

This is the author's final, peer-reviewed manuscript as accepted for publication. The publisher-formatted version may be available through the publisher's web site or your institution's library.

Classification of image pixels based on minimum distance and hypothesis testing

Santosh Ghimire and Haiyan Wang

How to cite this manuscript

If you make reference to this version of the manuscript, use the following information:

Ghimire, S., & Wang, H. (2012). Classification of image pixels based on minimum distance and hypothesis testing. Retrieved from <http://krex.ksu.edu>

Published Version Information

Citation: Ghimire, S., & Wang, H. (2012). Classification of image pixels based on minimum distance and hypothesis testing. *Computational Statistics and Data Analysis*, 56(7), 2273-2287.

Copyright: © 2012 Elsevier B.V.

Digital Object Identifier (DOI): [doi:org/10.1016/j.csda.2012.01.005](https://doi.org/10.1016/j.csda.2012.01.005)

Publisher's Link: <http://www.sciencedirect.com/science/journal/01679473/56/7>

This item was retrieved from the K-State Research Exchange (K-REx), the institutional repository of Kansas State University. K-REx is available at <http://krex.ksu.edu>

Classification of Image Pixels based on Minimum Distance and Hypothesis Testing

Santosh Ghimire¹ and Haiyan Wang^{2*}

¹*Department of Mathematics, Kansas State University, Manhattan, KS 66506*

²*Department of Statistics, Kansas State University, Manhattan, KS 66506*

Abstract: In this article, we introduce a new method of image pixel classification. Our method is a nonparametric classification method which uses combined evidence from the multiple hypothesis testings and minimum distance to carry out the classification. Our work is motivated by the test-based classification introduced by [Liao and Akritas \[2007\]](#). We focus on binary and multiclass classification of image pixels taking into account of both equal and unequal prior probability of classes. Experiments show that our method works better in classifying image pixels in comparison with some of the standard classification methods such as linear discriminant analysis, quadratic discriminant analysis, classification tree, polyclass method, and Liao and Akritas's method. We apply our classifier to perform image segmentation. Experiments show that our test-based segmentation has excellent edge detection and texture preservation property for both grey scale and color images.

AMS 2000 subject classifications: Primary 68U10, 97K80, 62H35; secondary 62G10.

Keywords and phrases: Image processing, image classification, hypothesis testing, minimum distance, image segmentation.

1. Introduction

Images can be considered as a finite collection of regions and thus can be realized by groups of pixel values representing different regions in the image. The pixels representing a particular feature or color in the image show more homogeneity in terms of distribution of pixel values. Groups of similar image pixels can be formed by comparing pixels with each other and to pixels of known identity. The groups so formed are called image pixel classes. These classes then represent different informational categories of interest and can follow any distribution.

Image pixels classification is a process of assigning pixels to different classes in the image. It is widely used in medical diagnosis and remote sensing. Some of the applications of multispectral image pixels classification in remote sensing are identification of objects in satellite images,

*Corresponding author e-mail: hwang@ksu.edu

land-use analysis, mineral exploration, and determination of earth surface composition where the knowledge of reflectance properties of various types of material is also needed for the classification. Image pixels classification has been very helpful in medical diagnosis such as chromosome karyotyping, comparison of normal and non normal blood vessels, categorization of database of x-ray images, study of anatomical structure, computer-integrated surgery, quantification of tissue volumes, treatment planning, etc. Some of the other applications of pixel classification include astronomy, face recognition, traffic control systems, agricultural imaging, computer vision etc.

Commonly used statistical methods that can be implemented for image pixels classification are linear discriminant analysis (LDA) (Hastie et al. [2009]), quadratic discriminant analysis (QDA) (Hastie et al. [2009]), classification tree (Breiman et al. [1998]), polyclass method (Stone et al. [1997]), maximum likelihood, and Bayes classifier. Commonly used computer-based classifiers include nearest-neighbor classifier, K-nearest-neighbor, neural networks, and support vector machine (Vapnik [1982]). All the aforementioned computer-based classifiers are nonparametric in that they make no assumptions on the distributions of the data to be classified. On the other hand, we have mixed bag of classifiers in the given statistical classifier methods. Classification tree and polyclass method are nonparametric whereas LDA, QDA, maximum likelihood, and Bayes classifier are parametric classifiers making assumptions about the distribution of classes. For example, LDA and QDA require that the distribution of values for all classes to be Gaussian. Similarly, maximum likelihood and Bayes classifier generally assume that the pixel intensities are independent samples from a mixture of Gaussian distributions. However in practice, image pixel values can follow any distribution. The computer-based approaches are straightforward and intuitive but barely consider the randomness of the data in each class. A classification method based on hypothesis testings was developed by Liao and Akritas [2007]. This is a powerful nonparametric classification method which can allow the variation within a class be taken into account through the test-statistics without making distributional assumptions. However the implementation of their method in the context of images reveals that the method can fail to correctly classify many image pixels in the given image due to small p-values. Here we introduce a minimum distance into the test-based classification and come up with a new classifier for image pixels. This new classification method eliminates the drawback of Liao & Akritas's method and works better than commonly used classification methods.

The pixels classification will be employed to perform segmentation of color images. Image segmentation is a process of dividing an image into different homogenous regions so that the image can be represented differently making it easy to study and analyze. In fact, image segmentation can be viewed as image pixels classification based on the spatial features and color of the images. Segmentation extracts information about the structure of objects in the image and is helpful in separating and observing various parameters of interest within the image data. There are several ap-

proaches for grey scale image segmentation. Commonly used approaches include histogram based approach (Dutta and Chaudhari [2009]), clustering approach (Coleman and Andrews [1979]), watershed transformation approach (Vincent and Soille [1991]), classifier method, region-based approach, edge detection-based approach, artificial neural networks etc. However there are not many literature available for color image segmentation and most of the available methods for color image are based on grey scale image segmentation approach. Readers can refer to Haralick and Shapiro [1985] and Pal and Pal [1993] for surveys on image segmentation techniques.

Classifier methods work well in image segmentation for images with quantifiable features. They can be employed in multichannel images and are efficient to employ in comparison with other approaches. In classifier-based image segmentation, training data are manually obtained to be used as references for segmentation of the entire image. As pixel values in image classes can follow any distribution, nonparametric classifiers in general are expected to produce more realistic results for a wide variety of data than the parametric classifiers. In this article, we describe how our nonparametric classifier can be used to produce accurate segmentation of color images.

The rest of the article is organized as follows. Section 2 gives detail about the formation of classes, training data and test points. Section 3 describes our method for the binary classification of image pixels where we consider both equal and unequal prior probabilities of classes. In Section 4, we extend to multiclass classification for equal and unequal prior probabilities of classes. Section 5 discusses the classification of pixel values in color images. In Section 6, we provide implementation and exhibit the detailed aspects of our classification method in grey scale images. Section 7 is devoted to the comparisons of several methods of classification with our method. Section 8 discusses the segmentation of color images using our method of classifier followed by a summary.

2. Training data, classes and test points

Here, we give a description about the formation of training data, classes and test points in a given image. This will be frequently used in the implementation of different classification methods in later sections. In an image, we can define our classes of interest by selecting the regions marked with different colors in it. We use some data that is known a priori to belong to the involved classes to train the system about these classes and learn the class parameters. This data is referred to as training data. We take a rectangular part of the regions representing the class to acquire the training data of that class. We do this by choosing two points in the region which will be the end points of the main diagonal of the rectangle. The rectangle so formed is simply a submatrix of the given matrix (image is a matrix). Next, we put it into a vector form by adjoining each column of the submatrix below its preceding column. Then we treat this vector of pixels as training data from the corresponding class. In the classification of images, we classify a randomly selected pixel, known as test point, in the image as belonging to one of the defined classes. In the implementation of all

the considered classification methods, we will randomly select equal number of test points from each of the region representing the different classes. This process of selecting equal number of test points from the corresponding class will be helpful to determine the misclassification rate of the method.

