remote sensing

MDPI

Article

Matlab Software for Supervised Habitat Mapping of Freshwater Systems Using Image Processing

Johnathan M. Bardsley [1,*], Marylesa Howard [2] and Mark Lorang [3]

[1] Mathematical Sciences Department, University of Montana, Missoula, MT 59812, USA
[2] Signal Processing and Applied Mathematics Group, Nevada National Security Site, North Las Vegas, NV 89030, USA; howardmm@nv.doe.gov
[3] Freshwater Map, Bigfork, MT 59911, USA; mark@freshwatermap.com
* Correspondence: johnathan.bardsley@umontana.edu

Abstract: We present a software package for the supervised classification of images useful for cover-type mapping of freshwater habitat (e.g., water surface, gravel bars, vegetation). The software allows the user to select a representative subset of pixels within a specific area of interest in the image that the user has identified as a cover-type habitat of interest. We developed a graphical user interface (GUI) that allows the user to select single pixels using a dot, line, or group of pixels within a defined polygon that appears to the user to have a spectral similarity. Histogram plots for each band of the selected ground-truth subset aid the user in determining whether to accept or reject it as input data for the classification processes. A statistical model, or classifier, is then built using this pixel subset to assign every pixel in the image to a best-fit group based on reflectance or spectral similarity. Ideally, a classifier incorporates both spectral and spatial information. In our software, we implement quadratic discriminant analysis (QDA) for spectral classification and choose three spatial methods—mode filtering, probability label relaxation, and Markov random fields—to incorporate spatial context after computation of the spectral type. This multi-step interactive process makes the software quantitatively robust, broadly applicable, and easily usable for cover-type mapping of rivers, their floodplains, wetlands often components of these functionally linked freshwater systems. Indeed, this supervised classification approach is helpful for a wide range of cover-type mapping applications in freshwater systems but also estuarine and coastal systems as well. However, it can also aid many other applications, specifically for automatic and quantitative extraction of pixels that represent the water surface area of rivers and floodplains.

Keywords: habitat mapping; remote sensing; supervised classification

Citation: Bardsley, J.M.; Howard, M.; Lorang, M. Matlab Software for Supervised Habitat Mapping of Freshwater Systems Using Image Processing. *Remote Sens.* **2021**, *13*, 4906. https://doi.org/10.3390/rs13234906

Academic Editor: Guillaume Ramillien

Received: 10 November 2021
Accepted: 30 November 2021
Published: 3 December 2021

1. Introduction

Passive remote sensing is the act of making observations from afar of light reflected from an object. This image data type is typically collected using a sensor mounted on an aircraft or spacecraft [1]. Statistically grouping the data is required to classify the imagery in cover-types of similarity before extracting other quantitative information about the object or similar objects within the field of view.

Supervised classification refers to a class of methods used in the quantitative analysis of remote sensing image data. These methods require that the user provide the set of cover types in the image—e.g., water, cobble, deciduous forest, etc.—as well as a training field for each cover type. The training field typically corresponds to an area in the image that contains the cover type, and the collection of all training fields is known as the training set or ground-truth data. The ground-truth data is then used to assign each pixel to its most probable cover type.

Our objective in this paper is to present MATLAB software for the classification of remotely sensed images. Several well-known software packages exist for classifying remote sensing imagery, e.g., ArcGIS, however, these packages are expensive and tend

to be complicated enough that a GIS specialist is required for their use. MATLAB is also expensive, but it is widely used in the imaging science and scientific computing communities, and our package is stripped-down and easy-to-use, requiring no special expertise. This software will provide the applied scientist who has access to MATLAB a software package containing a few of the highly effective classification methods presented in [1]. The software is also unique in that it allows the user to create ground-truth data from a particular image via a graphical user interface (GUI). The software includes spectral-based classification as well as classifiers that incorporate the spatial correlation within typical images. We emphasize that we are not presenting any new classification methods in this paper, and so we do not provide comparisons with other algorithms.

A remotely sensed image contains both spectral and spatial information. In our software package, we advocate first computing a spectral-based classification using quadratic discriminant analysis (QDA) (also called maximum likelihood classification [1]). QDA is popular due to its simplicity and compact training process [2]. It assumes images are multivariate normally distributed, which suggests a natural thresholding method. A vast number of other supervised classification schemes exist [3], however in [4] it is noted that simpler methods, such as QDA, often achieve over 90% of the predictive power obtained by the most advanced methods, though such good performance is not guaranteed. Moreover, in the comparison studies of algorithms found in [2], it is shown that the QDA classifier excels over artificial neural networks and decision trees in design effort, training time, and classification accuracy; and in [5,6] it was shown that although neural/connectionist algorithms make no assumption on statistical distribution, as does QDA, training time can be lengthy and the choice of design architecture is rather complex.

Nonetheless, a major drawback of QDA is that it ignores the spatial arrangement of the pixels within an image [7]. The correlation that exists between neighboring pixels within an image (see, e.g., [8–10]) should, in many instances, be incorporated into a classification scheme [1,11]. The effect of spatial autocorrelation and its relation to spatial resolution in an image has been explored in remote sensing [9,10,12], but less has been done to directly incorporate spatial autocorrelation into classification. In our software, we use the pre-classification filters and neighborhood exploitation methods described in [1,12] for dealing with spatial autocorrelation. Each of these techniques may or may not reclassify a given pixel based on information about its neighbors. Finally, in order to test the effectiveness of our classification schemes, we have implemented k-fold cross validation.

The classification techniques used in remote sensing come from the field of mathematical pattern recognition and are extensively covered in work within that discipline. Particular references that give more background and more variation in algorithms than we have presented above are found in [13–15]. We emphasize, again, that our goal here is to present a useful software package, not to present new classification algorithms, which would require comparisons and a longer list of references. However, we do point the reader to some recent references, first to habitat mapping algorithms incorporating Moran's I spatial autocorrelation [16,17]. Spatial relationships of pixels in the image are also in object-based image analysis (OBIA), which is also utilized in underwater habitat mappying studies [18–20].

This paper is arranged as follows. First, in Section 2, we derive both non-spatial and spatial algorithms for supervised classification and present *k*-fold cross validation, as well as a technique for thresholding. In Section 3, classification is carried out on a variety of images using a sampling of the methods from Section 2. A discussion and summary of the methods and the results is given in Section 5.

2. Methods

We begin with a presentation our spectral-based classifier, QDA. The spatial-based techniques follow, and then we present a thresholding method, as well as *k*-fold cross validation for error assessment.

2.1. Spectral-Based Classification: Quadratic Discriminant Analysis

We begin by letting ω_i be the ith of the M spectral classes, or cover types, in the image to be classified. When considering to which spectral class, or cover type, a particular pixel vector $\boldsymbol{x} \in \mathbb{R}^k$ belongs, we denote the conditional probability of $\boldsymbol{x}$ belonging to ω_i by $p(\omega_i|\boldsymbol{x})$, for $i = 1, \ldots, M$. We are interested in the spectral class that attains the greatest probability given $\boldsymbol{x}$. Thus classification will yield $\boldsymbol{x} \in \omega_i$ when

$$p(\omega_i|\boldsymbol{x}) > p(\omega_j|\boldsymbol{x}) \qquad \text{for all} \quad j \neq i. \tag{1}$$

It remains to define $p(\omega_i|\boldsymbol{x})$, which can be related to the conditional probability $p(\boldsymbol{x}|\omega_i)$ via Bayes' theorem

$$p(\omega_i|\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)p(\omega_i)/p(\boldsymbol{x}), \tag{2}$$

where $p(\omega_i)$ is the probability that the ith spectral class occurs in the image; and $p(\boldsymbol{x})$ is the probability density associated with the spectral vector $\boldsymbol{x}$ over all spectral classes. In Bayes' terminology, $p(\omega_i)$ is referred to as the prior probability and is used to model *a priori* knowledge.

