# IDENTIFYING FLUORESCENCE MICROSCOPE IMAGES IN ONLINE JOURNAL ARTICLES USING BOTH IMAGE AND TEXT FEATURES

*Juchang Hua[1,2,4], Orhan N. Ayasli[5], William W. Cohen[1,4] and Robert F. Murphy[1,2,3,5]*

Center for Bioimage Informatics[1], Departments of Biological Sciences[2] and Biomedical Engineering[3], Machine Learning Department[4] and Computer Science Department[5], Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

## ABSTRACT

We have previously built a Subcelluar Location Image Finder (SLIF) system, which extracts information regarding protein subcellular location patterns from both text and images in journal articles. One important task in SLIF is to identify fluorescence microscope images. To improve the performance of this binary classification problem, a set of 7 edge features extracted from images and a set of "bag of words" text features extracted from text have been introduced in addition to the 64 intensity histogram features we have used previously. An overall accuracy of 88.6% has been achieved with an SVM classifier. A co-training algorithm has also been applied to the problem to utilize the unlabeled dataset and it substantially increases the accuracy when the training set is very small but can contribute very little when the training set is large.

***Index Terms***— Image classification

## 1. INTRODUCTION

In biological research, results are usually reported via journal articles, which contain a mixture of methods, results, conclusions and more importantly, illustrations of images and plots. An important task of automated information retrieval is to process the varied, unstructured information in journal articles and organize them in a systematic, structured database. Extensive work has been done to do this for the text in journal articles [1, 2]. Since much of the useful information in an article is contained in the figures, we have previously described the first system to extract information from both text and images in biological journal articles [3-5]. One particular focus of this system, the Subcellular Location Image Finder (SLIF), is to retrieve information about the subcellular location patterns of proteins, the main source of which are fluorescence microscope images (FMIs). The automated identification of FMIs is therefore a crucial step in SLIF. Recently, other systems for classifying biological journal figures have been described [6-8].

The most similar study [6] is a fusion classifier to classify images in biological literature. The classifier is constructed on top of SVM classifiers trained on image and text features. However, FMI was not one of the five categories in this study.

The starting point for the work described here is an FMI classifier described previously. It was trained with a k-Nearest Neighbor (KNN) algorithm using a set of 64-bin image histogram features [5]. The classifier was trained on figures extracted from PDF files in PubMed Central. However, we have observed that this previously trained classifier works poorly when applied to a large collection of PNAS papers. The precision dropped to around 50% and a lot of non-FMI, especially gel images, were misclassified. The work described below therefore addresses two tasks. The first is to improve the FMI classification with extended image features and a set of "bag of words" text features. Different classification algorithms are also tested to achieve the best result. The second is to determine whether the use of a co-training algorithm to exploit unlabeled data can improve performance.

## 2. METHODS

### 2.1 Image Acquisition and Labeling

The current version of the SLIF database contains 15,180 papers from volumes 94-99 of the Proceedings of the National Academy of Sciences. There are about 64,000 figures in this dataset which are automatically split by our system into their component panel images. The figure splitting is accomplished by a recursive boundary detecting algorithm [3]. From this collection, we randomly selected 1073 figures and constructed a dataset consisting of all panels for 175 figures and only one panel from each of the rest of the figures. The dataset contain 1993 panels in total.

Visual inspection was performed to label these panels as FMI or non-FMI. During this process, both the panel images and the figure captions were made available to the inspectors. To reduce the systematic error, each panel was labeled by one inspector (O.N.A.) and checked by two of us (R.F.M. and J.H.). Of the 1993 panels in the dataset, 820 (41%) were considered to be FMI and about 19% are gel

images. The labeled dataset is available from http://murphylab.web.cmu.edu/software.

### 2.2 Feature Calculation

Previously, normalized 64-bin histograms on image pixel intensities were used to identify FMIs [5]. These features tell apart the FMIs, which usually have large dark backgrounds and small bright objects, from other common image types such as plots or graphs. However, they fail to tell the difference between FMI and gel images, which have very similar distributions in image histograms. Examples are shown in Figure 1. Seven features based on image edge detection were therefore added to the feature set because of the obvious fact that gel images usually have strong edges and these edges have a horizontal or vertical orientation. Five of these features (SLF7.9 to SLF7.13) have been described previously and used for classification of subcellular patterns in FMI [9]. These five include one that measures the fraction of above-threshold pixels that are on an edge and four that measure the homogeneity of edge direction. We added two more features that specifically measure the horizontal and vertical edge content using a Sobel filter. The two features are the ratio of horizontal edge pixels to non-horizontal edge pixels and the ratio of vertical edge pixels to non-vertical edge pixels.

