**BMC Bioinformatics**

BioMed Central

Research article

Open Access

# Boosting accuracy of automated classification of fluorescence microscope images for location proteomics

Kai Huang[1,3] and Robert F Murphy*[1,2,3]

Address: [1]Department of Biological Sciences, Carnegie Mellon University,4400 Fifth Avenue, Pittsburgh, PA 15213 USA, [2]Department of Biomedical Engineering, Carnegie Mellon University,4400 Fifth Avenue, Pittsburgh, PA 15213 USA and [3]Center for Automated Learning and Discovery, Carnegie Mellon University,4400 Fifth Avenue, Pittsburgh, PA 15213 USA

Email: Kai Huang - khuang@andrew.cmu.edu; Robert F Murphy* - murphy@cmu.edu

* Corresponding author

Published: 18 June 2004

*BMC Bioinformatics* 2004, **5**:78 doi:10.1186/1471-2105-5-78

Received: 26 January 2004

Accepted: 18 June 2004

This article is available from: http://www.biomedcentral.com/1471-2105/5/78

## Abstract

**Background:** Detailed knowledge of the subcellular location of each expressed protein is critical to a full understanding of its function. Fluorescence microscopy, in combination with methods for fluorescent tagging, is the most suitable current method for proteome-wide determination of subcellular location. Previous work has shown that neural network classifiers can distinguish all major protein subcellular location patterns in both 2D and 3D fluorescence microscope images. Building on these results, we evaluate here new classifiers and features to improve the recognition of protein subcellular location patterns in both 2D and 3D fluorescence microscope images.

**Results:** We report here a thorough comparison of the performance on this problem of eight different state-of-the-art classification methods, including neural networks, support vector machines with linear, polynomial, radial basis, and exponential radial basis kernel functions, and ensemble methods such as AdaBoost, Bagging, and Mixtures-of-Experts. Ten-fold cross validation was used to evaluate each classifier with various parameters on different Subcellular Location Feature sets representing both 2D and 3D fluorescence microscope images, including new feature sets incorporating features derived from Gabor and Daubechies wavelet transforms. After optimal parameters were chosen for each of the eight classifiers, optimal majority-voting ensemble classifiers were formed for each feature set. Comparison of results for each image for all eight classifiers permits estimation of the lower bound classification error rate for each subcellular pattern, which we interpret to reflect the fraction of cells whose patterns are distorted by mitosis, cell death or acquisition errors. Overall, we obtained statistically significant improvements in classification accuracy over the best previously published results, with the overall error rate being reduced by one-third to one-half and with the average accuracy for single 2D images being higher than 90% for the first time. In particular, the classification accuracy for the easily confused endomembrane compartments (endoplasmic reticulum, Golgi, endosomes, lysosomes) was improved by 5–15%. We achieved further improvements when classification was conducted on image sets rather than on individual cell images.

**Conclusions:** The availability of accurate, fast, automated classification systems for protein location patterns in conjunction with high throughput fluorescence microscope imaging techniques enables a new subfield of proteomics, location proteomics. The accuracy and sensitivity of this approach represents an important alternative to low-resolution assignments by curation or sequence-based prediction.

## Background

High-throughput methods for proteomics have become necessary in order to characterize the tens of thousands of proteins expressed in each multicellular organism. Although high-throughput methods for structural proteomics [1] and expression proteomics [2] are available and a number of large-scale projects have been initiated to make use of them, far less attention has been paid to automating the determination of the subcellular locations of proteins.

As an alternative to experimental determination, various approaches to predicting subcellular location from protein sequences have been described. These have used amino acid composition [3,4], N-terminal signal peptide sequences [5], functional domain composition [6,7], and combined evidence from sequence properties and expression levels [8]. The limitation in these approaches to date has been the absence of a systematic, high-resolution framework for describing subcellular location. That is, since only a small number (4–12) of broad location classes have been used to train these systems, they cannot distinguish the many different distributions that proteins can have within a single organelle (or the many combinations of these distributions that may occur). The performance of these approaches cannot be significantly improved unless a large number of subcellular location patterns are determined experimentally.

The spatial distribution of each protein in a cell type can be detected by various methods for fluorescently tagging a protein followed by fluorescence microscope imaging. In recent years, automated classification systems that can be used to assign proteins to the major subcellular location classes have been described [9-11]. The combination of such methods with high throughput imaging systems [12] holds enormous promise for automating the systematic determination of this critical aspect of protein function. The information that can be obtained from such systems can potentially be used to train improved systems for predicting location from primary sequence.

High-throughput automated recognition of protein subcellular distributions requires both robust image features and accurate classifiers. Several different image feature sets have been derived for 2D and 3D fluorescence microscope images [9,11,13,14], which capture information such as texture, object morphology and geometrical properties. Advanced feature reduction techniques have been used to select the most discriminative features from these feature sets [15]. Ten major subcellular location patterns in 2D images and eleven in 3D images, including two that cannot be discriminated by visual inspection, can be distinguished using these features.

Our group has considered two variations on this recognition problem, one in which only gray level images of the distribution of a single protein are available, and the other in which a parallel image of the DNA distribution in the same cell is also available. Our initial work made use of an artificial neural network for classification. With this classifier, we have achieved 86% average classification accuracy on 2D HeLa cell images without parallel DNA images and 88% accuracy with them [14]. For 3D HeLa cell images, we have achieved 91% average accuracy for the same patterns [11]. The goal of the work described here was to improve upon these results using state-of-the-art classifiers and new features. The classifiers evaluated in this paper are described briefly below.

### *Support Vector Machines (SVM)*

Support Vector Machines [16] are linear classifiers that find a hyperplane between two classes which maximizes the minimum distance between the hyperplane and the data points (the maximum margin hyperplane). The hyperplane can be sparsely represented by a small amount of data lying on the boundary of the maximum margin. Theoretically, maximum-margin classifiers can minimize the classification structural risk, the upper bound on the error expected for a test set [17].

SVM were first characterized in a two-class learning environment. However, they can be extended to solve multi-class learning problems by using one of three methods.

- Max-win classification [17] trains $N$ support vector machines, each of which separates class $i$ from non-$i$. The predicted class from the machine generating the highest output score is selected as the prediction.

- Pair-wise classification [18] trains a separate binary classifier for distinguishing each possible pair of classes. Each of these binary classifiers is given a vote and the class with the most votes is selected as the predicted class.

- DAG (Directed Acyclic Graph) classification [19] puts the binary classifiers trained in the pair-wise classification method into nodes of a rooted binary DAG. The output of each SVM is used to trace the graph until a leaf node that specifies the final prediction is reached. The rationale behind this method is that achieving a maximum margin classifier at each node of the classifier graph guarantees an upper bound on the generalization error [19].

### *Ensemble Methods*

All current machine learning models have some constraints or local minima in certain application domains given limited training data [20], resulting in the fact that different learning models might perform better for different application domains. Therefore it is useful to build an ensemble expert that can combine the outputs from different classifiers in a way such that the local limitations of each individual model can be overcome in the ensemble by other models that do not have the same constraints. In practice, ensemble methods have been shown to perform better than any of the base classifiers if the output errors of each base classifier are not fully dependent [20-23]. However, it is still an open question whether an ensemble model consisting of base classifiers with independent errors performs better than one consisting of base classifiers with partially dependent errors [24]. Many different ways of creating ensemble classifiers have been described [20,25] and we summarize some of these approaches here.

### *AdaBoost*

AdaBoost is an example of ensemble methods that create new and improved base classifiers by iteratively manipulating the training dataset [26]. It generates base classifiers, such as a neural network or decision tree, at each training iteration. Every newly created base classifier is fed an adjusted distribution in which each data point in the training set is assigned a weight computed from its classification result for the previous base classifier. The rationale behind Adaboost is to force a new base classifier to focus on the data points that were incorrectly classified previously such that those hard examples can be better classified in the new base classifier [26].

### *Bagging*

Bagging, which stands for bootstrap aggregation, is another way of manipulating training data for ensemble methods. In each bagging iteration, the training examples are bootstrapped (resampled with replacement), to generate a different training set for the base classifier [20]. On average, 63.2% of the total original training examples are retained in each bootstrapped set [20]. The rational behind bagging is that unstable base classifiers, such as neural networks and decision trees whose behavior could be significantly changed by small fluctuations in the training dataset, are more likely to be stabilized after being trained with different input data and combined afterwards [20,25].

### *Mixtures-of-Experts*

An ensemble method can also use a divide-and-conquer strategy to separate the training examples into many partitions, train a local base classifier (an expert) for each partition, and then combine the outputs of these local experts in a supervised-learning way [25,27]. Mixtures-of-Experts [27-29] is one kind of these ensemble methods.

Mixtures-of-Experts models the target data generation process as [27]:

$$P(Y \mid X) = \sum_{z} P(Z \mid X) P(Y \mid X, Z)$$

where $Y$ stands for the target data, $X$ is the input data, and $Z$ is a representation of hidden experts that relate an input data point to a target. The target generation process can be regarded as two steps [27,28]: one is to assign every input data point to a specific expert and the other is to compute the target data given the input data point and its related expert. A gating network is used in Mixtures-of-Experts to model P(Z | X), the first step above, and local expert networks are used to model P(Y | X, Z), which is the second step above [27].

### *Majority-voting Ensembles*

The majority-voting ensemble classifier is one of the simplest ensemble forms that can combine the outputs of multiple classifiers. Instead of using the complicated schemes described above, we can simply choose the predicted class by plurality from a classifier pool [20,25]. Assuming that the errors made by the classifiers are not highly correlated, the samples that are not accurately classified by one classifier have a good chance to be correctly classified by a plurality of the other classifiers. The majority-voting ensemble classifier overcomes the difficulty of choosing one classifier from a pool of classifiers with similar performance. Different weighting schemes have been described to form the plurality prediction from the classifier pool [20,30].

## Results and Discussion

### *Subcellular Image Datasets*

In order to demonstrate the feasibility of recognizing the major subcellular structures in fluorescence microscope images of single cells, we have previously created large collections of 2D and 3D immunofluorescence images of HeLa cells [9,11]. The subcellular location patterns in these collections include endoplasmic reticulum (ER), the Golgi complex, lysosomes, mitochondria, nucleoli, actin microfilaments, endosomes, microtubules, and nuclear DNA. We designed sets of numerical features to describe the pattern in the images, which we term SLF (for Subcellular Location Features). When these features were used to train neural network classifiers, from 73–99% of the images could be correctly classified (depending on the pattern) [11,14]. Figure 1 depicts the most typical 2D and 3D image of each pattern from correctly classified images using these neural network classifiers. (The most typical image was selected using TypIC [31], an algorithm that ranks images by their distance to the centroid of the feature space.) The goal of the work described here was to improve on the previous performance.