Substituting (1) into (2), and assuming that $p(\omega_i)$ is the same for all i, the inequality (1) takes the form

$$p(\boldsymbol{x}|\omega_i) > p(\boldsymbol{x}|\omega_j) \qquad \text{for all} \quad j \neq i. \tag{3}$$

We assume the probability distributions for the classes are multivariate normal, so that

$$p(\boldsymbol{x}|\omega_i) = ((2\pi)^N |\Sigma_i|)^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)' \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{m}_i)\right), \tag{4}$$

where $\boldsymbol{m}_i \in \mathbb{R}^k$ and $\Sigma_i \in \mathbb{R}^{k \times k}$ are the mean vector and covariance matrix, respectively, of the data in class ω_i.

To simplify the above equation, we use the discriminant functions $g_i(\boldsymbol{x})$ defined to be the natural logarithm of (3), which preserves the inequality since the logarithm is a monotone increasing function. Thus our method classifies $\boldsymbol{x}$ as an element of ω_i when

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \qquad \text{for all} \quad j \neq i, \tag{5}$$

where

$$g_i(\boldsymbol{x}) = -\ln |\Sigma_i| - (\boldsymbol{x} - \boldsymbol{m}_i)' \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{m}_i). \tag{6}$$

From here, a variety of different classifiers can be derived based on assumptions on the discriminant function and the probability model. However, we have found quadratic discriminant analysis (QDA) to be both very effective, computationally efficient, as well as straightforward to implement. QDA is given by (5), (6), with the mean and covariance in (6) approximated using the empirical mean and covariance of the data in ground-truth class i. The effective k-nearest neighbor method [3] was not implemented due to its greater computational burden.

2.2. Spatial-Based Techniques

In QDA, the classification of a particular pixel does not depend on that of it's neighbors. In certain instances, it is clear that neighbor information should be taken into account. For instance, in an image containing a river, the pixels in the "water class" should be distributed in a spatially uniform manner in the classified image. In order to take into account such prior information, additional techniques are needed.

Four separate approaches were taken for incorporating spatial autocorrelation of the remotely sensed data into classification: pre-classification filtering, probabilistic label relaxation, post-classification filtering, and Markov random Fields. Each of the four are described in detail in the following subsections.

2.2.1. Pre-Classification Filtering

Pre-classification filtering can be viewed as a denoising technique (low pass filter) for the image to be classified. The filtering is implemented prior to classification and hence does not directly incorporate spatial context into the classification scheme. However, it does impose a degree of homogeneity among neighboring pixels, increasing the probability that adjacent pixels will be classified into the same spectral class.

It requires a neighborhood matrix. In our implementation, we include a first-, second-, and third-order neighborhood:

$$N_1 = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad N_2 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad N_3 = \frac{1}{36}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 1 & 4 & 8 & 4 & 1 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{7}$$

Convolution of an image by one of these neighborhood matrices yields a new image in which each pixel value has been replaced by a weighted average of the neighboring pixels (including itself) of the original image, with the weighting given explicitly by the values in the chosen neighborhood matrix. If the neighborhood is too large comparative to the degree of spatial autocorrelation, the quantitative distinction between cover types could be lost, so the hope is that the neighborhoods used are large enough to incorporate neighboring information, but not so large that individuality of pixels is lost.

To incorporate pre-classification filtering into the algorithm, the `conv2` function in MATLAB is used. Each band of the image was convolved separately with the neighborhood matrix and the same size matrix was asked to be returned so the buffer placed on the image was not returned as well.

2.2.2. Post-Classification Filtering

To filter post-classification, a classification labeling must first be performed. The classified map is then filtered with a defined neighborhood to develop a degree of spatial context. This is a similar idea to the pre-classification filtering method, however, the `conv2` function from MATLAB cannot be used in this case due to the categorical response of the classified map; e.g., the spectral classes are not ordered such that a pixel in class 1 is closer in form to a pixel in class 2 than it is to a pixel in class 5. Thus a weighted average of neighboring pixels will not give us information about the central pixel.

Instead we must filter using a different approach [21]. First, a window size is chosen to define the neighborhood size—in our implementation, we use a 3×3 window surrounding the central pixel—and each pixel is relabeled by selecting the spectral class that occurs most often in the neighborhood window.

To implement this in MATLAB, a 1-pixel buffer was placed around the entire image by replicating its closest pixel, since it is probable that the boundary pixel's adjacent pixel is of the same spectral class. The `nlfilter` function was then used on the classified map with the `mode` function, and the buffer pixels were removed.

2.2.3. Probabilistic Label Relaxation

While post-classification filtering can be effective, it only uses the classified image to incorporate spatial context. Given that we have the values of the discriminant function (and hence the corresponding probabilities), it is tempting to want to use this information as well. Probability label relaxation (PLR) is a technique that uses this information to incorporate spatial homogeneity in the classification [1].

The PLR algorithm begins with a classification—in our case using QDA—as well as with the corresponding set of probabilities $p_{\boldsymbol{x}}(\omega_i)$ that pixel $\boldsymbol{x}$ belongs to spectral class ω_i. A neighborhood is then defined for $\boldsymbol{x}$ and neighboring pixels are denoted by $\boldsymbol{y}$. The neighborhood should be large enough that all pixels considered to have spatial

correlation with pixel $\boldsymbol{x}$ are included [1]; in our implementation, we simply use the pixels to the left, right, above, and below.

A neighborhood function $Q_{\boldsymbol{x}}(\omega_i)$ is then defined so that all pixels in the neighborhood have some influence over the probability of pixel $\boldsymbol{x}$ belonging to ω_i. The probabilities defined by $p_{\boldsymbol{x}}(\omega_i)$ are then updated and normalized using $Q_{\boldsymbol{x}}(\omega_i)$. This is done iteratively so that the PLR algorithm has the form

$$p_{\boldsymbol{x}}^{k+1}(\omega_i) = \frac{p_{\boldsymbol{x}}^{k}(\omega_i)Q_{\boldsymbol{x}}^{k}(\omega_i)}{\sum_i p_{\boldsymbol{x}}^{k}(\omega_i)Q_{\boldsymbol{x}}^{k}(\omega_i)}. \tag{8}$$

It remains to define the neighborhood function $Q_{\boldsymbol{x}}$. We begin with the compatibility coefficient defined by the conditional probability $p_{\boldsymbol{xy}}(\omega_i|\omega_j)$, i.e., the probability that ω_i is the correct classification of pixel $\boldsymbol{x}$ given that ω_j is the correct classification for neighboring pixel $\boldsymbol{y}$. We assume that $p_{\boldsymbol{xy}}(\omega_i|\omega_j)$ is independent of location, so that it is only the relative location of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$, and whether or not it is one of its neighbors, that matters. Then, using the previously computed classification, we can compute $p_{\boldsymbol{xy}}(\omega_i|\omega_j)$ by looking at the ratio of occurrences of a pixel $\boldsymbol{x}$ and its neighbors $\boldsymbol{y}$ being classified into ω_i and ω_j, respectively. A location dependent definition for $p_{\boldsymbol{xy}}(\omega_i|\omega_j)$ can also be given, but we do not do that in our implementation.

