Journal of Computing and Security

July 2014, Volume 1, Number 3 (pp. 225–238)

http://www.jcomsec.org

Semantic Segmentation of Aerial Images Using Fusion of Color and Texture Features

Mahdie Rezaeian [a,*]
Rasoul Amirfattahi [a]
Saeid Sadri [a]

[a] *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: 20 December 2013
Revised: 19 April 2014
Accepted: 14 May 2014
Published Online: 21 February 2015

Keywords:
Aerial Images, Semantic Segmentation, Classification, Local Binary Patterns, Feature Fusion, Artificial Neural Network, Support Vector Machine, Random Forest

ABSTRACT

This paper presents a semantic method for aerial image segmentation. Multi-class aerial images are often featured with large intra-class variations and inter-class similarities. Furthermore, shadows, reflections and changes in viewpoint, high and varying altitude and variability of natural scene pose serious problems for simultaneous segmentation. The main purpose of segmentation of aerial images is to make subsequent recognition phase straightforward. Present algorithm combines two challenging tasks of segmentation and classification in a manner that no extra recognition phase is needed. This algorithm is supposed to be part of a system which will be developed to automatically locate the appropriate site for Unmanned Aerial Vehicle (UAV) landing. With this perspective, we focused on segregating natural and man-made areas in aerial images. We compared different classifiers and explored the best set of features for this task in an experimental manner. In addition, a certainty based method has been used for integrating color and texture descriptors in a more efficient way. The experimental results over a dataset comprised of 25 high-resolution images show the overall binary segmentation accuracy rate of 91.34%.

1 Introduction

Segmentation is the process of partitioning an image into non-overlapping meaningful regions such that each region is uniform based on a specific homogeneity measure and no union of any adjacent segments fulfills the homogeneity criterion [1] . Most of the tasks based on processing of aerial images demand a full description of the scene. The main purpose of segmentation of aerial images is to make subsequent recognition phase straightforward. High and varying altitude, effects of shadow and reflection, varying natural scene, and changes in viewpoint pose a great challenge to the segmentation of aerial images. Furthermore, most recognition methods benefit from structure information. However, even in high resolution aerial images, there is no enough detail and structure information. Therefore, common recognition methods are not directly applicable to aerial images. These considerations add to the complexity of the segmentation problem which is already yet to be fully resolved in the computer vision and image processing community. As a result, any attempt that can reduce the online computation demand while making the system robust against the above challenges is of significant importance. For this

[*] Mahdie Rezaeian.

Email addresses: m.rezaeian@ec.iut.ac.ir (M. Rezaeian), fattahi@cc.iut.ac.ir (R. Amirfattahi), sadri@cc.iut.ac.ir (S. Sadri)

ISSN: 2322-4460 © 2014 JComSec. All rights reserved.

purpose, a semantic segmentation approach is proposed that incorporates invariant and robust descriptors to encode distinctive image information.

Semantic or class specific segmentation is the task of labeling image pixels (or areas) to a set of semantic classes such that resultant segments represent high-level information [2] like land cover types in this work. It combines two challenging tasks of segmentation and classification in a manner that no extra recognition phase is needed. This way, not only different areas of the test image are segmented but also similar segments take the same label. Classification can be performed either unsupervised or supervised. In unsupervised or clustering techniques, samples in the feature space are not labeled. To distinguish different classes, samples are divided based on some similarity/ dissimilarity measure. Due to the variable number of classes and scene variability in aerial images, unsupervised techniques may result in over- or under-segmentation. On the other hand, supervised approaches need a training phase prior to segmentation. During this phase, a set of input features is computed for each sample image of favorite classes. These features form a database of predefined classes. A label is assigned to each sample in the database which shows the class that the sample belongs to. Finally, representative features of an unknown sample are compared with a database of a certain number of classes and the most likely label is computed. The following section presents a brief review of recent researches in aerial image segmentation.

2 Related work

Ojala and Pietikäinen [3] used pixel-classification as the final step of their segmentation algorithm to improve the localization of the boundaries. Hu et al. [4] applied the proposed method by Ojala, and Pietikäinen on seven aerial images and reported overall misclassification rate of 15.7%. They used an adaptive weighted combination of texture, intensity, and color features. Each image was segmented into four classes including water, residential area, wood and crop.

Permuter et al. [5] segmented natural and man-made areas in aerial images on the basis of Gaussian mixture models (GMMs). They tested the performance of their method over a database including seven gray-scale aerial images with average classification accuracy of 85.2%.

Yang and Newsam [6] evaluated Gabor texture features and Scale-Invariant Feature Transform (SIFT) descriptors for extracting 11 land cover classes. They applied these spatial features to maximum a posteriori (MAP) and support vector machine (SVM) classifiers. Their results are shown in Table 1.

