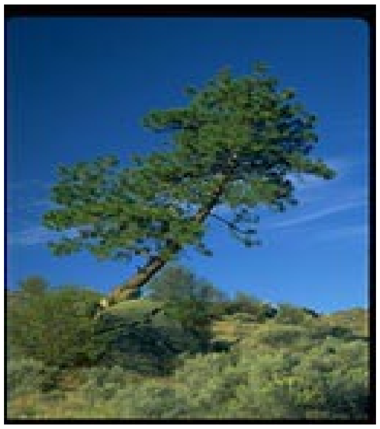
final report for

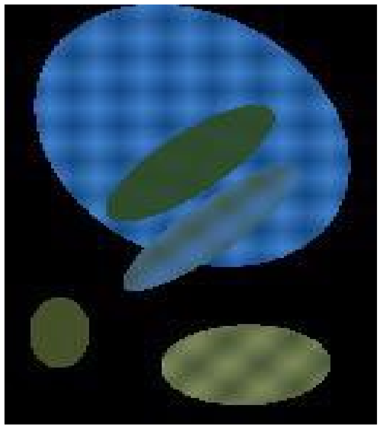
Applications of High Performance Statistical Modeling to Image Analysis of Forest Structure

submitted to:

Mr. Victor A. Rudis
USDA Forest Service
Southern Research Station
P. O. Box 928
Starkville, MS 39760-0928
Tel: 662-338-3109, Fax: 662-338-3101
Email: vrudis@usfs.msstate.edu

August 15, 1998





submitted by:

Z. Long, J. Picone
Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
Box 9571
413 Simrall, Hardy Road
Mississippi State, Mississippi 39762
Tel: 662-325-3149
Fax: 662-325-3149
Email: {long, picone}@isip.msstate.edu

EXECUTIVE SUMMARY

We had developed an extensive image database and a Principal Components Analysis (PCA) image classification system by the end of last year, which represented the major accomplishments for our efforts on the Scenic Beauty Estimation (SBE) problem. Since then, we have continued to improve that SBE system. Our focus has now shifted from SBE classification to object recognition, a more challenging task.

We incorporated the NTSC YIQ scale and the colors brown and yellow as features, because they are also important factors in scenic image analysis. These YIQ features, however, have failed to improve performance. While the new color features of brown and yellow helped improve the performance of the PCA system. For example, blue and brown combined gave us our best result of 31.4% misclassification rate on the first data set of Pre-Phase 01 images. The improvement was justified by the fact that brown and yellow play a very important role in human perception of the scenic beauty of forestry images.

We investigated two other algorithms: Independent Component Analysis (ICA) and Support Vector Machine (SVM). These algorithms have produced excellent results in many classification problems. We implemented ICA in MATLAB, a useful mathematical software tool. For SVM, we took advantage of a public-domain program, *SVMlight*. ICA combined with the feature set of “rgb” (red + green + blue) gave us an overall error rate of 33.4% on Pre-Phase 01 data. SVM gave an average of 32.2% error on the same database when we chose the feature set to be “rgb + long lines.” Compared with the best error performance of the PCA system, which was 37.7% achieved by using a feature vector of “rgb + entropy”, the improvement was significant.

Another objective of this project is to recognize important objects from a forestry scene. As a precursor to real object recognition, we focused on the image segmentation problem. We developed a front end tool, ISIP Object Labeler, to aid in manually segmenting a total of 1784 images. Then, before we proceeded to implementing an automatic segmentation system, we carefully designed two kinds of preliminary experiments to see how well the system would work. For the first one, we split images horizontally into three equal strips and applied our PCA system on only one of the strips. It was encouraging to observe that we obtained much better results on some specific data sets, such as an error rate of 27.8% generated by examining the top region using features of “rgb” combined with entropy on data set 2. In the second kind of so-called “cheating” experiments, where we trained and tested the images on the regions manually segmented out, although performance did not improve significantly, we did find better classification results on some specific regions.

We built the automatic segmentation system on the basis of the software resources we created for the classification task. Using the same object-oriented philosophy, we added new classes and changed the process flow accordingly. A significant difference of this system from the classification system is that we are now processing images at a block level. So far, we have implemented a first draft of this segmentation software. The best evaluation result was a 61.4% error performance. A careful analysis revealed that the high error rates were due to a bad classifier for foliage blocks. Therefore, we currently are searching to find a good foliage classifier to improve the system’s performance. A more careful analysis of the effect of block sizes will be carried out. During the next year of the project we expect to use more powerful classification paradigms like Hidden Markov Models to aid automatic segmentation and object recognition.

TABLE OF CONTENTS

ABSTRACT

The objective of this project is to automate the extraction of diversified information from forestry images in order to help analyze forest structure. By the end of last year, we had developed an extensive image database and an image classification system dealing with the problem of Scenic Beauty Estimation (SBE) [1]. Since then, we have worked mainly in two directions. One direction involved improving the Principal Components Analysis (PCA) classification system we already developed. The second direction involved implementing an automatic image segmentation system. In the first portion of this research, we integrated new features of NTSC YIQ [2], brown and yellow [3] into the system. We also investigated Independent Component Analysis (ICA) and the Support Vector Machines (SVM) [4]. In the second part of this research program, we developed a manual segmentation tool and created a manually segmented image database with the aid of that tool [5]. Based on this database, we implemented an automatic image segmentation system. Now we are in the process of improving that system.

1. INTRODUCTION

The need for automatic determination of the scenic beauty of a forest scene has been the prime source of motivation for this project. In our prior efforts, a PCA system was built, where commonly used features like RGB colors, entropy, number of trees, etc., were used to classify images according to their scenic beauty quality. Meanwhile, an extensive image database has been developed to assist in the classification [1].

This project is a continuation of our previous work. OUr primary goal is to expand the research scope and use various signal processing techniques to extract automatically more diversified information from forestry images. To this end, we focused our efforts mainly on two aspects. First, we investigated some new features and several advanced pattern recognition algorithms in an effort to improve the PCA system we developed. We incorporated YIQ, another common color model widely used in television broadcasts, which is different from the RGB model we used before. We adjusted the normal method of histogram computation to meet the special case for YIQ and used the histogram as features. But the outcome from these features was not as good as what we obtained from the RGB model. We also computed brown and yellow colors by combining red, green and blue in some particular manner according to the additive color theory. As shown in the research paper [7], these colors are determinative in human perception of forestry scenery. Our experiments proved this by yielding quite a few really exciting classification results.

With respect to algorithm research, we investigated Independent Component Analysis (ICA) and the Support Vector Machines (SVM) [4]. ICA is similar to PCA in its linear property, but is superior in that it is not constrained to be orthogonal. We predicted that this orthogonal-free property would help to produce a better discrimination among the features, hence a better classification. This was confirmed by the significant drop in error classification rate provided by our ICA system. Another promising algorithm widely used in image classification problem is SVM. It was giving excellent performance in nonlinear problems such as human face recognition. Since we believed that this SBE problem would more likely be nonlinear in nature than linear, we built an SVM system with the aid of a public-domain software tool (*SVMlight*). The overall error performance was indeed much better than that of the PCA system. It was interesting to note that there were only slight differences between the performance of ICA and SVM.

Presently, our efforts are focused on recognizing important objects from a forest scene to assist in scenic beauty estimation, as well as forest structure analysis. Our initial attempt involved segmenting images automatically. To prepare necessary data, we developed a segmentation tool with which we manually segmented the USFS database of 1784 images into various regions typical of any forestry scene, such as trees, foliage, bushes, grass, sky, and background sky. Then we designed a series of cheating experiments using these manual segmentations as a first step toward the automatic segmentation. We observed some outstanding performance of the SBE system on that manual segmentation data, which implied promise for a successful automatic segmentation system. Consequently, we implemented a block-level classification system, which was capable of classifying image blocks into one of the six regions we defined in manual segmentation. The system was worked reasonably well, although lots of important issues need to be settled before it will perform in a satisfactory manner. We are currently in the process of improving this automatic segmentation system. We expect to make progress towards a real object recognition system in the near future.

2. SCENIC BEAUTY ESTIMATION

One of the major goals of this project is to assess automatically the scenic beauty quality of forestry images. We had done extensive research on this topic previously. Our accomplishments on this can be best represented by the PCA classification system and the USFS image database we created. However, there were still some important issues remaining beyond those achievements; hence, there is a need to look into those issues and, accordingly, to strengthen the power of the existing system.

2.1. New Features

For the new features, we mainly studied the color model of YIQ and the color features such as brown and yellow. We did these things because colors are supposed to be one of the most important features for forestry image analysis.

2.1.1. YIQ

We built our PCA classification system on the commonly used RGB scale. To study the benefits of using alternate color models, we integrated the luminance-chrominance scale of YIQ as features. YIQ is widely used in television broadcasts, where Y stands for luminance, I for hue, and Q is the saturation value.

We started by computing the YIQ values for all images across the database. We observed from the corresponding feature distributions that the I and Q values are concentrated in a very small portion of the possible ranges they could occupy. This is shown in Figure 1 for the case of I. Clearly the histogram depicts a non-uniform, Gaussian like distribution in a small range. If we do not take this into consideration when we extract the YIQ features, we would run into intractable computational problems in PCA.