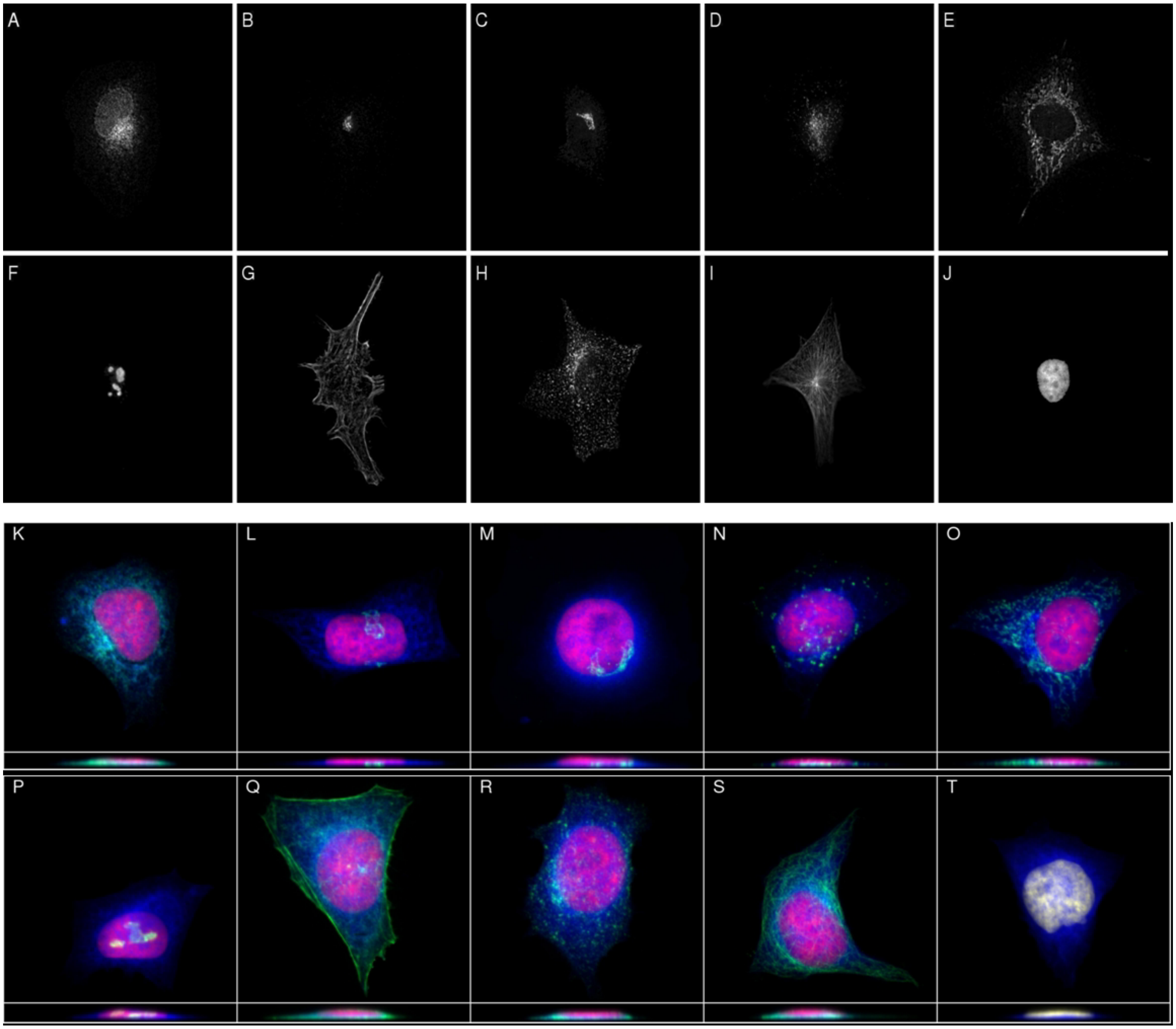

**Figure 1**
Representative images of each pattern from correctly classified images using previous neural network classifiers. Ten patterns from the 2D/3D HeLa cell image collection are depicted: endoplasmic reticulum (A/K), giantin (B/L), gpp130 (C/M), LAMP2 (D/N), mitochondria (E/O), nucleolin (F/P), actin (G/Q), transferrin receptor (H/R), tubulin (I/S), and DNA (J/T). Each false color in the 3D images represents the fluorescence intensity from labeling the target protein (green), total DNA (red), and total protein (blue). Projections that are summed upon the Z or Y axis are shown. The feature sets SLF13 (2D) and SLF10 (3D) were used both for classification and for choosing a typical image.

### *Comparison of Classifier Characteristics*

Our first goal was to measure the performance of the different classifiers described above with the SLF that we have used previously. Before embarking on this, we here try to provide some insight into the differences between the classifiers. For this purpose, we can describe them using three characteristics. The first characteristic is the complexity of the decision boundaries that a classifier can generate. Figure 2 illustrates differences in this characteristic for a pair of Golgi proteins, giantin and gpp130, that are difficult to distinguish computationally (and essentially impossible to distinguish visually [14]). There is

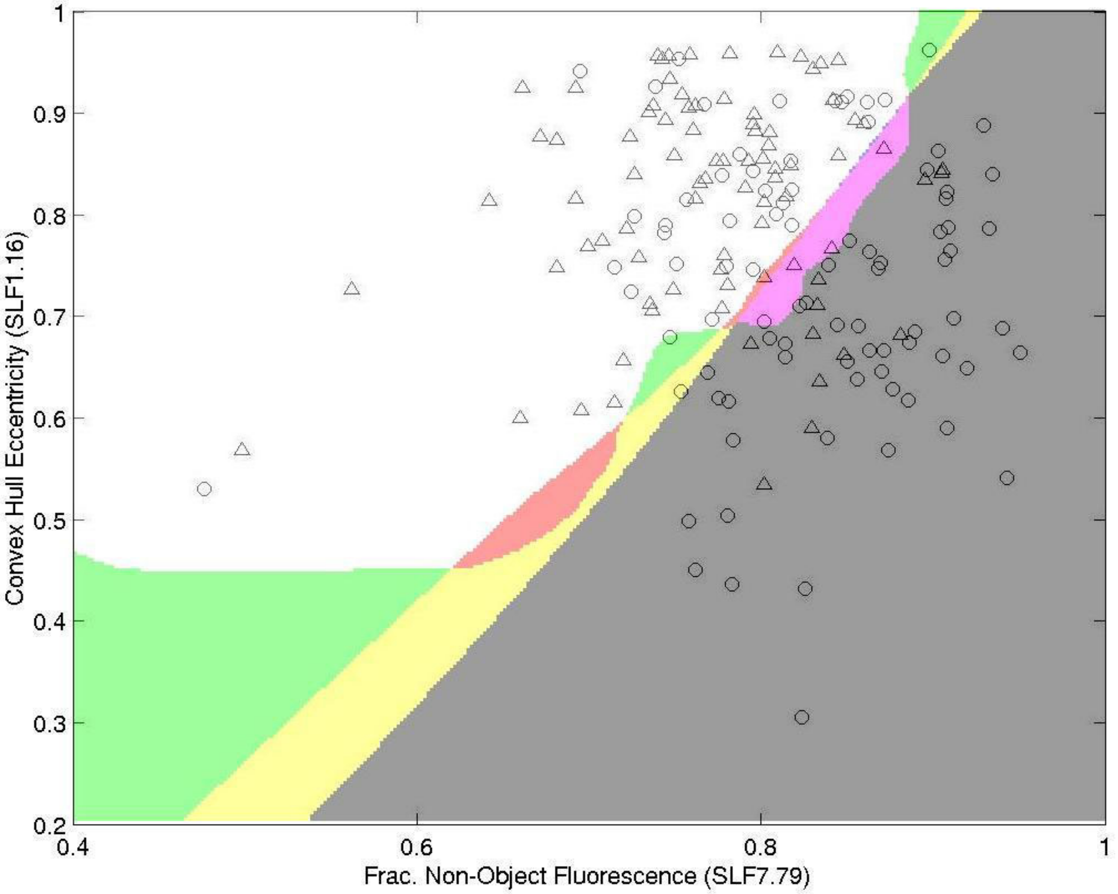

**Figure 2**
Decision boundaries of various classifiers for distinguishing the patterns of two Golgi proteins. A scatterplot of the two most informative features for distinguishing images of giantin (circle) and gpp130 (triangle). These were chosen from SLF7DNA by SDA. Various classifiers were trained using just these features and maps of the class assigned to various points in the feature plane by each trained classifier were created (a zero (white) pixel corresponded to gpp130). A false color map was created by combining the decision maps for exponential-rbf kernel SVM (green), the majority-voting ensemble classifier (red), and neural networks (blue).

significant overlap in the feature distributions for these classes but for illustration we selected the most discriminative pair of features for the two Golgi patterns using stepwise discriminant analysis (SDA) from the combination of SLF7 and the 6 DNA features (SLF7DNA). These were the fraction of fluorescence not in any object (SLF7.79) and the convex hull eccentricity (SLF1.16). The values of these features for each image are plotted in Figure 2 along with colored regions illustrating the boundary between the two classes found by three different classifiers. While the boundaries for the neural network and mixture-of-experts classifiers are nearly linear, the boundary for the exponential-rbf-kernel SVM is much more convoluted. The accuracy of the classifiers increases with the extent of curvature, from 68.6% for the neural network, to 71.5% for the mixture-of-experts, to 75% for the exponential-rbf-kernel SVM.