3. Binary Classification

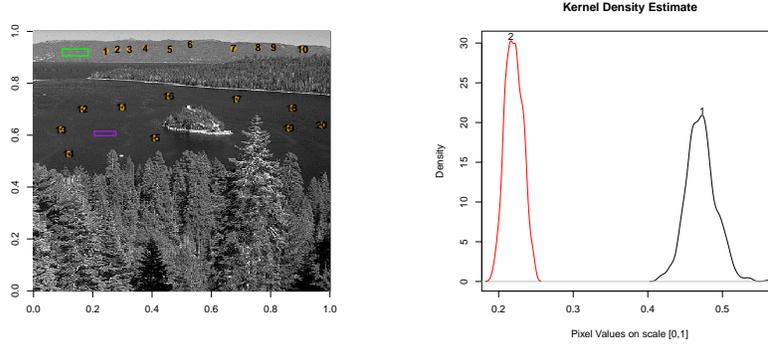
Let us consider two image pixel classes with means μ_1, μ_2 and x_0 be a randomly selected test point in the image. Denote $(x_{11}, x_{12}, x_{13}, \dots, x_{1n_1})$ and $(x_{21}, x_{22}, x_{23}, \dots, x_{2n_2})$ as the training data from class 1 and class 2 respectively. Liao and Akritas [2007] (denote as LA) suggested a classification scheme based on the following two applications of a test. The null hypothesis can be that two samples have equal mean, or more generally, class 1 and class 2 have identical distribution. The t-test would correspond to equal mean hypothesis whereas Wilcoxon rank sum test is appropriate for the other hypothesis.

- Test 1: Place x_0 with the observations from class 1 and use $(x_0, x_{11}, x_{12}, \dots, x_{1n_1})$ and $(x_{21}, x_{22}, \dots, x_{2n_2})$ to test the null hypothesis H_0 .
- Test 2: Place x_0 with the observations from class 2 and use $(x_{11}, x_{12}, \dots, x_{1n_1})$ and $(x_0, x_{21}, x_{22}, \dots, x_{2n_2})$ to test the null hypothesis H_0 .

We denote p-values from test 1 and test 2 by $PV_1(x_0), PV_2(x_0)$ and use p_1, p_2 to denote the prior probability of classes. A small $PV_1(x_0)$ and a large $PV_2(x_0)$ suggests that putting this observation in class 1 will maintain the difference of the two classes. On the other hand, putting this observation in class 2 will blur the boundary between the two classes. Thus, $PV_1(x_0)/[PV_1(x_0) + PV_2(x_0)]$ can be assumed as the relative test-based probability that the test point x_0 is not from class 1 so that $(1 - PV_1(x_0)/[PV_1(x_0) + PV_2(x_0)])$ works as the probability that x_0 is from class 1. LA classify x_0 as from class 1 if $PV_2(x_0)p_1 > PV_1(x_0)p_2$. They classify x_0 as from class 2 if $PV_1(x_0)p_2 > PV_2(x_0)p_1$. In practice, we have found out that this classification scheme tends to misclassify an observation for image data when both p-values, $PV_i(x_0)$, are too small. For example, consider the image in Figure 1(a). We choose mountain region and water region as class 1 and class 2. Training data for these classes are formed by following the procedure in Section 2. In the image, size (number of pixels) of training data for class 1 and class 2 are 630 and 380 respectively. The image has size 512×512 and hence the proportion of the training data are 0.00240 and 0.00144 respectively. For the classification purpose, we select 20 test points labeled with numbers such that first 10 of them are chosen from region representing class 1 and the rest are taken from class 2 region as shown in Figure 1(a). Kernel density estimate is used to obtain the density plot of pixel values in each class and is shown in Figure 1(b). The plot shows that classes so formed are distinct

and well separated.

FIG 1.



(a) Image with training data and test points. (b) Density plot of classes.

The selected test points are classified using the LA method and are shown in Table 1 which shows that their method has misclassified test points 2,4,5 and 7-10.

TABLE 1
Classification by Liao & Akritas method in image Figure 1(a)

TP	LA	PV1	PV2	Obs.
1	class 1	7.493683e-157	4.621872e-156	0.4745098
2	class 2	7.523930e-157	5.808925e-157	0.4352941
3	class 1	7.516353e-157	1.054541e-156	0.4549020
4	class 2	7.523384e-157	6.037376e-157	0.4392157
5	class 2	7.523930e-157	5.441142e-157	0.4274510
6	class 1	7.493683e-157	1.385811e-156	0.4588235
7	class 2	7.523649e-157	5.621821e-157	0.4313725
8	class 2	7.524070e-157	5.360952e-157	0.4235294
9	class 2	7.524086e-157	5.188463e-157	0.4039216
10	class 2	7.523992e-157	5.297517e-157	0.4196078
11	class 2	8.472896e-157	5.187623e-157	0.2392157
12	class 2	1.835294e-156	5.174205e-157	0.2235294
13	class 2	2.407138e-156	5.179641e-157	0.2196078
14	class 2	1.436675e-156	5.182781e-157	0.2274510
15	class 2	1.835294e-156	5.174205e-157	0.2235294
16	class 2	7.984479e-157	5.188075e-157	0.2431373
17	class 2	1.835294e-156	5.174205e-157	0.2235294
18	class 2	6.155528e-156	5.187333e-157	0.2039216
19	class 2	7.182948e-156	5.188463e-157	0.1803922
20	class 2	7.182948e-156	5.188463e-157	0.1764706

PV1= p-value from test 1, PV2= p-value from test 2,
 TP= Test point, Obs= Test point pixel,
 LA= Liao & Akritas's classification result

Next we analytically explain why the LA method may fail to classify some test points accurately. Let us consider two image pixel classes, class 1 and class 2 and consider a test point x_0 which we would like to classify using the LA method. We illustrate with the t-test case. Let $(x_{11}, x_{12}, \dots, x_{1n_1})$ and $(x_{21}, x_{22}, \dots, x_{2n_2})$ be the training data for the class 1 and class 2 respectively. We take $n_1 = n_2 = n$. Then the test statistics for the Test 1 described earlier is given by,

$$T_1 = \frac{\frac{x_0 + x_{11} + x_{12} + \dots + x_{1n}}{n+1} - \frac{x_{21} + x_{22} + \dots + x_{2n}}{n}}{\sqrt{\frac{sd_1^2}{n+1} + \frac{\widehat{\sigma}_2^2}{n}}},$$

where $\hat{\sigma}_2^2$ is the sample variance of class 2 and $sd_1^2 = [(x_0 - \hat{\mu}_1)^2 + \sum_{i=1}^n (x_{1i} - \hat{\mu}_1)^2]/n$ with $\hat{\mu}_1 = \bar{x}_1. + (x_0 - \bar{x}_1.)/(n+1)$. Similarly the test statistics for the test 2 mentioned earlier is

$$T_2 = \frac{\frac{x_{11} + x_{12} + \dots + x_{1n}}{n} - \frac{x_0 + x_{21} + x_{22} + \dots + x_{2n}}{n+1}}{\sqrt{\frac{sd_2^2}{n+1} + \frac{\hat{\sigma}_1^2}{n}}},$$

where $\hat{\sigma}_1^2$ is the sample variance of class 1 and $sd_2^2 = [(x_0 - \hat{\mu}_2)^2 + \sum_{i=1}^n (x_{2i} - \hat{\mu}_2)^2]/n$ with $\hat{\mu}_2 = \bar{x}_2. + (x_0 - \bar{x}_2.)/(n+1)$. Note that for large n , $sd_1^2 \approx \hat{\sigma}_1^2$ and so $T_1 \approx t + (x_0 - \bar{x}_1.)/\sqrt{V_1}$, where

$$V_1 = (n+1)sd_1^2 + \hat{\sigma}_2^2(n+1)^2/n \text{ and } t = \frac{\bar{x}_1. - \bar{x}_2.}{\sqrt{(\hat{\sigma}_1^2 + \hat{\sigma}_2^2)/n}}.$$

Similarly, $T_2 \approx t - (x_0 - \bar{x}_2.)/\sqrt{V_2}$, where $V_2 = (n+1)sd_2^2 + \hat{\sigma}_1^2(n+1)^2/n$. The two statistics T_1 and T_2 differ in the second terms. Note that

$$V_1 - V_2 = (x_0 - \bar{x}_1.)^2 - (x_0 - \bar{x}_2.)^2 + (\hat{\sigma}_2^2 - \hat{\sigma}_1^2)(2n+2)/n,$$

which was obtained by writing

$$\begin{aligned} sd_1^2 &= (x_0 - \bar{x}_1.)^2/(n+1) + \hat{\sigma}_1^2(n-1)/n, \\ sd_2^2 &= (x_0 - \bar{x}_2.)^2/(n+1) + \hat{\sigma}_2^2(n-1)/n. \end{aligned}$$

It can be seen that when $|x_0 - \bar{x}_1.|$ and $|x_0 - \bar{x}_2.|$ are close for x_0 between $\bar{x}_1.$ and $\bar{x}_2.$ but $|x_0 - \bar{x}_1.| < |x_0 - \bar{x}_2.|$, we have $V_1 > V_2$ if σ_2 is much larger than σ_1 (since $\hat{\sigma}_i^2$ are consistent unbiased estimators of σ_i , $i = 1, 2$). Consequently, $PV_1(x_0) > PV_2(x_0)$ and LA would classify x_0 to class 2 even though x_0 is closer to class 1.