As a solution to this problem, we compute the YIQ values on an exponential scale, and then

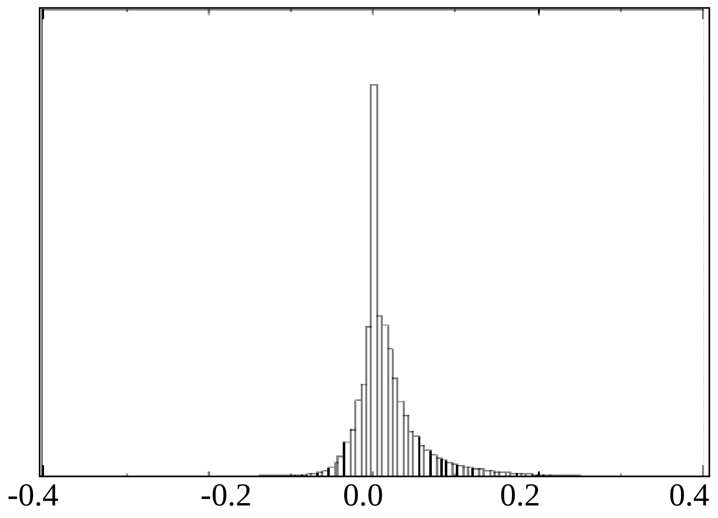


Figure 1. Histogram of I values based on the USFS training data. Note the narrow range actually occupied by the I feature.

generate a histogram of these exponentially scaled values to determine their range. A histogram for the YIQ values is recomputed on the basis of this range. This, however, does not guarantee that all the bins in the histogram are non-zero, which could potentially cause matrix-inversion problems. We employed Gaussian smoothing to alleviate this problem.

The system with YIQ values, alone, as its feature set generated a 51% classification error on the evaluation data set 1 of the Pre-Phase 01 images. As a comparison, the counterpart system, using only RGB as features, produced an error rate of 40% for the same data set. Most likely, this is an indication that the YIQ model may not be a better choice than the more commonly used RGB scale. However, further analysis is required to evaluate the efficacy of these YIQ features for scenic beauty estimation.

2.1.2. Brown And Yellow

Colors are supposed to have important effects on human perception of the scenic beauty of an image. In [7] the following rules have been demonstrated: The color green is most likely to appear in a summer scene, yellow in fall, but blue and brown have the highest chance to show in winter. In non-winter seasons, the presence of green and yellow colors enhances the scenic beauty of a forest scene, while blue and brown do the opposite. From these results, it is evident that brown and yellow are also key color features in evaluating the scenic beauty quality of an image. Therefore, we deem it necessary to add the brown and yellow features into our color measurement scheme and evaluate its effectiveness on the SBE problem.

We computed the histogram for the brown and yellow colors, as we did for red, green and blue, previously. To calculate the intensity of brown and yellow for a given pixel, we take advantage of the theory of additive color synthesis [8], which allows us to create any color by mixing the three primary colors, red, green, and blue, in various proportions.

Particularly, we generated the color brown by mixing one part blue, one part green, and four parts red. Similarly, we obtained yellow by mixing equal parts of red and green. Since we desired to measure the intensity of these colors, as we did in previous experiments for the primary colors, we implemented a normalized version of the mixing formulae:

$$brown = \frac{1}{6} \times blue + \frac{1}{6} \times green + \frac{4}{6} \times red \tag{1}$$

$$yellow = \frac{1}{2} \times red + \frac{1}{2} \times green \tag{2}$$

where red, green and blue stand for the corresponding intensity values of the given pixel.

We investigated the effects of these new features on our PCA image classification system. We evaluated various combinations of features involving brown and yellow, and compared these to some existing baseline results. These results are shown in Table 1. We believe it is a very encouraging sign that brown and yellow, combined with other primary colors, produce good classification results. For example, blue and brown in combination gave us our best results yet for a PCA-based measure on the first data set. We believe this can be attributed to the fact that blue is an indication of the presence of sky, while brown is related to the presence of trees in the image.

Although the excellent results on the first data set did not hold up well over the entire evaluation, we believe this probably resulted from the varied nature of the test sets with respect to scene composition. Therefore, if we are doing classification on regions of similar scene composition, we will benefit from these new color features. Obviously, we will go under such restrictions when we do image segmentation.

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
red + green + blue	40.3	42.4	38.1	33.1	**38.5**
yellow	54.7	38.6	45.6	67.5	51.6
brown	37.7	38.6	41.9	55.0	43.3
yellow + brown	40.3	46.2	40.6	45.6	43.2
rgb + yellow + brown	39.6	39.9	39.4	66.9	46.5
blue + brown	**31.4**	50.6	35.0	56.9	43.5
rgb + brown	47.2	46.2	48.1	38.8	45.1
grn + blue + ylw + ll	39.6	38.0	45.0	53.1	43.9
grn + blue + ylw + brn + ll	39.6	52.5	48.1	40.0	45.1
blue + brn + ll	37.1	52.5	33.1	58.1	45.2

Table 1. Results of SBE evaluations using brown and yellow.

2.2. New Algorithms

The pattern recognition algorithm involved is one of the most important elements of the image classification system. Naturally, we would like to test other algorithms than principal component analysis in the hope of finding some that give better performance. We studied several advanced algorithms and chose Independent Component Analysis (ICA) and the Support Vector Machine (SVM). We believe both algorithms are promising for the SBE problem under discussion.

2.2.1. ICA

Before we describe the principles of ICA, we would like to discuss PCA briefly, because it is closely related to ICA and it is used in the baseline system which we will use for performance comparison between various systems.

PCA is a statistical normalization approach. It is a straightforward method of classification using a linear classifier; hence, it has the advantages of linear classifiers, such as simplicity of implementation and robustness to poorly estimated statistical models [9]. In PCA, it is assumed that the first principal component of a sample vector lies parallel to the direction along which there is the largest variance over all samples. This direction corresponds to the eigenvector associated with the largest eigenvalue. The *k*th principal component is chosen to be the linear combination of the input features that has the largest variance, under the constraint that it is also uncorrelated to the previous *k*-1 principal components. This approach, depicted in Figure 2, works well when the direction along which there is maximum variation also contains the information about the class discrimination.

The principle behind ICA [10] is very similar to that of PCA, but the main difference is that the resulting transformation is no longer orthogonal. This is depicted in Figure 2. Similar to PCA, ICA finds a linear coordinate system for the data, transforming it using a linear matrix transformation. Hence, the complexity of the runtime analysis is equivalent to PCA.

However, unlike PCA, the transformation is free to be non-orthogonal, as shown in Figure 2. Furthermore, the computation of this transform involves higher order statistics of the data. In the PCA case, which involves an underlying assumption that the data is multivariate Gaussian, only

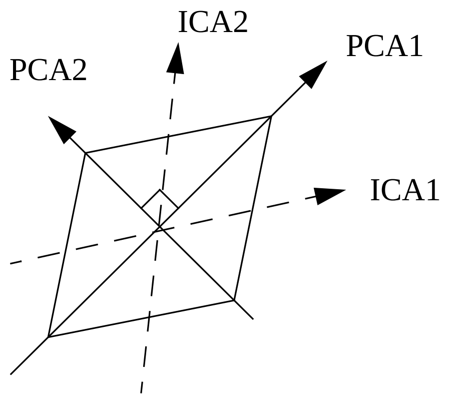


Figure 2. A comparison between transformations found by PCA and ICA when the data is uniformly distributed within the diamond-shaped region.

the mean and covariance (second-order statistics) are used.

Our approach to finding the ICA transformation, described in [11], involves finding a linear transformation $\overline{W}$, that minimizes the mutual information of the transformed data. Since a closed-form solution is not tractable, we use a stochastic gradient ascent algorithm. When the nonlinear function is the same (with respect to scaling and shifting) as the cumulative density functions of the independent components, it can be shown that a nonlinear infomax procedure also minimizes the mutual information between the components of the transformed data.

However, in practice, we must pick a nonlinear function for the stochastic ascent without any detailed knowledge of the probability density functions of the underlying independent components. Our procedure is to maximize $H[g(u)]$, where $g(\mu)$ is a sigmoidal function, and to assume the pdf's of μ are super-Gaussian (similar to a Gaussian, but more "peaky" with longer tails). This approach leads to an ICA solution when such a solution exists.

The specific form of the stochastic gradient ascent that is used involves a learning rule that changes weights according to the gradient of the entropy [11]:

$$\Delta W \propto \frac{\partial H(y)}{\partial W} W^T W = (I + \hat{y} u^T) W$$

in which $\hat{y}_i = \frac{\partial}{\partial y_i} \frac{\partial y_i}{\partial u_i} = \frac{\partial}{\partial u_i} In \frac{\partial y_i}{\partial u_i}$ is computed as $\hat{y}_i = 1 - 2y_i$, and $y_i = (1 + e^{-u_i})^{-1}$.

Our implementation of ICA was based on the information maximization theory. If the output entropy is maximized, then the components of the output vector must be statistically independent.

Assuming we have a training set $\{x\}$. We need to find a transformation $u = Wx$, with the properties mentioned above. First, the data x is prewhitened by $\{x\} \leftarrow 2W_z(\{x\} - \langle x \rangle)$, where $W_z = \langle xx^T \rangle^{-1/2}$. The matrix is then initialized to the identity matrix, and trained accordingly.

We use a context-dependent transformation in this case. Hence, three separate transformations are trained, and the test images are classified by using a class-specific Euclidean distance computation in the transformed space.

We evaluated ICA on the four standard subsets of the 637 Pre-Phase 01 images as we did for all previous evaluations. Among the 45 features (new features discussed earlier not included), we examined the following combinations:

- all: all 45 features
- rgb: red + green + blue
- rgb + ll: rgb + long lines
- rgb + ll + ent: rgb + ll + entropy
- rgb + ll + fract: rgb + ll + ent + fractal dimensions.

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
all	35.2	33.5	31.9	33.1	33.4
rgb	34.6	33.5	32.5	33.1	33.4
rgb + ll	34.6	33.5	33.1	33.1	33.6
rgb + ll + ent	34.6	34.2	33.1	32.5	33.6
rgb + ll + fract	35.9	33.5	32.5	33.1	33.8

Table 2. Error performance of ICA on all four evaluation data sets.

The results are shown in Table 2. The best error rate achieved by ICA is 33.4%, using a simple combination of R, G, and B. This is extremely promising in that it utilizes the simplest feature vector, and seems to learn all other derived information, such as entropy and fractal dimension.