A second classifier characteristic is the dependence of performance on the size of the training set. This can be divided into two parts: ability to learn from limited training data, and insensitivity to the presence of outliers when training data is plentiful. Both of these can only be described in relative terms by comparing the performance

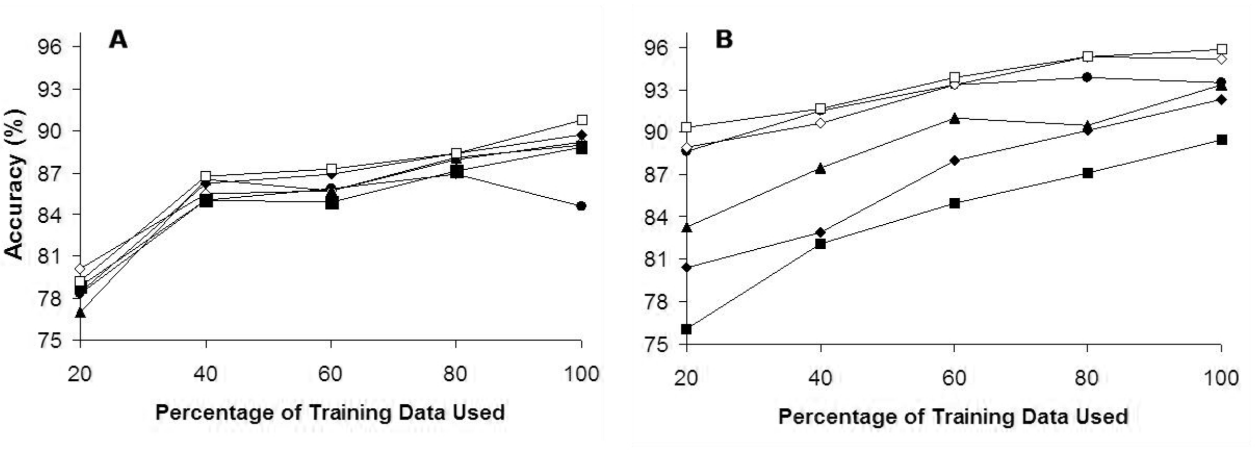

**Figure 3**
Dependence of classifier performance on amount of training data. The average performances of neural network (filled circle), SVM (open diamond), AdaBoost (filled triangle), Bagging (filled square), Mixtures-of-Experts (filled diamond), and majority-voting ensemble (open square) classifiers are shown as a function of the amount of training data given to the classifier. Average performance is defined as the average fraction of images in ten (2D) or eleven (3D) classes that were correctly classified over ten cross-validation trials. A) Results for 2D images using feature set SLF13. B) Results for 3D images using feature set SLF10.

of different classifiers on the same data. Figure 3 shows a comparison of the major classifiers for both 2D and 3D images as a function of the amount of training data provided. For low numbers of training images, performance is similar for all 2D classifiers but varies significantly for the 3D classifiers. In particular, bagging does poorly for low numbers of training images while the majority-voting ensemble does quite well. As the amount of training data is increased, all but one of the classifiers improve their accuracy monotonically. The exception is the neural network classifier, for which the accuracy decreases. We presume that this is due to a heightened chance of including outlier images as the size of the training set increases.

The third characteristic is the sensitivity of performance to the presence of uninformative features. This is illustrated in Figure 4, in which classification accuracy is displayed as a function of increasing numbers of features. For this purpose, the features in SLF13 and SLF10 were ranked in order of their discriminative ability by SDA. Classifiers were trained using increasing numbers of the sorted features, and when these were exhausted additional features (which can be regarded as noisy or less informative features) were randomly included from the 90-dimensional SLF7DNA and 28-dimensional SLF9. For the 2D images (Figure 4A), all classifiers except neural networks handled the increasing number of input features with little impairment from uninformative features. The decrease in accuracy for neural networks at high numbers of features, along with the large fluctuations in their performance, implies that feature selection is essential for neural network classification in the SLF7DNA feature space. For 3D images (Figure 4B), neural networks performed the best and the SVM classifier showed a significant drop in accuracy as the number of features increased. When compared to Figure 4A, the stability of neural networks as well as the instability of SVM in the SLF9 feature space implies that the ability of a classifier to adapt to more noisy features depends on the feature space itself. As will be shown later, image set classification is much less dependent on the feature space.

Table 1 summarizes the above three characteristics for all major classifiers. It also shows the information content (the number of parameters times the bits required for each parameter) of the model that each classifier builds from the training data. We note that the classifier with the lowest information content, the neural network, is also the most effective classifier at learning from limited training data.

### *Evaluation of Eight Classifiers*

With this background, we proceeded to evaluate the different classifiers using two different feature sets for 2D HeLa images, namely SLF8 and SLF13, and two different feature sets for 3D HeLa images, namely SLF10 and SLF14. Two feature sets were used for each image collection so that performance with and without DNA features could be compared. Eight classifiers, including a one-hidden-layer neural network, linear-kernel SVM, rbf-kernel SVM,

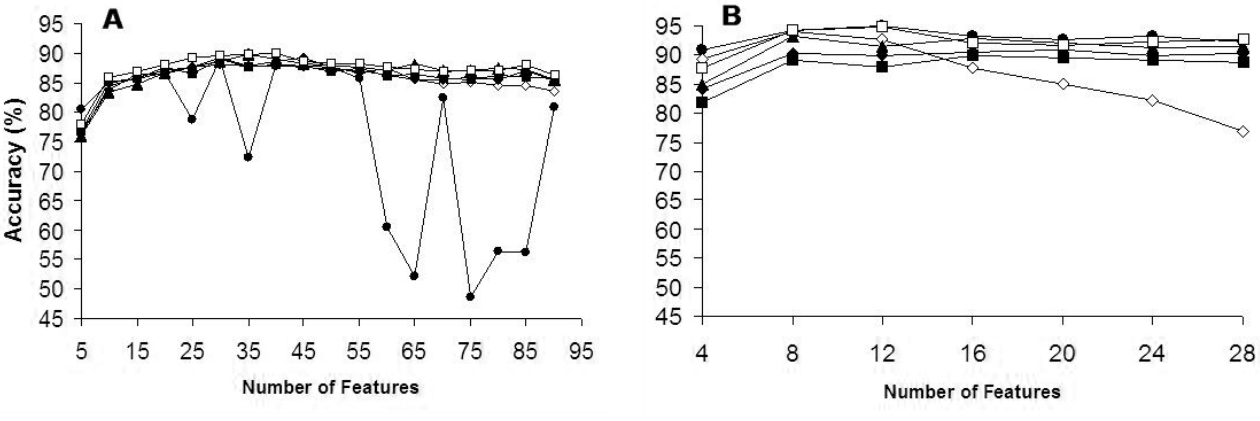

**Figure 4**
Dependence of classifier performance on number of input features. The average performances of neural network (filled circle), SVM (open diamond), AdaBoost (filled triangle), Bagging (filled square), Mixtures-of-Experts (filled diamond), and majority-voting ensemble (open square) classifiers are shown as a function of the number of features used to train the classifier. Average performance is defined as the average fraction of images in ten (2D) or eleven (3D) classes that were correctly classified over ten cross-validation trials. The features in SLF7DNA (A) or SLF9 (B) were ranked in order of their ability to discriminate the classes using SDA and increasing numbers of the features were used to train classifiers.

**Table 1: Comparison of the characteristics of the classifiers used in this study. The results are derived from the data in Figures 2-4.**

| Classifier | Ability to generate nonlinear decision boundary | Ability to learn well from limited training data | Insensitivity to outliers in training data | Insensitivity to uninformative features | Log information content* (2D/3D) |
|---|---|---|---|---|---|
| Neural Networks | Low | High | Medium | Medium | 10.0/10.0 |
| Exponential-rbf-kernel SVM | High | High | High | Low | 14.2/13.9 |
| AdaBoost | Medium | Med | High | High | 13.5/13.4 |
| Bagging | Medium | Low | High | High | 14.8/12.0 |
| Mixtures-of-Experts | Medium | Low | High | High | 13.5/13.5 |
| Majority-voting Ensemble | Medium | High | High | High | 14.7/14.6 |

* The natural logarithmic of the information content was calculated as described in the Methods section for the feature set SLF13 (2D) and SLF10 (3D). The classifiers were configured as detailed in Table 2.

exponential-rbf-kernel SVM, polynomial-kernel SVM, AdaBoost, Bagging, and Mixtures-of-Experts, were evaluated by 10-fold cross validation.

Table 2 shows the results for each classifier with its optimal parameters for each of the four feature sets. To address the statistical significance of the experiments, we conducted a paired t-test between the 10-fold cross validation results of each of the eight classifiers and those for the neural network classifier topology we described previously. Previous results have shown that the inclusion of features calculated from a parallel DNA image can improve classification accuracy. The parallel DNA channel provides an additional landmark for locating protein subcellular distributions. For instance, a nucleolar protein would be expected to overlap completely with the DNA channel, while a mitochondrial protein would be localized around but not in the nucleus. As expected, both SLF13 and SLF10 performed better than their no-DNA counterparts SLF8 and SLF14, respectively. However, the benefit from the DNA features was much larger for 3D image classification (about 6%) than for 2D image classification (about 1%). The reason for this is unclear, but could be due either to the greater amount of information present in 3D images, or to the presence of redundancy between the non-DNA and DNA features for 2D but not 3D images.

**Table 2: Comparison of eight classifiers for 2D and 3D image classification. The average classification accuracy on test data (from 10-fold cross-validation) is shown for the optimal parameters settings (shown in parentheses) for each classification approach. The parameters are: nhu – number of hidden nodes in neural network, stop-fract – the fraction of the training data used to stop neural network training, C – error penalty in SVMs, sigma – kernel variance in SVMs, nboost – total number of iterations in AdaBoost, nbag – total number of iterations in Bagging, nhug – number of hidden nodes in the gating network of Mixtures-of-Experts, and ne – total number of experts in Mixtures-of-Experts. The accuracies across the 10-fold cross-validation trials were compared to those for the previously described neural network configuration (nhu = 20, stop-fract = 0.3) by a paired t-test (88% for SLF13, 86% for SLF8, 93% for SLF10, and 84% for SLF14). The best performances are underscored and highlighted for each feature set. *CPU times listed for each classifier are for training and testing for all images in each cross-validation trial (training times include times calculating features), which were measured on an Athlon 1.7 GHz processor with 1.5 GB memory running Redhat Linux 7.1.**