In lieu of above finding, we modify the LA classification criterion as follows.

- If $\max(PV_1, PV_2) \geq 0.001(\text{threshold})$, i.e., at least one of the test p-values is larger than the threshold value, then a test point x_0 belongs to class 1 if $PV_2(x_0)p_1 > PV_1(x_0)p_2$. Similarly, the test point x_0 belongs to class 2 if $PV_1(x_0)p_2 > PV_2(x_0)p_1$.
- If $\max(PV_1, PV_2) < 0.001(\text{threshold})$, i.e., both of the test p-values are smaller than the threshold value, then a test point x_0 is classified to class 1 if the distance of x_0 to class 1 is less than the distance of x_0 to class 2. The x_0 is classified to class 2 if the distance of x_0 to class 2 is less than the distance of x_0 to class 1.

The distance of a point x_0 to a class can take one of the traditional forms such as complete linkage, single linkage, average linkage, etc., or simply, the distance between x_0 and the central

tendency of class pixel values. In our experiments, we employ the distance of x_0 to the mean pixel values of each class.

In hypothesis testings, 0.001, 0.01, 0.05, 0.1 are typical significance levels used by practioners to declare a significant result. In our experiments, we take the most conservative one, i.e. 0.001, out of these four values as the threshold for significance levels. It could be a good idea to use cross-validation with the training data to choose the best value among these such that the cross-validation error rate is minimized for the training data. We are reluctant to take the threshold to be values other than those listed above. For example, 0.06 is not a reasonable significance level to use as a threshold for significance.

If the prior probability of classes are equal then $p_1 = p_2 = 1/2$. For the unequal prior case, we can define prior probability of classes as follows. Define $\lambda = (\mu_1 + \mu_2)/2$.

- If μ_1 is less than μ_2 , then

Prior of class 1 = Proportion of pixels in the training data that are less than λ ;

Prior of class 2 = 1 – Prior of class 1.

- If μ_2 is less than μ_1 , then

Prior of class 2 = Proportion of pixels in the training data that are less than λ ;

Prior of class 1 = 1 – Prior of class 2.

Another way to define prior probability of classes is;

$$\text{Prior of class 1} = N_1/(N_1 + N_2) \quad \text{and} \quad \text{Prior of class 2} = N_2/(N_1 + N_2),$$

where N_1 and N_2 are the numbers of pixel values in the training data for classes 1 and 2 respectively.

In general, in the case of image data, we find that the first prior definition is more informative than the second. This can be observed by the separation distance of classes in the density plots. Due to this reason, we will employ the first definition to calculate the prior probability of classes in all the classification methods.

4. Multiclass classification

Here, we extend the idea of binary classification to multiclass classification. Assume that there are k pixel classes in the image with means $\mu_1, \mu_2, \dots, \mu_k$ and prior probabilities p_1, p_2, \dots, p_k , respectively. We use f_i, F_i to denote the probability density function and cumulative density function of class i . Let x_0 be a test point which we would like to classify. Hypothesis testings are done as

many times as the number of classes by placing the test observation in one of the classes every time. Suppose that we have training data with observation $(x_{11}, x_{12}, \dots, x_{1n_1})$, $(x_{21}, x_{22}, \dots, x_{2n_2})$, \dots , and $(x_{k1}, x_{k2}, \dots, x_{kn_k})$ from the classes 1, 2, \dots , and k , respectively. We perform a series of hypothesis testing in which we test to see the sample evidence that x_0 belongs to each of the classes based on the training data. For these hypothesis testings, we choose Kruskal-Wallis test for generality since this test allows arbitrary distribution for each sample. The tests are as follows.

- Test 1: Place x_0 with the observations from class 1. Assume

$$(x_0, x_{11}, x_{12}, \dots, x_{1n_1}) \sim f_1(x),$$

$$(x_{21}, x_{22}, \dots, x_{2n_2}) \sim f_2(x),$$

$$\vdots$$

$$(x_{k1}, x_{k2}, \dots, x_{kn_k}) \sim f_k(x),$$
 and test the null hypothesis H_0 that all the distribution functions are identical.
- Test 2: Place x_0 with the observation from class 2. Assume

$$(x_{11}, x_{12}, \dots, x_{1n_1}) \sim f_1(x),$$

$$(x_0, x_{21}, x_{22}, \dots, x_{2n_2}) \sim f_2(x),$$

$$\vdots$$

$$(x_{k1}, x_{k2}, \dots, x_{kn_k}) \sim f_k(x),$$
 and test the null hypothesis H_0 that all the distribution functions are identical and similarly,
- Test k: Place x_0 with the observation from class k. Assume

$$(x_{11}, x_{12}, \dots, x_{1n_1}) \sim f_1(x),$$

$$(x_{21}, x_{22}, \dots, x_{2n_2}) \sim f_2(x),$$

$$\vdots$$

$$(x_0, x_{k1}, x_{k2}, \dots, x_{kn_k}) \sim f_k(x),$$
 and test the null hypothesis H_0 that all the distribution functions are identical.

Let $PV_1(x_0), PV_2(x_0), \dots$ and $PV_k(x_0)$ denote the p-values of the Test 1, Test 2, \dots , and Test k respectively. We mainly use these p-values and the distance of x_0 to the classes. Depending upon the test p-values, we now state our detailed classification rule as follows.

- If $\max_{1 \leq i \leq k} PV_i \leq 0.001$ (threshold), i.e., all the test p-values are small, then we use minimum distance classification. We assign x_0 to the class with the smallest d_i , where $d_i = \min_{1 \leq i \leq k} D_i$ and D_i is the distance of the observation x_0 to the mean of class i .
- If $\min_{1 \leq i \leq k} PV_i \geq 0.001$ (threshold), i.e., all the test p-values are large, then x_0 is classified using the following steps.
 - Step 1: We calculate the prior probabilities p_i of classes and eliminate the class with

the largest $(1 - p_i) \times PV_i(x_0)$.

- Step 2: We repeat Step 1 until there are two classes of pixels left.
- Step 3: For the remaining two pixel classes, we classify x_0 to the class with the smaller $(1 - p_i) \times PV_i(x_0)$.
- If m ($1 < m < k$) of the test p-values are less than or equal to the threshold (0.001), then we eliminate the $k - m$ classes that have p-values larger than the threshold. We use the minimum distance rule to determine the class label from these m classes.
- If $m = 1$, we assign the observation to that class with the p-value less than the threshold.

We note that when all the test p-values are larger than the threshold, then x_0 is classified to the class obtained by eliminating classes, one at a time and comparing $(1 - p_i) \times PV_i(x_0)$ as explained above. We follow this stepwise elimination of classes instead of classifying x_0 to the class with the smallest $(1 - p_i) \times PV_i(x_0)$ to avoid masking phenomenon as is explained in [Hastie et al. \[2009\]](#) and [Liao and Akritas \[2007\]](#). The other reason to follow the stepwise elimination of classes is that the prior probabilities of classes could be updated after a class is eliminated.

If the prior probability of classes are equal, then we use $p_1 = p_2 = \dots = p_k = 1/k$. For the unequal priors, we can define the prior probabilities of classes as follows. Let $\mu_{(1)}, \mu_{(2)}, \dots, \mu_{(k)}$ be the ordered means of the classes. Then,

Prior of class i = Proportion of pixels larger than $[\mu_{(i-1)} + \mu_{(i)}]/2$ and smaller than $[\mu_{(i)} + \mu_{(i+1)}]/2$.