2.2.2. SVM

A support vector machine (SVM) [12] is a new technique for classification. The main objective of this technique is to construct an optimal hyperplane, which uses a small part of the training set as support vectors. If the training vectors are separated without errors by an optimal hyperplane, the expected error rate on a test sample is bounded by the ratio of the expected number of the support vectors to the number of training vectors. Since this ratio is independent of the dimension of the problem, good generalization is guaranteed if we can find a small set of support vectors.

First, we will have to look at the simplest case: a linear machine trained on separable data. Suppose we have 2 classes. Labeling the training data as $\{x_i, y_i\}$, $i = 1, \ldots, l$, $y_i \in \{-1, 1\}$, $x_i \in R^d$, we would like to find a separating hyperplane $w \cdot x + b = 0$, where w is normal to the hyperplane. This hyperplane separates the positive examples from the negative ones. Let d_+ (d_-) be the shortest distance from the separating hyperplane to the closest positive (negative) example, then $d_+ + d_-$ is called the “margin” of the separating hyperplane.

Here, the support vector machine just looks for the separating hyperplane with the largest margin. This has been shown [13] to be equivalent to the following problem. Given the inequalities:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \qquad \forall i$$

find hyperplanes of H$_1$: $x_i \cdot w + b = 1$ and H$_2$: $x_i \cdot w + b = -1$, which give the maximum margin by minimizing $||w||^2$, subject to the constraints shown in the inequality above.

To solve the problem using the Lagrangian method, we have to maximize $L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$ with respect to α_i, subject to constraints $\sum_i \alpha_i y_i = 0$ and the positivity of α_i [13]. The solution is given by $w = \sum_i \alpha_i y_i x_i$.

Once a support vector machine has been trained in this way, a test vector may be classified by simply determining on which side of the decision hyperplane it falls. For non-separable data, the constraints should be modified as follows:

$$x_i \cdot w + b \geq 1 - \xi_i \qquad y_i = 1$$

$$x_i \cdot w + b \leq -1 + \xi_i \qquad y_i = -1$$

$$\xi_i \geq 0 \qquad \forall i$$

The solution is similar to that described previously.

The above SVM algorithm on a linearly separable data set can be extended to nonlinearly separable data. The general idea is that we can map the nonlinearly separable data into a higher dimensional space in which we can still do a linear separation on the mapped data. To do the mapping, we need to find a suitable “kernel function.” A detailed discussion of this approach is found in [13].

Our SVM implementation follows the three-way classification strategy similar to what was done with ICA. One classifier is trained for each of the three classes. Each classifier is trained by considering the in-class data versus the rest of the data. For each test image, we calculate the distance of its feature vector to each of the three classifiers. Classification was achieved by comparing the distances of each image’s vector from the three hyperplanes.

The SVM implementation was accomplished using the public-domain *SVMlight* [8] package. The radial basis function kernel was chosen because it was found to give the best performance on a number of related classification tasks.

To facilitate a comparison between all the algorithms, we evaluated SVM the same way we did for ICA, *i.e.*, we chose the same evaluation data sets and feature sets. The results are described in Table 3. We achieved a 32.2% overall error rate by SVM, using the feature combination of rgb plus long lines. We also noted that the results on this feature set display a very small variance across all the data sets. This indicates that SVM makes effective use of long line information in combination with RGB.

A comparison of ICA and SVM to the baseline PCA system is shown in Figure 3. We see from the plot that ICA and SVM provide improved performance over PCA. The best overall error

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
all	34.6	34.2	31.2	31.9	33.0
rgb	35.2	34.8	32.5	32.5	33.8
rgb + ll	32.1	32.3	31.9	32.5	32.2
rgb + ll + ent	35.2	34.2	31.9	32.5	33.4
rgb + ll + fract	35.2	35.2	32.5	31.2	33.5

Table 3.Error performance of SVM on the same four data sets.

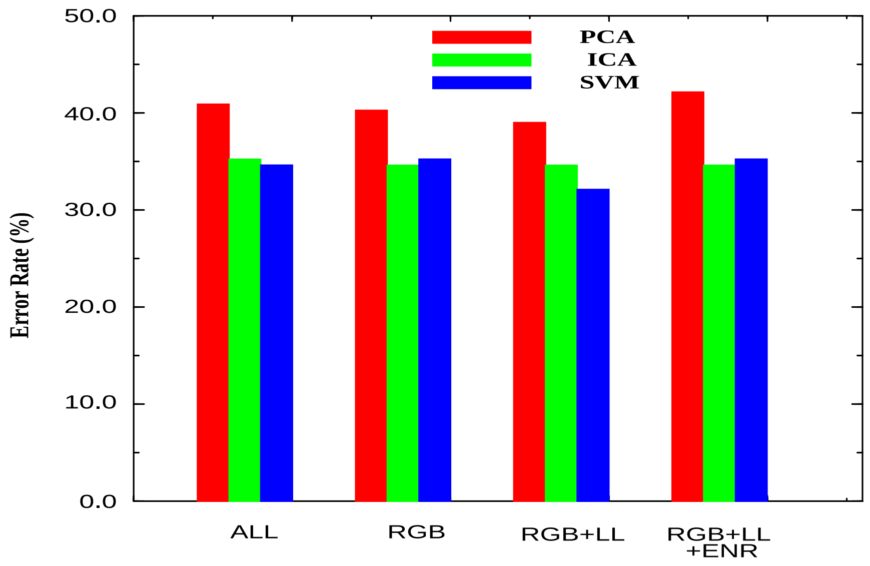


Figure 3. A comparison between ICA, SVM and PCA on evaluation set 1 of the Pre-Phase 01data.

performance of ICA is 33.4%, while for SVM, it is 32.2%. Both are superior to the best results we have obtained with PCA, which is 37.7% with rgb combined with entropy as the feature set. Neither ICA nor SVM is better on all experimental conditions, however. This suggests there is still some work ahead on dimensionality reduction and feature set enhancement. These encouraging results also suggest a possible system, where we may combine these two techniques to produce a better classification scheme. By using ICA as a front-end to SVM, we will be able to supply optimally separated distributions to the SVM, thereby constructing a better nonlinear decision region. We expect that the classification capability of the combined system will be a great improvement over the system using either technique, alone. It is important to note that, as encouraging as these numbers are, the performance of intelligent guessing (always guess msbe) is still extremely close to our best classification system.

3. MANUAL SEGMENTATION

Since most object recognition algorithms need transcribed training data to build statistical models, we segmented the digital images into different regions and labeled these regions as tree, bush, grass, etc. We developed a software tool to facilitate this task. A total of 1784 images, consisting of all USFS images processed to date, has been manually segmented. This large database of segmented images will form the basis for the development of advanced object classification algorithms. As a first step, we designed some preliminary experiments to gain insight into the image segmentation problem.

3.1. Manual Segmentation

We have developed a segmentation tool named Image Object Labeler (IOL). This tool was rapidly prototyped in Tcl/Tk, and is fairly portable across Unix and PC environments. It was designed to allow a user to select a certain type of object, and to draw a closed polygon around the region defining that object. Each image can be labeled multiple times. A list of images is traversed using a scrolling dialog box. This allows the user to iterate on a set of images until all segmentations are consistent. An example of a labeled image is shown in Figure 4.

For now, we have defined six labels for the image transcription. These labels which are necessary for transcribing forestry images are described in Table 4.

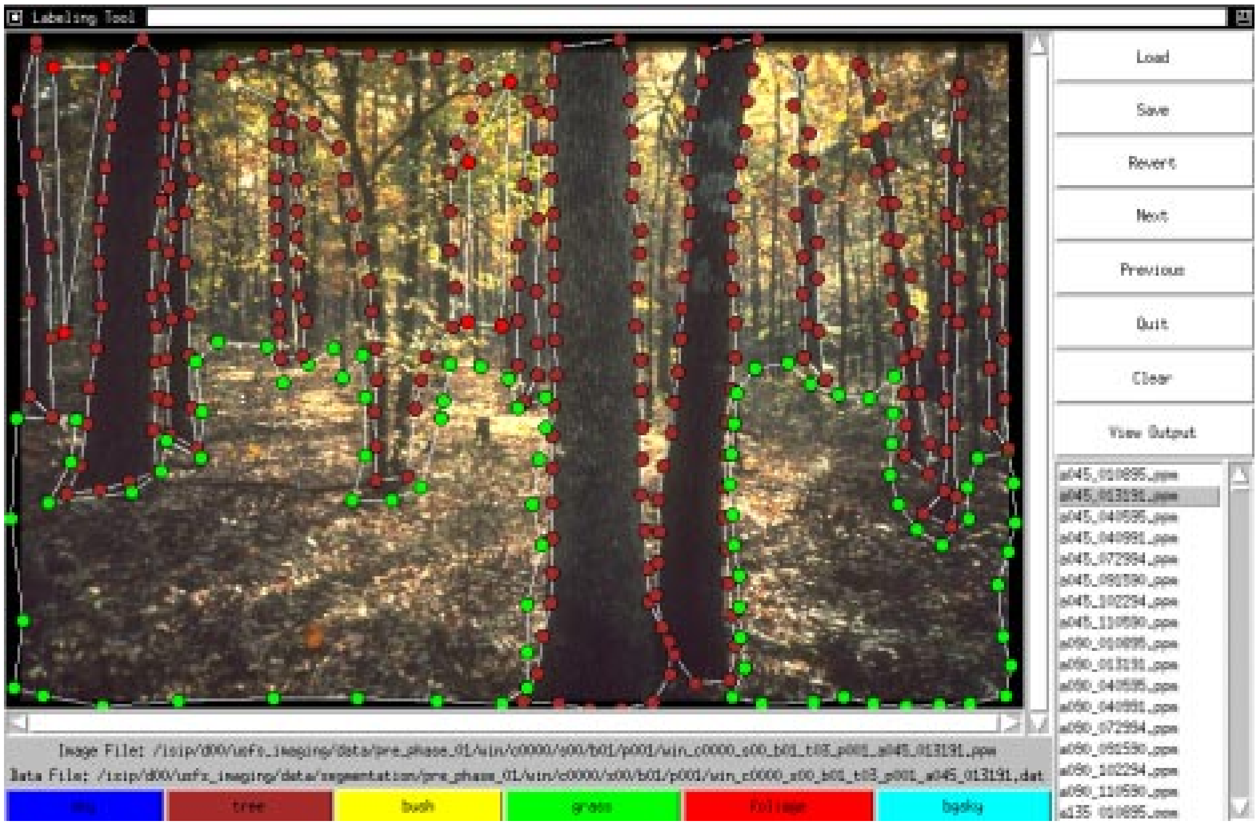


Image File: /isip/d00/usfs_imaging/data/pre_phase_01/win/c0000/s00/b01/p001/win_c0000_s00_b01_t03_p001_a045_013191.ppm
Data File: /isip/d00/usfs_imaging/data/segmentation/pre_phase_01/win/c0000/s00/b01/p001/win_c0000_s00_b01_t03_p001_a045_013191.dat

Figure 4. An example of a labeled image using our segmentation tool.