| Feature Set | Classifier | Classification accuracy (%) | Average training time* (s) | Average testing time* (s) | P-value |
|---|---|---|---|---|---|
| SLF13 (2D DNA) | Neural Network (nhu = 16, stop-fract = 0.1) | 87.8 | 116.3 | **0.001** | 0.43 |
| | SVM (linear, DAG, C = 1) | 87.9 | **0.7** | 0.088 | 0.36 |
| | SVM (rbf, DAG, sigma = 8, C = 16) | 89.4 | 1.1 | 0.470 | **0.03** |
| | SVM (exprbf, maxwin, sigma = 4, C = 4) | 89.2 | 3.5 | 0.530 | **0.04** |
| | SVM (poly, maxwin, degree = 2, C = 0.01) | 88.6 | 4.7 | 0.140 | 0.21 |
| | Adaboost (nhu = 8, nboost = 64) | 88.9 | 55.2 | 0.018 | 0.10 |
| | Bagging (nhu = 64, nbag = 32) | 88.9 | 111.0 | 0.078 | 0.09 |
| | Mixtures-of-Experts (nhu = 16, nhug = 64, ne = 16) | **89.7** | 38.3 | 0.010 | **0.02** |
| SLF8 (2D) | Neural Network (nhu = 16, stop-fract = 0.3) | 86.1 | 139.1 | **0.001** | 0.53 |
| | SVM (linear, DAG, C = 1) | 84.9 | **0.7** | 0.075 | 0.83 |
| | SVM (rbf, maxwin, sigma = 8, C = 64) | 87.9 | 11.4 | 1.600 | 0.15 |
| | SVM (exprbf, maxwin, sigma = 8, C = 16) | 88.1 | 4.0 | 0.540 | **0.02** |
| | SVM (poly, maxwin, degree = 2, C = 0.01) | 86.7 | 5.2 | 0.170 | 0.37 |
| | Adaboost (nhu = 32, nboost = 128) | **88.2** | 412.0 | 0.190 | 0.12 |
| | Bagging (nhu = 64, nbag = 64) | 87.2 | 238.2 | 0.160 | 0.17 |
| | Mixtures-of-Experts (nhu = 32, nhug = 16, ne = 4) | 87.0 | 11.6 | 0.002 | 0.22 |
| SLF10 (3D DNA) | Neural Network (nhu = 32, stop-fract = 0.1) | **95.3** | 740.3 | **0.001** | 0.06 |
| | SVM (linear, DAG, C = 8) | 93.3 | **0.3** | 0.043 | 0.47 |
| | SVM (rbf, maxwin, sigma = 2, C = 64) | 95.0 | 2.3 | 0.230 | 0.08 |
| | SVM (exprbf, DAG, sigma = 1, C = 1) | 95.2 | 0.5 | 0.081 | 0.06 |
| | SVM (poly, maxwin, degree = 2, C = 1) | 93.1 | 2.0 | 0.067 | 0.51 |
| | Adaboost (nhu = 32, nboost = 32) | 93.2 | 43.2 | 0.016 | 0.46 |
| | Bagging (nhu = 64, nbag = 4) | 89.4 | 6.8 | 0.003 | 0.99 |
| | Mixtures-of-Experts (nhu = 32, nhug = 64, ne = 16) | 92.2 | 45.8 | 0.007 | 0.74 |
| SLF14 (3D) | Neural Network (nhu = 32, stop-fract = 0) | 88.4 | 172.0 | 0.001 | **0.02** |
| | SVM (linear, DAG, C = 32) | 86.5 | **1.0** | 0.047 | 0.12 |
| | SVM (rbf, maxwin, sigma = 2, C = 32) | 86.6 | 4.6 | 0.290 | 0.17 |
| | SVM (exprbf, maxwin, sigma = 2, C = 8) | **89.1** | 1.4 | 0.170 | 0.05 |
| | SVM (poly, maxwin, degree = 2, C = 2) | 87.3 | 8.3 | 0.068 | 0.05 |
| | Adaboost (nhu = 64, nboost = 64) | 87.7 | 144.3 | 0.085 | **0.03** |
| | Bagging (nhu = 64, nbag = 256) | 82.2 | 505.7 | 0.340 | 0.82 |
| | Mixtures-of-Experts (nhu = 16, nhug = 8, ne = 2) | 83.8 | 2.9 | **0.001** | 0.59 |

The various parameters selected for each classifier on different feature sets suggest that the right configuration of a classifier is highly dependent on the input data. Given limited data, a classifier can only be optimized to an extent that its performance is maximized on this specific set of data. However, we can still observe some common trends in parameter selection. The pairwise method for combining binary SVMs was never selected in our experiments, although the pairwise method was reported to give good performance in some application domains. There is an assumption in this method that the binary classifier should generate an unknown prediction for classes that it cannot recognize. In other words, a binary classifier trained between classes *a* and *b* should give equal weights to both *a* and *b* given a data point from a third class *c*. However, this assumption seemed not to hold in our experiments. None of the pairwise methods gave an average accuracy over 20% for any of the four feature sets (data not shown). On the contrary, both maxwin and DAG methods gave much better results and it is hard to tell which one of the two was more advantageous. The close performance of linear-kernel SVMs with other nonlinear-kernel SVMs in most experiments suggested that the classification boundaries in the original input spaces of all four feature sets were not far from linear. This can be confirmed by the optimal choices of the polynomial degrees in the polynomial-kernel SVMs. A polynomial degree of 2 was selected for all four feature sets. The smaller the degree, the more linear the final classification boundary would be. SVMs, ranked in the top two for all four feature sets, performed generally very well among all classifiers in both 2D and 3D classifications. Ensemble methods displayed more varieties in their performances. Although two ensemble methods, Mixtures-of-Experts and AdaBoost, performed the best on the feature set SLF13 and SLF8 respectively, these methods did not perform well in 3D classifications. Among the ensemble methods, AdaBoost showed the most consistent performance.

Table 2 also shows cpu times for training and testing each classifier. Although the training and testing time of a classifier can be affected by the number of features, the data itself, and the implementation details, we observe some trends from the experimental results. As expected, the more hidden nodes in a neural network, the longer it will take to train, which can be observed in neural networks, Adaboost, Bagging, and Mixtures-of-Experts. It is also obvious that longer iterations of boosting, bagging, and larger number of experts in Mixtures-of-Experts account for more time in training classifiers. SVMs are the fastest to train among all eight classifiers, although relatively slow in the test phase. Despite various changes in its parameters, the training time of SVM stays consistent for each kernel function. Neural network is the fastest classifier in the testing phase in three out of the four experiments conducted. Three ensemble methods also show faster performance in testing than SVMs, partially because the building blocks of these methods in the experiments are also neural networks.

To determine the significance of improvements in performance by individual classifiers over our previous neural network approach, we conducted a paired t-test [32], on the 10-fold cross validation results of each classifier against those of the previously configured neural networks. From Table 2, we can see that three, one, and two of the eight individually configured classifiers performed significantly better than the previous neural network classifiers on SLF13, SLF8, and SLF14 feature sets respectively. None of the eight classifiers could perform statistically better than the previous neural network classifier on SLF10. If a single classifier has to be selected for each of the four feature sets, a support vector machine classifier with exponential-rbf kernel function can be used given its reasonable accuracy and speed, although some fine tuning of its parameters might need to be conducted.

### *Optimal Majority-voting Ensembles of Eight Classifiers*

More improvement might be achieved by taking a majority-voting ensemble model of all possible combinations of the eight tested classifiers. Since all eight classifiers were trained differently and were based on either different kernel functions or different theoretical justifications, their error can be regarded as mostly independent, which makes constructing a larger ensemble model possible. Table 3 shows the statistics of pairwise-classifier-error correlation coefficients between all 8 classifiers on six different feature sets. As expected, all pairs of classifiers did not show strong error correlation. The mean error correlation coefficients from all six experiments were less than 0.15.

**Table 3: Analysis of error correlation between different classifiers. For each of the classifiers listed in Table 2, a list of all test images that were incorrectly classified was generated. The correlation coefficient between the incorrectly classified images was calculated for each pair of classifiers, and the minimum, maximum, mean and standard deviation of these correlations were calculated for all pairs. This process was repeated for each feature set.**

| Feature Set | Pairwise-Classifier-Error Correlation Coefficients | | | |
|---|---|---|---|---|
| | Min | Max | Mean | St. dev. |
| SLF8 (2D) | 0.01 | 0.22 | 0.09 | 0.06 |
| SLF15 (2D) | 0.00 | 0.22 | 0.10 | 0.06 |
| SLF13 (2D DNA) | 0.00 | 0.25 | 0.10 | 0.06 |
| SLF16 (2D DNA) | 0.00 | 0.26 | 0.10 | 0.07 |
| SLF14 (3D) | 0.00 | 0.25 | 0.11 | 0.07 |
| SLF10 (3D DNA) | 0.01 | 0.41 | 0.13 | 0.10 |

**Table 4: Improvement in classification accuracy using majority voting ensembles. Optimal unweighted majority-voting ensemble classifiers were formed by selecting classifiers from all 8 classifiers for each feature set listed and the average classification accuracy for 10-fold cross-validation was calculated. A paired-t test was performed for each ensemble classifier against the previous neural network classifier for each feature subset (SLF15 and SLF16 were compared against the previous classifier for SLF8 and SLF13, respectively). Each ensemble classifier was also compared against the optimal classifier for each feature set listed in Table 2 (SLF15 and SLF16 were compared with the individual optimal classifiers for SLF8 and SLF13, respectively).**

| Feature Set | Classifiers in the Optimal Majority-voting Ensemble | Average classification accuracy (%) | P-value of paired t test with previous results | P-value of paired t test with optimal single classifier | Classification Accuracy Upper Bound* (%) |
|---|---|---|---|---|---|
| SLF8 (2D) | Exprbf-kernel SVM<br>AdaBoost<br>Bagging | 89.4 | 0.003 | 0.08 | 95.5 |
| SLF15 (2D) | Rbf-kernel SVM<br>Exponential-rbf-kernel SVM<br>Polynomial-kernel SVM | 91.5 | 0.0006 | 0.01 | 96.1 |
| SLF13 (2D DNA) | Rbf-kernel SVM<br>AdaBoost<br>Mixtures-of-Experts | 90.7 | 0.003 | 0.03 | 95.6 |
| SLF16 (2D DNA) | Neural Network<br>Linear-kernel SVM<br>Exprbf-kernel SVM<br>Polynomial-kernel SVM<br>AdaBoost | 92.3 | 0.003 | 0.02 | 96.6 |
| SLF14 (3D) | Neural Network<br>Linear-kernel SVM<br>Exprbf-kernel SVM<br>Polynomial-kernel SVM<br>AdaBoost | 89.8 | 0.02 | 0.29 | 96.3 |
| SLF10 (3D DNA) | Linear-kernel SVM<br>Rbf-kernel SVM<br>Exprbf-kernel SVM<br>Mixtures-of-Experts | 95.8 | 0.02 | 0.35 | 98.2 |

* The upper bound of classification accuracy for a feature set is defined as the percentage of all images that could be correctly classified by at least one of the eight tested classifiers using that feature set.

There are many possible voting schemes that can be employed in a majority-voting ensemble. Some experiments have shown that a relatively simple method such as unweighted majority voting works as well as those more complicated trainable weighted voting methods [30]. Therefore, we constructed an unweighted majority-voting ensemble of all possible combinations of the 8 classifiers for each feature set. Table 4 shows the optimal majority-voting classifiers found for each feature set. The accuracies on both SLF8 and SLF13 feature sets were improved by 1% by combining three classifiers for each: exponential-rbf-kernel SVM, AdaBoost, and Bagging for SLF8; rbf-kernel SVM, AdaBoost, and Mixtures-of-Experts for SLF13. Less than 1% improvement, however, were observed on the SLF10 and SLF14 feature sets. Two of the top three classifiers for each feature set (Table 2) were selected in all optimal majority-voting ensembles. The same paired t-test was conducted between the majority-voting classifier and the previously configured neural network classifier for each feature set. Statistically significant improvements were obtained for all four feature sets. Compared to the individual optimal classifiers listed in Table 2 for each feature set, the majority-voting classifiers for SLF13 also showed statistically significant improvement. Although only a marginal improvement was observed by using a majority-voting classifier for each feature set in general, it eliminates the subjective errors resulting from having to choose a single classifier for a classification problem.

Assuming we were able to select the best classifier for each individual image, we can calculate the upper bounds of classification accuracy on the current feature sets. Over 95% of all images can be correctly classified by at least one of the eight tested classifiers for each of the four feature sets. SLF10, containing only 9 features, gave the closest performance to its upper bound among the feature sets. The accuracy upper bounds presumably represent the fraction of cells whose patterns were distorted by mitosis, cell death or acquisition errors.