Xu et al. [7] presented a Bag-of-Visual Words (BOV) representation for a four-class land-use segmentation problem. By means of a combination of spectral and texture features with SVM classifier they achieved the overall classification accuracy of 93.12% over 882 “single-class” images.

Kluckner et al. [8] proposed a novel feature representation based on covariance matrices and Sigma Points. For accurate semantic classification into five classes, they applied multiple appearance cues including color, edge responses, and height information to multi-class random forest (RF) classifier. The experimental results of their method are given in Table 1.

Nitze et al. [9] compared the performance of machine learning techniques including Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest (RF) in the task of agricultural crop classification in remote sensing images. The best result they achieved was 88.1% overall accuracy for SVM. Their results are given in Table 1.

Yuan et al. [10] presented a systematic benchmarking of aerial image segmentation. They compared six major segmentation algorithms including JSEG [11] , mean shift, the multi-resolution region merging algorithm (MSEG), statistical region merging (SRM), the graph-based region merging algorithm (Felz-Hutt), and oriented watershed transform ultra-metric contour maps with globalPb as contour detector (gPb-owt-ucm). They used one boundary and two region based metrics to quantitatively evaluate and compare segmentation task. Since their evaluation method is completely different from other researches presented in Table 1. , we avoid bringing their research in the table and refer interested readers to their paper for more details.

Table 1 provides a brief literature review on class based segmentation of high resolution aerial images.

This paper provides: 1) semantic segmentation of aerial images using color and texture descriptors, 2) testing four representative learning algorithms, 3) comparison between LBP- based texture descriptors for segmentation of aerial images, and 4) the fusion of color and texture features to improve segmentation results.

The rest of the paper is organized as follows. In Section 3 the proposed segmentation method is described. In Section 4, the segmentation results obtained by four representative classifiers and color histograms are compared. The texture descriptors tested in this work are explained and compared in Section 5. Section 6 shows experimental results of the proposed segmentation algorithm using color and texture fusion and final conclusion is presented in Section 7.

Table 1. Brief literature review on class based segmentation of aerial images.

<table>
<tr><th>Author</th><th>Year</th><th>Method</th><th>Overall classification Accuracy</th></tr>
<tr><td>Hu et al. [4]</td><td>2005</td><td>LBP/C + saturation/hue distribution + grey-level histograms</td><td>84.29%</td></tr>
<tr><td>Permuter et al. [5]</td><td>2006</td><td>energies of wavelet subbands and DCT regions + mean and covariance of the RGB and LAB values</td><td>85.2%</td></tr>
<tr><td>Yang and Newsam [6]</td><td>2008</td><td>SIFT &amp; Gabor + MAP & SVM as classifier</td><td><table><tr><th></th><th>SIFT</th><th>Gabor</th></tr><tr><td><b>MAP</b></td><td>84.5%</td><td>73.9%</td></tr><tr><td><b>SVM</b></td><td>76.2%</td><td>89.8%</td></tr></table></td></tr>
<tr><td>Xu et al. [7]</td><td>2010</td><td>the mean and standard deviations of three spectral bands (i.e., RGB) + 48 texture features computed from 12 GLCMs + SVM</td><td>93.12%</td></tr>
<tr><td>Kluckner et al. [8]</td><td>2010</td><td>covariance matrices and Sigma Points + randomized forest</td><td><table><tr><td>Building</td><td>Water</td><td>Grass</td><td>Tree</td><td>Street</td></tr><tr><td>92.7%</td><td>85.8%</td><td>94.4%</td><td>92.6%</td><td>95.3%</td></tr></table></td></tr>
<tr><td>Nitze et al. [9]</td><td>2012</td><td>five different vegetation indices including ground cover, NDVI, MTVI2, NDVIRE, MTCI</td><td><table><tr><th>ANN</th><th>SVM</th><th>RF</th></tr><tr><td>87.1%</td><td>88.1%</td><td>87.4%</td></tr></table></td></tr>
</table>