Label	Description
tree	a woody perennial plant having a single, usually elongated, main stem, generally with few or no branches on its lower part
foliage	a cluster of leaves, flowers, and branches
bush	a low densely branched shrub or a close thicket of shrubs suggesting a single plant
grass	grass or land covered with growing grass, or other green-colored ground cover
sky	a region typically containing sky and few obstructions such as trees; strictly limited to the top of the image
background sky	a region with sky in the background and other objects in the foreground, but still clearly identified as sky

Table 4. The six categories used to label images.

The segmentation process begins by creating a list of images to be segmented. The button named "load" is used to load both the image file list to be labeled and the data file list which will keep the results of the labeling. When an image is chosen and displayed, we can label objects as listed at the bottom of the window. As an example, to label a tree in the image, just click the button named "tree" first, then proceed to draw a polygon around the tree. After finishing labeling all objects in the image, use the "save" button to write the data into the corresponding data file. Then click "next" to get the next image file in the file list, or "previous" to return to the previous image file. In the labeling process, we use "clear" to clear all the labels and use "revert" to go back to the most recently saved file. The "view output" button provides a corresponding look at the output data. As we can see, the user interface is simple and intuitive.

The time needed to segment an image varies from image to image. For some simple images which just contain one or two objects, such as a foliage plus a background sky, only 2 minutes are required. On the other hand, for the really complicated images, it takes as much as 8 minutes to label all the objects in one image. The average time used to segment an image is 4 minutes or so. This implies it will take about 120 hours to segment the entire database, or approximately 6 weeks at 20 hours per week.

An example of the data file format is shown in Figure 5. The data file begins with the name of the image file, followed by the object name, then the coordinates of all the label points used to mark this object. There can be multiple instances of any object in the file, and the order of the objects is arbitrary (each object is written in the order it was created). The description of an image is terminated by either a new image file name or the end of the file.

With the tool introduced above, we manually segmented all USFS image data that was provided. The images are located under the directories /isip/d00/usfs_imaging/data/pre_phase_01/ and /isip/d00/usfs_imaging/data/phase_01/, respectively. The first set consists of 637 images taken from the Winona Ranger District. The latter set consists of 1147 images taken from the Ouachita and Ozark National Forest. There is a total of 1784 images.

```
/isip/d00/usfs_imaging/data/phase_01/cd_0012/img0002.ppm
Class tree
954  32
954  386
...
1106  30
Class foliage
186  32
...
```

Figure 5. An example of the output file format used by the segmentation tool to store image segmentation information. A simple ASCII file format is used for portability.

3.2. Preliminary Experiments

Once we have created the manually segmented image database, we are ready to begin development of a system capable of leveraging our manual segmentations. This is being done in two steps. First, we want to assess the effectiveness of this information through “cheating” experiments where we estimate scenic beauty given knowledge of the segmentations. Second, we will develop schemes to automatically segment the images, and evaluate the accuracy of these schemes based on the manually derived information. Here, we review progress on the former task. We leave the automatic segmentation for the later part of this report.

In our manual segmentation approach described previously, each image was segmented into regions of similar appearance, and assigned one of the following tags: tree, foliage, bush, grass, sky and background sky. Given the striking visual differences between most of these regions, we believe that if we build region-specific models, and classify images according to these regions, we will attain a significant improvement in our ability to automatically classify forestry images. However, instead of proceeding directly to implementation of such a segmentation-based scenario, we first need to evaluate the theoretical limits such approaches can achieve. We refer to this type of experiment as a “cheating” experiment, since we assume the manual segmentation information is already known to the system during the classification process.

To implement a cheating experiment, we need to incorporate manual segmentations into the system. This requires non-trivial modifications to our software. Hence, we designed a simple image splitting scheme as a first step: each image was equally split into three horizontal strips. The top region was designated “sky.” The middle region was designated “trees and foliage.” The bottom region represented “grass and ground.” We evaluated our ability to classify each image solely based on one of these regions using the feature vector “blue + brown.” We compared these results with those generated on the non-split images, or the entire images. The comparisons are shown in Table 5. We still used a PCA based approach for all these experiments.

Observe that the average performance across all data sets for the top region is slightly better than a comparable result for the entire image. In other words, by examining only the top one-third of each image, using features of blue and brown, we were able to achieve our best overall performance. Clearly, this approach was doing a better job of weighting the relative importance of such features as sky and brush.

Region	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
top	44.0	38.0	43.8	45.0	**42.7**
middle	37.7	46.1	71.3	60.0	52.5
bottom	50.3	35.4	53.1	42.5	45.3
entire	31.4	50.6	35.0	56.9	**43.5**

Table 5. Results of SBE evaluations using “blue + brown” features on equally split image regions.

To further explore the significance of this result, we continued experiments using region-specific features. For each region, we chose features believed to be relevant to the type of imagery expected in that region. These experiments are summarized in Tables 6-8. It is interesting to note that we obtained much better results on some specific data sets. For example, by examining the top region using features of rgb combined with entropy, we achieved a 27.8% error rate on data set 2, which is the best performance obtained on any data set thus far. Building on this encouraging result, we chose the best error rates for each region and compared them with the best average error rate obtained previously for the entire images. This is summarized in Table 9.

Obviously, the best results of this region-specific measurement scheme were comparable to those of our previous best systems that processed the entire images. Though this approach was a very simple first step, we believe it justifies examining segmentation-based approaches, or the cheating experiments mentioned earlier, in more detail. To carry out the cheating experiments, where we classify images only on a specific region prescribed by the user, we need to add the following capabilities to the software: reading in the segmented regions as polygons; running evaluation only on the prescribed region. Therefore, we designed and implemented a polygon class, which

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
blue	48.4	31.6	43.1	50.0	43.3
rgb	51.6	30.4	66.9	36.9	46.5
rgb + ent	54.1	27.8	30.6	35.6	37.0

Table 6. Results of SBE evaluations using only the top region of each image.

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
green	49.7	58.9	65.0	60.6	58.5
brown	44.7	57.0	66.9	51.9	55.1
rgb	43.4	40.5	39.4	41.9	41.3
rgb + ent	42.8	40.5	31.3	40.6	38.8

Table 7. Results of SBE evaluations using only the middle region of each image.

Features	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Average
brown	37.7	65.8	46.3	41.3	47.8
ylw + brn	44.0	52.5	45.0	41.3	45.7
rgb	45.3	32.9	41.3	33.1	38.2
rgb + ent	38.4	32.3	43.8	36.9	37.9

Table 8. Results of SBE evaluations using only the bottom region of each image.

Region	Features	Error Rate
top	rgb + ent	37.0
middle	rgb + ent	38.8
bottom	rgb + ent	37.9
entire	rgb + ent	37.7

Table 9. A comparison of performance using region-specific measures to a baseline system.

can load the manual segmentation data and determine if a given pixel is inside a certain polygon or not. Further, we modified the image_analysis class to incorporate this polygon class, which mainly involved extension of the computational methods. Also, we made other necessary changes to the driver program and the parameter files to support experimentation on either a specified region or an entire image.

We carried out a series of cheating experiments with several combinations of features. The experiments were based on data set 1 of Pre-Phase 01 images. Overall, running our standard SBE classification system on a particular region, such as background sky, produced worse results than running the same system on entire images did. This is clearly revealed by the results in Table 10. The reason is most likely that the feature sets we tested were not doing as good a discriminating job on the specific regions as they did on the entire images. What is interesting, however, is that using a feature of blue plus brown on regions labeled background sky (bgsky) halves the errors for HSBE image classification. Here, we show the confusion matrix generated from this experiment in Table 11. For comparison, we also present in Table 12 the corresponding matrix we got from the experiment with the same feature set on entire images. Such significant reduction in errors for individual classes was also found in other evaluation experiments using different feature sets (Refer to the appendix for detailed information.). This observation shows that, if we can use region-specific images, and do a good job of automatically segmenting images, we can probably do much better in characterizing the images. With proper region-specific classifiers, we believe that this 50% reduction in error rate can hold up. This encouraged us to proceed to the implementation of an automatic image segmentation system.

Features	Tree	Foliage	Bush	Grass	Bgsky	Entire Image
rgb + ent	47.8	46.5	52.8	47.2	55.3	**39.6**
rgb + ent + ll	48.4	49.1	51.6	46.5	57.2	**42.1**
blue + brown	45.9	45.3	54.7	42.1	56.6	**31.4**

Table 10. Results of the cheating experiments.

Automatic	LSBE	MSBE	HSBE
LSBE	5	12	0
MSBE	19	45	9
HSBE	4	46	19

Table 11. Confusion matrix for the cheating experiment on background sky region with blue plus brown as features. Note the errors for the HSBE image classification decreased by half.