Another way to define prior probabilities of a class is,

$$\text{Prior of class } i = N_i / (N_1 + \dots + N_k),$$

where N_i represents the number of pixel values in the training data for the i^{th} class.

If classes t_1, t_2, \dots, t_r are excluded, the prior probabilities of classes for the next step can be adjusted as, $p_i = N_i / (\sum_{m=1}^k N_m - \sum_{j=1}^r N_{t_j})$. The first definition of priors is more informative than the second for the image data, which could be seen by the separation distance of different classes in the density plots. Considering this fact, we will calculate prior probability of classes using the first definition for all the classification methods.

5. Classification of pixel values in color images.

In this section, we discuss our method of classification for color image pixels. We can consider two approaches of classification of color image pixels. In the first approach, we consider the three grey scale images of the original image corresponding to the RGB components or channels. We

implement our classification method, discussed in Sections 3 and 4, to classify pixels in each component where we employ a univariate test in the hypothesis testings. After the classification of the test points in each component, we assign the final classification labels using a majority of votes. For example, if a test point is classified as coming from class 1 in the red component and as belonging to class 2 in the green and blue components, the final classification for it will be in class 2. In the case of tie, we randomly assign the test point to one of the classes.

The other approach considers every image pixel as a 3-dimensional vector consisting of corresponding pixel values for red, green, and blue components. The RGB model is mainly based on the Young-Helmholtz theory of trichromatic color vision (Paul [1981]) and Maxwell's theory of color triangle (Paul [1981]). The RGB space is a three-dimensional orthogonal coordinate system in the sense that the three axes, representing the red, green, and blue color intensities, are perpendicular to each other. In this color space, a color is simply formed by superimposing the three colored light beams which are called components of the color. The spectrum of the final color so formed is obtained by adding together the spectra from each of the three lights, wavelength by wavelength. In this sense, the RGB model is an additive model. Hence we can conclude that the three components are independent of each other. Due to this reason the multivariate test is equivalent to three independent univariate tests if the tests are under identical assumptions. We could perform hypothesis testings of equal multivariate distribution employing a multivariate test to obtain test p-values. Being in a three dimensional space, we use the Euclidean distance to measure the distance between the test points and the mean of classes. Once we have the p-values and the distance, we use the same decision rule discussed in Sections 3 and 4 to obtain the final classification of each pixel. In practice, however, nonparametric multivariate tests are not as stable as univariate tests for the same number of observations in the training data. Moreover, they need a large sample size to perform well. For example, when the sample size is small it is harder to get good density estimate in multivariate case than in the univariate case. Even when the large sample size is satisfied, there are additional difficulty with multivariate tests. For example, to allow the pixel value from any distribution, the multivariate test should not be restricted to multivariate normal distribution. Consequently, a nonparametric version of the Kruskal-Wallis rank test is desired. However, ranks for multivariate data are not uniquely defined. Ranking within each component is an example; spatial ranks and affine ranks are two other examples (chapter 6 of Thomas and Joseph [2010]). In recent years, ranking based on data depth was also studied (Yijun and Xuming [2006]). We defer detailed examination and comparison of these multivariate tests in image pixel classification to a separate study. Due to these reasons, we use the first approach for the classification of color images.

Before the implementation of our method and other classification methods, we remark that we choose not to use cross validation or bootstrap method to estimate extra-sample prediction error for the following reasons:

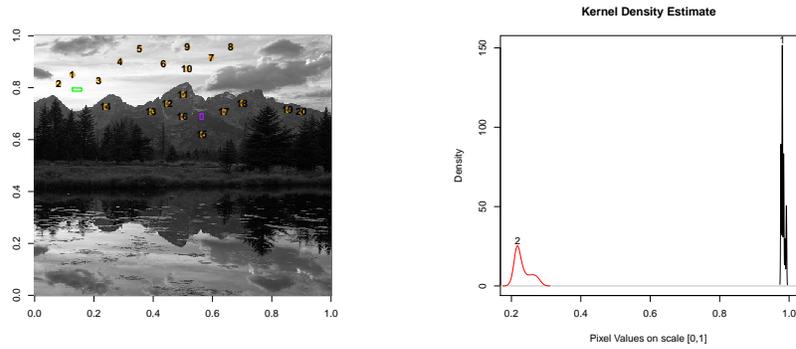
- Our training and testing data are randomly selected.
- In order to do cross-validation or bootstrap method to estimate extra-sample prediction error, we need the true class memberships for all pixels in the image. However this information is not clearly available to us for the entire image. Instead, the class memberships of some of the pixels are obvious, such as the pixels in a homogeneous region. Hence we are comfortable to manually decide the class of some of pixels but not for all pixels.
- In addition to difficulty of unavailability of true classmemberships of all the pixels, k-fold cross-validation tends to use more pixels as training data and less pixels as test data. In order to estimate the extra-sample prediction error, k need to be large. Leave one out cross-validation (LOOCV) is unbiased estimate for generalization error. This requires all pixels but one to be used as training data. In our reported case, we only used a very small portion of pixels to train the model.
- If our goal is to estimate the extra-sample generalization for one particular method, it would be necessary to conduct LOOCV. However, here our purpose is to compare different methods trained on the same training data and testing on the same test data. Due to the identical data used, the classification results from different methods do indicate how well each method performs relative to others. We admit that some methods might be sensitive to the training and test data selected. In our case, the training and test data are randomly selected from homogeneous regions. In images the between-class variations of pixels values are much bigger than within-class variations (which can be seen from the density plots). So we expect the classification results would not fluctuate a lot for different training and testing data selected.

6. Implementation

In this section, we implement our method along with the LA method for binary and multiclass classification of some grey scale images and compare their performance. Consider the image in Figure 2(a) and choose two classes, namely sky and mountain. Then following the procedure described in Section 2, training data are formed. In this image, size of the training data are 136 and 84 for class 1 and class 2. The size of the original image is 625×500 . Then the proportion of the training data are 0.00043 and 0.00026. Kernel density estimate is used to obtain the density plot of classes. We select 20 test points, labeled with numbers, in the image in such a way that the first 10 test points are selected from class 1 and the rest are from class 2. Prior probabilities of classes are obtained by using the first definition in Section 3. The density estimates of pixel values are plotted in the right panel of Figure 2 showing that the classes so formed are distinct.

The classification results of the LA method and our method are presented in Table 2. Our method correctly classified all the test points in Table 2 whereas the LA method has misclassified test points 3 – 10. So our method performs better than the LA method for binary classification of image pixels

FIG 2.



(a) Image with training data and test points. (b) Kernel Density estimate of classes.

TABLE 2

Classification of test points in image Figure 2(a)

TP	LA	Our	PV1	PV2	D1	D2	Pr1	Pr2	Obs.
1	class 1	class 1	2.489520e-36	5.717360e-35	0.009775	0.760224	0.258220	0.741779	0.992156
2	class 1	class 1	2.361023e-36	7.614957e-36	0.001989	0.748459	0.258220	0.741779	0.980392
3	class 2	class 1	2.506775e-36	1.748560e-36	0.009832	0.740616	0.258220	0.741779	0.972549
4	class 2	class 1	2.506775e-36	1.748560e-36	0.076499	0.673949	0.258220	0.741779	0.905882
5	class 2	class 1	2.506775e-36	1.748560e-36	0.103950	0.646498	0.258220	0.741779	0.878431
6	class 2	class 1	2.506775e-36	1.748560e-36	0.107872	0.642577	0.258220	0.741779	0.874509
7	class 2	class 1	2.506775e-36	1.748560e-36	0.201989	0.548459	0.258220	0.741779	0.780392
8	class 2	class 1	2.506775e-36	1.748560e-36	0.213754	0.536694	0.258220	0.741779	0.768627
9	class 2	class 1	2.506775e-36	1.748560e-36	0.135323	0.615126	0.258220	0.741779	0.847058
10	class 2	class 1	2.506775e-36	1.748560e-36	0.037283	0.713165	0.258220	0.741779	0.945098
11	class 2	class 2	2.506775e-36	1.748560e-36	0.676499	0.073949	0.258220	0.741779	0.305882
12	class 2	class 2	2.506775e-36	1.748560e-36	0.672577	0.077871	0.258220	0.741779	0.309803
13	class 2	class 2	2.506775e-36	1.748560e-36	0.652970	0.097479	0.258220	0.741779	0.329411
14	class 2	class 2	2.506775e-36	1.748560e-36	0.656891	0.093557	.258220	0.741779	0.325490
15	class 2	class 2	2.526761e-35	1.748560e-36	0.786303	0.035854	0.258220	0.741779	0.196078
16	class 2	class 2	2.506775e-36	1.748560e-36	0.672577	0.077871	0.258220	0.741779	0.309803
17	class 2	class 2	2.613017e-36	1.748480e-36	0.700028	0.050420	0.258220	0.741779	0.282352
18	class 2	class 2	2.506775e-36	1.748560e-36	0.692185	0.058263	0.258220	0.741779	0.290196
19	class 2	class 2	5.001380e-36	1.747767e-36	0.743166	0.007282	0.258220	0.741779	0.239215
20	class 2	class 2	4.542750e-36	1.748480e-36	0.735323	0.015126	0.258220	0.741779	0.247058