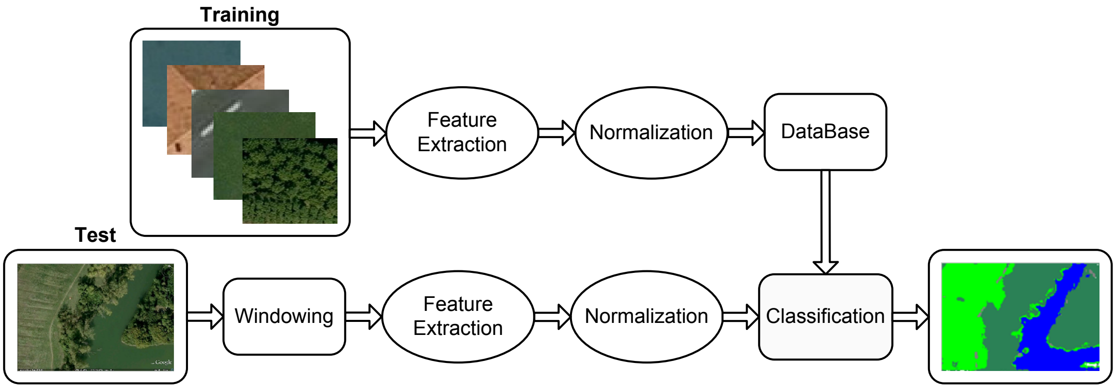

Figure 1. Proposed image segmentation scheme.

3 Proposed method

Before detailing our approach, it is necessary to point out that most of previous works in aerial image segmentation area, such as those reviewed in this work, have used high quality and high resolution satellite images. Here, to investigate the performance of the proposed method, we assumed that proper images are available and applied our proposed method to color images obtained from Google Earth. However, in real application a preprocessing and image restoration or enhancement step may be necessary. Figure 1 shows the block diagram of the proposed segmentation algorithm.

Different steps of the proposed algorithm are described in the following sections. The image data used in the experiments was obtained from the city of Venice, Italy using Google Earth. We collected 80 images from different scenes. Training samples were cropped from 55 images and the rest 25 images were set aside for the segmentation task.

Table 2. Computation formulas for some typical distance metrics.

Distance metric	Formula
Minkowsi (L_k-norm) [12]	$D_k(A,B) = (\sum_{i=1}^{S} \lvert A_i - B_i \rvert^k)^{\frac{1}{k}}$
L1-norm (Manhattan)	$D_1(A,B) = \sum_{i=1}^{S} \lvert A_i - B_i \rvert$
L2-norm (Euclidean)	$D_2(A,B) = \sqrt{\sum_{i=1}^{S} (A_i - B_i)^2}$
Non-InterSection [13]	$D(A,B) = \sum_{i=1}^{S} (A_i - \sum_{i=1}^{S} min_i(A_i, B_i)$
Chi-square	$D(A,B) = \sum_{i=1}^{S} \frac{(A_i - B_i)^2}{A_i + B_i}$

3.1 Training

In this paper, the problem is restricted to detecting 5 classes include *"tree"*, *"grass"* and *"water body"* as natural and *"building"* and *"road"* as man-made sites. To train the classes, homogeneous rectangular areas of different sizes on images of dataset were selected. Then each area is labeled to one of the predefined classes. We provided 160 images of each class and therefore in total we had 800 training images. After choosing training images for each class, a set of features is computed for them. To make comparison of different samples possible, it may be necessary to compute a suitable transformation like normalization for the feature vectors.

3.2 Test

In order to extract features from the test image, a simple strategy is to do measurements in a window. This operation has two stages:

- *Feature extraction:* in this stage the window is placed and moved over the image pixels and a set of descriptors is calculated for each window.
- *Classification:* the calculated descriptors of the window are compared with those in the dataset and the most likely label is assigned to the window.

The window size depends on image resolution and fineness of its texture. In general, window size should be large enough to reflect the local characteristics of the test image in feature vectors. On the other hand, large window causes increase in computations and loss of image details and boundary localization accuracy. For more accurate segmentation, sweeping windows have overlap. Adjacent pixels in natural images generally have similar characteristics and belong to the same class. So, larger overlapping means more redundant computation. Besides, partly cover between sweeping windows will results in a more uniform and accurate segmentation. Briefly, windows overlap specifies computation time and classification accuracy. We heuristically subdivide the input image to overlapping windows of size 45×45 with sweeping step of 10 pixels. This means areas of 10×10 are assumed to have same label.

4 Comparison between different classifiers

There are different types of machine learning algorithms applicable in segmentation task. After introducing a brief description of some prominent cases of these, the segmentation results using these algorithms have been provided for comparison.

4.1 k-Nearest Neighbor (k-NN)

k-Nearest neighbor (k-NN) works based on the intuitive principle that in a feature space, the samples with similar characteristics generally exist close together. Because of possible outliers, a judgment solely based on the nearest neighbor may result in error. To achieve robustness against outliers, k-NN classifier finds the k closest feature samples in the training set and returns the most frequently class label within the k-subset [12]. The two parameters of a k-NN classifier are k and a distance metric. k is set to 10 with respect to the number of the trained samples per class. The choice of the distance metric varies according to the features. The formulas for computation of the distance metrics used in this work are given in Table 2. The parameter S is the length of the two vectors which their distance is to be computed.