Automatic	LSBE	MSBE	HSBE
LSBE	2	2	0
MSBE	24	97	18
HSBE	2	4	10

Table 12. Confusion matrix for the experiment on entire images with blue plus brown as features.

4. AUTOMATIC SEGMENTATION

Automatic segmentation of forestry images is a very important step in the progress towards our goal of object recognition. We were supposed to build up the automatic system on the basis of all previous work on manual segmentation. We first implemented the system at a block level; *i.e.*, we processed the images block by block, classifying each block into one of the prescribed regions. The evaluation results indicated that much improvement was needed to get the system working as we expected. Currently, we are in the process of system analysis and improvement.

4.1. Software Design

Our initial attempt at the automatic segmentation task is to segment an image on a block-by-block basis. First, we estimate models at a block level for all six regions available from the manual segmentation. Then we classify each block into one of the six regions. Though we are still using the PCA algorithm for this block classification, we need to revise our PCA code, because we implemented the algorithm on an image-by-image basis previously, and that made a somewhat significant difference in the process flow of the software.

To keep as much consistency with the previous code as possible, we modified the class of image_analysis slightly, overloading some computational methods to deal with the case of block-level computation. And we designed a new class named image_segmentation. This class is responsible for all issues specific to block-based training and testing. We also introduced the concept of windowing into the segmentation system. After the image under investigation has been divided into equally-sized blocks, or frames, we window each frame with a rectangular window. For simplicity, we zero-pad those windows which happen to fall on the edge of the image. This windowing capability has been integrated into the image_segmentation class. Another issue involved in block-level computation is how we determine if a block belongs to a specific region or not. This concerns us because, in many cases, there will be only part of a block falling into the supposed region. What we do in this system is to simplify the problem, deciding only if the block center is inside the region. We also had to design a new utility named segment_image, by which we were able to accomplish the automatic segmentation task. To show the segmentation results intuitively, we write the block classification results into an image file, where we display colorful blocks, with each color mapping to one specific region.

4.2. Performance

To evaluate the performance of the segmentation system, we need to be concerned about not only the feature and data sets as we did for SBE classification system evaluation, but also the windowing issue, as well. For an initial test, we set a frame size of 64 by 64. We believe this frame size is suitable for the ppm images sized at 1536 by 1024. As to the window size, we need to investigate extensively to find an optimal value. However, it is reasonable to try a 50% overlap between neighboring windows at the beginning stage. We thus chose the window size to be 128 by 128. Under this windowing scheme, we evaluated the automatic segmentation system with three different feature sets on set 1 of the Pre-Phase 01 data. Here, we compared the automatic segmentation results to our reference manual segmentations. The best segmentation result we achieve is a 61.4% block classification error rate by using blue combined with brown as the feature set. Besides investigating different feature vectors, we also studied the effect of the window size on the classification error rate. We repeated the experiment with the feature set of blue plus brown for a smaller window size of 96 by 96, which gave us a window overlap of 33.3%. The evaluation results are shown in Table 13. Also a segmentation example is shown here in Figure 6. For more such examples, please see the appendix.

Although the segmentation system did not show very good performance in terms of the block classification error rate, from the more intuitive segmentation images, we do note a trend toward a successful system. As we see in Figure 6, the blocks were generally classified in a reasonable manner, with sky and background sky in the top, grass in the bottom, etc. This is a good sign.

To improve the system performance, we explored further to see what was responsible for the high error rates. When we examined the confusion matrices from the above experiments (refer to the appendix for details), we noticed that a significant portion of the errors occurred within the foliage blocks. Currently, for each specific region, the errors made by the best system, which uses rgb combined with entropy as its feature vector, can be classified as: tree - 47.3%, foliage - 86.1%, bush - 44.1%, grass - 52.9%, background sky - 26.6%, sky - 10.0%. Obviously, a better foliage classifier will be the key to the improvement of the system. To this end, we studied the foliage classifier from a statistical viewpoint. In our PCA system, we compute distances, or scores as they

Features	Frame Size	Window Size	Error Rate
blue + brown	64 x 64	128 x 128	70.8
blue + brown + sl	64 x 64	128 x 128	70.2
rgb + ent	64 x 64	128 x 128	61.4
blue + brown	64 x 64	96 x 96	72.0

Table 13. Evaluation results for the block-level automatic segmentation system.

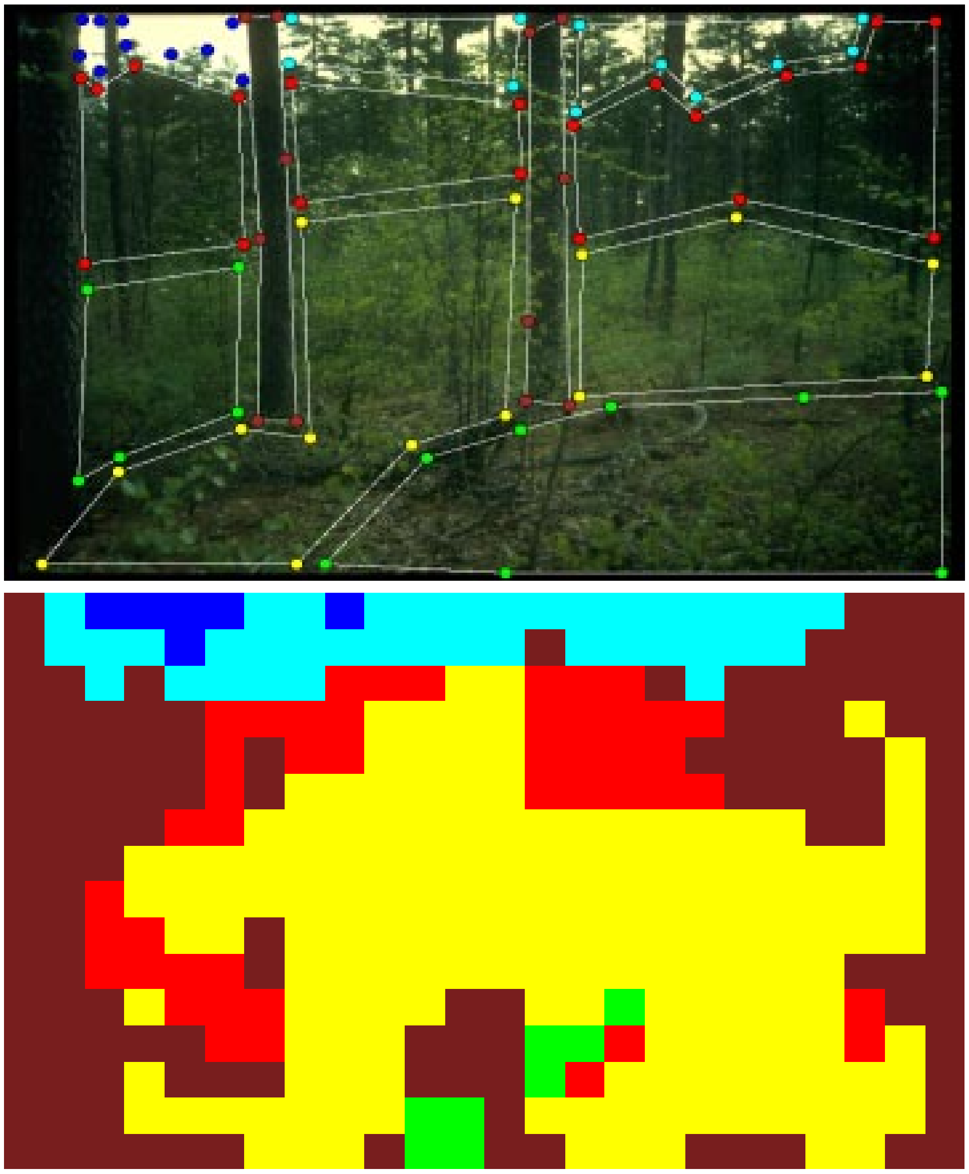


Figure 6. An example of the automatic segmentation. The manual segmentations have been added to the original image. In both manual segmentations and automatic segmentations, we use various colors to mark different regions. The mapping relationship is: brown - tree, red - foliage, yellow - bush, green - grass, dark blue - sky, light blue - background sky. Note the accuracy of the segmentations for sky and background sky.

are known in pattern recognition research, of a test feature vector to all the reference vectors and assign the block a label corresponding to the reference it is closest to. We computed the distribution of scores the foliage classifier generated in each experiment, false-alarms and accuracy. The plots look very similar. We present one plot here in Figure 7. The remaining plots can be found in the appendix. From these plots, it is clear that the foliage classifiers we used in these experiments are not capable of discriminating foliage blocks and non-foliage blocks. This explains the high errors which occurred. Currently, we are computing all 65 features on the foliage regions and studying their statistical characters. We would like to choose those significant features for the foliage classifier that can help better discrimination. If PCA can not succeed in this, we will try better techniques such as Linear Discriminant Analysis (LDA).

From the results, we also note that the window size does not have a significant bearing on the classification rate. By decreasing the window size from 128 by 128 to 96 by 96, we observe an increase of 1 percent in the error rate. That should not be deemed significant. Therefore, our efforts to improve the system will focus on researching feature sets.