The probabilities $p_{\boldsymbol{y}}(\omega_j)$, which we have in hand, are then modified via the equation

$$\sum_j p_{\boldsymbol{xy}}(\omega_i|\omega_j)p_{\boldsymbol{y}}(\omega_j). \tag{9}$$

In the approach for computing $p_{\boldsymbol{xy}}(\omega_i|\omega_j)$ mentioned in the previous paragraph, this amounts to simple matrix multiplication. Our neighborhood function is then defined as a spatial average of the values of (9), i.e.,

$$Q_{\boldsymbol{x}}(\omega_i) = \sum_{\boldsymbol{y}} d_{\boldsymbol{y}} \sum_j p_{\boldsymbol{xy}}(\omega_i|\omega_j)p_{\boldsymbol{y}}(\omega_j) \tag{10}$$

where $d_{\boldsymbol{y}}$ are the weights for each neighboring pixel $\boldsymbol{y}$. Generally the weights will all be the same—as is the case for us—unless there is prior knowledge about stronger spatial autocorrelation in one direction than another.

It is recommended that algorithm (8) be iterated until the probabilities have stabilized. There is no predetermined number of iterations for this process to terminate, though in practice it has been observed that beyond the first several iterations the relaxation process can begin to deteriorate [22]. Richards [1] mentions that five to ten iterations will typically produce satisfactory results.

2.2.4. Markov Random Fields

The neighborhood function used in PLR, though intuitive, can only be justified heuristically. Another approach that incorporates spatial context, and that can be justified mathematically, from a Bayesian viewpoint, is Markov random fields (MRF).

Suppose the image of interest has pixel data $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where N is the number of pixels. Let ω_{in} denote the ith spectral class of $\boldsymbol{x}_n$. Then the set of class labels for the entire image can be represented as $\Omega = \{\omega_{i1}, \ldots, \omega_{iN}\}$, where i can be one of $i = 1, \ldots, M$. Our task is to compute a random field Ω that well-approximates the "true" class labeling Ω^* of the image under investigation.

We now take a Bayesian point of view and assume that Ω is a random array with (prior) distribution $p(\Omega)$. In this context, Ω is sometimes referred to as a random field. Through classification, we seek the random field Ω that correctly identifies the ground-truth, and thus hopefully correctly identifies the entire image. For this we maximize the posterior probability $p(\Omega|\boldsymbol{X})$, i.e., from Bayes' theorem we compute

$$\widehat{\Omega} = \arg\max_{\Omega}\{p(\Omega|\boldsymbol{X}) \propto p(\boldsymbol{X}|\Omega)p(\Omega)\}.$$

Since spatial autocorrelation typically exists in images, we incorporate it into the posterior probability. Considering a neighborhood $\mathcal{N}$ around a pixel vector $\boldsymbol{x}_n$, we denote the labels on this neighborhood $\omega_{\delta n}$, $\delta \in \mathcal{N}$ and our objective is now to maximize $p(\omega_{in}|\boldsymbol{x}_n, \omega_{\delta n})$. Using probability theory we can manipulate the conditional probability as follows

$$\begin{aligned} p(\omega_{in}|\boldsymbol{x}_n, \omega_{\delta n}) &= p(\boldsymbol{x}_n, \omega_{\delta n}, \omega_{in}) / p(\boldsymbol{x}_n, \omega_{\delta n}) \\ &= p(\boldsymbol{x}_n|\omega_{\delta n}, \omega_{in}) p(\omega_{\delta n}, \omega_{in}) / p(\boldsymbol{x}_n, \omega_{\delta n}) \\ &= p(\boldsymbol{x}_n|\omega_{\delta n}, \omega_{in}) p(\omega_{it}|\omega_{\delta n}) p(\omega_{\delta n}) / p(\boldsymbol{x}_n, \omega_{\delta n}). \end{aligned} \tag{11}$$

To simplify (11), note that $p(\boldsymbol{x}_n|\omega_{\delta n}, \omega_{in})$ is the probability distribution of $\boldsymbol{x}_n$ conditional on both its labeling as the ith class and its given neighborhood labelings $\omega_{\delta n}$ for $\delta \in \mathcal{N}$. We assume that this conditional distribution is independent of the neighborhood labelings, i.e., $p(\boldsymbol{x}_n|\omega_{\delta n}, \omega_{in}) = p(\boldsymbol{x}_n|\omega_{in})$. Then we also have that the measurement vector $\boldsymbol{x}_n$ is independent of its neighborhood labeling, and hence, $p(\boldsymbol{x}_n, \omega_{\delta n}) = p(\boldsymbol{x}_n) p(\omega_{\delta n})$. Substituting these expressions into Equation (11) then yields

$$\begin{aligned} p(\omega_{in}|\boldsymbol{x}_n, \omega_{\delta n}) &= p(\boldsymbol{x}_n|\omega_{in}) p(\omega_{it}|\omega_{\delta n}) p(\omega_{\delta n}) / p(\boldsymbol{x}_n) p(\omega_{\delta n}) \\ &= p(\boldsymbol{x}_n|\omega_{in}) p(\omega_{it}|\omega_{\delta n}) / p(\boldsymbol{x}_n) \\ &\propto p(\boldsymbol{x}_n|\omega_{in}) p(\omega_{it}|\omega_{\delta n}). \end{aligned}$$

The conditional prior probability $p(\omega_{in}|\omega_{\delta n})$ is the probability for class i of pixel n given the neighborhood labels $\omega_{\delta n} \in \mathcal{N}$. This conditionality forces our random fields of labels to be known as Markov Random Fields (MRF). Knowing that we have a MRF gives us the ability to express $p(\omega_{it}|\omega_{\delta n})$ in the form of a Gibbs distribution

$$p(\omega_{it}|\omega_{\delta n}) = \frac{1}{Z} e^{-U(\omega_{in})},$$

where $U(\omega_{in})$ is based on the Ising model

$$U(\omega_{in}) = \sum_{\delta n} \beta[1 - \delta(\omega_{in}, \omega_{\delta n})].$$

The term $\delta(\omega_{in}, \omega_{\delta n})$ is the Kroneker delta, which takes value 1 if the arguments are equal and 0 otherwise. We will take $Z = 1$ as suggested in [1]. The parameter $\beta > 0$, fixed by the user, controls the neighborhood effect. We can now take $p(\boldsymbol{x}_n|\omega_{in})$ to be given by (4).

Taking the natural log of $p(\omega_{in}|\boldsymbol{x}_n, \omega_{\delta n})$, as in our derivation of QDA, yields the discriminant function for classification

$$g_{in}(\boldsymbol{x}_n) = -\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{m}_i)' \Sigma_i^{-1} (\boldsymbol{x}_n - \boldsymbol{m}_i) - \sum_{\delta n} \beta[1 - \delta(\omega_{in}, \omega_{\delta n})]. \tag{12}$$

Here β is sometimes referred to as a regularization parameter, and it must be specified by the user.

Notice that before the Kroneker delta can be computed in the above MRF discriminant function, an initial classification must be computed. We do this using the QDA classifier. Thus the MRF discriminant function modifies a given set of labels based on a neighborhood effect. This process should be iterated until there are no further changes in labeling [1]; 5–10 iterations is typically sufficient.

2.3. Error Assessment

2.3.1. Thresholding

The QDA method can only classify a pixel into one of the originally given spectral classes. The question arises, what happens if the ground-truth data is incomplete, i.e., there are missing classes? In such cases, it is useful to have an additional spectral class designed

for pixels that, with some degree of certainty, do not fit into one of the original spectral classes noted by the user. The technique of thresholding addresses this problem.

Thresholding compares the probability value of $p(\boldsymbol{x}|\omega_i)$ to a user specified threshold. If the probability value is lower than the threshold, the pixel will be assigned to a *discard* class. Thus, thresholding extracts the pixels with a low probability of belonging to any of the training classes and places them into the *discard class*.

With thresholding, our classification schema now returns $\boldsymbol{x}$ as an element of ω_i when

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \quad \forall\, j \neq i \quad \text{and} \quad g_i(\boldsymbol{x}) > T_i, \tag{13}$$

where T_i is the threshold associated with spectral class ω_i. It is only when the above does not hold for any i that $\boldsymbol{x}$ is then classified into the discard class.