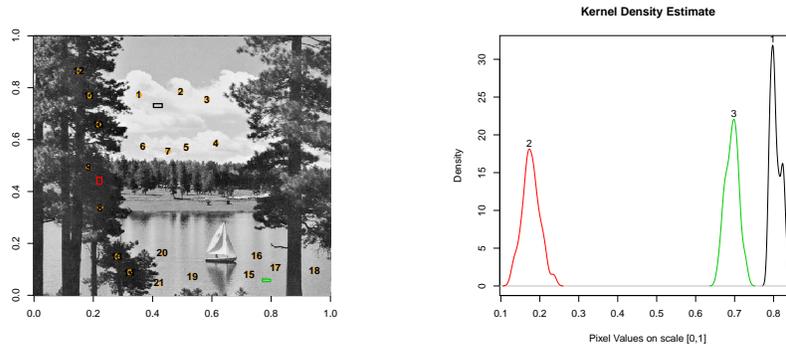
TP= Test point, Obs= Test point pixel, Our= Classification by our method, LA= Liao & Akritas's method
 PV_i= p-value from test i, Pr_i=Prior probability of class i, D_i= Distance of TP from mean of class i,

in the given image.

Next, we compare the performance of our method and LA method for multiclass classification of grey scale image pixels. Consider the image in Figure 3(a), a 512 × 512 image, in which sky, tree, and water are defined as three classes with training data formed accordingly described in Section 2. First 7 test points are from class 1; next 7 are from class 2 and the last 7 are from class 3 as shown in Figure 3(a). Right panel of Figure 3 shows the kernel density estimate of pixel values of the classes. Size and proportion of the training data for class 1 are 136 and 0.00051. Similarly 150 and 0.00057 are the size and proportion of training data for class 2. Finally, 96 and 0.00036 are size and proportion of class 3 training data in the given image. We evaluate the prior probabilities of the classes using the first definition in Section 4.

We applied the LA method and ours to classify the selected test points and the classification results are displayed in Table 3. Our method classifies all the test points accurately but LA method

FIG 3.



(a) Image with training data and test points. (b) Kernel Density estimate of classes.

TABLE 3

Classification of test points in image Figure 3(a)

TP	LA	Our	PV1	PV2	PV3	D1	D2	D3	Pr1	Pr2	Pr3	Obs.
1	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0411	0.6666	0.1490	0.1899	0.4731	0.3370	0.8431
2	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0372	0.6627	0.1451	0.1899	0.4731	0.3370	0.8392
3	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0725	0.6980	0.1803	0.1899	0.4731	0.3370	0.8745
4	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0725	0.6980	0.1803	0.1899	0.4731	0.3370	0.8745
5	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0647	0.6902	0.1725	0.1899	0.4731	0.3370	0.8667
6	class 1	class 1	1.198e-73	4.586e-72	3.920e-73	0.0568	0.6823	0.1647	0.1899	0.4731	0.3370	0.8588
7	class 1	class 1	1.193e-73	9.965e-73	1.870e-73	0.0019	0.6235	0.1058	0.1899	0.4731	0.3370	0.8000
8	class 3	class 2	1.532e-72	1.246e-73	2.538e-73	0.6294	0.0039	0.5215	0.1899	0.4731	0.3370	0.1725
9	class 3	class 2	2.966e-73	1.248e-73	1.089e-73	0.5588	0.0666	0.4509	0.1899	0.4731	0.3370	0.2431
10	class 3	class 2	4.327e-72	1.248e-73	4.324e-73	0.6568	0.0313	0.5490	0.1899	0.4731	0.3370	0.1451
11	class 3	class 2	2.966e-73	1.248e-73	1.089e-73	0.4961	0.1294	0.3882	0.1899	0.4731	0.3370	0.3059
12	class 3	class 2	2.966e-73	1.248e-73	1.089e-73	0.5588	0.0666	0.4509	0.1899	0.4731	0.3370	0.2431
13	class 3	class 2	1.532e-72	1.246e-73	2.538e-73	0.6294	0.0039	0.5215	0.1899	0.4731	0.3370	0.1725
14	class 3	class 2	5.301e-72	1.248e-73	4.797e-73	0.6725	0.0470	0.5647	0.1899	0.4731	0.3370	0.1294
15	class 3	class 3	1.262e-73	3.092e-73	1.089e-73	0.0803	0.5451	0.0274	0.1899	0.4731	0.3370	0.7216
16	class 3	class 3	2.435e-73	1.542e-73	1.089e-73	0.1235	0.5019	0.0156	0.1899	0.4731	0.3370	0.6784
17	class 3	class 3	2.055e-73	1.847e-73	1.089e-73	0.1117	0.5137	0.0039	0.1899	0.4731	0.3370	0.6902
18	class 3	class 3	1.741e-73	2.200e-73	1.089e-73	0.1039	0.5215	0.0039	0.1899	0.4731	0.3370	0.6980
19	class 3	class 3	2.925e-73	1.267e-73	1.089e-73	0.1431	0.4823	0.0352	0.1899	0.4731	0.3370	0.6588
20	class 3	class 3	2.185e-73	1.731e-73	1.089e-73	0.1156	0.5098	0.0078	0.1899	0.4731	0.3370	0.6863
21	class 3	class 3	2.185e-73	1.731e-73	1.089e-73	0.1156	0.5098	0.0078	0.1899	0.4731	0.3370	0.6863

TP= Test point, Obs= Test point pixel,

PVi= p-value from test i, Di = Distance of TP from mean of class i,

Pri=Prior probability of class i, Our= Classification by our method, LA=Liao & Akritas's method

misclassifies 7 test points. So our method works better than the LA method in the given image. Extensive experiments and comparisons were conducted for binary and multiclass classification on image pixel values. The results can be found in Ghimire [2011]. The general conclusion from these experiments is that the LA method works fine if all the p-values are large and could misclassify a test point when multiple p-values are small. Our method, on the other hand, handles both situations very well.

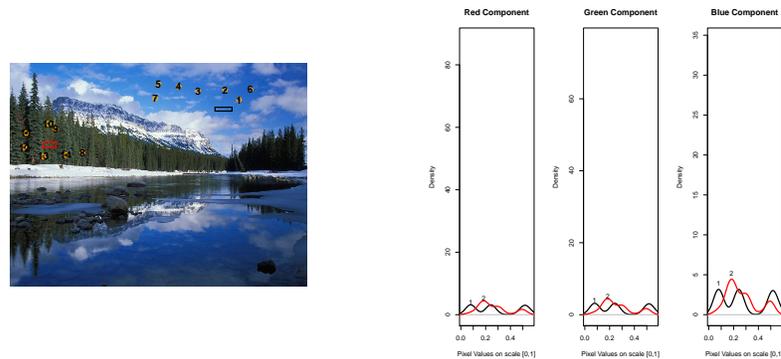
7. Comparison with other methods

In this section, we compare our method of classification with some of the standard classification methods, both statistical and computer-based, in classifying image pixels in color images. The standard statistical classifiers include LDA, QDA, polychlor, classification tree. Liao and Akritas's

method will also be included in the comparison. We will compare our method with a computer-based method, namely support vector machine.