5. SUMMARY

We have improved our PCA classification system on the SBE problem. This involved research on new features and new algorithms. We have used an alternate chrominance-luminance color model, YIQ, for the classification of images. The results indicate that the YIQ model may not be a good replacement for the more commonly used RGB color model. We have studied a new color computation and classification scheme using brown and yellow. These new color features provided performance comparable to our previous best systems. We have also applied two new

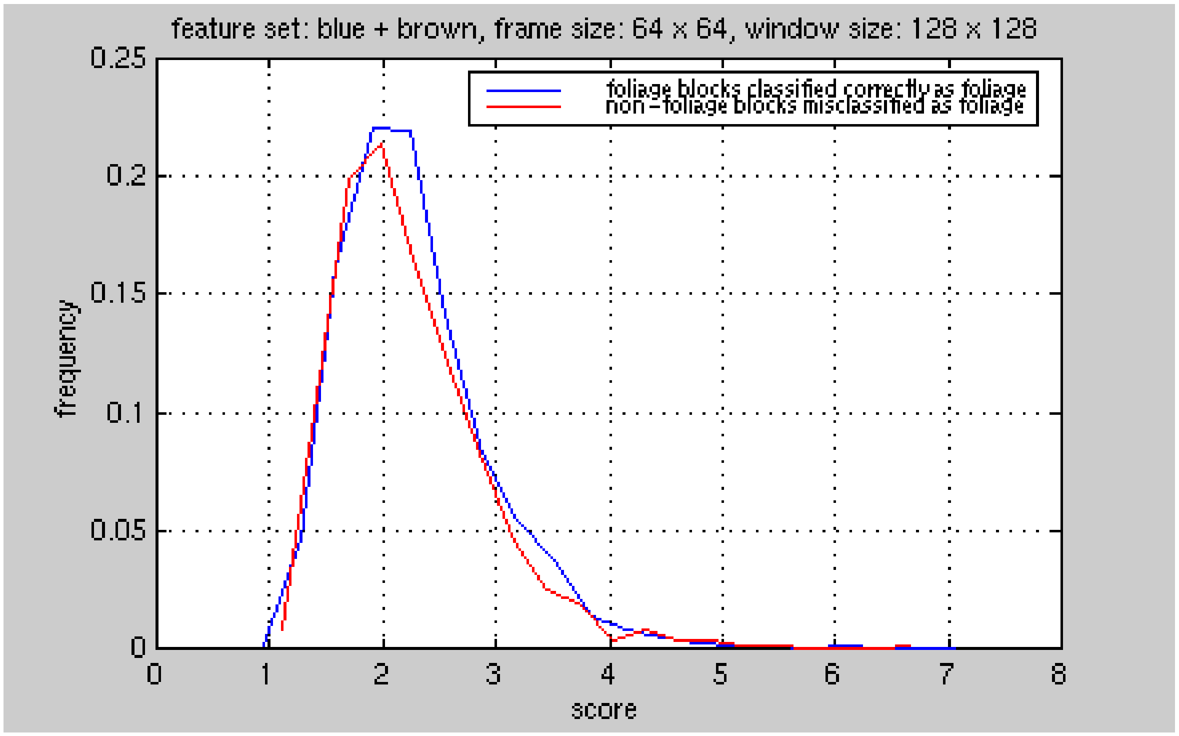


Figure 7. Plots of distributions of scores generated by foliage block classifier in the experiment using a feature set of blue combined with brown. Note the poor discrimination ability of this classifier.

classification techniques, ICA and SVM, to the SBE problem. We have shown that both ICA and SVM perform better than the standard classification scheme of PCA. The overall performance, however, is still far from human performance.

Other than system improvement, we have also made significant progress towards object recognition. We have developed an image segmentation tool that allows rapid and accurate manual segmentation of images. The entire USFS image database has been segmented using a six-way classification scheme. We have designed software that makes use of this information to train image classifiers. We also described a series of cheating experiments doing classification on this manually-segmented image database. By processing only the upper one third portion of each image, we were able to slightly improve on our previous best result. And by processing the background sky region with the feature of blue and brown combined together, we reduced the error classification rate by half for HSBE images. We described our initial implementations of software that estimates models with this manual segmentation information and does classification on a block-by-block basis. We evaluated the performance of this automatic segmentation system. A careful analysis revealed that the current system is weak in classifying foliage blocks. If we find significant features to build a foliage classifier, we will see great improvement on the performance of this automatic segmentation software.

6. FUTURE WORK

We will first work on generating a good classifier for foliage blocks. We are now computing all features on the foliage region for the whole database and analyzing their statistical characteristics. We hope in this way to find some significant features that do a good job on foliage classification. Then, we will incorporate syntactic information into the system to enhance the system performance further. We will also carry out a more careful analysis of the effect of block sizes on the system performance. Once we accomplish a system doing satisfactory image segmentation, our focus will shift to automatic interpretation of the segmentations in ways that will move us closer to true object recognition. We expect to build object recognition capabilities into the software using successful statistical pattern recognition paradigms such as two-dimensional Hidden Markov Models (HMM).

7. ACKNOWLEDGMENTS

This work was supported by the United States Department of Agriculture (USDA) through the United States Forest Service (USFS) Southern Research Station. We would like to express our appreciation for the continuous support and cooperation provided by Mr. Victor Rudis in this project. We would like to thank Nirmala Kalidindi for her excellent work on the PCA classification system, which laid a firm basis for our work. Our appreciation also goes to Aravind Ganapathiraju, Neeraj Deshmukh, Jonathan Hamaker, Issac Alphonso, Rick Duncan and all other fellow members of ISIP for their technical support during the course of this project.

8. REFERENCES

[1] N. Kalidindi, V. Ramani and J. Picone, “Scenic Beauty Estimation of Forestry Images,” *Final Project Report for the Southern Forest Experiment Station, United States Forest Service*, Institute for Signal and Information Processing, Mississippi State University, December 1998.

[2] V. Ramani and J. Picone, “Applications of High Performance Statistical Modeling to Image Analysis of Forest Structure,” *Quarterly Status Report for the Southern Research Station, USDA Forest Service*, Institute for Signal and Information Processing, Mississippi State University, November 1998.

[3] Z. Long and J. Picone, “Applications of High Performance Statistical Modeling to Image Analysis of Forest Structure,” *Quarterly Status Report for the Southern Research Station, USDA Forest Service*, Institute for Signal and Information Processing, Mississippi State University, May 1999.

[4] X. Zhang, V. Ramani, Z. Long, Y. Zeng, A. Ganapathiraju and J. Picone, “Scenic Beauty Estimation Using Independent Component Analysis and Support Vector Machines,” *Proceedings of IEEE Southeastcon*, pp. 274-277, Lexington, Kentucky, USA, March 1999.

[5] Z. Long and J. Picone, “Applications of High Performance Statistical Modeling to Image Analysis of Forest Structure,” *Quarterly Status Report for the Southern Research Station, USDA Forest Service*, Institute for Signal and Information Processing, Mississippi State University, February 1999.

[6] N. Kalidindi, A. Le, J. Picone, V. A. Rudis, “Scenic Beauty Estimation Database”, Institute for Signal and Information Processing and USDA-FS Southern Research Station, Forest Inventory and Analysis Unit, Starkville, MS, USA, 1996.

[7] W. Yhang, J. H. Gramann and V. A. Rudis, *Color Visibility by Season and Silvicultural Treatment: Effects on the Scenic Beauty of Southern U.S. Pine-Oak Forests*, Texas A&M University, College Station, TX and USDA-FS Southern Research Station, Forest Inventory and Analysis Unit, Starkville, MS, USA, October 1998 (manuscript draft).

[8] J. Scruggs, “Color Theory — Additive Color Synthesis,” *http://www.bway.net/~jscruggs/add.html*, LTI Labs, New York City, USA, May 1999.

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, California, USA, 1990.

[10] T. Bell, “Source Separation and Learning Non-orthogonal Bases for Signals Using Independent Component Analysis,” *http://www.clsp.jhu.edu/ws98/presentations/workshop/bell/abstract.html*, presented at the 1998 Summer Workshop on Language Engineering, Center for Language and Speech Processing, John Hopkins University, Baltimore, Maryland, USA, July 1998.

[11] A. J. Bell and T. J. Sejnowski, “An Information-Maximization Approach to Blind Separation and Blind Deconvolution,” *Technical Report No. INC-9501*, Institute for Neural Computing, University of California at San Diego, San Diego, California, USA, February 1995.

[12] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, New York, USA, 1995.

[13] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *http://svm.research.bell-labs.com/SVMdoc.html*, Bell Laboratories, Lucent Technologies, Holmdel, New Jersey, USA, December 1998.

9. APPENDIX

9.1. Detailed SBE Experiments Results

Algorithm	Features	Set 1	Set 2	Set 3	Set 4	Average
PCA	red	51.6	49.4	57.5	55.0	53.4
PCA	green	38.4	34.8	50.0	61.9	46.3
PCA	blue	35.8	46.2	40.6	53.1	43.9
PCA	yellow	54.7	38.6	45.6	67.5	51.6
PCA	brown	37.7	38.6	41.9	55.0	43.3
PCA	rgb	40.3	42.4	38.1	33.1	38.5
PCA	ylw + brn	40.3	46.2	40.6	45.6	43.2
PCA	rgb + ylw + brn	39.6	39.9	39.4	66.9	46.5
PCA	blue + brn	31.4	50.6	35.0	56.9	43.5
PCA	rgb + brn	47.2	46.2	48.1	38.8	45.1
PCA	rgb + ll	39.0	40.5	35.6	36.3	37.9
PCA	grn + blue + ylw + ll	39.6	38.0	45.0	53.1	43.9
PCA	grn + blue + ylw + brn + ll	39.6	52.5	48.1	40.0	45.1
PCA	blue + brn + ll	37.1	52.5	33.1	58.1	45.2
PCA	rgb + ll + sl	40.3	42.4	35.0	34.4	38.0
PCA	rgb + sl	37.8	42.4	38.1	33.1	37.9
PCA	rgb + ent	39.6	41.8	33.8	35.6	37.7
PCA	rgb + ll + ent	42.1	43.0	32.5	36.3	38.5
PCA	ent + fract	56.6	42.4	51.9	57.5	52.1
PCA	rgb + ll + ent + fract	43.3	46.8	33.8	33.8	39.4
PCA	entropy	49.7	44.9	64.4	50.6	52.4
PCA	rgb + all	42.1	40.5	41.9	34.4	39.7
PCA	rgb + all + fract	43.4	43.0	45.0	33.8	41.3
SVM	all	34.6	34.2	31.2	31.9	33.0
SVM	rgb	35.2	34.8	32.5	32.5	33.8
SVM	rgb + ll	32.1	32.3	31.9	32.5	32.2
SVM	rgb + ll + ent	35.2	34.2	31.9	32.5	33.4
SVM	rgb + ll + fract	35.2	35.2	32.5	31.2	33.5
ICA	all	35.2	33.5	31.9	33.1	33.4
ICA	rgb	34.6	33.5	32.5	33.1	33.4
ICA	rgb + ll	34.6	33.5	33.1	33.1	33.6
ICA	rgb + ll + ent	34.6	34.2	33.1	32.5	33.6
ICA	rgb + ll + fract	35.9	33.5	32.5	33.1	33.8

Table 14. A cumulative summary of our SBE results.