4.2 Artificial Neural Network (ANN)

Neural networks consist of a number of simple processors or perceptrons [14]. An ANN is typically characterized by its architecture and its learning process. Here, we used a feed forward Multi-Layer-Perceptrons (MLP) consisting of one input layer, one hidden layer and one output layer. The number of hidden layers depending on the complexity of the problem and input features is selected via trial and error. The number of neurons comprising the input layer is equal to the sum of the lengths of the input features. According to our experience it is better to choose the number of hidden neurons to be the square root of the input neurons. The number of output neurons depends on the number of classes. For binary classification, a single output neuron is sufficient. The goal of the training procedure is to find a set of weights that allow the network to perform correctly on the training examples. In the MLP network used in our study, the relationship between the input neurons (i_m) and the output neuron (o) is determined by:

$$o = f[\sum_n w_n g(\sum_m w_{nm} i_m + \theta_{in}) + \theta_{hid}] \qquad (1)$$

The activation function for both the hidden and output layers is sigmoid function: $f(x) = g(x) = \frac{1}{1+e^{-x}}$. The activation function defines the output of neurons in terms of their weighted inputs. In Equation (1) w_n is the weight from the nth hidden neuron to the output neuron, w_{nm} is the weight from mth input neuron to the nth hidden neuron, θ_{in} and θ_{hid} are the input and hidden biases, respectively. Network weights are iteratively adjusted and computed based on backpropagation (BP) with momentum terms, as follows:

$$\Delta_{w_{jk}}(t+1) = \alpha \delta_k Z_j + \mu \Delta_{w_{jk}}(t) \qquad (2)$$

Where $Z_j = g(\theta_j + \sum_k w_{jk} x_k$, x_k is the activity level of the kth neuron in the previous layer and w_{jk} is the weight of the connection between the jth and kth neurons. δ_k is the error between the desired and actual ANN output value. α is the learning rate, μ is the momentum and t is the number of iterations. The momentum term determines the effect of past weight changes on the current weight update. BP is a gradient descent method and can get stuck in local minima. The momentum term prevents the network from trapping into local minima and speeds up the convergence of the network. In this study, the learning rate and momentum were optimized through trial and error and the weights were randomly initialized within [-1,1].

4.3 Support Vector Machine (SVM)

In this method, data is transformed to a P-dimensional vector using a kernel function so that it becomes separable using a P-1dimensional hyper plane. There are many possible hyper planes but, a rational choice is to choose the hyper plane creating maximum margin between the separating hyper plane and the samples on either side of it [15]). SVM algorithm originally solves binary classification problems. In this work, generalization to multi-class classification is accomplished by training multiple one-against-the-others classifiers. The kernel function is *homogeneous polynomial* of degree 3 and the soft margin constant which determines the upper bound on the Lagrange multipliers is 1000. Interested readers are referred to [16] for more details on SVM.

4.4 Random Forest (RF)

Random forest is a combination of decision trees such that each tree is formed based on a set of random decision functions selected independently. In the training phase, the distribution of different classes for each tree is computed, then in the test phase the class that is the mode of the class's output by individual trees is selected for assigning label to a new sample [17] . Each node in a tree (except leaf nodes) divides the data into two disjoint subsets. The decision function used in this study is a simple comparison with a normally distributed random threshold which is generated between the minimum and maximum values of the trained features. In the training phase, each random split function is scored using the Shannon entropy [18]. Shannon entropy quantifies the homogeneity of the labeled samples in the child nodes:

$$H = -\sum_c \frac{n_c}{N} log \frac{n_c}{N} \qquad (3)$$

Where N is the number of the samples passing through the current node and n_c is the number of samples among the N inputs belonging to class c. For perfectly homogeneous data containing only a single class, the entropy is 0. Therefore, efficient decision rules can be selected based on the condition of minimum entropy. In our tests, the number of the tress is 100.

4.5 Comparison of Classifiers

In this section, the four abovementioned classifiers are evaluated for semantic segmentation using the histogram of RGB components as the color descriptor. The feature vector is constructed by concatenating three histograms of three color channels in RGB space. Each histogram has 32 bins and the color descriptor is a vector of size 3×32=96 in length. As mentioned before, the parameters of each classifier are tuned to achieve the best result. The results are reported from average over 5 runs (except for k-NN). Parameters were kept fixed for all the experiments.

Table 3 shows the confusion matrix of the pixel-wise semantic labeling. The diagonal elements of this matrix show the correct per-class semantic segmentation rates. Other elements show the "inter- class" misclassification rate. In addition to the confusion matrix, the quantitative segmentation results of binary segmentation are given in Figure 2. In the binary case, the segmentation is considered as the problem of seg-

Table 3. Comparison of segmentation error between four classifiers using RGB histograms.