Consider the image in Figure 4(a) which is a 602×452 image. Let us define the vegetation and sky region as our two classes and rectangles are formed as described in Section 2 to obtain the training data. The training data for class 1 has size 315 and has proportion 0.00115. Similarly, 338 and 0.00124 are the size and proportion of class 2 training data. Density plots of RGB components of classes are obtained by using kernel density estimates and are shown in Figure 4(b). As before, some test points are selected from each of the regions representing classes. For the classification of the selected test points by the given statistical classifiers, we consider each of the grey scale components of the image, namely, red, green, and blue, and employ our method and the other methods to classify the test points in each of these components. Componentwise classification of the test points are shown in Table 4 along with the pixel values which is scaled to range $[0, 1]$.

FIG 4.



(a) Image with training data and test points. (b) Kernel Density estimate of classes.

After the classification of test points in each component, we use the majority of votes discussed in Section 5 to obtain the final classification of test points. As we have 2 classes and 3 components, we will not have any tie while employing the majority of votes. Next, we employ support vector machine to classify the selected test points in the image and the its classification result is presented in Table 5 along with the final classification of chosen statistical classifiers.

We note that first 7 of the 14 selected test points in the given image were taken from class 1 (sky) and the rest of the test points were from class 2 (vegetation). Table 5 shows that our method of classification has no misclassification while other methods have misclassifications.

Next, we compare the multiclass classification performance of our method with other methods in Figure 5(a). In this image we consider three classes which are the grass region as class 1, the sky region as class 2 and the tree region as class 3. The training data are formed and some test points are selected in the image. Componentwise kernel density estimates of classes are shown in Figure

TABLE 4
Classification of test points in RGB components in image Figure 4

Compt	TP	LA	OUR	LDA	QDA	TREE	POLY	Value
Red	1	class 1	class 1	class 1	class 1	class 1	class 1	0.08
	2	class 2	class 1	0.09				
	3	class 2	class 1	0.09				
	4	class 2	class 1	class 1	class 1	class 2	class 2	0.10
	5	class 2	class 1	class 1	class 1	class 2	class 2	0.10
	6	class 2	class 1	0.09				
	7	class 1	0.08					
	8	class 2	0.14					
	9	class 2	0.16					
	10	class 2	0.15					
	11	class 1	class 1	class 1	class 1	class 2	class 2	0.05
	12	class 2	0.22					
	13	class 2	0.18					
	14	class 2	0.18					
Green	1	class 1	class 1	class 1	class 2	class 2	class 2	0.21
	2	class 1	class 1	class 1	class 2	class 2	class 2	0.21
	3	class 1	class 1	class 2	class 2	class 2	class 2	0.22
	4	class 1	class 1	class 2	class 2	class 2	class 2	0.22
	5	class 1	class 1	class 2	class 2	class 2	class 2	0.22
	6	class 1	class 1	class 1	class 1	class 2	class 2	0.19
	7	class 1	class 1	class 1	class 2	class 1	class 1	0.23
	8	class 1	class 1	class 1	class 1	class 2	class 2	0.17
	9	class 1	class 1	class 1	class 1	class 2	class 2	0.17
	10	class 1	class 1	class 1	class 1	class 2	class 2	0.16
	11	class 1	class 1	class 1	class 1	class 2	class 2	0.04
	12	class 2	class 1	0.26				
	13	class 1	class 1	class 1	class 2	class 2	class 2	0.22
	14	class 1	class 1	class 1	class 2	class 2	class 2	0.22
Blue	1	class 2	0.43					
	2	class 2	class 2	class 2	class 2	class 2	class 2	0.41
	3	class 2	0.44					
	4	class 2	0.43					
	5	class 2	0.44					
	6	class 2	0.38					
	7	class 1	0.53					
	8	class 2	0.19					
	9	class 2	0.15					
	10	class 2	0.12					
	11	class 2	0.04					
	12	class 2	0.31					
	13	class 2	0.27					
	14	class 2	0.22					

5(b). As shown in the image, first five test points are selected from class 1, the next five from class 2 and so on. Size of the training data for the defined classes are 275, 270 and 100 respectively. The given image has size 512×512 so that proportion of the training data are 0.00104, 0.00102 and 0.00038 respectively.

We perform the componentwise classification of the selected test points to employ the chosen statistical classifiers and is displayed in Table 6. To obtain the final classification of the test points, we use the rule of majority of votes described earlier. As before, in case of tie, we randomly assign the test point to one of the classes. After the classification by statistical classifiers, we classify the selected test points by support vector machine. The classification result of all the methods are presented in Table 7.

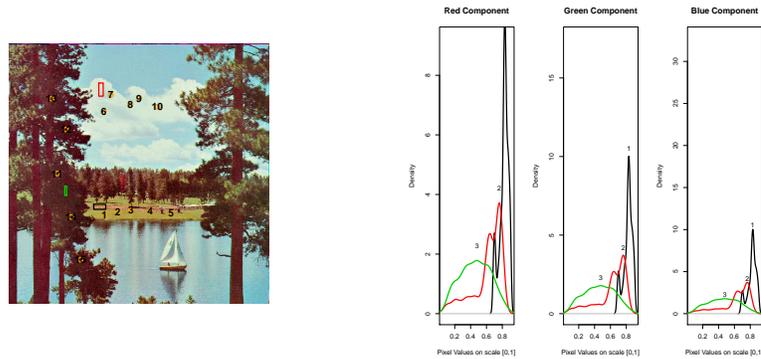
From Table 7, we observe that only a test point, namely, 15, is misclassified by our method. The other methods have more misclassified test points. Additional experiments were performed to compare our method with the other methods in different color images (see Ghimire [2011]). These experiments confirm that our method has a smaller misclassification rate than other methods. From the above discussion, we can conclude that our method performs better than other methods for

TABLE 5
Final classification of test points in Figure 4

TP	LA	OUR	LDA	QDA	TREE	POLY	SVM
1	class 1	class 1	class 1	class 2	class 2	class 2	class 1
2	class 2	class 1	class 1	class 2	class 2	class 2	class 2
3	class 2	class 1	class 2	class 2	class 2	class 2	class 1
4	class 2	class 1	class 2				
5	class 2	class 1	class 2				
6	class 2	class 1	class 1	class 2	class 2	class 2	class 2
7	class 1						
8	class 2						
9	class 2						
10	class 2						
11	class 1	class 1	class 1	class 2	class 2	class 2	class 2
12	class 2						
13	class 2						
14	class 2						

LA=Liao & Akritas's method, OUR= Our method
LDA=Linear Discriminant Analysis, QDA=Quadratic Discriminant Analysis
TREE=Classification Tree, POLY=Polyclass method,
SVM=Support Vector Machine, TP=Test Point

FIG 5.



(a) Image with training data and test points. (b) Kernel Density estimate of classes.

color image pixel classification.

8. Segmentation of images and properties of the method.

In this section, we employ our method of classifier to segment color images. Formally, we can define image segmentation as follows. If the domain of the image is given by Ω then the segmentation problem is to determine the sets $S_k \subset \Omega$ whose union is the entire domain Ω . Thus, the sets that make up a segmentation must satisfy $\Omega = \cup_{k=1}^K S_k$ where $S_k \cap S_j = \emptyset$ for $k \neq j$, and each S_k is connected (Dzung L. Pham and Prince [2000]). We first briefly discuss the implementation of our classifier method for segmentation. For this, we define our classes (segments) in the image where the classes are simply some quantifiable features in the image. Training data for these classes are then obtained by forming the rectangles in the regions representing the classes. Pixel values corresponding to three components are then combined to form the training data. Using these training data, we classify all the pixels in the image using our method of classification discussed in Sections