### *Inclusion of New Texture Features*

Having presumably identified the limits of classification accuracy using the existing SLF, we next examined whether adding new features could improve these

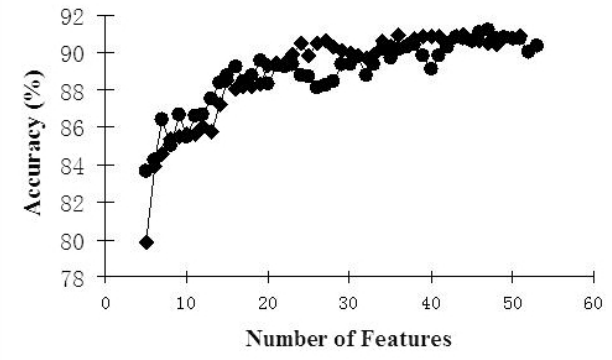

**Figure 5**
Selection of feature subsets including Gabor and Daubechies features. Classifiers were trained using increasing numbers of features from the ranked list selected by SDA from either the 180-feature set including DNA features (filled circle) or the 174-feature set without DNA features (filled diamond) and performance evaluated by 10-fold cross validation. The classifiers used were the optimal majority-voting ensemble classifiers for SLF13 and SLF8 respectively (see Table 4).

accuracies. In looking for new features, we noted that eleven of the features selected for SLF8 and nine of the features selected for SLF13 were Haralick texture features [14]. To explore the possibility that other types of texture features might provide additional descriptive power, a wavelet-based multiresolution filtering technique was used to derive two additional sets of texture features, namely Gabor texture features [33] and Daubechies 4 wavelet features [34]. Gabor texture features were reported to outperform a number of other wavelet transformations in a task of analyzing images with natural textures [35,36].

60 Gabor texture features and 30 Daubechies 4 wavelet features were added to the current 2D image features. In addition to the six features derived from parallel DNA images and the original 84 features from the SLF7 feature set, we ended up with 180 features in total for 2D image description. We have shown that feature reduction can significantly improve classification accuracy and reduce the training and testing time [15]. Among a group of eight different feature reduction methods, stepwise discriminant analysis (SDA) was rated as the best [15]. We therefore applied SDA on both the 180-feature set including 6 DNA features and the 174-feature set without DNA features. 53 ranked features were returned from the 180-feature set and 51 features from the 174-feature set. The optimal majority-voting ensemble classifiers for SLF13 and SLF8 listed in Table 4 were applied to the 53-feature set and 51-feature set respectively to evaluate the sequential inclusion of these ranked features. Figure 5 shows the sequential inclusion of the top-ranked features in both sets evaluated by the ensemble classifiers. The top 47 out of 53 features selected by SDA from the 180-feature set gave the highest accuracy on the optimal majority-voting ensemble classifier from SLF13, and we defined these as a new feature set, SLF16. The top 44 features selected by SDA from the 174-feature set gave the highest accuracy on the optimal majority-voting ensemble classifier from SLF8. We defined a new feature set containing these 44 features as SLF15.

Both SLF16 and SLF15 were then evaluated using all eight classifiers. Table 3 shows the pairwise-classifier-error correlation coefficients for these two new feature sets. Again, all 8 classifiers gave few dependent errors on both feature sets. Optimal majority-voting ensemble classifiers were created and found to include neural network, linear-kernel SVM, exponential-rbf-kernel SVM, polynomial-kernel SVM, and AdaBoost for SLF16, and rbf-kernel SVM, exponential-rbf-kernel SVM, and polynomial-kernel SVM for SLF15 (Table 4). We achieved a 92.3% average accuracy for 2D image classification by using SLF16 and 91.5% by using SLF15. These two majority-voting classifiers for SLF16 and SLF15 performed statistically better than the previously configured neural networks and the individual classifiers for SLF13 and SLF8 respectively. The benefits of including the new texture features can be represented by a 2% improvement on classifying 2D protein fluorescence microscope images both with and without DNA features. Table 4 also showed that the accuracy upper bounds for SLF16 and SLF15 are higher than those of SLF13 and SLF8 feature sets respectively.

To gain insight into the basis for the improvement, we compared the distributions in the two feature spaces of those images that were misclassified by the neural network classifier using SLF13 but were correctly classified by the ensemble classifier using SLF16. The ratio of the average distance of these images from their class mean over the average distance of all images from their class mean changed from 0.08 in the SLF13 feature space to 0.07 in that of SLF16. Thus the new features apparently moved the outlier images close enough to their class means that they could be correctly classified.

### *Improved distinction between similar classes*

That the best classifiers described above represent improvement over previous results not only in average classification accuracy but also in the ability to discriminate similar patterns is shown by the confusion matrixes in Tables 5A and 5B. Compared to the best previous results on 2D and 3D classification [11,14], the recognition accuracies on most patterns are significantly improved. The Golgi proteins giantin and Gpp130, which

**Table 5: Confusion matrix for 2D HeLa cell (A) and 3D HeLa cell (B) images using optimal majority-voting ensemble classifier with feature set SLF16 (A) and SLF10 (B) respectively. (Due to rounding, each row may not sum to 100). The average accuracy was 92.3% for 2D images (A) and 95.8% for 3D images (B).**

A)

| | DNA | ER | Gia | Gpp | Lam | Mit | Nuc | Act | TfR | Tub |
|---|---|---|---|---|---|---|---|---|---|---|
| DNA | 98.9 | 1.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ER | 0 | 96.5 | 0 | 0 | 0 | 2.3 | 0 | 0 | 0 | 1.2 |
| Gia | 0 | 0 | 90.8 | 6.9 | 0 | 0 | 0 | 0 | 2.3 | 0 |
| Gpp | 0 | 0 | 14.1 | 82.4 | 0 | 0 | 2.4 | 0 | 1.2 | 0 |
| Lam | 0 | 0 | 1.2 | 0 | 88.1 | 1.2 | 0 | 0 | 9.5 | 0 |
| Mit | 0 | 2.7 | 0 | 0 | 0 | 91.8 | 0 | 0 | 2.7 | 2.7 |
| Nuc | 0 | 0 | 0 | 0 | 0 | 0 | 98.8 | 0 | 1.3 | 0 |
| Act | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| TfR | 0 | 1.1 | 0 | 0 | 12.1 | 2.2 | 0 | 1.1 | 81.3 | 2.2 |
| Tub | 1.1 | 2.2 | 0 | 0 | 0 | 1.1 | 0 | 0 | 1.1 | 94.5 |

B)

| | Cyt | DNA | ER | Gia | Gpp | Lam | Mit | Nuc | Act | TfR | Tub |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cyt | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DNA | 0 | 98.1 | 0 | 0 | 0 | 0 | 0 | 1.9 | 0 | 0 | 0 |
| ER | 0 | 0 | 96.6 | 0 | 0 | 0 | 0 | 0 | 1.7 | 0 | 1.7 |
| Gia | 0 | 0 | 0 | 98.2 | 0 | 1.9 | 0 | 0 | 0 | 0 | 0 |
| Gpp | 0 | 0 | 0 | 4 | 96.0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lam | 0 | 0 | 0 | 1.8 | 1.8 | 96.4 | 0 | 0 | 0 | 0 | 0 |
| Mit | 0 | 0 | 0 | 3.5 | 0 | 0 | 94.7 | 0 | 1.8 | 0 | 0 |
| Nuc | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| Act | 0 | 0 | 1.7 | 0 | 0 | 0 | 1.7 | 0 | 94.8 | 1.7 | 0 |
| TfR | 0 | 0 | 0 | 0 | 0 | 5.7 | 3.8 | 0 | 1.9 | 84.9 | 3.8 |
| Tub | 0 | 0 | 3.7 | 0 | 0 | 0 | 0 | 0 | 0 | 1.9 | 94.4 |

cannot be distinguished by visual inspection [14], can be distinguished over 82% of the time in 2D images using SLF16 and 96% in 3D images using SLF10. These are 12% and 15% higher than the previous best performance for 2D and 3D images respectively. The transferrin receptor pattern (TfR), which has a similar distribution to that of the lysosomal protein LAMP2 (Lam), was the least accurately classified pattern in both 2D and 3D. Its recognition was improved by 6% compared to previous results for 2D images, but not at all for 3D images. This suggests the need for future work to improve its recognition.

The better performance achieved from the new features and the ensemble classifiers can be further seen by inspecting images that were misclassified by the original neural network classifier but could be correctly classified using the new texture features (Figure 6). We note that the images in Figure 6 are much less typical of the patterns expected for the major organelles than those in Figure 1. For instance, mitochondria typically locate around the nucleus (Figure 1E) but the image shown in Figure 6E includes staining over the nucleus. The same is true for the tubulin pattern. The blurry image shown in Figure 6I is much less typical compared to the branched pattern shown in Figure 1I. Presumably, the new features that rely on texture are not confused by these visible differences. As a further example, the nucleolin pattern shown in Figure 6F only contains one big object, while that of Figure 1F includes a few round objects. Although the morphological and geometric features would be very different for these images, similar texture information can be observed in both images. Furthermore, the relatively independent errors (Table 3) among the classifiers of a majority-voting ensemble contribute to a more robust prediction. For instance, linear-kernel SVM, one of the five classifiers in the best performing ensemble classifier of SLF16 (Table 4), predicted the image of the transferrin receptor pattern in Figure 6 as tubulin, while all other classifiers in the ensemble made the accurate prediction. This error would not be avoided if the linear-kernel SVM was selected as the only classifier. Therefore, the accurate recognition of these less typical images can be attributed to the new texture features that capture more frequency information from the fluorescence distribution and the ensemble classifier that enriches the prediction confidence by combining outputs from multiple classifiers.