		Building	Grass	Road	Tree	Water
k-NN	**Building**	76.80%	5.90%	6.90%	9.20%	1.20%
	Grass	1.20%	81.30%	4.90%	12.60%	0
	Road	2.70%	2.80%	81.20%	12.40%	0.90%
	Tree	3.10%	4.30%	3.30%	87.90%	1.40%
	Water	0.60%	9.90%	1.30%	4.60%	83.60%
ANN	**Building**	78.20%	0.90%	7.50%	13.40%	0
	Grass	1.30%	74.20%	7.10%	17.40%	0
	Road	2.00%	2.00%	82.00%	12.90%	1.10%
	Tree	3.40%	5.00%	3.90%	86.20%	1.50%
	Water	0	13.10%	1.80%	5.10%	80.00%
SVM	**Building**	77.90%	0	3.30%	18.80%	0
	Grass	2.00%	63.70%	4.40%	29.90%	0
	Road	4.70%	1.20%	76.30%	16.90%	0.90%
	Tree	3.40%	3.20%	2.50%	89.70%	1.20%
	Water	1.40%	13.10%	0.70%	5.90%	78.90%
RF	**Building**	46.90%	0.20%	10.90%	42.00%	0
	Grass	0.90%	64.20%	8.90%	26.00%	0
	Road	1.30%	0.30%	55.70%	31.70%	11.00%
	Tree	0.60%	3.70%	2.90%	92.60%	0.20%
	Water	0.50%	7.20%	31.30%	8.80%	52.20%

regating *man-made* and *natural* areas. In Figure 2 error type I is the percentage of image pixels belonging to building or road classes and labeled as natural and error type II pertains to those pixels of the three natural classes labeled as man-made.

The segmentation errors are calculated over 25 aerial images. These values are calculated based on pixel-wise comparison between manually generated ground-truth and the computerized results.

As shown in Table 3 and Figure 2, among the four classifiers, k-NN classifier yielded the best result. ANN and SVM exhibited nearly the same mean overall accuracies as k-NN and RF produced the worst results. In the next step, to improve the segmentation accuracy we took advantage of texture information.

5 Texture description

Texture descriptors are values and arrays that encode important distinctive information about a texture area. Texture descriptors can be applied to a rectangular or alternatively free-form image area. In an ideal case, they should be compact and invariant to changes in viewpoint, rotation angle, illumination and scale. Among different available texture descriptors, some variations of Local Binary Patterns (LBP) have the advantage of being invariant to rotation, small scale changes and monotonic changes in intensity. LBP can be very compact, easily computed and compared against. All these characteristics make LBP a perfect choice for our application. Since the segmentation algorithm presented in this work relies on LBP to describe texture, a brief description of LBP and its variations is presented in the following subsections.

