq m(B, A)$, because the distance threshold procedure allows many-to-one feature matches. We can then define similarity between two images as $d(A, B) = (m(A, B) + m(B, A))/2$. Now we can easily build a k-nearest-neighbor (KNN) classifier. This approach has performed very well on a sign recognition task [12] in which the goal was to identify specific objects stored in a database.

The disadvantage of using the number of matches as a similarity measure is that image matching fails to generalize for the entire class consisting of highly variable organisms. This is problematic in this application due to the high in-class variability. The accuracy achieved by this method on our domain was only 25% using 1 nearest neighbor. Using more neighbors decreased the accuracy by 1-2%. To mitigate this problem, we adopted a image distance more suitable to this task, the Hausdorff Average.

3.2 Hausdorff Average

The one-sided Hausdorff distance [9] between two sets of points in a space is defined as

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||, \tag{1}$$

where A and B are the two sets of points, and $|| \cdot ||$ is a norm for points in the sets.

In general, under this formulation $h(A, B) \neq h(B, A)$. To address this, the bi-directional Hausdorff distance is defined as

$$\hat{h}(A, B) = \max(h(A, B), h(B, A)). \tag{2}$$

The Hausdorff distance is often used for object detection, where an image is represented by a set of edge points. In our case, we use the Hausdorff distance to compare sets of points in a high-dimensional feature space, rather than in the image plane. Specifically, we use a variation of the Hausdorff distance, known as the Hausdorff Average, defined as

$$ha(A, B) = \frac{\sum_{a \in A} \min_{b \in B} ||a - b||}{|A|}, \tag{3}$$

where $|A|$ is the cardinality of A, and $a_i \in A$. The Hausdorff Average has been shown to be the most stable variation of the Hausdorff distance under image distortions [17]. Intuitively, the distance between two images is made greater whenever a local feature in one image is not close to any of the local features in the other image, and vice versa.

The Hausdorff average allows us to compare two images represented by the corresponding sets of local features, which also can be used to build a k-nearest-neighbor classifier. On our data set the accuracy of a KNN classifier using 5 nearest neighbors was 45.66%.

3.3 Maximum Likelihood Classifier

While the accuracy of the Hausdorff-based classifier is encouraging compared to the feature matching technique described in Section 3.1, we believe that classes of images can be better represented by estimating a probability distribution over local features present in those images. Once the distributions for each class of images are estimated, we can build a maximum-likelihood classifier.

Since we have little a priori knowledge about structure in our data, we will use non-parametric density estimation. We start by gathering local features from training images of a particular class into a single set. Then for every local feature, a Gaussian kernel is placed in the feature space with its mean at the feature. The probability density function (PDF) of the class is then defined as the normalized sum of all the kernels. In theory, it is possible to estimate the distribution over local features for each individual image. However, a union of the features from all training images of a class gives us a much larger number of samples, resulting in a better density estimate.

We set the covariance of the kernels using Parzen Windows [6]. The approach keeps the kernels isotropic, and the standard deviations of all kernels the same. Thus, there is only one parameter σ, which is set such that the mean log likelihood of every point is maximized using a leave-one-out scheme.

After the PDFs of all classes are estimated, we can build a maximum-likelihood classifier. Let $C = \{C_1, C_2, ..., C_n\}$ be the set of image classes. Let $Q = \{q_1, q_2, ..., q_m\}$ be a query image, and $q_i \in Q$ be one of its constituent local features. First, we compute the likelihood of the query given each class:

$$\log p(Q|C_i) = \frac{1}{m} \sum_{j=1}^{m} \log p(q_j|C_i), \qquad (4)$$

where $p(q_j|C_i)$ is given by the PDF of C_i. Summing the log likelihoods for each class corresponds to an assumption that the local features found in each image are generated independently. We can then output the most likely class label for Q.

Furthermore, the posterior probabilities for each class $p(C_i|Q)$ can be easily computed by normalizing the likelihoods:

$$p(C_i|Q) = \frac{p(Q|C_i)}{\sum_{j=1}^{n} p(Q|C_j)}. \qquad (5)$$

We assume uniform priors, because one of our objectives is to estimate relative proportions of the populations of different plankton species.

One of the difficulties of using non-parametric density estimation is that in higher dimensions one needs a very large number of sample points. In the case of local features, this problem is somewhat alleviated by the fact that there are many more local features than there are images. Furthermore, in our implementation we first reduce the dimensionality of the SIFT features from 128 to 16 using Principal Components Analysis.