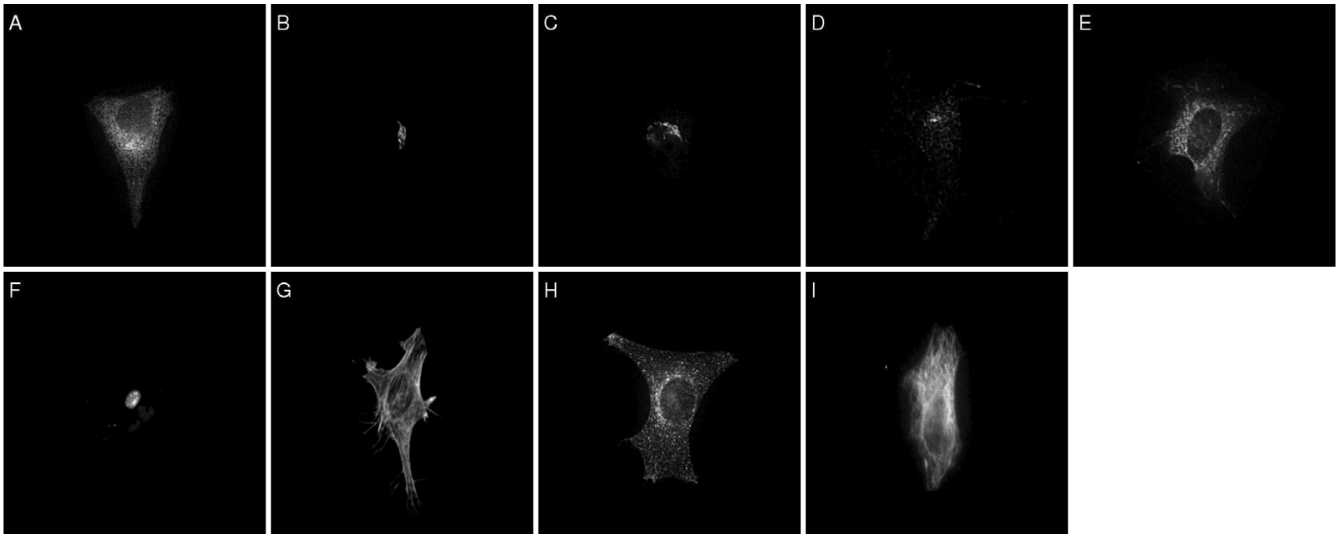

**Figure 6**
Example 2D images that were misclassified by the original neural network classifier but could be correctly classified using the best performing ensemble classifier using SLF16. From among the images incorrectly classified by the neural network for each class, the image that was most frequently classified accurately during training of the ensemble classifier was chosen (a random choice was made in the case of ties). ER (A), giantin (B), gpp130 (C), LAMP2 (D), mitochondria (E), nucleolin (F), actin (G), transferrin receptor (H), and tubulin (I). The only DNA image that was misclassified by the original neural network classifier was also missed by the ensemble.

Figure 7 shows example 2D images that could not be correctly classified by any of the eight classifiers using the new feature set SLF16. The giantin and gpp130 images in Figure 7 show patterns that are much more diffuse than typical Golgi images. This is presumably due to the onset of Golgi breakdown prior to mitosis. The LAMP2 and mitochondria patterns in Figure 7 may also be affected by mitosis (or perhaps cell death) given the extensive fluorescence in the nucleus and cytoplasm that are not usually observed (Figure 1). In future work, we plan to seek more robust features that can minimize the effect of mitotic changes and capture the common information that exists between the images in Figure 7 and those in Figure 1. Of course, we expect that some images that are perturbed by cell death and other experimental artifacts may never be correctly classified.

### *Tradeoffs between classification accuracy and computational cost for feature sets*

To further investigate the utility of our classification systems, we examined the processing time ("cost") of each feature. Image preprocessing steps including background subtraction, cropping, thresholding, and filtering need to be conducted before calculating features for 2D images. For 3D images, similar preprocessing steps were employed except that cropping is replaced by seeded watershed segmentation [11]. Since many features are related to each other, we divided the calculation of each feature to two parts, the setup cost and the computation cost. For instance, SLF7.80–7.84 are object skeleton features [14] that all require object finding as the setup cost before calculating each individual feature. Therefore, the total cost of SLF7.80–7.84 is the common setup cost plus the cost of calculating each individual feature. Since Gabor and Daubechies wavelet features involve decomposing an input image with filters that have different scales, we consider using them as a group for image classification. Table 6 shows the time costs and performances of the six feature sets used in our experiments as well as other processing costs. The large increases in costs from SLF13 to SLF16 and SLF8 to SLF15 are from the computation of both Gabor and Daubechies wavelet features. In 3D image classification, preprocessing and segmentation dominate the total costs of a feature set. Given the relationship between performance and computation cost for each of the six feature sets, a practical system can trade performance against cost and select the optimal feature set for a given purpose.

### *Classification of sets of images*

Previous work has shown that classifying sets of images by plurality voting can dramatically increase the overall accuracy of recognizing subcellular patterns [9]. We therefore evaluated two strategies of image set classification by

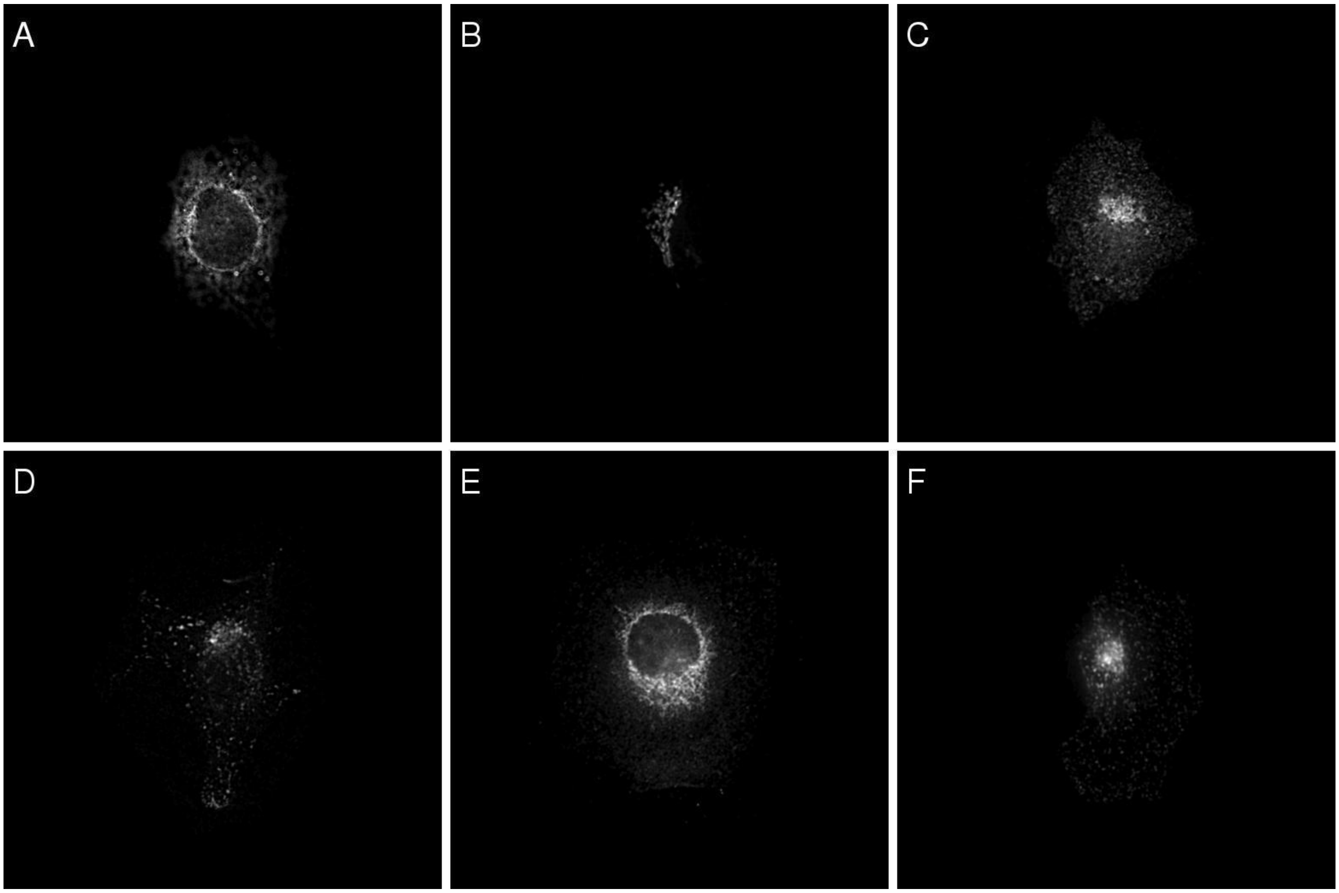

**Figure 7**
Example 2D images that could not be correctly classified by any individual classifier using feature set SLF16. The image that was most frequently classified incorrectly during training of the ensemble classifier was chosen (a random choice was made in the case of ties). ER (A), giantin (B), gpp130 (C), LAMP2 (D), mitochondria (E), and transferrin receptor (F). All images in the other classes could be correctly classified by at least one of the eight classifiers.

**Table 6: Processing costs for calculating feature sets as well as the performance of each feature set. The costs were averaged on ten random images selected from all classes of both 2D and 3D fluorescence microscope images. Costs shown for feature sets include preprocessing (and segmentation) costs.**

| Operation | Number of Features | CPU time (s) | Overall Accuracy (%) |
|---|---|---|---|
| 2D preprocessing | N/A | 0.6 | N/A |
| 3D preprocessing | N/A | 21.1 | N/A |
| 3D segmentation | N/A | 6.8 | N/A |
| SLF8 (2D) | 32 | 13.2 | 89.4 |
| SLF15 (2D) | 44 | 68.3 | 91.5 |
| SLF13 (2D DNA) | 31 | 10.8 | 90.7 |
| SLF16 (2D DNA) | 47 | 66.3 | 92.3 |
| SLF14 (3D) | 14 | 31.5 | 89.8 |
| SLF10 (3D DNA) | 9 | 32.0 | 95.8 |

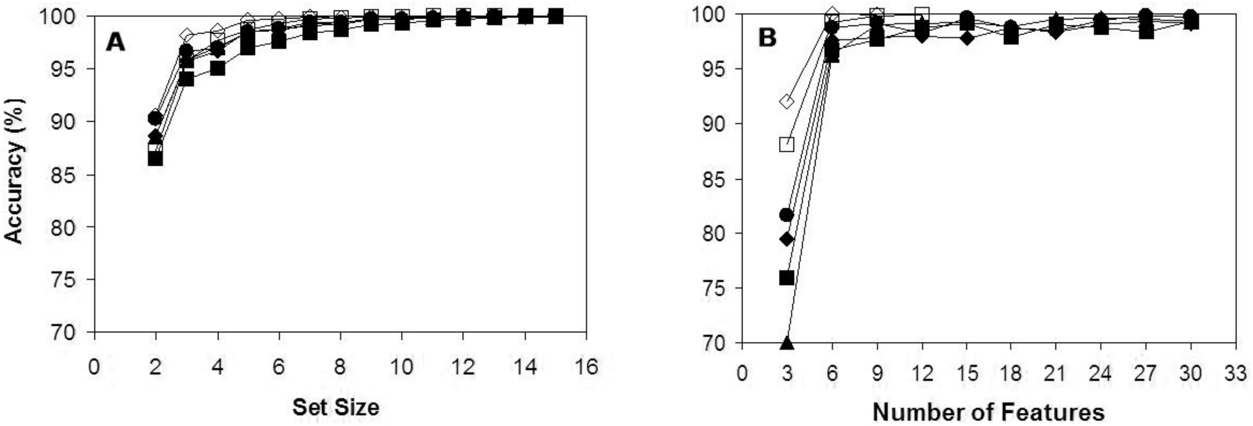

**Figure 8**
Dependence of image set classification accuracy on set size and feature set size. Panel A depicts classification accuracy for sets of images drawn from the same class. The accuracy obtained using plurality voting was averaged on 1000 random trials of image sets of various sizes drawn from each class in the test set by using the optimal majority-voting classifier for feature set SLF8 (filled square), SLF15 (filled triangle), SLF13 (filled diamond), SLF16 (filled circle), SLF14 (open square), and SLF10 (open diamond). Panel B depicts classification accuracy for reduced feature subsets using plurality voting. The accuracy obtained using plurality voting was averaged on 1000 random trials of sets of 10 images drawn from each class in the test set for various numbers of features from SLF8 (filled square), SLF15 (filled triangle), SLF13 (filled diamond), SLF16 (filled circle), SLF14 (open square), and SLF10 (open diamond).