To determine T_i, we examine the discriminant function; for QDA, it is given by (6) and the second inequality in (13) becomes

$$(\boldsymbol{x} - \boldsymbol{m}_i)'\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) < -T_i - \ln|\Sigma_i|. \tag{14}$$

Assuming $\boldsymbol{x}$ is normally distributed with mean $\boldsymbol{m}_i$ and covariance $\boldsymbol{\Sigma}_i$, the left hand side of (14) is known to have a χ^2 distribution with k degrees of freedom (recall, $\boldsymbol{x} \in \mathbb{R}^k$) [23]. The corresponding χ^2_k value can be found using MATLAB's `chi2inv` function for the desired confidence, yielding a corresponding value for T_i.

2.3.2. *K*-Fold Cross Validation

Once a classification has been made, we are interested in how accurately it performed. Since the image in its entirety is unknown, we look to the ground-truth data. First, the ground-truth data set is partitioned into K approximately equal-sized parts. Remove the first part and set it aside to be used as the testing set while the remaining $K-1$ sets become the adjusted training set. The classifier is created with the adjusted training set and is then used to classify the testing set. For example, if $K = 10$, this is equivalent to using 90% of the ground-truth to classify the remaining 10% of the ground-truth. The classified set is then compared to the ground-truth pixels, respectively, and the percent of misclassified pixels is noted. This process is repeated K times (folds) so that each of the K sets is used exactly once as a testing set. The average percent of misclassified pixels is reported.

The recommended K for K-fold cross validation (CV) is $K = 5$ or 10 [3]. If instead, K is taken to be the total number of ground-truth pixels, N, this process is known as the leave-one-out cross validation method. This is approximately unbiased, however it may result in high variance due to the similarities between the N training sets and can be computationally costly. With $K = 5$ or 10, the variance is lower, but bias may play a role in the error assessment. If the classifier is highly affected by the size of the training set, it will overestimate the true prediction error [3].

3. Results: Numerical Experiments

We test our algorithms on a variety of images. Since ground classification in remote sensing is the application that motivated this work, we begin with these types of images.

3.1. Multi-Banded Imagery

The classification methods were tested on two different flood plain images from Quickbird. The first image is of the Middle Fork of the Flathead River in the Nyack flood plain bordering Glacier National Park, Montana (left image in Figure 1). It is an 8-bit RGB-infrared (4 banded) image of 2,208,129 pixels. The 8-bit training data set consisted of 1540 pixels. The location of the training data can be seen in the right-hand image in Figure 1. Training data for the Nyack flood plain consists of the following common cover types that could be easily identified by non-expert users: pasture land, willows, cottonwoods, conifers, water, cobble, regenerative riparian trees, deciduous forest, and road.

To classify this image, we prefilter using neighborhood matrix N_1 and implemented QDA both with and without post-classification smoothing. The straight spectral classification is given on the left in Figure 2; it took approximately 9.5 s in our MATLAB implementation. The 10-fold cross validation error was approximately 3%.

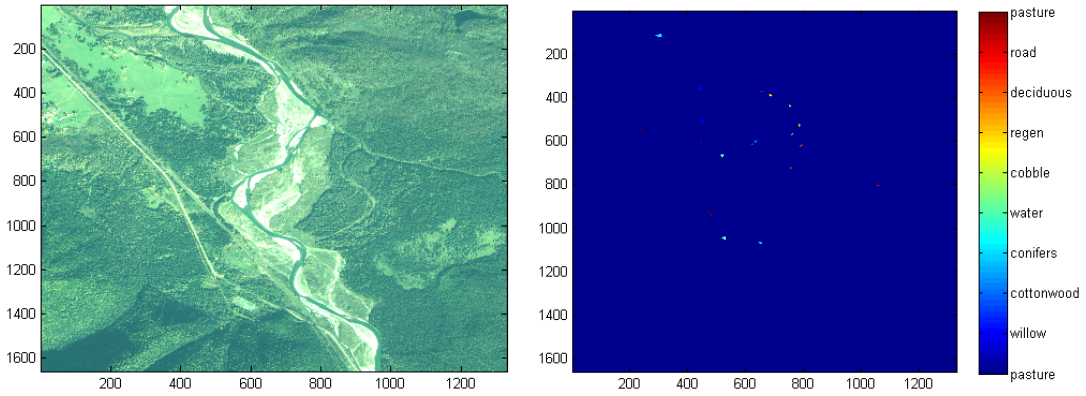

Figure 1. On the left is an image of the Nyack flood plain located on the Middle Fork of the Flathead River, Montana. On the right, the polygons denote the location of the training data set, taken from the image on the left.

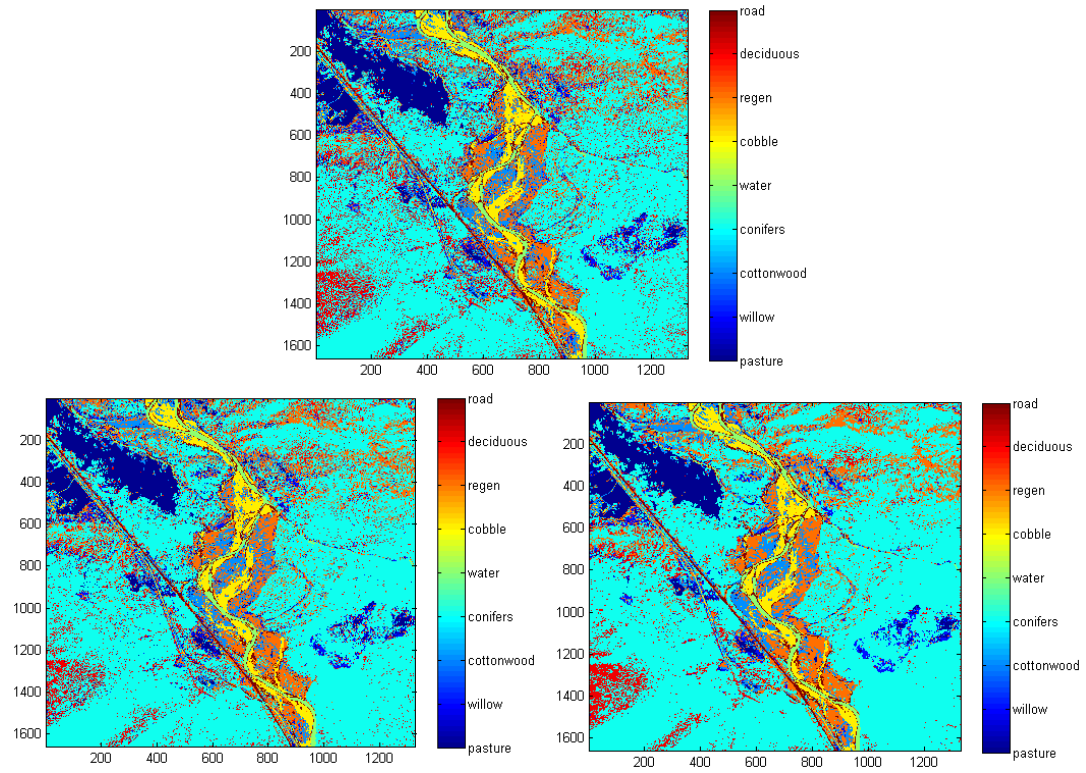

Figure 2. (**Above**) is the QDA classification. The lower figures are the classifications obtained using probability label relaxation (**left**) and Markov random fields (**right**), respectively, applied to the QDA classification.