TABLE 6
 Classification of test points in image Figure 5

Compt	TP	LA	OUR	LDA	QDA	TREE	POLY	Value	
Red	1	class 1	class 1	class 1	class 1	class 1	class 1	0.80	
	2	class 1	0.79						
	3	class 1	0.77						
	4	class 1	0.71						
	5	class 2	0.65						
	6	class 3	class 2	0.60					
	7	class 1	0.71						
	8	class 2	0.60						
	9	class 2	0.62						
	10	class 1	class 2	class 1	class 1	class 1	class 1	0.69	
	11	class 3	class 2	class 3	0.29				
	12	class 3	class 2	class 3	0.31				
	13	class 3	class 3	class 3	class 3	class 2	class 3	class 2	0.43
	14	class 3	class 2	class 3	0.29				
	15	class 1	0.71						
Green	1	class 1	class 1	class 1	class 1	class 1	class 1	0.87	
	2	class 1	0.87						
	3	class 1	0.85						
	4	class 2	class 2	class 1	class 2	class 2	class 2	0.81	
	5	class 2	class 1	class 1	class 2	class 2	class 2	0.82	
	6	class 2	0.69						
	7	class 2	class 2	class 1	class 2	class 2	class 2	0.78	
	8	class 2	0.75						
	9	class 2	0.71						
	10	class 2	class 2	class 1	class 2	class 2	class 2	0.77	
	11	class 3	0.12						
	12	class 3	0.14						
	13	class 3	class 2	class 3	0.29				
	14	class 3	0.06						
	15	class 2	class 2	class 1	class 2	class 2	class 2	class 2	0.79
Blue	1	class 1	class 1	class 1	class 1	class 1	class 1	0.83	
	2	class 2	class 1	0.81					
	3	class 2	class 1	class 1	class 1	class 1	class 2	0.81	
	4	class 2	class 1	0.81					
	5	class 1	0.83						
	6	class 2	0.65						
	7	class 2	class 2	class 1	class 2	class 2	class 2	0.75	
	8	class 2	0.71						
	9	class 2	0.67						
	10	class 2	class 2	class 1	class 2	class 2	class 2	0.75	
	11	class 3	class 2	class 2	0.08				
	12	class 3	class 2	class 2	0.11				
	13	class 3	class 3	class 3	class 3	class 2	class 2	class 2	0.21
	14	class 3	class 2	class 2	0.08				
	15	class 2	0.68						

3-5, considering one dimension (RGB) of pixels at a time. For the final classification of each pixel in the image, we employ the majority of votes classification described earlier. In this way, all the image pixels are classified resulting in the complete segmentation of the image.

We first perform segmentation using our classifier method in the image shown in the upper leftmost panel of Figure 6. The objects in this image has many ill-defined edges. We would like to illustrate with this example that our classifier method works well in the segmentation of images with too many edges or ill defined edges. The size of the given image is 256×384 where we define 6 classes as the segments and training data are obtained for these classes as before. Size of the training data for the given six classes are 702, 936, 936, 957, 1485, and 1188 respectively. Then the proportion of the training data of the classes are 0.00714, 0.00952, 0.00952, 0.00973, 0.01510, and 0.01208 respectively. The size of the Kernel density estimate for each component was obtained and plotted in the middle panel of the first row in Figure 6. Then we perform segmentation using our method and the segmented image is displayed in the upper right panel of the Figure 6. It can be seen that our method has segmented the image accurately. From the density plot of classes, we see that majority of the classes are overlapping but our segmentation looks okay even if the classes

TABLE 7
Final classification of test points in image Figure 5

TP	LA	OUR	LDA	QDA	TREE	POLY	SVM
1	class 1	class 1	class 1	class 1	class 1	class 1	class 1
2	class 1						
3	class 1						
4	class 2	class 1					
5	class 2	class 1	class 1	class 2	class 2	class 2	class 2
6	class 2	class 3					
7	class 2	class 2	class 1	class 2	class 2	class 2	class 2
8	class 2						
9	class 2						
10	class 2	class 2	class 1	class 2	class 2	class 2	class 2
11	class 3	class 3	class 3	class 3	class 2	class 3	class 2
12	class 3	class 3	class 3	class 3	class 2	class 3	class 2
13	class 3	class 3	class 3	class 2	class 2	class 2	class 3
14	class 3	class 3	class 3	class 3	class 2	class 3	class 2
15	class 2	class 2	class 1	class 2	class 2	class 2	class 2

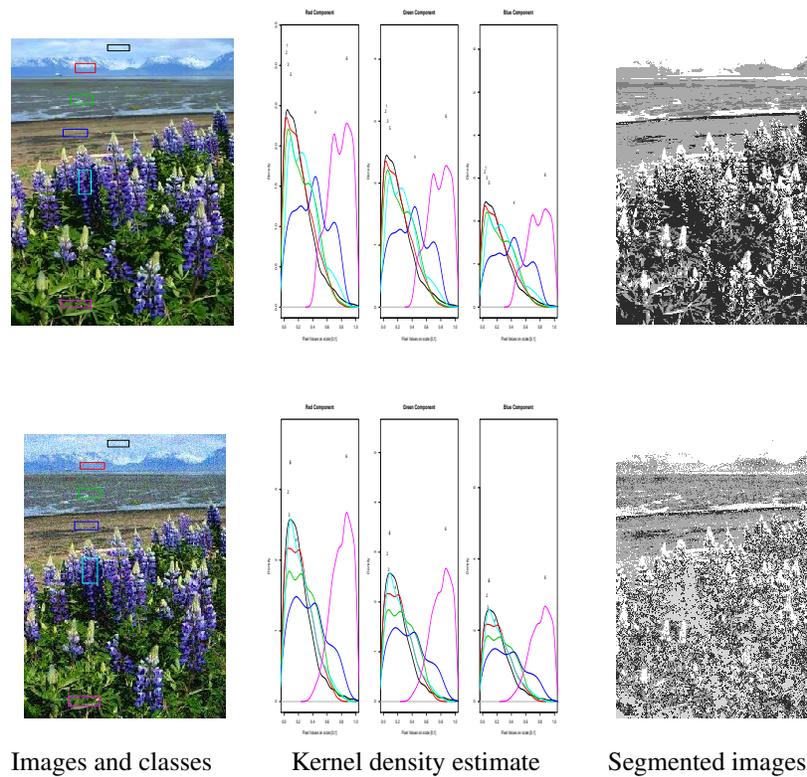
LA= Liao & Akritas's method, OUR= Our method,
LDA=Linear Discriminant Analysis, QDA=Quadratic Discriminant Analysis
TREE=Classification Tree, POLY=Polyclass method,
SVM=Support Vector Machine, TP=Test Point

are overlapping indicating that segmentation can also be done by considering smaller number of classes in the image.

Robustness of a segmentation method on noisy images is a desired property for image processing. We consider the image from the previous example and add Gaussian noise with mean 0 and variance 0.06 (for image pixel values in $[0, 1]$) into it. The image in the bottom leftmost panel of Figure 6 shows this noisy image along with the training data for the defined classes. In the noisy image, size of the training data for the defined classes are 840, 960, 1218, 1116, 2100, 1722 and hence the proportion are 0.00854, 0.00976, 0.01239, 0.01135, 0.02136, 0.01751 respectively. We segment this noisy image with our classifier method. The result is shown in the lower rightmost panel of Figure 6. From the segmented image, we see that the segmentation result is satisfactory though it seems to be much noisier than in previous case.

Next, we perform segmentation in another image given in the upper leftmost panel of Figure 7. The given image has dimensions 481×321 . The purpose of this example is to show that our classifier method produces a closed curve or boundary among different quantifiable features in a segmented image. So, we form training data and obtain density estimates as before and employ our method. Here the size of the training data for classes are 252, 1080, 1215, 546, 1014 and the proportion to the original image are 0.00163, 0.00699, 0.00786, 0.00386, and 0.00656 respectively. The segmented image is shown in the upper right panel of Figure 7. One can see that a clear boundary among each object or feature of the segmented image was identified accurately. After this, we add some noise in the given image whose pixel values are scaled to range $[0, 1]$ and as before we segment this noisy image. The training data of classes in the noisy image has size 120, 1368, 1248, 1209, and 1395 respectively. Then the proportion of the training data of the defined classes are 0.00777, 0.00886, 0.00808, 0.00783, and 0.00903. The image in the bottom leftmost panel is the image with Gaussian noise with mean 0 and variance 0.06. Our method still produces