In addition to the 71 image features, "bag of words" text features were also extracted for each panel. One text feature is created for each word present in any of the training captions (a total of 20,627 words). In SLIF, significant efforts have been made to connect specific panels with specific information in caption. First, an OCR package was used to detect a panel label (such as "A" at the corner of a panel) in a panel. Then a caption processing program was used to detect the image pointer in the caption (such as "(A)" in front of a sentence) and divide the caption into "scopes" [3]. The text features value for each panel is the number of times that the corresponding word appeared in the the scope whose image identifier matches that panel's label and all the words in the rest of the caption which refers to the whole figure.

### 2.3 Image Classification

In order to show the contribution of these features, classifiers were trained on the labeled dataset with the following feature sets:

1. 64 histogram image features
2. 71 image features (with 7 new features)
3. text features only
4. all image and text features

Four different classification algorithms were used in the study. They are Support Vector Machine (SVM) [10], Boosted Decision Tree, Boosted Stump and K-Nearest Neighbor. SVM is a generalized linear classifier which searches for a decision boundary after transforming the feature space with a kernel function. In this study, we used a linear kernel for SVM with parameter values of 20 for *C*, the penalty factor and 0.01 for *epsilon*, the width of insensitive zone. Boosting, which is also known as "AdaBoost" [11], is a meta-algorithm to improve the performance of "weak" classifiers such as Decision Tree. It adaptively trains a new classifier on the data points which are misclassified in the previous one and a majority voting mechanism is used in the process of classification. In this study, both Decision Tree and Decision Stump (a decision tree of only one split) were boosted 10 times. The Decision Tree classifier uses a maximum depth of 5. The KNN algorithm looks for the $k$ training examples that are closest to the testing example and lets these training examples vote for a classification label. We used $k = 5$ in this study. All these algorithms are implemented in MinorThird, an open source Java package (http://minorthird.sourceforge.net/).

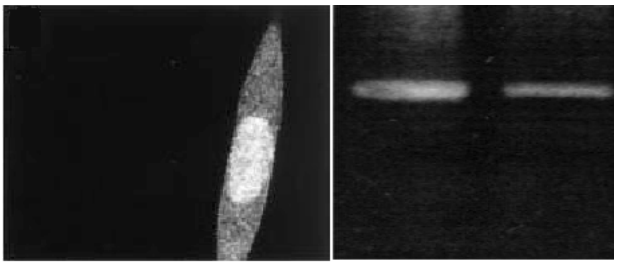

**Figure 1**. Comparison of fluorescence microscope and gel images. The left panel is an FMI of a CHO cell while the right panel shows an image of a gel. Note the strong horizontal edge content of the gel image.

### 2.4 Co-training with Unlabeled Dataset

Unlabeled data are usually much easier to obtain in a machine learning problem, and our FMI classification problem is one example. A co-training method has been proposed to take advantage of unlabeled data [12]. This algorithm starts with a labeled set *L* and an unlabeled set *U*. Then it iterates the following steps. First, *L* is used to train two distinct classifiers $h_1$ and $h_2$. $h_1$ is only based on the image features and $h_2$ is only based on text features. Second, both of these two classifiers are applied to a small unlabeled set *C*, which contains *c* examples randomly chosen from *U*. The most confident *p* positive labeled and *n* negative labeled examples are then added to *L* by both classifiers. Finally, *2p+2n* examples are randomly chosen from *U* to replenish *C*. Such a process repeats for a given number of times or until there are not enough examples in *U*. A final classifier is then trained on the expanded *L*. In this paper, we use $c$=10, $p$=1and $n$=3. Both classifiers are trained with SVM algorithms. To evaluate the performance of this algorithm, we use a portion of the labeled data as *L* and the rest of them for testing. *U* consists of randomly chosen panel images from the SLIF database.

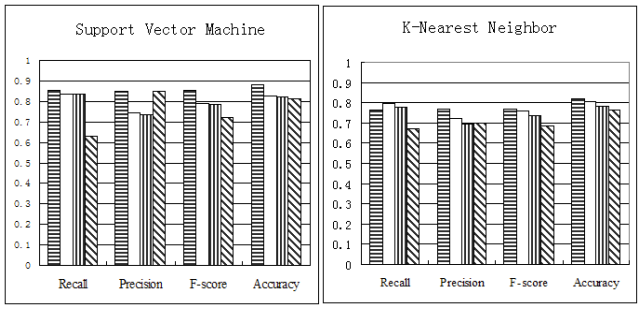


**Figure 2.** Results for different features sets and classifiers. The Recall, Precision, F-score and Error Rate are reported for each algorithm. From left to right, the columns show results with all image and text features, all image features, histogram image features only, and text features only.