This classification technique differs significantly from most methods that use local features in that it does not explicitly compute feature correspondences. For example, Helmer and Lowe [8] propose a probabilistic object recognition method that models an object as a collection of parts, and looks for most likely matches between model parts and image features. The NPD approach, on the other hand represents a class of objects as a probability distribution over the feature space, and computes the likelihood of an image feature, without expicitly assigning it to a particular model part.

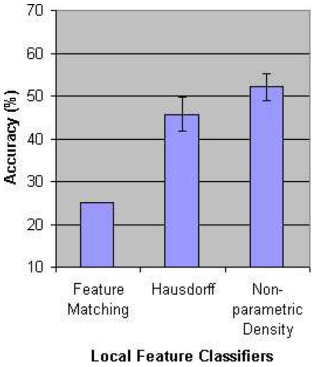


Figure 3: Local Features Classifiers

Comparative results are shown for the three techniques for local feature classification described here in Figure 3.

4 Combination Methods

The key contribution of this paper is combining the different information provided by local and global features. We explore two methods for achieving this. The first is the classical method of stacking and the second is using a classification hierarchy. Both significantly improve results over methods that use either global or local features alone.

4.1 Stacking

Ensemble methods are learning algorithms that have been shown to improve performance by combining the outputs of multiple component classifiers. Ensemble methods for classification have been shown to have better accuracy than the component classifiers if the component classifiers are accurate and diverse [5]. An accurate classifier is one that outperforms random guessing, and diverse classifiers are those that commit independent errors. One of the main areas of ensemble research is how to induce independence between the component classifiers. Inducing independence can be achieved by manipulating the training set, manipulating the input features, or injecting randomness in the learning algorithm.

As opposed to techniques in which a fixed ensemble strategy is used, meta-learning techniques employ a meta-classifier that generalizes over the space of outputs from base level classifiers. In *stacking* [16] the outputs of constituent classifiers are concatenated and used as an input feature vector for a meta-classifier. Stacking is perhaps the most intuitive technique for meta-learning, but has found surprisingly low adoption in the vision community.

We discuss here two main variations of stacking. In the first, the input to the meta-classifier is a concatenation of class labels produced by each of the component classifiers. In the second variation, each component classifier outputs a posterior distribution over class labels, rather than a single label. Distributions from the component classifiers are concatenated and used as input to the meta-classifier. Stacking with probability distributions, in essence, trains on an estimate of classification confidence from the base level classifier. Any classifier can be used at the base level if we only require a single category label, but stacking with probability distributions restricts us to classifiers that output distributions over class labels. The choice of meta-classifier is not restricted in any way. In our experiments with stacking we have used SVM as the meta-classifier.

Our local feature classifier produces posterior distributions over labels as described in Section 3.3. We have experimented with two variations of base level classifiers for global features. In the first, we used non-parametric density estimation to build several maximum-likelihood classifiers using different global features. Using stacking to combine only the global features, we achieved an accuracy of 50.32%. Because SVM had a higher accuracy for global features, our second technique uses SVM classifiers at the base level. The meta-classifier takes a vector consisting of ones in the elements corresponding to the base-level classification, and zero elsewhere. Classification accuracy for these experiments is summarized in Figures 4(a) and 4(b).

4.2 Classification Hierarchy

Given that local and global features provide different kinds of information about an image, it is possible that a pair of classes not separable in global feature space will be distinguished by local features. In this section, we propose a 2-tier hierarchical classification system that uses global and local features in succession.

At the top level, classes that are not separable by global features are merged into super-classes. The global feature classifier is then trained on these super-classes. A local feature classifier is then trained to distinguish between the original classes contained in each superclass. When a query image is classified as belonging to a super-class, it is passed to the local feature classifier, which in turn determines to which of the classes the image belongs. The reasoning behind this is that at the top level the images are categorized into broader, more separable, groups, and at the bottom level the classifier has to distinguish between fewer classes. The global feature classifier is used at the top level because it is faster, resulting in a speedup for the overall system.