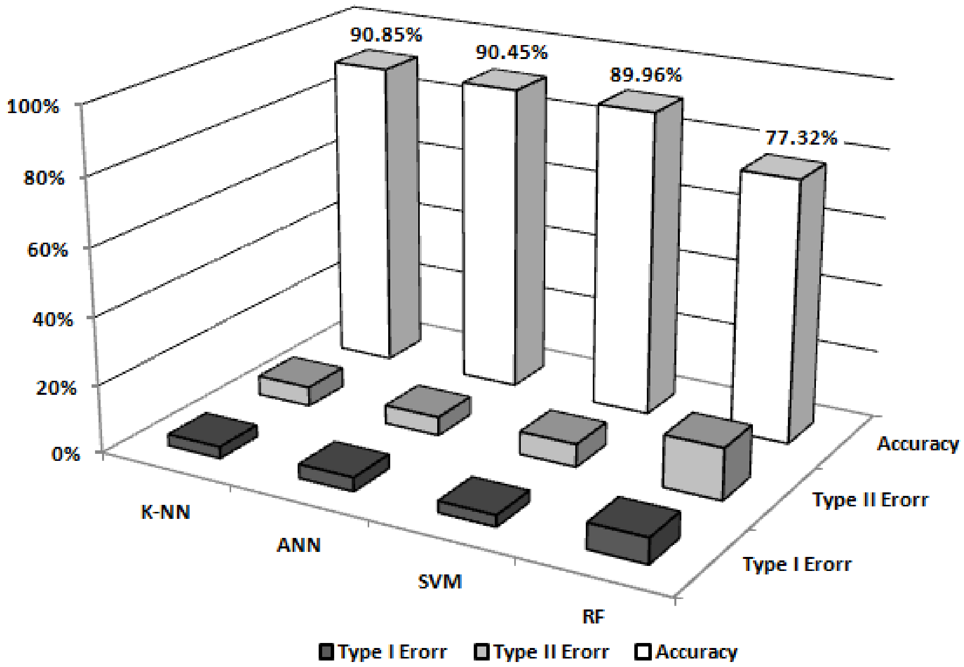

Figure 2. Comparing classifiers using RGB histograms

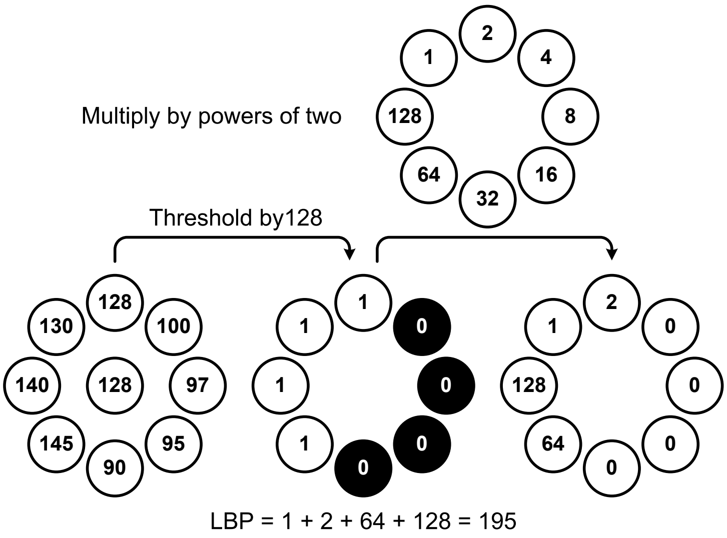

Figure 3. LBP code

5.1 Local Binary Pattern

To explain LBP descriptors we first describe LBP codes. LBP codes are defined for each pixel in the area which is to be described. The LBP code for each pixel is defined over a circular symmetric neighborhood about the pixel for which the LBP code is to be computed. For each pixel on the circular neighborhood of size P and radius R, a LBP bit is assigned. This results in a P bit binary LBP code which is achieved by thresholding the intensity value of each pixel on the circular neighborhood at the value of the central pixel. If the value of the neighboring pixel is less than the central pixel, its LBP bit code is assigned to be zero. Otherwise the LBP bit code is assigned to be unity. This procedure is illustrated in Figure 3. This procedure is performed for every pixel in the area (for example rectangular window). This results in a LBP image. Then the histogram of LBP codes for this window is computed and considered as a texture descriptor for this area. As discussed above, LBP is defined for a circular neighborhood of size P. If P does not divide 360^o, for those samples which are not at the center of pixels the gray values are estimated by interpolation.

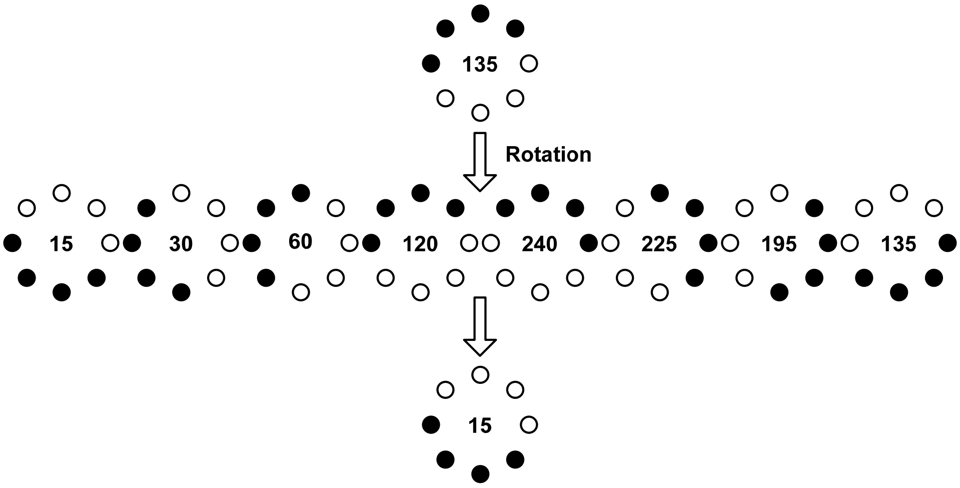

Figure 4. Rotation invariant LBP by rotating neighborhoods to their minimum value. Black and white circles represent zeros and ones respectively.

5.2 Rotation invariance

The LBP patterns obtained in the previous step are not rotation invariant. This is because a rotation in the texture results in a circular shift in the LBP histogram. To make LBP rotation invariant, one idea is to iteratively apply a circular bit-wise right shift on the binary code and choose the minimum code. This is further explained in Figure 4. Since this procedure alters LBP codes in the pixel level, it makes LBP descriptors both locally and globally rotation invariant. We desire a texture descriptor to be globally rotation invariant. However, local rotation invariance is undesired as it disregards local orientation of texture. This issue will later be further discussed in Section 5.6.