Implementing 5 iterations of the Markov random field (MRF) took approximately 49 s, including the QDA classification step. The β parameter in (12) was taken to be 10. Eight iterations of probability label relaxation (PLR) took approximately 55 s. Note the increased spatial homogeneity obtained when PLR, and more noticeably when MRF, are used. We emphasize that this may not always be desirable in practice, but if the user knows, prior to classification, that spatial homogeneity exists, this can be enforced using PLR or MRFs in an efficient manner.

We also test the software on an image of the Kwethluk flood plain—an RGB image containing 132,126 pixels—given on the left in Figure 3, and on the right is the ground-

truth data, which was hand chosen using the graphical user interface (GUI) discussed in Section 3.4, which will be introduced in detail below. We note that the chosen ground-truth is not necessarily optimal in the sense of yielding the most accurate classification, but was chosen to test the software.

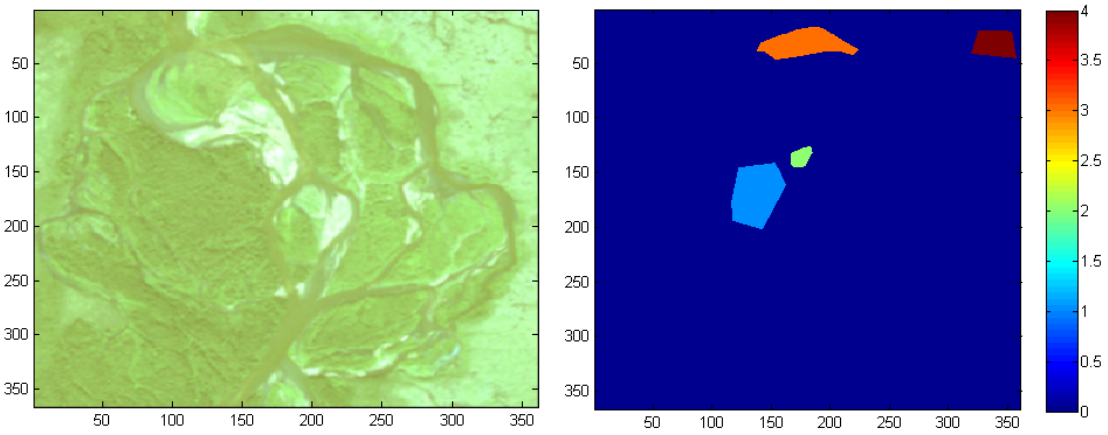

Figure 3. On the left, is an image of a flood plain on the Kwethluk River, Alaska. On the right, the polygons denote the location of the training data set, taken from the image on the left; 1 denotes conifer trees, 2 denotes cobble, 3 denotes water, and 4 denotes deciduous trees.

To classify the image, we implement QDA both with and without post-classification smoothing. The straight spectral classification is given on the left in Figure 4; it took approximately 0.32 s in our MATLAB implementation. The 10-fold cross validation error was approximately 4.6%. Implementing QDA plus 8 iteration of PLR took approximately 1 s. QDA plus 8 MRF iterations with $\beta = 10$ took approximatively 1.8 s.

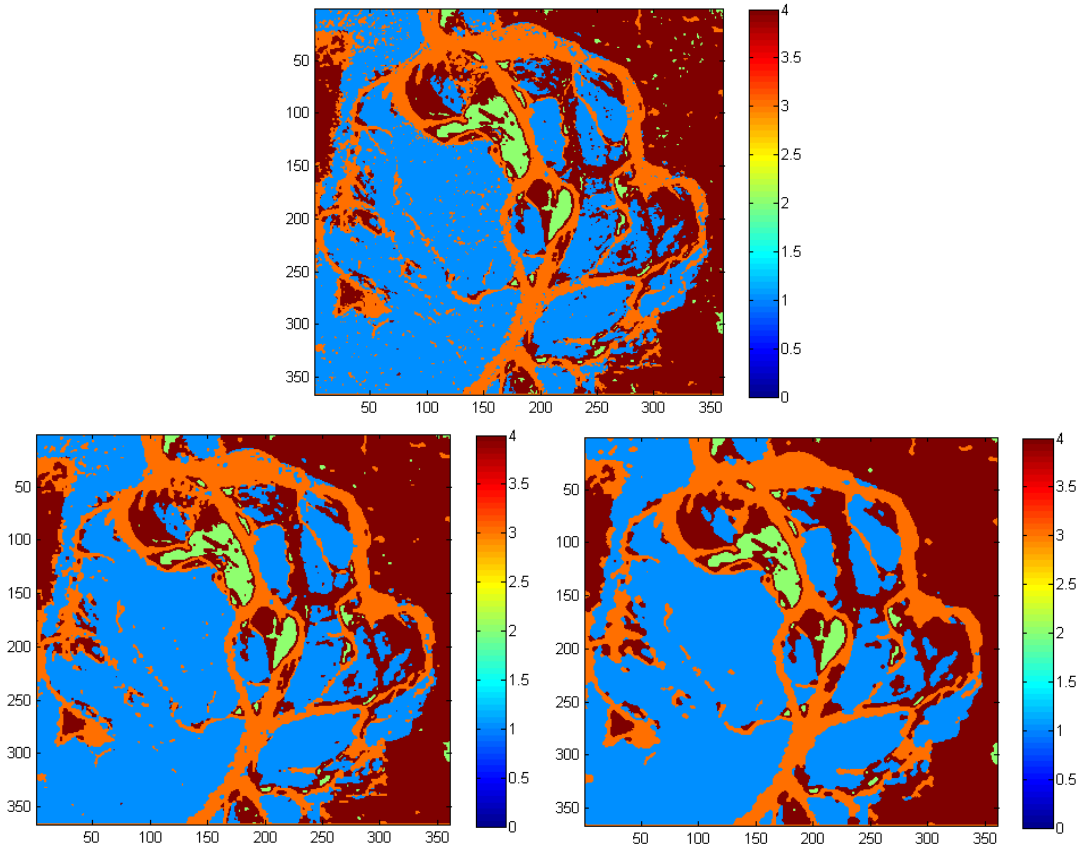

Figure 4. (**Above**) is the QDA classification. The lower figures are the classifications obtained using probability label relaxation (**left**) and Markov random fields (**right**), respectively, applied to the QDA classification.

3.2. Thresholding

To use thresholding on the QDA classifier, we must first calculate T_i. From Equation (14), since the Nyack data set is four-banded, we know the left hand side has a χ^2 distribution with 4 degrees of freedom. For a 99% acceptance rate, $T_i - \ln|\Sigma_i| = 13.2767$ is the value left of which lies 99% of the area under the χ^2 curve with 4 degrees of freedom. Therefore, $T_i = -13.2767 - \ln|\Sigma_i|$, and this is used in Equation (14) for classification.

Because the multi-variate Gaussian assumption can be inaccurate for the spectral information corresponding to certain cover types, thresholding is not effective in all instances. However, for certain types of problems it is useful. For example, suppose that we want to extract the water from the Nyack flood plain image in Figure 1. In this case, it seems unnecessary to ground-truth every cover type appearing in the given image, since we are not interested in any cover type other than water. Thus using only water as ground-truth input, the technique of thresholding at the 99% level was combined with QDA to produce Figure 5. The dark blue on the left image represents the water that has been extracted. Ground-truth for the water extraction (light blue polygons) is displayed in the right image. We created the ground-truth using the GUI discussed in Section 3.4. The 10-fold CV error was 0% indicating all of the supplied ground-truth was classified as water and not discarded.