Table 1: Confusion matrix for SVM with global features

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**21**	5	3	1	1	2	0	1	2	96	0	0	0	1
2	4	**33**	0	0	6	3	0	1	3	30	0	5	1	0
3	3	2	**22**	0	0	0	11	1	1	54	3	0	1	2
4	3	2	0	**10**	8	1	0	0	1	2	0	7	0	0
5	1	4	0	2	**94**	2	0	0	0	5	0	10	13	0
6	1	2	1	0	11	**4**	1	0	0	39	1	1	7	0
7	0	3	11	0	0	1	**29**	0	3	70	21	0	1	3
8	3	3	2	0	0	0	0	**83**	2	1	1	1	0	1
9	5	0	1	2	2	0	3	0	**89**	24	3	1	0	3
10	12	11	10	0	1	2	21	3	6	**339**	16	0	4	8
11	0	0	3	0	0	0	16	1	2	18	**67**	0	0	1
12	1	13	1	5	11	0	1	6	3	5	0	**155**	1	0
13	3	0	0	0	19	5	0	0	0	23	0	1	**30**	0
14	4	2	2	0	0	0	2	1	17	33	5	1	1	**10**

Table 2: Confusion matrix for NPD with local features

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**18**	0	0	0	17	0	0	0	0	41	0	56	1	0
2	2	**13**	0	0	8	0	1	0	1	8	0	50	1	1
3	1	0	**47**	2	4	0	7	0	7	24	0	7	0	0
4	0	0	0	**0**	15	1	0	0	0	2	0	16	0	0
5	0	0	0	0	**87**	0	0	0	0	2	0	39	2	0
6	1	0	1	0	23	**6**	0	0	0	12	1	20	4	0
7	3	0	10	0	6	0	**74**	0	4	27	2	10	3	2
8	0	0	0	0	0	0	0	**89**	0	0	0	8	0	0
9	4	0	0	0	6	1	3	0	**57**	29	3	24	0	6
10	7	1	5	1	29	2	14	0	3	**274**	0	79	16	0
11	1	0	1	0	0	0	3	0	1	12	**88**	1	0	1
12	0	0	0	0	7	0	0	21	0	1	0	**173**	0	0
13	0	0	0	0	42	0	0	0	0	3	0	28	**8**	0
14	1	1	1	0	3	0	0	0	8	21	1	28	0	**14**

For this combination method we used the best performing classifier for global features (SVM) and for local features (NPD). The confusion matrices for the component classifiers are shown in Tables 1 and 2. The groupings have been constructed by iteratively merging two classes A and B, such that the percentage of instances of A classified as B is the highest. This process stops, when the percentage falls below a threshold.

The merging procedure on the VPR data set resulted in the creation of two super-classes, consisting of 2 and 4 classes, respectively. The accuracy of the SVM classifier on the resulting 10-class problem increased to 69%, and the overall accuracy increased to 60%.

5 Application Domain

In this paper we are attempting to classify gray-scale images of zooplankton acquired by the Video Plankton Recorder (VPR) [2]. The VPR consists of a single video camera and synchronized strobe that images the contents of a small volume of water at a rate of 60 Hz. In this study the camera imaged 5.1 ml per image, with a field of view of 17.5 mm wide x 11.7 mm tall x 25 mm deep. Images were transmitted to the surface via fiber-optic cable where time-code from a GPS system was added and data were archived on S-VHS videotape. In the lab, the video signal was routed through a PC-based image processing system (Imaging Technologies) that digitized each image and located objects meeting user-defined criteria for size, brightness, and focus. Objects meeting these criteria (termed regions of interest or ROIs) were cropped and written to disk as individual TIFF files. A subset of images was manually classified into zooplankton categories that ranged from individual species to broader groups, depending on how many taxonomic classification features were present.

One issue with the data produced by the VPR is that the video frames are interlaced. Since the camera system is typically towed at over 4 m/s, each video field represents a complete scene. When ROIs are extracted by the image processor, each image has only half of its horizontal scan lines (either odd or even). Therefore the images have to be interpolated to recover their proper aspect ratio.

The images we used to test our classification techniques show zooplankton belonging to 13 categories and one phytoplankton category (Table 3). Microscopic plants (phytoplankton) are often radially-symmetrical, and may appear similar when viewed from different angles. On the other hand, the animals in our images are 3D objects, whose appearance very much depends on their orientation relative to the camera. Most of the animals also have various appendages that may be extended or retracted, and posses articulated exoskeletons that can twist and bend, resulting in many degrees of freedom of motion. In other words, they are capable of a wide range of articulated motion, resulting in great variety of possible appearances. These facts make the task of classifying these images very challenging.