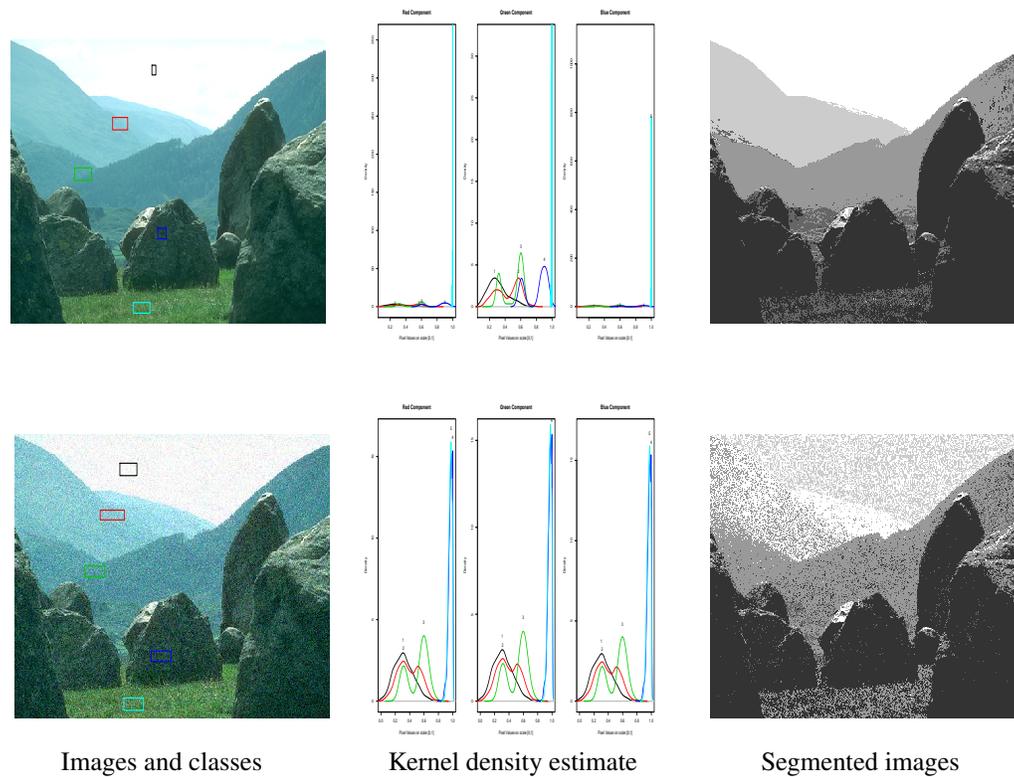
FIG 6.



a clear boundary among different features in the noisy segmented image. But we note that this segmented image is much noisier than in previous case. From the density plot as shown in the middle panel of Figure 7, we note that classes 1 and 2 overlap mostly. But the boundaries between these classes in the segmented image are very well detected. Also there is well defined boundary between classes 4 and 5 although they overlap in the density plot. The mass of the pixel values of class 3 is well separated from the rest of the classes.

Finally, we perform an automatic segmentation of color images by our method. Here, by automatic segmentation we mean the segmentation of images using the training data from a different image. So for automatic segmentation, we consider the image shown in the leftmost panel of Figure 8 and form training data for the defined classes. The size of the training data for the classes are 702, 936, 936, 957, 1485, and 1188. Here the image has dimensions 256×384 so that the proportion of the training data are 0.00714, 0.00952, 0.00952, 0.00973, 0.01510, and 0.01208 respectively. Using the training data of this image, we employ our method to segment a different image, namely the image in upper leftmost panel of Figure 7. The segmented image is displayed in the rightmost panel of Figure 8. In the segmented image we observe that the boundaries among the first three classes are weak and are almost indistinguishable. From the density plot of classes, displayed in

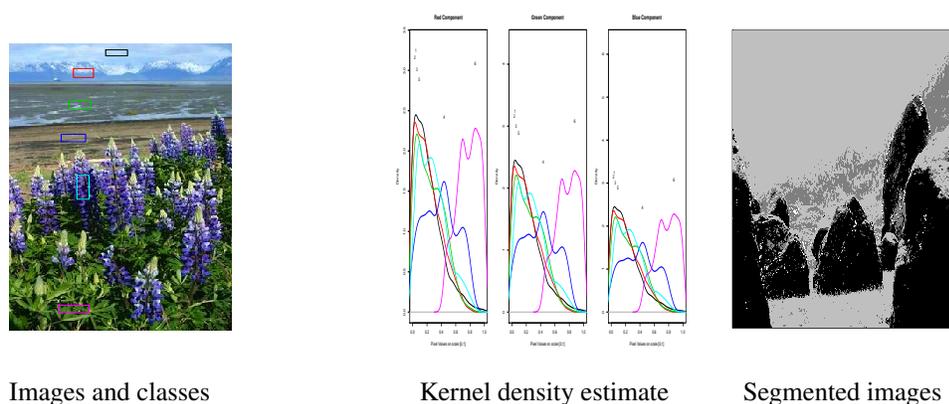
FIG 7.



the middle panel of Figure 8, we observe that the first 3 classes overlap mostly which makes the program to be almost blind to the boundaries of the first 3 classes. Due to this reason, there is a weak boundary among some features in the segmented image. In this way, using training data from an image, we can segment many other images resulting in the automatic segmentation of images.

From the above implementation, we see that our method is a straightforward classification method and can easily be implemented. Manual selection of training data allow us to have better classification accuracy. Vaguely defined classes from another image can also be used to guide the classification. However, some of the boundaries may not be accurately identified due to different behavior of the training data compared to the image to be segmented. As the pixel values within a well-defined class are relatively close, our hypothesis testing based method would produce similar decisions for these pixels. This guarantees that the segmented images are contiguous. Moreover, our method can also be implemented in images where homogeneity criteria is hard to define.

FIG 8.



Images and classes

Kernel density estimate

Segmented images

9. Summary

In this work, we introduced a new method of image pixels classification that works well in classifying pixels in grey scale and color images. The classifier uses p-values from hypothesis testings and distance of test points from the mean of classes to make decisions. In the classification of image pixels, we observed that test p-values are small due to the size of training data. Theoretically, the p-values in a valid test follow the uniform $(0, 1)$ distribution. Hence two test p-values both smaller than a significance level do not provide a different level of evidence to reject the null hypothesis. Consequently, p-values alone are not adequate to make correct classifications. So we introduced a combination of hypothesis testing and minimum distance in our classification. In the implementation of our method, we observed that distance was mostly used for the classification if the pixel values of classes were separated. The minimum distance classification works well when there is low variability within classes and classes are distinct. But for classes with large within variability, the minimum distance alone may not give the correct classification and we need to use p-values. Using our classifier method, we can also segment images. Our method is computationally efficient and works well in noisy images too. The method produces a clear boundary among different features which is essential in segmentation and the method works well in the images with too many edges or ill defined edges. The drawback of our method is that the training data have to be obtained manually.

References

- Breiman, L., Friedman, H. J., Olshen, A. R., and Stone, J. C. (1998). *Classification and regression tree*. Chapman and Hall/CRC.
- Coleman, G. and Andrews, H. (1979). Image segmentation for clustering. *Proceeding of IEEE*, 67:773–785.

- Dutta, S. and Chaudhari, B. B. (2009). Homogenous region based color image segmentation. *Proceedings of the world congress on engineering and computer science*, 2.
- Dzung L. Pham, C. X. and Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337.
- Ghimire, S. (2011). Image pixels classification using minimum distance and hypothesis testings. *Master's Thesis, Department Of Statistics, Kansas State University*.
- Haralick, R. M. and Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132.
- Hastie, T., Tibshirani, R., and J., F. (2009). *The Elements of Statistical Learning*. Springer-Verlag, New York.
- Liao, S.-M. and Akritas, M. (2007). Test-based classification: A linkage between classification and statistical testing. *Statistics and Probability Letters*, 77:1269–1281.
- Pal, N. R. and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294.
- Paul, S. D. (1981). *Colour vision in the nineteenth century: the Young-Helmholtz-Maxwell theory*. A. Hilger (Bristol and Philadelphia, PA).
- Stone, J. C., Hansen, H. M., Kooperberg, C., and Truong, K. Y. (1997). Polynomial splines and their tensor products in extended linear modeling. *Annals of Statistics*, 25:1371–1470.
- Thomas, H. P. and Joseph, M. W. (2010). *Robust Nonparametric Statistical Methods*. Second edition, Chapman and Hall/CRC.
- Vapnik, V. (1982). *The nature of Statistical Learning Theory*. Springer-Verlag, New York.
- Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on pattern analysis and machine intelligence*, 13.
- Yijun, Z. and Xuming, H. (2006). On the limiting distributions of multivariate depth-based rank sum statistics and related tests. *Annals of Statistics*, 34:2879–2896.