9.2. Confusion Matrices

9.2.1. Confusion Matrices For The Cheating Experiments

Here are the confusion matrices for the cheating experiments. For comparison, we also give the confusion matrices for those experiments with the same conditions, but on entire images.

Automatic	LSBE	MSBE	HSBE
LSBE	8	2	0
MSBE	17	75	15
HSBE	3	26	13

Table 15. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Error Rate: 39.6%. This experiment is on the entire image set.

Automatic	LSBE	MSBE	HSBE
LSBE	7	22	1
MSBE	20	67	18
HSBE	1	14	9

Table 16. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: tree; Error Rate: 47.8%.

Automatic	LSBE	MSBE	HSBE
LSBE	11	23	2
MSBE	15	62	14
HSBE	2	18	12

Table 17. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: foliage; Error Rate: 46.5%

Automatic	LSBE	MSBE	HSBE
LSBE	4	27	10
MSBE	21	60	7
HSBE	3	16	11

Table 18. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bush; Error Rate: 52.8%

Automatic	LSBE	MSBE	HSBE
LSBE	9	24	1
MSBE	12	65	17
HSBE	7	14	10

Table 19. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: grass; Error Rate: 47.2%

Automatic	LSBE	MSBE	HSBE
LSBE	7	10	0
MSBE	16	42	6
HSBE	5	51	22

Table 20. Experiment Condition: Feature Set: rgb + entropy; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bgsky; Error Rate: 55.3%

Automatic	LSBE	MSBE	HSBE
LSBE	5	2	0
MSBE	20	75	16
HSBE	3	26	12

Table 21. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Error Rate: 42.1%. This is the experiment on the entire image set.

Automatic	LSBE	MSBE	HSBE
LSBE	7	20	1
MSBE	20	66	18
HSBE	1	17	9

Table 22. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: tree; Error Rate: 48.4%

Automatic	LSBE	MSBE	HSBE
LSBE	9	27	2
MSBE	16	63	17
HSBE	3	13	9

Table 23. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: foliage; Error Rate: 49.1%

Automatic	LSBE	MSBE	HSBE
LSBE	6	25	10
MSBE	19	63	10
HSBE	3	15	8

Table 24. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bush; Error Rate: 51.6%

Automatic	LSBE	MSBE	HSBE
LSBE	9	22	0
MSBE	12	68	20
HSBE	7	13	8

Table 25. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: grass; Error Rate: 46.5%

Automatic	LSBE	MSBE	HSBE
LSBE	7	11	0
MSBE	16	39	6
HSBE	5	53	22

Table 26. Experiment Condition: Feature Set: rgb + entropy + long lines; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bgsky; Error Rate: 57.2%

Automatic	LSBE	MSBE	HSBE
LSBE	2	2	0
MSBE	24	97	18
HSBE	2	4	10

Table 27. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Error Rate: 31.4%. This is the experiment on the entire images.

Automatic	LSBE	MSBE	HSBE
LSBE	9	27	2
MSBE	16	66	15
HSBE	3	10	11

Table 28. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: tree; Error Rate: 45.9%

Automatic	LSBE	MSBE	HSBE
LSBE	8	22	3
MSBE	18	75	21
HSBE	2	6	4

Table 29. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: foliage; Error Rate: 45.3%

Automatic	LSBE	MSBE	HSBE
LSBE	10	37	12
MSBE	14	51	5
HSBE	4	15	11

Table 30. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bush; Error Rate: 54.7%

Automatic	LSBE	MSBE	HSBE
LSBE	5	14	0
MSBE	14	70	11
HSBE	9	19	17

Table 31. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: grass; Error Rate: 42.1%

Automatic	LSBE	MSBE	HSBE
LSBE	5	12	0
MSBE	19	45	9
HSBE	4	46	19

Table 32. Experiment Condition: Feature Set: blue + brown; Data Set: set 1 of Pre-Phase data; Algorithm: PCA; Region: bgsky; Error Rate: 56.6%

9.2.2. Confusion Matrices For The Automatic Segmentation Evaluations

Automatic	tree	foliage	bush	grass	bgsky	sky
tree	2619	5100	2567	3332	179	0
foliage	122	1788	528	384	35	0
bush	855	5024	4595	3096	1	0
grass	985	1962	2818	3001	14	0
bgsky	697	3299	508	420	1464	1
sky	49	77	167	140	266	9

Table 33. Experiment Condition: Feature Set: blue + brown; frame size: 64 x 64; window size: 128 x 128; Error Rate: 70.8%.

Automatic	tree	foliage	bush	grass	bgsky	sky
tree	2437	4913	2381	2978	182	0
foliage	170	1952	575	457	40	0
bush	822	4796	4428	2971	3	0
grass	1156	2315	3167	3441	14	0
bgsky	695	3201	475	388	1464	1
sky	47	73	157	138	256	9

Table 34. Experiment Condition: Feature Set: blue + brown + short lines; frame size: 64 x 64; window size: 128 x 128; Error Rate: 70.2%.

Automatic	tree	foliage	bush	grass	bgsky	sky
tree	2807	4494	1717	2454	181	0
foliage	260	2402	609	259	98	0
bush	911	6412	6252	2578	16	0
grass	831	1479	2208	4888	7	0
bgsky	504	2450	391	169	1437	1
sky	14	13	6	25	220	9

Table 35. Experiment Condition: Feature Set: rgb + entropy; frame size: 64 x 64; window size: 128 x 128; Error Rate: 61.4%.

Automatic	tree	foliage	bush	grass	bgsky	sky
tree	2839	5091	2701	3486	187	0
foliage	85	1276	426	306	33	0
bush	895	5436	4776	3259	13	0
grass	711	1753	2488	2551	16	0
bgsky	752	3633	664	646	1460	0
sky	45	61	128	125	250	10

Table 36. Experiment Condition: Feature Set: blue + brown; frame size: 64 x 64; window size: 96 x 96; Error Rate: 72.0.

9.3. Distribution Plots Of The Scores By The Foliage Block Classifier

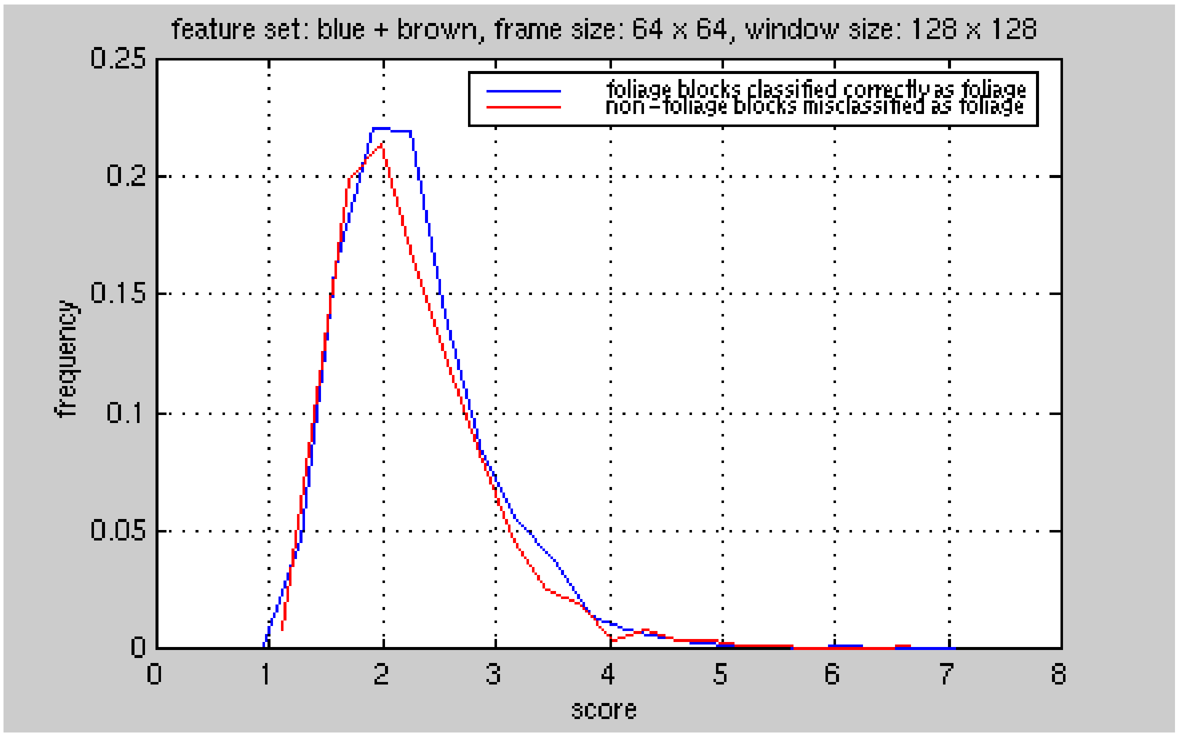


Figure 8. Distribution plot of scores generated by foliage block classifier of the automatic segmentation system. This is for the evaluation experiment No. 1. The conditions are listed in the title of the plot.