Table 3: Taxonomic Categories of the VPR Data Set

Category Name	Taxonomic Group	# images
Calanus finmarchicus	copepod species	132
Chaetognaths	zooplankton phylum	86
Conchoecia Ostracods	ostracod genus	100
Ctenophores	zooplankton phylum	34
Euphausiids	zooplankton order	131
Hyperiid Amphipods	zooplankton suborder	68
Pteropods	zooplankton order	142
Diatom Rods	phytoplankton class	97
Larvaceans	zooplankton class	133
Small Copepods	zooplankton class	433
Unidentified Cladocerans	zooplankton order	108
Siphonophores	zooplankton suborder	202
Euchaeta norvegica	copepod species	81
Siphonulae	developmental stage of zooplankton suborder	78

6 Results

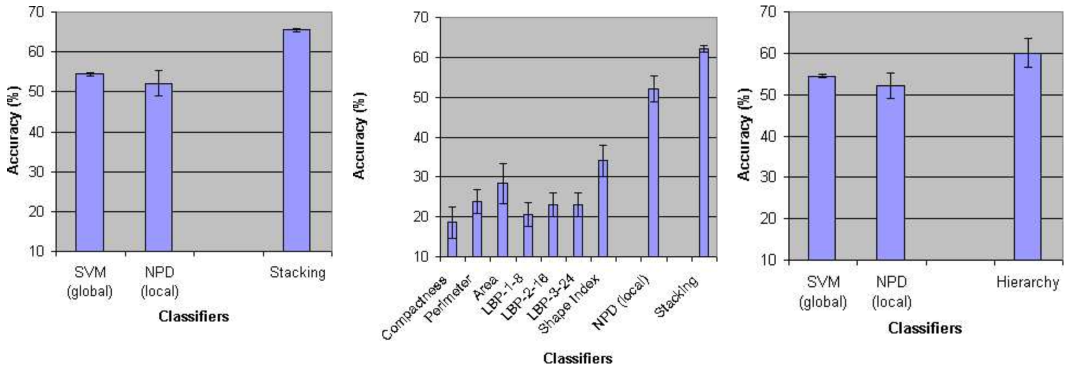
The results of combining local and global features with stacking, as described in Section 4.1, are shown in Figures 4(a) and 4(b). In each figure, the accuracy of the component classifiers is displayed followed by the accuracy of the stacking technique.

In Figure 4(a), two component classifiers were used, increasing the accuracy to 65.5%. The first component is an SVM classifier trained on global features. We used the SVM implementation included in the Weka toolkit [21]. The classifier output was converted into a vector with a value of 1 assigned to the predicted label, and a value of 0 to all the others. The second classifier was NPD with local features.

In Figure 4(b), 8 NPD component classifiers were used, increasing the accuracy to 62%. One of them was trained with local features, while the rest used global features. The global features (described in Section 2) were compactness, perimeter, area, three kinds of local binary patterns (LBP), and shape index. The local binary pattern features used pixels sampled at a radius of 1, 2, and 3, and sample sizes of 8, 16, and 24, respectively.

In both cases posterior distributions for all images in the VPR data set were produced by the component classifiers using 10-fold cross-validation. The mean accuracy and the standard error for the NPD classifiers were computed using the results of the folds. The mean accuracy of the SVM component classifier was computed by running 10-fold cross-validation 10 times with different permutations of the data. This resulted in much tighter standard error bounds. The resulting distributions were concatenated to form meta-features, which were used to train and test a meta-classifier (SVM) again using 10-fold cross-validation. The mean accuracy and the standard error were computed the same way as for the component SVM classifier.

Figure 4(c) shows the results of combining local and global features using the hierarchy of classifiers (Section 4.2). At the top level of the hierarchy two super-classes were created, one merging classes 4 (Ctenophores) and 12 (Siphonophores), and the other containing classes 3 (*Conchoecia* Ostracods), 7 (Pteropods), 10 (Small Copepods), and 11 (Unidentified Cladocerans). The number of categories at the top level has been reduced to 10. An SVM global feature classifier is trained on these new categories yielding an accuracy of 69%. When the local feature classifier is applied to the super-classes, an overall accuracy of 60% is achieved.



(a) Stacking using a SVM global feature classifier, and a NPD local feature classifier.

(b) Stacking using many NPD global feature classifiers, and a NPD local feature classifier

(c) Hierarchical classification using SVM global feature classifier, and a NPD local feature classifier.

Figure 4: Classification results.