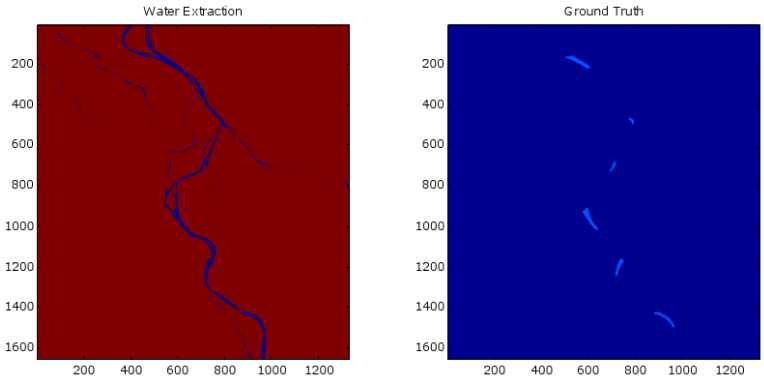

Figure 5. (**Left**): River (blue) has been extracted using 99% threshold under QDA. (**Right**): Light blue polygons denote location of water ground-truth from original image

3.3. Classifying General Images

More generally, the software can be used to segment any RGB image—an important problem in image processing—very efficiently. As an example, consider the color image in Figure 6, the ground-truth image obtained using the ground-truth GUI of Section 3.4, and the resulting QDA classification that cannot be improved upon, visually, using a spatial technique such as MRF or PLR. This classification took approximately 2.5 s.

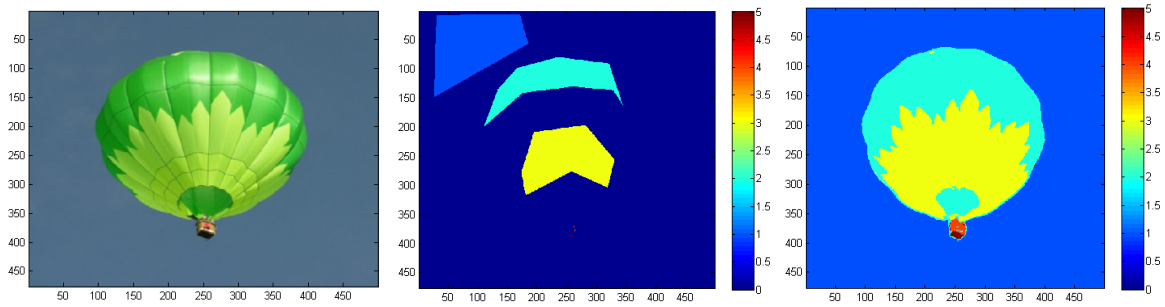

Figure 6. On the left is the original RGB image. In the middle is the ground-truth obtained using the GUI; class 4 and 5, corresponding to the basket, are hardly visible. On the right is the classification obtained using QDA.

Finally, the software can be used on gray-scale and binary images as well for denoising, which is an important problem in image processing. For our gray-scale example, we use

the noisy gray-scale image of Mickey Mouse on the left in Figure 7. For this example, no ground-truth data was used, which is allowed in the codes provided the mean vectors and the covariance matrices (both scalars in this case) are provided. For our classification, we chose the five means 1, 90, 188, 200, and 255, and the variance values were taken to be 1 for every class. The denoising is then accomplished using 10 iteration of the MRF algorithm with $\beta = 7000$, which took approximately 1 s. Implementing the mode filtering, on the other hand, yielded similar results but at approximately 15 s, a significant increase in computational time.

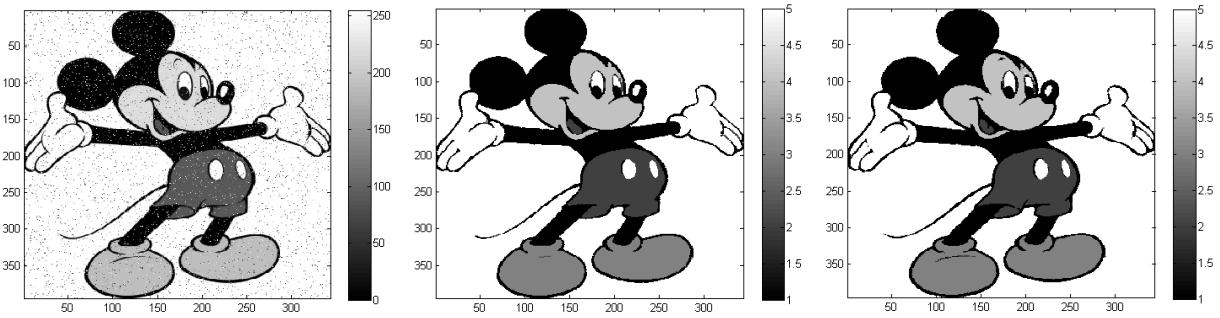

Figure 7. On the (**left**) is the noisy image of Mickey. The denoised images were obtained using MRFs (**middle**) and mode filtering (**right**).

Our binary example is given in Figure 8. In this case, the means were taken to be 1 and 0, and the variances were both taken to be 1. Ten MRF iterations with $\beta = 1$ yielded the denoised image in the middle in Figure 8 and took approximately 2 s. On the right in Figure 8 is the denoised image obtained using mode filtering, which took approximately 55 s to compute.

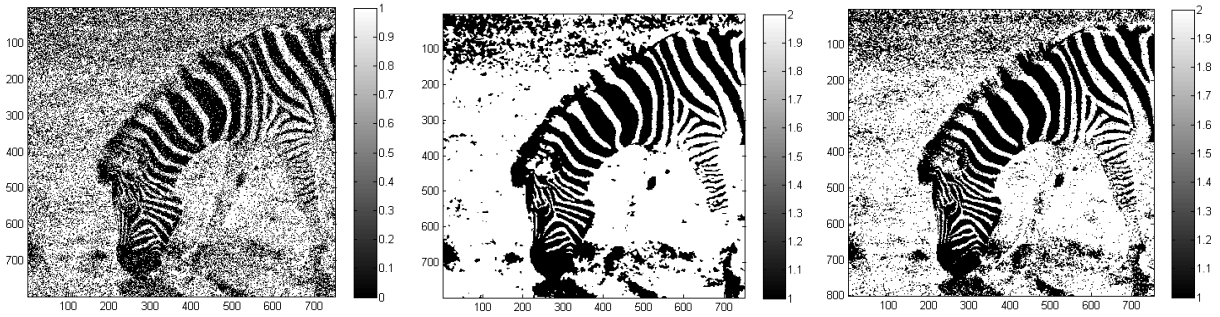

Figure 8. On the (**left**) is the noisy binary image of a Zebra. The denoised images were obtained using MRFs (**middle**) and mode filtering (**right**).

3.4. Accessing and Using the Software and Ground-Truth GUI

The software can be downloaded here: https://github.com/bardsleyj/ClassifyImages (created 5 Novebmer 2021). The accompanying file ReadMe.txt describes in detail how to use the software and the graphical user interface (GUI). The GUI, coupled with the classification codes, makes the classification of images extremely simple and efficient. A snapshot of the GUI can be seen in Figure 9.