## 3. RESULTS

Ten-fold cross-validation was performed to evaluate each of the four algorithms on each of the four feature sets described above. In this process, the labeled dataset was randomly divided into 10 parts of equal size. In each of the 10 trials, 9 parts were used for training a classifier and 1 part was used for testing. In order to avoid the effect of the strong similarity of the panels in a given figure, all panels from a given figure were either all put into the training set or the testing set during the splitting of the data. For each cross-validation, the number of True Positives, False Positives, True Negatives and False Negatives were counted. Figure 2 shows the performance of SVM and KNN by comparing the Recall (TP/(TP+FN)), Precision (TP/(TP+FP)), F-score (2/(1/Recall+1/Precision)) and Accuracy ((TP+TN)/total). Table 1 shows the confusion matrix of the SVM classifier trained on all features. The performances of all four algorithms are shown in Table 2.

With all of the four algorithms we used, the precision increases when both image and text features are used. Although the recall using both features is slightly less than that using image features only in KNN and Boosted Decision Tree algorithms, the improvement of using both features is unanimous in all algorithms when comparing F-score or error rate. The best result is an overall accuracy of 88.6% when SVM is used for both image and text features and the precision and recall are 85.3% and 85.1% respectively. The trade off between precision and recall is shown in Figure 5. It also shows the precision and recall of the previous system.

| True label | Predicted by classifier | |
|---|---|---|
| | FMI | Non-FMI |
| FMI | 85.3% | 14.7% |
| Non-FMI | 9.39% | 90.61% |

**Table 1**: Confusion matrix for 10-fold cross-validation using an SVM classifier on both image and text features. The overall accuracy is 88.6%.

| | | Recall | Prec. | F-score | Accur. |
|---|---|---|---|---|---|
| SVM | All | **0.853** | **0.851** | **0.852** | **0.886** |
| | Image | 0.838 | 0.747 | 0.790 | 0.828 |
| | Hist | 0.838 | 0.735 | 0.783 | 0.821 |
| | Text | 0.629 | 0.850 | 0.723 | 0.814 |
| KNN | All | 0.767 | 0.771 | 0.769 | 0.822 |
| | Image | 0.798 | 0.723 | 0.759 | 0.804 |
| | Hist | 0.776 | 0.695 | 0.744 | 0.782 |
| | Text | 0.670 | 0.701 | 0.685 | 0.762 |
| Boosted Decision Tree | All | 0.680 | 0.800 | 0.735 | 0.810 |
| | Image | 0.740 | 0.720 | 0.730 | 0.790 |
| | Hist | 0.742 | 0.712 | 0.727 | 0.787 |
| | Text | 0.580 | 0.740 | 0.650 | 0.760 |
| Boosted Stump | All | 0.739 | 0.837 | 0.785 | 0.844 |
| | Image | 0.770 | 0.725 | 0.747 | 0.798 |
| | Hist | 0.754 | 0.713 | 0.733 | 0.789 |
| | Text | 0.599 | 0.763 | 0.671 | 0.773 |

**Table 2**. The performance of four classification algorithms on four different feature sets. The best result is obtained with SVM on all features (Precision=0.853, recall=0.851,F-value=0.852,overall accuracy = 0 .886).

To determine whether these results could be improved by co-training, we performed experiments using different numbers of training images. We used an unlabeled dataset consisting of 10,000 panels randomly chosen from the SLIF database. The same image and text features were extracted for each. In the first experiment, 50% of the labeled data were used for co-training and different numbers of iterations were repeated to expand the training set. In the second experiment, only 10% of the labeled data were used for co-training. The results are reported in Table 3. When the training set was 50% (about 1,000 images) co-training did not help the classification. But when only a limited number of training data (10%, about 200 images) were used, the