7 Conclusions and Future Work

In this paper, we have presented two methods for combining local and global features. We argue that global and local features should be used for recognition in applications where an object detector is available. We have shown, through experimental results, that combining these two types of features is reduces error by more than 20%. Although the local and global feature sets used in this experiment both largely describe texture we have nevertheless shown that both provide different kinds of information about the image. We expect that classification accuracy would increase further if we were able to add more shape descriptors or contour features. We have also presented a novel method for classification using local features that has outperformed several image matching methods. This method is able to generalize for the entire class and thus is capable of partially overcoming the high in-class variability present in our data set. In future work we plan to extend the hierarchal classification model and introduce a new combination method through the use of kernels.

We plan to extend the classification hierarchy approach to be more flexible with respect to the class groupings. Using unsupervised clustering and the entropy of each cluster is a promising technique for revealing the confused classes in global feature space. By training local feature classifiers on the classes that belong to clusters with low entropy, we allow the individual classes to be present in more than one superclass, which gives the model more flexibility. Since the clusters formed would be easily separable in global feature space, we obtain an accurate top level classifier. We also plan to analyze the effect of a top level global feature classifier vs a top level local feature classifier.

Several recent papers have suggested approaches to tie image matching with local features to the support vector framework [7][10][19]. The support vector framework is a general method for classification derived from inner products over feature vectors [15]. A key feature of the SVM framework is that it allows for the replacement of strict inner products in the original feature space with *Mercer kernels*, functions that are equivalent to inner products between projections of the original vector into a, usually higher dimensional, feature space. Though the data may not be well separated in the lower dimensional space, their projection into higher dimensions may be. Positively weighted linear combinations of Mercer kernels are Mercer kernels themselves. We plan to develop a new SVM kernel consisting of a linear combination of a kernel to match local features and a kernel applied to a global image descriptor.

Acknowledgments

This material is based upon work supported in part by the National Science Foundation under Grants ATM-0325167 and IIS-0100851. The first author is supported by a fellowship from Eastman-Kodak Company Research Labs.

References

[1] M. Blaschko, G. Holness, M. Mattar, D. Lisin, P. Utgoff, A. Hanson, H. Schultz, E. Riseman, M. Sieracki, W. Balch, and B. Tupper. Automatic in situ identification of plankton. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2005.

[2] C. S. Davis, S. M. Gallager, M. S. Berman, L. R. Haury, and J. R. Strickler. The video plankton recorder (VPR): design and initial results. *Archiv für Hydrobiologie Beiheft Ergebnisse der Limnologie*, 36:67–81, 1992.

[3] C. S. Davis, Q. Hu, S. M. Gallager, X. Tang, and C. J. Ashjian. Real-time observation of taxa-specific plankton distributions: an optical sampling method. *Marine Ecology Progress Series*, 284:77–96, 2004.

[4] 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.

[5] T. Dietterich. Ensemble methods in machine learning. In *1st Intl. Workshop on Multiple Classifier Systems*, 2000.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc, 2001.

[7] J. Eichhorn and O. Chapelle. Object categorization with SVM: kernels for local features. Technical report, Max Plank Institute for Biological Cybernetics, Tübingen Germany, 2004.

[8] S. Helmer and D. G. Lowe. Object recognition with many local features. In *Workshop on Generative Model Based Vision*, 2004.

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

[10] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Intl. Conf. on Machine Learning*, 2003.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[12] M. A. Mattar, A. R. Hanson, and E. G. Learned-Miller. Automatic sign classification for the visually impaired. Technical Report UM-CS-2005-014, UMass Amherst, 2005.

[13] 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.

[14] S. Ravela. *On Multi-Scale Differential Features and their Representations for Image Retrieval and Recognition*. PhD thesis, UMASS Amherst, 2002.

[15] B. Schölkopf and A. Smola. *Learning from Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2001.

[16] A. K. Seewald. *Towards Understanding Stacking - Studies of a General Ensemble Learning Scheme*. PhD thesis, Austrian Research Institute for Artificial Intelligence (FAI), 2003.

[17] M. Shapiro and M. Blaschko. Stability of Hausdorff-based distance measures. In *Proc. of IASTED Visualization, Imaging, and Image Processing*, 2004.

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

[19] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision*, 2003.

[20] J. Weinman, A. Hanson, and A. McCallum. Sign detection in natural images with conditional random fields. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 549–558, 2004.

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

[22] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE PAMI*, 24(1), 2002.