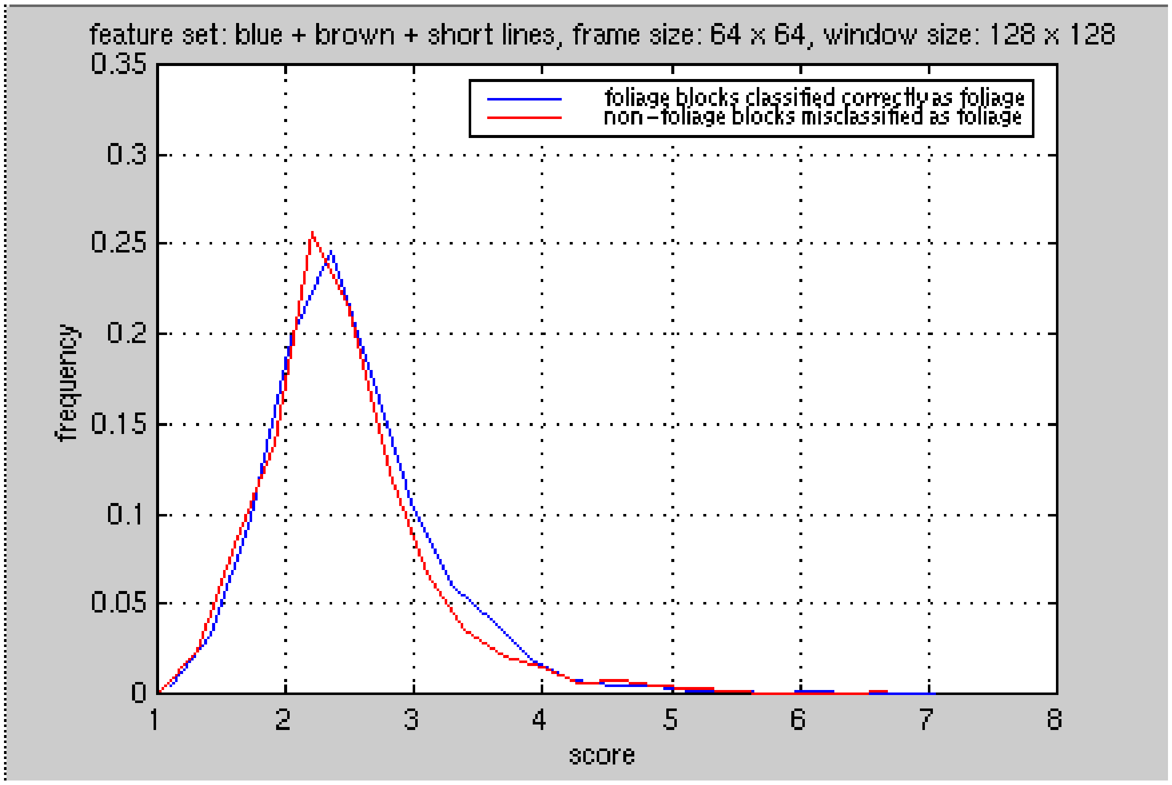


Figure 9. Score distribution plot for experiment No. 2.

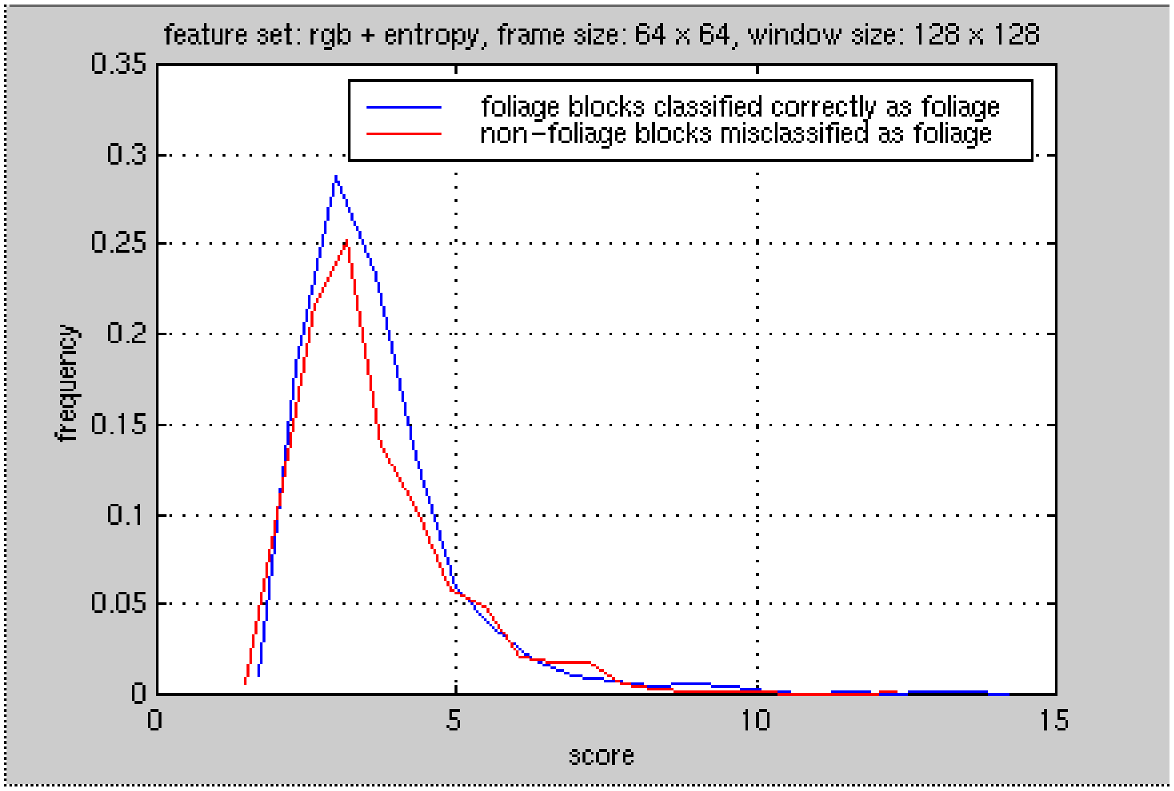


Figure 10. Score distribution plot for experiment No. 3.

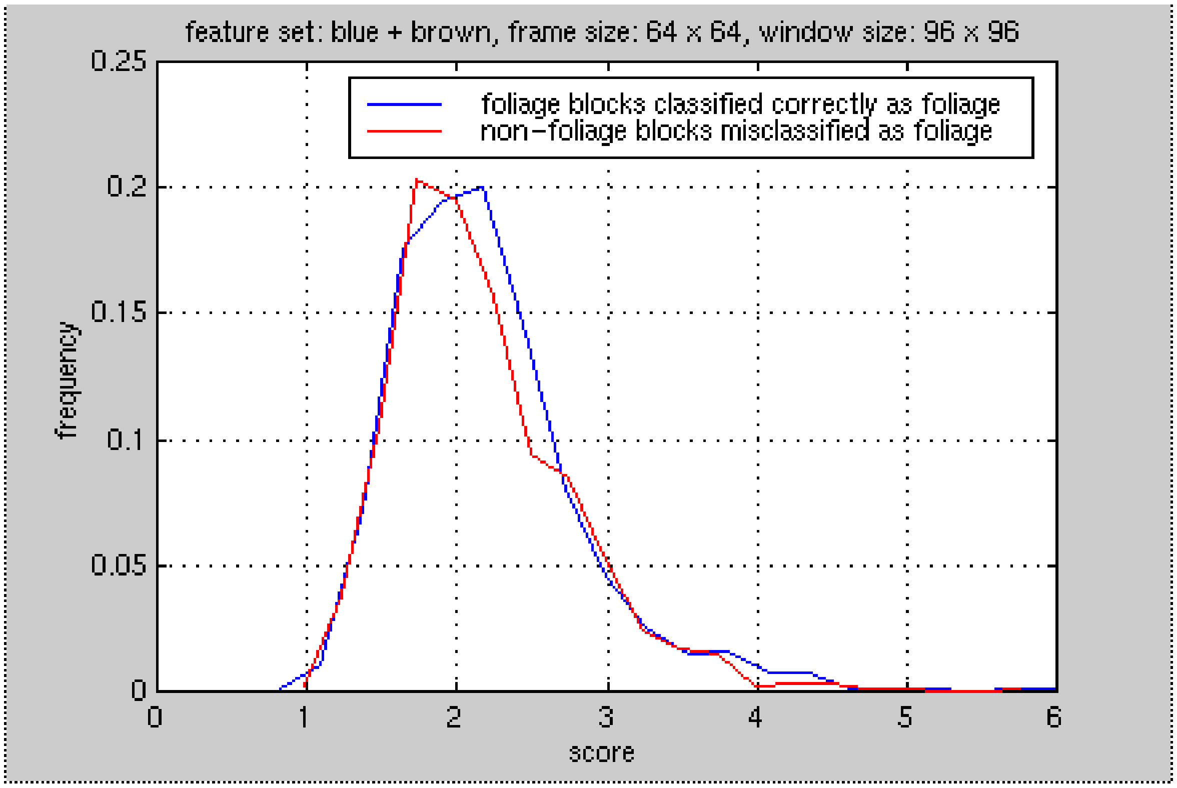


Figure 11. Score distribution plot for experiment No. 4. This experiment used a smaller window size.

9.4. Automatic Segmentation Examples

Some examples of the automatic segmentations are shown here. Note we added the manual segmentations in the original images. For both manual segmentation and automatic segmentation, we use various colors to distinguish different regions. The mapping relationship is as follows.

region	color
tree	brown
foliage	red
bush	yellow
grass	green
bgsky	light blue
sky	dark blue

Table 37. Mapping relationship between colors and segmented regions.