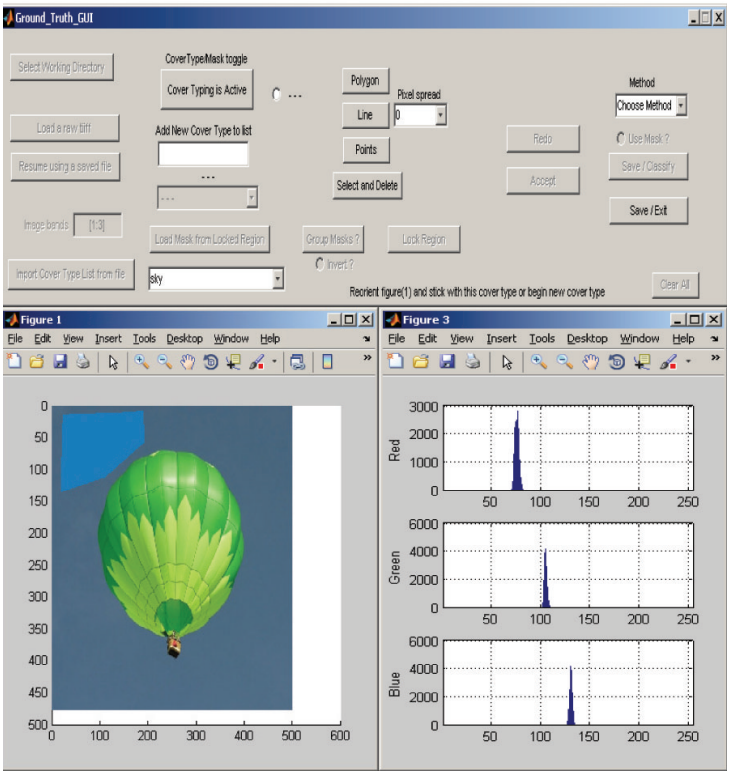

Figure 9. A snapshot of the GUI. On the lower-right are the histograms for each band of the spectral values for the chosen pixels.

4. Discussion

Assessment of habitat abundance and spatial distribution is a fundamental first order analytical requirement for many ecological questions regarding the structure and function of freshwater systems. Cover type mapping of habitat areas began shortly after introducing aerial photographic imagery to map landscape cover over broad areas. For 25 decades it was conducted by manually drawing polygons around standard features, common vegetation types or exposed substrate, and water surfaces of rivers and streams. The approach was helpful in early attempts to determine the available area of critical habitat for organisms and define corridors of connected habitat. Our software makes habitat mapping approachable to a wide range of practitioners working in freshwater systems (e.g., ecologists, fisheries biologists, wetland scientists) by allowing them to interact with the imagery in a long held and intuitive way of manually identifying points, drawing lines or polygons around habitats recognizable to those subject experts.

Subjective human interpretation regarding the visual choices of habitat similarity observed in the imagery limited cover type mapping. The time involved in creating the polygons and accurately determining polygon area due to distortion in images and mosaics also limited the approach. Hence, the onset of more modern multi-spectral imagery, airborne and satellite, coupled with automated rectification procedures enhanced habitat analysis. However, developing automatic and objective cover-type mapping approaches has been evolving since the onset of remote sensing of landscapes. Our software solves these obstacles and hence provides a robust, accurate and easy to use objective method of freshwater habitat assessment using remote sensing imagery as well as historical black and white aerial photographs. The ability to link habitat analysis through cover type mapping of historical aerial photographs to recent satellite imagery improves our understanding of change freshwater systems have undergone over the past decades of climate change.

To test our software, we consider images of river flood plains, which are among the most dynamic and threatened ecosystems on the planet [24]. Flood plains are endangered worldwide because most have been permanently inundated by reservoirs which also reduce

geomorphic forming flows or are disconnected by dikes and armored by bank stabilization structures [25]. River ecosystems are in a state of constant change, both on the seasonal and yearly scales, as well as over the course of decades [26]. Remote sensing of flood plain imagery has become a valuable tool for conserving and protecting ecosystem goods and services that are important to human well being through modeling river dynamics [26,27]. Modeling this change is crucial to river ecologists. In particular, analyzing cover type changes over time allows the river ecologist to address physical and ecological questions at scales relevant to large river ecology and management of those ecosystem components. To perform such an analysis, the collection of remotely sensed images is required, as is the classification of the collected imagery for which our software provides a key tool for practitioners to use that is easy to use and free to download.

For example, we demonstrated that increased spatial homogeneity can be obtained when PLR, and more noticeably when MRF, are used (Figure 2). If the user knows, prior to classification, that spatial homogeneity exists, this can be enforced using PLR or MRFs in an efficient manner. When testing our software using imagery from river floodplains we purposefully choose ground truth data that was not necessarily optimal in the sense of yielding the most accurate classification, but was chosen to test the software as shown with the example from the Kwethulk River in Alaska (Figure 3). Ground truth pixels that represent a cover type (e.g., groups of similar trees) are isolated by enclosing within a polygon or by intersecting with a line or point. If the image requires being zoomed in to view ground-truth pixels for selection, this adjustment needs to be completed before selecting the preferred button for delineating ground-truth data. Therefore, smaller more precise polygons could have been added as well as simply increasing the number of ground truth data using lines or even individual points would increase the accuracy of the classification.

Thresholding is a method to increase classification accuracy. The QDA method can only classify a pixel into one of the originally given spectral classes. The question arises, what happens if the ground-truth data is incomplete resulting in missing classes? The technique of thresholding addresses this problem by allowing or expanding additional spectral classes designed for pixels that, with some degree of certainty, do not fit into one of the original spectral classes noted by the user. Once a classification has been made, we are interested in how accurately it performed. K-fold cross validation is the method we have employed in our software. The classified set is then compared to the ground-truth pixels, respectively, and the percent of misclassified pixels is noted.

Many questions and issues to be solved in freshwater systems require application of remote sensing data. Researchers in academia, government agencies and private sector consulting and engineering firms are pushing hard to solve global water problems related to climate change. Remote sensing of river discharge is a rapidly evolving application [28,29]. Mapping of river channel bathymetry is generally determined by applications of airborne LIDAR data but also through acoustic based sonar with some attention to also mapping the flow field [27,29–33]. Our software fills a need for objective, repeatable and robust cover type mapping of rivers especially when coupled with other methods and approaches to use remote sensing to address the structure and function of freshwater systems.

5. Conclusions

In this paper, we outline a MATLAB software package for the supervised classification of images. The software uses quadratic discriminant analysis (QDA) for spectral classification and contains three methods for spatial smoothing after a spectral classification has been computed. These spatial methods include mode filtering, probability label relaxation, and Markov random fields. Also included is a thresholding technique based on the multi-variate Gaussian assumption used in the motivation of QDA, as well as k-fold cross validation for error analysis.

The software contains a graphical user interface (GUI) that enables the user to generate subsets, known as ground-truth data, for building statistical classifiers. Current versions of

MATLAB do not contain such software, hence the use of the ground-truth GUI is essential in allowing our classification software to be widely and easily useable.

The software is tested on a number of images—two examples from remote sensing and three generic images—in order to illustrate its usefulness on a broad range of image processing applications. The examples also show that the software is useful, easy to use, and efficient on large-scale problems.