5.3 Compactness

It is a reasonable idea to give more importance to those LBP patterns which are seen in the real word more frequently. Ojala et al. [19] showed that more than 94% of LBP patterns seen in the real world are “uniform”. By uniform LBP codes, they mean those binary patterns that have only up to two transitions from zero to one or vice versa. Non-uniform LBP codes, however, have more than two transitions from zero-bits to one-bits or vice versa. Figure 5 demonstrates this definition.

All non-uniform LBP patterns are assigned a single LBP code. Therefore, the assigned code to uniform rotation invariant LBP patterns can be simply the number of one-bits in the pattern [20] . We refer to this way of encoding LBP patterns that takes advantage of dividing LBP patterns to uniform and non-uniform ones as “LBP^{riu}”. This makes the LBP code to be more compact and removes extra nearly zero bins in the LBP histogram. Although, it reduces the number of bins in the histogram, it increases the efficiency of LBP descriptor and elevates the need for histogram quantization.

5.4 Scale Invariance

LBP operator can be performed in different scales. This means by varying the parameters P and R, we can have different LBP patterns which encode texture information in different scales. Histograms of such LBP operators are combined to have a multi-resolution LBP descriptor. In fact, if the metrics used to compare LBP histograms have additive property, then the histograms can be safely concatenated to model the joint distribution of “assumingly” independent texture events [21].

5.5 Adding Contrast to LBP

Although contrast is an important quality of texture, LBP operator ignores the values of gray level differences. Adding contrast information can improve segmentation accuracy. Indeed, texture can be considered as a two-dimensional phenomenon which is characterized by two qualities i.e. spatial structure or pattern and contrast [3] . Pattern is independent from grayscale but contrast is not; in addition, contrast does not change with rotation however pattern does. In view of that, these measures complement each other usefully.

Local contrast could be measured in a circular symmetric neighborhood like LBP [19]:

$$\begin{cases} VAR_{P,R} = \frac{1}{P}\sum_{P=0}^{P-1}(g_P-\mu)^2 \\ \mu = \frac{1}{P}\sum_{P=0}^{P-1} g_P \end{cases} \tag{4}$$

Where g_P refers to gray level of neighboring pixels. Let (P_1, R_1) and (P_2, R_2) represent neighborhood size

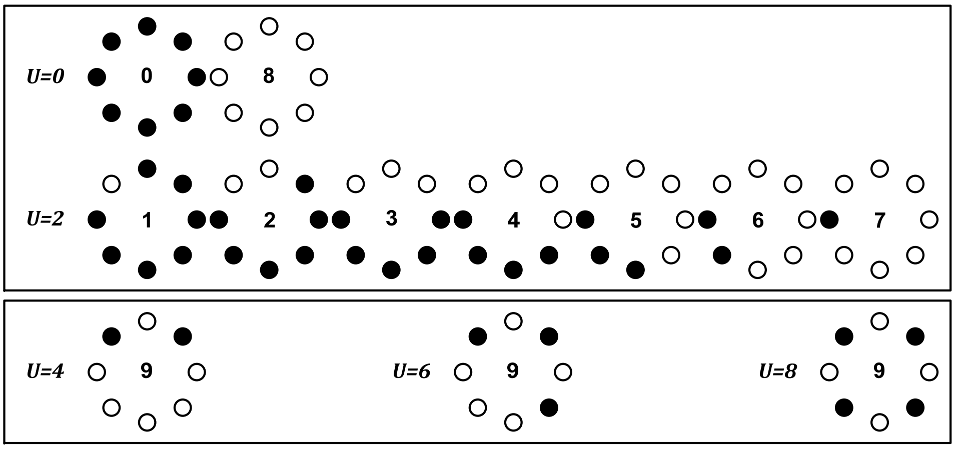

Figure 5. Examples of uniform and non-uniform patterns. The corresponding code for each uniform rotation invariant pattern is equal to the number of ones in the pattern. All other patterns assigned to a single code.

and radius for LBP and VAR respectively. Although there is no any restriction for calculating LBP and VAR in different neighborhoods and choosing different (P,R), usually $P_1 = P_2$ and $R_1 = R_2$ is chosen.

Joint distribution of LBP and VAR is a powerful tool for texture analysis. But according to the Equation (4) $VAR_{P,R}$ is a continuous quantity and should be quantized. In order to achieve optimum resolution, a comprehensive and numerous set of training images is needed to estimate the range of $VAR_{P,R}$ variations. Then for dividing this range into N equal parts, threshold values should be calculated. However, quantization with this method has three limitations. First, a training stage is necessary to determine threshold values. Second, because different classes may have very different contrast, quantization depends on training samples. Finally, third limitation is that finding optimal N based on feature vector size and the discrimination of the classes is difficult. If the number of quantization levels (N) is small, $VAR_{P,R}$ will be useless and classes will not be separated correctly. Conversely, choosing large N leads to histogram instability and increasing size of the feature vector.