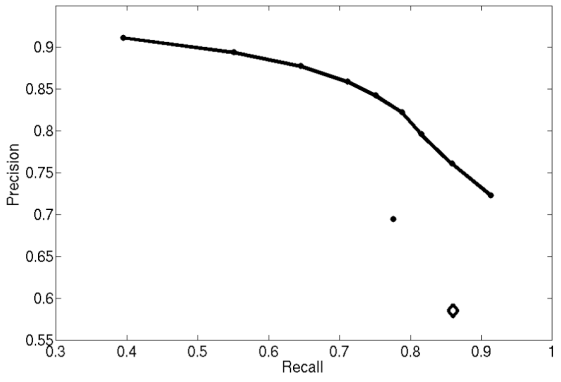


**Figure 3**. The precision and recall trade off for SVM classifier trained on both image and text features. The dot off the line shows the performance of a KNN classifier trained on histogram features only. The diamond shows the performance of the previously trained classifier on the new labeled dataset.

| Experiments | | Recall | Precision | Error Rate |
|---|---|---|---|---|
| 50% training | SVM | 0.829 | 0.836 | 0.132 |
| | Co-training | 0.826 | 0.828 | 0.137 |
| 10% training | SVM | 0.561 | 0.791 | 0.229 |
| | Co-training | 0.666 | 0.849 | 0.179 |

**Table 3**. Co-training results for different amounts of training data.

co-training algorithm clearly increases both the recall and precision compared to the case when no unlabeled data were used. However, it is still worse than the results from a large training set even without co-training. This indicates that the training examples in the large set adequately sample the types of panels in the entire dataset, and thus cotraining does not discover any variations of the original classes.

## 4. CONCLUSION

The introduction of edge and text features clearly improves the classification of FMI. The contribution is mainly an increase in accuracy with little or no loss in recall. This improvement is consistent in all four learning algorithms which have been tried in this study. The classification clearly out-performs the previous system. However, there is still room for improvement. The next step of the work is to study closely the image instances which are misclassified and to design new features which can do a better job of differentiating FMI from non-FMI.

The study of co-training shows the possibility of using the unlabeled dataset to improve the performance. However, the effect is only obvious when there are very limited amount of labeled data. When the labeled set is sufficiently large to cover the class distribution in the feature space, co-training can do very little to help and sometimes even decreases the performance due to the uncertainty of unlabeled data. However, cotraining might be helpful when a new dataset is analyzed, such as images from a different journal or research field.

## 5. ACKNOWLEDGEMENT

This work was supported in part by NIH grants R01 GM078622-01 and K25 DA017357-01. O.N.A. was supported by a Summer Scholar award from the Merck Computational Biology and Chemistry Program made possible by a grant from the Merck Company Foundation.

## 6. REFERENCES

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys (CSUR) 34:* 1-47, 2002.

[2] A.M. Cohen and W.R. Hersh, "The Trec 2004 Genomics Track Categorization Task: Classifying Full-Text Biomedical Documents," *Journal of Biomedical Discovery and Collaboration 1*, 2006.

[3] R.F. Murphy, Z. Kou, J. Hua, M. Joffe, and W.W. Cohen, "Extracting and Structuring Subcellular Location Information from on-Line Journal Articles: The Subcellular Location Image Finder," *Proceedings of the IASTED International Conference on Knowledge Sharing and Collaborative Engineering (KSCE 2004),* pp. 109-114, 2004.

[4] Z. Kou, W.W. Cohen, and R.F. Murphy, "Extracting Information from Text and Images for Location Proteomics," *Proceedings of the 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD03)*. pp. 2-9, 2003.

[5] R.F. Murphy, M. Velliste, J. Yao, and G. Porreca, "Searching Online Journals for Fluorescence Microscope Images Depicting Protein Subcellular Locations," *Proceedings of the 2nd IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE 2001)*. pp. 119-128, 2001.

[6] B. Rafkind, M. Lee, S.-F. Chang, and H. Yu, "Exploring Text and Image Features to Classify Images in Bioscience Literature," *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06*. pp. 73-80, 2006.

[7] H. Yu and M. Lee, "Accessing Bioscience Images from Abstract Sentences," *Bioinformatics 22:* 547-556, 2006.

[8] H. Shatkay, N. Chen, and D. Blostein, "Integrating Image Data into Biomedical Text Categorization," *Bioinformatics 22:* 446-453, 2006.

[9] R.F. Murphy, M. Velliste, and G. Porreca, "Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images," *Journal of VLSI Signal Processing 35:* 311-321, 2003.

[10] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning 20:* 1-25, 1995.

[11] Y. Freund and R.E. Shapire, "Experiments with a New Boosting Algorithm," *Proceedings of the 13th International Conference on Machine Learning*. pp. 148-156, 1996.

[12] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proceedings of the 11th Anual Conference on Computational Learning Theory (COLT 98)*. pp. 92-100, 1998.