using the six feature sets. Firstly, different image set sizes were tested for the six feature sets. Figure 8A shows the average performance of the majority-voting classifier for each feature set (Table 4) over 1000 random trials of image sets drawn from each class in the test set. The dominant predicted class in an image set was taken as the output, while random choice was made if several classes tied. The more images in a set, the more accurate the classifier will be. The smallest numbers of images in a set required to reach an average accuracy of 99% are 7, 9, 5, 6, 7, and 7 for SLF13, SLF8, SLF10, SLF14, SLF16, and SLF15 respectively. 3D image classification requires fewer images in a set to achieve 99% classification accuracy than for 2D. Secondly, we tested different numbers of features (in increments of 3) selected from each feature set given a fixed image set size of 10. The results are shown in Figure 8B. To achieve an average 99% classification accuracy in a 10-image set, the top 24, 30, 6, 6, 9, 9 features of SLF13, SLF8, SLF10, SLF14, SLF16, and SLF15 need to be included respectively. Again, 3D image classification shows more advantages in image sets classification than 2D image sets classification. The benefits of SLF16 and SLF15 can be represented by a large reduction of the numbers of features required to achieve an average 99% accuracy compared to SLF13 and SLF8 respectively. Unlike the fluctuation in classifier performance on individual images given increasing number of features in different feature spaces (Figure 4), the performance on image sets approaches a plateau as the number of features increases for all feature spaces. Image sets are often easily acquired in biological imaging experiments; therefore classifying sets of images has practical feasibility and can dramatically enhance system performance. By using only 9 and 6 features for 2D and 3D image sets that have 10 images, we can achieve essentially perfect classification accuracy.

## Conclusions

This paper addresses a supervised learning problem in the domain of protein subcellular location determination. Through employing new classifiers and features in our system, we reduced the error rates of our previous system by one third to one half for 2D and 3D image classification. The current system is able to give 92–93% classification accuracy on 10 major eukaryotic subcellular location patterns from 2D fluorescence microscope images with DNA features (compared to 88% previously) and 91–92% accuracy without DNA features (compared to 86% previously). Around 96% classification accuracy can be expected using the current system to recognize 11 major eukaryotic subcellular location patterns from 3D images with DNA features (compared to 91% previously) and 89–90% accuracy without DNA features. Patterns that cannot be distinguished by visual examination, Giantin and Gpp130, can now be distinguished 96% of the time in 3D images. In addition, essentially perfect results can be achieved by classifying sets of images using a small number of features. These results are the best currently available for the protein subcellular location domain.

This fast and automated image-based approach can be combined with high-throughput fluorescence microscope imaging techniques, forming a new subfield of proteomics which we have termed location proteomics. By taking fluorescence microscope images of all proteins expressed in a certain cell type, we can cluster proteins by their location similarity and create a Subcellular Location Tree organizing location families [37]. The system also has possible applications in medicine, including identification of proteins whose location patterns change as a result of disease and which may be either diagnostic markers or therapeutic targets.

## Methods

### *Training of the Classifiers*

Throughout this section, $K$ represents the total number of classes, $F$ stands for the number of input features to each classifier, and $P$ denotes the total number of parameters of a given model. We define the information content of each classifier, similar to the minimum description length [32], as the number of bits that need to be learned and delivered in order to evaluate the classifier on test datasets. Since each parameter is represented by a floating point number, the total number of bits is calculated as $32P$.

#### *Support Vector Machine*

We tested SVM with four different kernel functions as well as the three multi-class expansion methods. The kernel functions used were a linear kernel, a polynomial kernel, a Gaussian radial basis kernel, and an exponential radial basis kernel. The optimal values of the parameters for the last three kernels as well as the error penalty $C$ for each kernel were determined for a given dataset by 10-fold cross validation, and the highest scoring kernel was chosen. A Matlab SVM toolbox was downloaded from http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox/.

The parameters of an SVM classifier include the support vectors represented in the feature space, the weights associated with each support vector plus the bias, kernel parameters used in the kernel function, and the scale factor introduced by a multi-class expansion method. Let $S_i$ denote the number of support vectors in the ith model, $k$ denote the number of kernel parameters, and $m$ denote the number of models that need to be saved for a given multi-class expansion method. The total number of parameters of an SVM is then:

$$P = \sum_{i=1}^{m} (F * S_i + S_i + 1 + k)$$

where $m = K$ for the maxwin strategy and $m = K * (K - 1)/2$ for the DAG and pairwise strategies. Except for the linear kernel function that has only one kernel parameter (penalty C), all other three kernel functions have two kernel parameters (Table 2).

#### *AdaBoost*

A C++ machine learning library, Torch [38], was used as an implementation of AdaBoost.M1 with a one-hidden-layer neural network as the base classifier. Based on preliminary experiments (data not shown), we used a maximum iteration number of 300 and a learning rate decay of 0.1 for the training of the neural network leaving all other parameters at their defaults. The neural network was trained until the error rate on the training set was minimal or the maximum number of iterations was reached. We tested different numbers of hidden nodes in the base classifier as well as different numbers of boosting iterations.

The parameters of an AdaBoost classifier include the weights associated with each base classifier to form the final output and the weights of each neural network base classifier. Let $N_{hu}$ denote the number of hidden nodes in the base classifier and $N_{boost}$ denote the number of boosting iterations. We can calculate the total number of parameters of a given AdaBoost classifier as:

$$P = (N_{hu} * (K + F) + 1) * N_{boost}.$$

#### *Bagging*

We also used the Torch software as an implementation of bagging with a one-hidden-layer neural network as the base classifier (as described above for Adaboost). An equally weighted sum was used as the output. We tested different numbers of hidden nodes in the base classifier as well as different numbers of bagging iterations.

Similar to AdaBoost, the parameters of a bagging classifier include the weights of each base classifier. However, equal weights are associated to each base classifier to form the output. Therefore, we can calculate the total number of parameters as:

$$P = N_{hu} * (K + F) * N_{bag}$$

where $N_{hu}$ denotes the number of hidden nodes in the base classifier and $N_{bag}$ stands for the number of bagging iterations.

#### *Mixtures-of-Experts*

We also used the Torch software implementation of Mixtures-of-Experts with a one-hidden-layer neural network as both the gating and expert networks. We tested different numbers of hidden nodes in both the gating and expert networks and different numbers of experts.

The parameters of a Mixture-of-Experts classifier involve those of the gating network and the local expert networks. Let $N_e$ denote the number of local experts, $N_{hu}$ denote the number of hidden nodes in each local expert, and $N_{hug}$ represent the number of hidden nodes in the gating network. The total number of parameters of a given Mixtures-of-Experts classifier can be calculated as:

$$P = N_{hu} * (K + F) * N_e + N_{hug} * (K + K * N_e).$$

### *Netlab Neural Network*

For comparison, we also used the same neural network implementation as in our previous work. This was the gradient descent back-propagation neural network with one hidden layer implemented by the Netlab neural network toolbox http://www.ncrg.aston.ac.uk/netlab/, which was used with momentum and learning rate set to 0.9 and 0.001 respectively [9]. A single hidden layer network was used given our previous observation that a two-hidden layer network did not significantly improve classification accuracy for the 2D HeLa images [13]. The training set was further divided into "use for training" and "use to stop training" subsets. We tested different fractions of the training set in these subsets, as well as different numbers of hidden nodes. The total number of parameters of a neural network classifier can be calculated as:

$$P = N_{hu} * (K + F)$$

where $N_{hu}$ stands for the number of hidden nodes in the network.

## *Features*

### *Previous Subcellular Location Features*

We have previously designed numerical features from different sources to describe protein location patterns in fluorescence microscope images. These features are invariant to cell rotation and translation, and are robust over various cell types and fluorescence microscopy methods. To date, the best feature sets for describing 2D subcellular patterns are SLF8 and SLF13 [14], which are used when a parallel DNA image is not or is available, respectively. SLF8 was derived by feature selection starting from an 84-feature set, SLF7, that includes various moment, texture, morphological, and geometric features. SLF8 contains 7 Zernike moment features, 11 Haralick texture features, and 14 geometric and morphological features. SLF13 is selected from the combination of SLF7 with 6 features derived from a parallel DNA image that were originally defined as part of SLF2 [9]. SLF13 contains 6 Zernike moment features, 9 Haralick texture features, and 16 geometric and morphological features including the DNA features.

Our previous best feature set for 3D fluorescence microscope images is SLF10 (Velliste and Murphy, in preparation). It is derived from feature set SLF9 [11], which contains 28 geometric and morphological features, 14 derived from a protein pattern itself and 14 derived from its relationship to a parallel DNA image. SLF10 contains 9 features selected by stepwise discriminant analysis from this set (Velliste and Murphy, in preparation). For cases where a parallel DNA image is not available, we define here the 14 features from SLF9 that do not require a parallel DNA image as the feature set SLF14. Since this feature set is already fairly small, we did not do feature selection to reduce it further.

### *New Texture Features*

A 2D Gabor function is a 2D Gaussian modulated by a sinusoid [36]:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x{}^2} + \frac{y^2}{\sigma_y{}^2}\right) + i2\pi Wx\right\}$$

where $W$ is the frequency bandwidth of Gabor filters.

Given a Gabor filter with scale $m$ and orientation $n$, we convolved an image with this filter and took the mean and standard deviation of the resulting image as texture features. For this purpose, we used the Gabor texture feature extractor [36] downloaded from http://vision.ece.ucsb.edu/texture/software/index.html. The default parameters were used to construct a Gabor wavelet bank of five different scales and six different orientations. A total of 60 Gabor texture features were derived from each image.

Thirty conventional Daubechies 4 wavelet features were calculated up to the 10th level decomposition using both the scaling and wavelet functions of the Daubechies 4 wavelet transformation [34]. The scaling function and wavelet function can be viewed as a low-pass and a high-pass filter respectively. After decomposing an image column-wise and row-wise sequentially using the two filters, we obtained four convolved images, three of which captured the high-frequency information in the x, y, and diagonal directions and the remaining one stored the low-frequency information and was decomposed in the next iteration. The average energy of the three high-frequency components were taken as features at each level, and we therefore ended up with 30 features by decomposing an image 10 times. The Daubechies 4 wavelet transformation was implemented by using the Matlab (R12) Wavelet toolbox.