Author Contributions: Conceptualization, J.M.B. and M.L.; MATLAB code J.M.B. and M.H.; writing and editing, J.M.B. and M.L.; data curation, M.L.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: Support for M. Howard came from an NSF grant to JMB and MSL, Mathematical methods for habitat classification of remote sensing imagery from river flood plains, NSF-EPSCoR Large River Ecosystems Grant EPS-0701906. M. Howard was a PhD student at the the University of Montana at the time this was work done and was not working for Nevada National Security Site. Partial support for JMB and MSL came from the Gordon and Betty Moore Foundation (UM grant #344.01) The Salmonid Rivers Observatory Network: Relating Habitat and Quality to Salmon Productivity for Pacific Rim Rivers, Stanford PI, Hauer, Kimball, Lorang, Poole Co-PI's, and support for C. Footstalk to write the GUI came from a grant to MSL (UM grant PPL#443216) from Pennslyviana Power and Light.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Richards, J.; Jia, X. *Remote Sensing Digital Analysis, an Introduction*; Springer: Berlin, Germany, 2006.
2. Pal, M.; Mather, P. An assessment of the effectiveness of decision tree methods for land cover classification. *Remote Sens. Environ.* **2003**, *86*, 554–565. [CrossRef]
3. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 214–216.
4. Hand, D.J. Classifier technology and the illusion of progress. *Stat. Sci.* **2006**, *21*, 1–14. [CrossRef]
5. Foody, G.M.; Arora, M.K. An evaluation of some factors affecting the accuracy of classification by an artificial neural network. *Int. J. Remote Sens.* **1997**, *18*, 799–810. [CrossRef]
6. Wilkinson, G.G. Open questions in neuro-computing for earth observation. In *Neuro-Computational in Remote Sensing Data Analysis*; Springer: Berlin, Germany, 1997; pp. 3–13.
7. Tadjudin, S.; Landgrebe, D. *Classification of High Dimensional Data with Limited Training Samples*; Technical Report 98-08; Electrical and Computer Engineering Department, Purdue University: West Lafayette, IN, USA, 1998.
8. Congalton, T.G. Using spatial autocorrelation analysis to explore the errors in maps generated from remotely sensed data. *Photogramm. Eng. Remote Sens.* **1988**, *54*, 587–592.
9. Spiker, J.S.; Warner, T.A. Scale and spatial autocorrelation from a remote sensing perspective. In *Geo-Spatial Technologies in Urban Environments*; Springer: Berlin/Heidelberg, Germany, 2007; Chapter 10, pp. 197–213.
10. Yin, S.; Chen, X.; Yu, Z.; Sun, Y.; Cheng, Y. Scale dependence of autocorrelation from a remote sensing perspective. In Proceedings of the SPIE, Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Advanced Spatial Data Models and Analyses, Guangzhou, China, 28–29 June 2008; pp. 71461T-1–71461T-9.
11. Switzer, P. Extensions of linear discriminant analysis for statistical classification of remotely sensed satellite imagery. *Math. Geol.* **1980**, *12*, 367–376. [CrossRef]
12. Chen, D.M.; Wei, H. The effect of spatial autocorrelation and class proportion on the accuracy measures from different sampling designs. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 140–150. [CrossRef]
13. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2001.
14. Nilsson, N.J. *The Mathematical Foundations of Learning Machines*; McGraw-Hill: New York, NY, USA, 1990.
15. Swain, P.H.; Davis, S.M. (Eds.) *Remote Sensing: The Quantitative Approach*; McGraw-Hill: New York, NY, USA, 1978.
16. Li, J.; Tran, M.; Siwabessy, J. Selecting Optimal Random Forest Predictive Models: A Case Study on Predicting the Spatial Distribution of Seabed Hardness. *PLoS ONE* **2016**, *11*, e0149089. doi:10.1371/journal.pone.0149089. [CrossRef] [PubMed]
17. Diesing, M.; Stephens, D. A multi-model ensemble approach to seabed mapping. *J. Sea Res.* **2015**, *100*, 62–69. [CrossRef]
18. Janowski, L.; Wroblewski, R.; Dworniczak, J.; Kolakowski, M.; Rogowska, K.; Wojcik, M.; Gajewski, J. Offshore benthic habitat mapping based on object-based image analysis and geomorphometric approach. A case study from the Slupsk Bank, Southern Baltic Sea. *Sci. Total Environ.* **2021**, *801*, 149712. doi:10.1016/j.scitotenv.2021.149712. [CrossRef] [PubMed]

19. Diesing, M.; Mitchell, P.J.; O'Keeffe, E.; Gavazzi, G.O.A.M.; Bas, T.L. Limitations of Predicting Substrate Classes on a Sedimentary Complex but Morphologically Simple Seabed. *Remote Sens.* **2020**, *12*, 3398. doi:10.3390/rs12203398. [CrossRef]
20. Innangi, S.; Tonielli, R.; Romagnoli, C.; Budillon, F.; Di Martino, G.; Innangi, M.; Laterza, R.; Le Bas, T.; Iacono, C.L. Seabed mapping in the Pelagie Islands marine protected area (Sicily Channel, southern Mediterranean) using Remote Sensing Object Based Image Analysis (RSOBIA). *Mar. Geophys. Res.* **2019**, *40*, 333–355. doi:10.1007/s11001-018-9371-6. [CrossRef]
21. Townsend, F. The enhancement of computer classifications by logical smoothing. *Photogramm. Eng. Remote Sens.* **1986**, *52*, 213–221.
22. Richards, J.; Landgrebe, D.; Swain, P. On the accuracy of pixel relaxation labelling. *IEEE Trans. Syst. Man Cybern.* **1981**, *SMC-6*, 420–433.
23. Anderson, T.W. *An Introduction to Multivariate Statistical Analysis*; Wiley: New York, NY, USA, 1984.
24. Tockner, K.; Stanford, J.A. Riverine floodplains: Present state and future trends. *Environ. Conserv.* **2002**, *29*, 308–330. [CrossRef]
25. Tockner, K.; Lorang, M.S.; Stanford, J.A. River flood plains are model ecosystems to test general hydrogeomorphic and ecological concepts. *River Res. Appl.* **2010**, *26*, 76–86. [CrossRef]
26. Stanford, J.A.; Lorang, M.S.; Hauer, F.R. The shifting habitat mosaic of river ecosystems. *Verh. Internat. Verein. Limnol.* **2005**, *29*, 123–136. [CrossRef]
27. Lorang, M.S.; Whited, D.C.; Hauer, F.R.; Kimball, J.S.; Stanford, J.A. Using airborne multispectral imagery to evaluate geomorphic work across floodplains of gravel-bed rivers. *Ecol. Appl.* **2005**, *15*, 1209–1222. [CrossRef]
28. Durand, M.; Gleason, C.; Garambois, P.A.; Bjerklie, D.; Smith, L.; Roux, H.; Rodreguez, E.; Bates, P.D.; Pavelsky, T.V.; Monnier, J. An intercomparison of remote sensing river discharge estimation algorithms from measurements of river height, width, and slope. *Water Resour. Res.* **2016**, *52*, 4527–4549. [CrossRef]
29. Piegay, H.; Arnaud, F.; Belletti, B.; Bertrand, M.; Bizzi, S.; Carbonneau, P.; Dufour, S.; Liebault, F.; Ruiz-Villanueva, V.; Slater, L. Remotely sensed rivers in the Anthropocene: State of the art and prospects. *Earth Surf. Process. Landf.* **2020**, *45*, 157–188. [CrossRef]
30. Lorang, M.S.; Hauer, F.R.; Whited, D.C.; Matson, P.L. Using airborne remote-sensing imagery to assess flow releases from a dam in order to maximize re-naturalization of a regulated gravel-bed river. In *The Challenges of Dam Removal and River Restoration: Geological Society of America Reviews in Engineering Geology*; De Graff, J.V., Evans, J.E., Eds.; Geological Society of America: Boulder, CO, USA, 2013; Volume XXI, pp. 117–132. doi:10.1130/2013.4021(10). [CrossRef]
31. McKean, J.; Nagel, D.; Tonina, D.; Bailey, P.; Wright, C.W.; Bohn, C.; Nayegandhi, A. Remote sensing of channels and riparian zones with a narrow-beam aquatic-terrestrial LIDAR. *Remote Sens.* **2009**, *1*, 1065–1096. [CrossRef]
32. Marotz, B.; Lorang, M.S. Pallid sturgeon larvae: The drift dispersion hypothesis. *J. Appl. Ichthyol.* **2018**, *34*, 373–381. doi:10.1111/jai.13569. [CrossRef]
33. Kasvi, E.; Salmela, J.; Lotsari, E.; Kumpula, T.; Lane, S. Comparison of remote sensing based approaches for mapping bathymetry of shallow, clear water rivers. *Geomorphology* **2019**, *333*, 180–197. [CrossRef]