Since features derived from wavelet transformation are not invariant to cell rotation and translation, we conducted a principal axis alignment on each image before performing wavelet transformations. After background subtraction, thresholding, and filtering, the image was pivoted to its center of fluorescence. The angle between the coordinate axis and the principal axis of the cell was computed using secondary moments and the image was rotated to align the primary axis with the y axis. To align the direction of the minor axis, an additional 180° rotation was carried out when the x skewness was negative.

#### *Feature Selection*

Feature reduction is often beneficial for faster and more accurate classification, and we have previously observed that stepwise discriminant analysis (SDA) performs the best in a comparison of eight different feature reduction methods [15]. SDA considers the goodness of a feature as its ability to differentiate classes while at the same time clustering data from the same class as compactly as possible [39]. This goodness can be represented by F-statistics involving the ratio between the within-class and among-class covariance matrix. We implemented SDA using the STEPDISC function of SAS (SAS Institute, Cary, NC, USA), which employed a forward-backward scheme to select features based on their goodness measured by F-statistics. The 90 Gabor and Daubechies features were merged with SLF7 either without or with the 6 DNA features. The sets obtained from SDA were defined as SLF15 and SLF16, respectively.

#### *Feature Normalizaton*

All features in a given training set were normalized to have zero mean and unit variance and the same mean and variance were used to normalize the features for the corresponding test set.

### *Data*

To evaluate different classifiers, 2D and 3D HeLa image collections described previously were used. These two image data sets were created to include the most common protein subcellular distributions as well as a pair of proteins, giantin and Gpp130, that cannot be distinguished by visual comparison. We have previously reported the inability of a human subject to distinguish these proteins [14].

The 2D HeLa image set [9] representing 10 major subcellular distributions was acquired by introducing antibodies and molecular probes against proteins in major subcellular organelles as well as nuclear DNA. The nine proteins in this image set included an endoplasmic reticulum protein, f-actin in microfilaments, Giantin and Gpp130 in the Golgi complex, the nucleolar protein nucleolin, a mitochondrial outer membrane protein, LAMP2 in the lysosomes, endosomal transferrin receptor, and tubulin in microtubules. A set of three images separated by 0.23 microns were collected for each field, and nearest neighbor deconvolution was used to generate a single image corresponding to the central slice after removal of estimated out of focus fluorescence. The collection contains from 73 to 98 images for each of the 10 classes for a total of 862 single-cell images. Every image has a resolution of 382 × 512 pixels, each representing 0.23 × 0.23 microns. The most typical image of each pattern from correctly classified images by the original neural network classifier is shown in Figure 1.

The 3D HeLa image set [11] was created with a three-laser confocal laser scanning microscope by using probes for the nine proteins above as well as DNA and total protein. In addition to being used as a reference image for the proteins, the DNA and total protein images were also used to create two additional classes (DNA and Cytoplasm, respectively), resulting in a total 11 classes. Each of these 11 classes has a number of 3D images ranging from 50 to 58 for a total of 598 single-cell images. Every 3D image is a stack of 14 to 24 2D slices, each of which has a resolution of 1024 × 1024 voxels. Each voxel in the 3D stack represents 0.049 × 0.049 × 0.2 microns of the sample. The most typical image of each pattern from correctly classified images by the original neural network classifier is shown in Figure 1.

## Author's Contributions

K.H. played the major role in carrying out the specific experiments, and drafted the manuscript. R.F.M. conceived of the study and edited the manuscript.

## Acknowledgments

We thank William Dirks and Adrienne Wells for programming and preliminary work on applying wavelet features to protein location patterns, and Dr. Samy Bengio (IDIAP, Martigny, Switzerland) for helpful discussion about the Torch package. This work was supported in part by NIH grant R33 CA83219 and R01 GM068845, and by a research grant from the Commonwealth of Pennsylvania Tobacco Settlement Fund. K.H. was supported by a Graduate Fellowship from the Merck Computational Biology and Chemistry Program at Carnegie Mellon University through a grant from the Merck Company Foundation.

## References

1. Norin M, Sundstrom M: **Structural proteomics: developments in structure-to-function predictions.** *Trends Biotech* 2002, **20:**79-84.
2. Macbeath G: **Protein microarrays and proteomics.** *Nature Genetics* 2002, **32:**526-532.
3. Nakai K: **Protein sorting signals and prediction of subcellular localization.** *Adv Protein Chem* 2000, **54:**277-344.
4. Hua S, Sun Z: **Support vector machine approach for protein subcellular localization prediction.** *Bioinformatics* 2001, **17:**721-728.
5. von Heijne G, Nielsen H, Engelbrecht J, Brunak S: **Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites.** *Protein Eng* 1997, **10:**1-6.

6. Chou K, Cai Y: **Using function domain composition and support vector machines for prediction of protein subcellular location.** *J Biol Chem* 2002, **277:**45765-45769.
7. Mott R, Schultz J, Bork P, Ponting CP: **Predicting protein cellular localization using a domain projection method.** *Genome Res* 2002, **12:**1168-1174.
8. Drawid A, Gerstein M: **A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome.** *J Mol Biol* 2000, **301:**1059-1075.
9. Boland MV, Murphy RF: **A Neural Network Classifier Capable of Recognizing the Patterns of all Major Subcellular Structures in Fluorescence Microscope Images of HeLa Cells.** *Bioinformatics* 2001, **17:**1213-1223.
10. Danckaert A, Gonzalez-Couto E, Bollondi L, Thompson N, Hayes B: **Automated Recognition of Intracellular Organelles in Confocal Microscope Images.** *Traffic* 2002, **3:**66-73.
11. Velliste M, Murphy RF: **Automated Determination of Protein Subcellular Locations from 3D Fluorescence Microscope Images.** *2002 IEEE International Symposium on Biomedical Imaging (ISBI-2002)* Bethesda, MD, USA; 2002:867-870.
12. Price JH, Goodacre A, Hahn K, Hodgson L, Hunter EA, Krajewski S, Murphy RF, Rabinovich A, Reed JC, Heynen S: **Advances in Molecular Labeling, High Throughput Imaging and Machine Intelligence Portend Powerful Functional Cellular Biochemistry Tools.** *J Cell Biochem Supp* 2003:194-210.
13. Murphy RF, Boland MV, Velliste M: **Towards a Systematics for Protein Subcellular Location: Quantitative Description of Protein Localization Patterns and Automated Analysis of Fluorescence Microscope Images.** *Eighth International Conference on Intelligent Systems for Molecular Biology Volume 8*. San Diego; 2000:251-259.
14. Murphy RF, Velliste M, Porreca G: **Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images.** *J VLSI Sig Proc* 2003, **35:**311-321.
15. Huang K, Velliste M, Murphy RF: **Feature Reduction for Improved Recognition of Subcellular Location Patterns in Fluorescence Microscope Images.** *Proc SPIE* 2003, **4962:**307-318.
16. Cortes C, Vapnik V: **Support vector networks.** *Machine Learning* 1995, **20:**1-25.
17. Vapnik V: **Statistical Learning Theory.** New York City, Wiley; 1998.
18. Kressel U: **Pairwise Classification and Support Vector Machines.** *Advances in Kernel Methods - Support Vector Learning Volume 15*. Edited by: Scholkopf B, Burges C and Smola A J. Cambridge, Massachusetts, MIT Press; 1999:255-268.
19. Platt J, Cristianini N, Shawe-Taylor J: **Large Margin DAGs for Multiclass Classification.** *Adv Neural Inform Proc Systems* 2000, **12:**547-553.
20. Dietterich TG: **Ensemble methods in machine learning.** *Lecture Notes in Computer Science Volume 1857*. Springer-Verlag; 2000:1-15.
21. Hansen L, Salamon P: **Neural network ensembles.** *IEEE Trans Pattern Analysis and Machine Intell* 1990, **12:**993-1001.
22. Freund Y, Shapire RE: **Experiments with a new boosting algorithm.** *13th International Conference on Machine Learning* Morgan Kauffman; 1996:148-156.
23. Bauer E, Kohavi R: **An empirical comparison of voting classification algorithms: Bagging, boosting and variants.** *Machine Learning* 1999, **36:**525-536.
24. Kuncheva LI, Whitaker CJ, Shipp CA, Duin RPW: **Is independence good for combining classifiers?** *15th International Conference on Pattern Recognition* Barcelona, Spain; 2000:168-171.
25. Valentini G, Masulli F: **Ensembles of learning machines.** *Neural Nets WIRN Vietri-02* Edited by: Marinaro M and Tagliaferri R. Heidelberg , Germany., Springer-Verlag; 2002:3-19.
26. Schapire RE: **The Boosting Approach to Machine Learning: An Overview.** *MSRI workshop on Nonlinear estimation and Classification* 2002.
27. Waterhouse SR: **Classification and Regression using Mixtures of Experts.** *Department of Engineering, Jesus College* Cambridge, U.K., University of Cambridge; 1997.
28. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE: **Adaptive mixtures of local experts.** *Neural Computation* 1991, **3:**79-87.
29. Jordan MI, Jacobs RA: **Hierarchical mixtures of experts and the EM algorithm.** *Neural Computation* 1994, **6:**181-214.
30. Kittler J, Messer K: **Fusion of multiple experts in multimodal biometric personal identity verification systems.** *2002 IEEE International Workshop on Neural Networks for Signal Processing (NNSP 12)* 2002:3-12.
31. Markey MK, Boland MV, Murphy RF: **Towards objective selection of representative microscope images.** *Biophys J* 1999, **76:**2230-2237.
32. Mitchell TM: **Machine Learning.** WCB/McGraw-Hill; 1997.
33. Daugman JD: **Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression.** *IEEE Trans Acoustics Speech Sig Proc* 1988, **36:**1169-1179.
34. Daubechies I: **Orthonormal Bases of Compactly Supported Wavelets.** *Comm Pure Appl Math* 1988, **41:**909-996.
35. Ma WY, Manjunath BS: **A comparison of wavelet transform features for texture image annotation.** *IEEE International Conference on Image Processing Volume 2*. Washington D.C, U.S.A.; 1995:2256-2259.
36. Manjunath BS, Ma WY: **Texture features for browsing and retrieval of image data.** *IEEE Trans Pattern Analysis and Machine Intelligence* 1996, **8:**837-842.
37. Chen X, Velliste M, Weinstein S, Jarvik JW, Murphy RF: **Location proteomics - Building subcellular location trees from high resolution 3D fluorescence microscope images of randomly-tagged proteins.** *Proc SPIE* 2003, **4962:**298-306.
38. Collobert R, Bengio S, Mariéthoz J: **Torch: a modular machine learning software library.** IDIAP; 2002.
39. Jennrich RI: **Stepwise discriminant analysis.** *Statistical Methods for Digital Computers Volume 3*. Edited by: Enslein K, Ralston A and Wilf HS. New York, John Wiley & Sons; 1977:77-95.