To overcome these limitations Guo et al. [22] proposed a simple yet effective method for combining contrast to local LBP histograms. They use $VAR_{P,R}$ as an adaptive weight for calculating the histogram of LBP. This feature which is named LBPV is calculated as follows [22]:

$$LBPV_{P,R}(k)|_{k\epsilon[0,K]} = \sum_{x}\sum_{y} w(LBP_{P,R}(x,y),k) \tag{5}$$

$$w(LBP_{P,R}(x,y),k) = \begin{cases} VAR_{P,R}(x,y), LBP_{P,R}(\\ \qquad x,y) = k \\ 0 \quad \text{otherwise} \end{cases}$$

Consequently, LBPV is a simple representation of LBP / VAR two-dimensional distribution which is smaller and does not require quantization of VAR values.

5.6 Local Binary Pattern Histogram Fourier (LBP-HF)

Ahonen et al. [23] introduced a rotation invariant texture descriptor named Local Binary Pattern Histogram Fourier (LBP-HF) which is derived from the magnitude of discrete Fourier transform of uniform LBP histograms. Unlike the earlier local rotation invariant features discussed in Section 5.2 , the rotation invariance of LBP-HF descriptor is attained globally. This means that the descriptor is invariant against rotations of the whole image but at the same time, if only some parts of the image are rotated the descriptor will differ. Details on how to calculate the descriptor are given in [23].

5.7 Evaluating texture descriptors and similarity measures

In this section we evaluate different variations of LBP and determine the most suitable distance measure for each descriptor. The following texture descriptors were applied to the training images:

- $LBP^{riu}_{(8,1)}$ (applied on gray scale image)
- $LBP^{riu}_{(8,1)+(16,2)}$ (Multi-scale LBP applied on gray scale image)
- $LBP^{riu}_{(8,1)+(16,2)}$ (applied on RGB channels)
- $LBP^{riu}_{(8,1)+(16,2)}$ (applied on HS channels)

- LBPV (applied on gray scale image)
- LBP-HF (applied on gray scale image)

In the case of LBPV and LBP-HF descriptors, histograms of "uniform LBP" codes were computed in (8,1) and (16,2) neighborhoods to achieve scale invariance. To choose the most suitable descriptor among the above descriptors, we have evaluated them in identifying the right label for a test set of single class homogenous textured images (275 single class test images; that is 55 images per class) using k-NN classifier. For each texture descriptor, a number of different distance metrics are examined and the most effective one is selected. These metrics include L1-Norm, L2-Norm [12] , Histogram Intersection [13] and Chi square distance.

The results of applying each texture descriptor with the abovementioned distances are given in Table 4.

As shown in Table 4, the combination of LBP-HF with L2-Norm distance metric outperforms the other cases. LBP-HF is invariant against global rotations of input image. This characteristic is very important in segmentation of aerial images, however, the length of the feature vector should also be considered.

It is important to note that k-NN classifier can be replaced with any other classifier or machine learning approach such as Random forests, AdaBoost or SVM. Since k-NN has been sufficiently capable of classifying LBP-HF descriptors, we avoided the use of more complex classifiers. As a result, the success of algorithm can be attributed to the algorithm itself and not the complexity of the classifier.

5.8 Color and texture fusion

We tested our semantic segmentation algorithm on aerial images using the combination of color and texture descriptors. In view of the fact that it is unlikely that both color and texture descriptors make the same mistake, they can correct each other. As shown in Table 4 extracting LBP descriptors from color channels is not an efficient way for using color information. Color and texture features fusion can be carried out by simply connecting two feature vectors to each other. The other option is to classify color and texture separately and then evaluate the accuracy of each classifier to make final decision. This way, it is possible to set the parameter for each classifier independently in order to achieve the optimum result.

The color descriptor (a vector of size 96 constructed from 32-bin-histogram of R, G and B channels) and the texture descriptor of our choice, (here LBP-HF) are applied to each area of the test image. The decision of using LBP-HF among all different variations of LBP is based on the evaluation study explained in Section 5.7. The experiments show that the most successful combination is LBP-HF and L2-Norm. The pool of LBP-HF descriptors for the samples of the five classes accompanied with their semantic label is fed to a k-NN classifier. For the color classifier the similarity measure is set to Histogram Intersection. For k-NN classifier the maximum number of agreed votes among k neighboring samples in the feature space is used as a measure for accuracy. The process of how to make final decision is shown in Figure 6. The fusion weights of color and texture are the same.

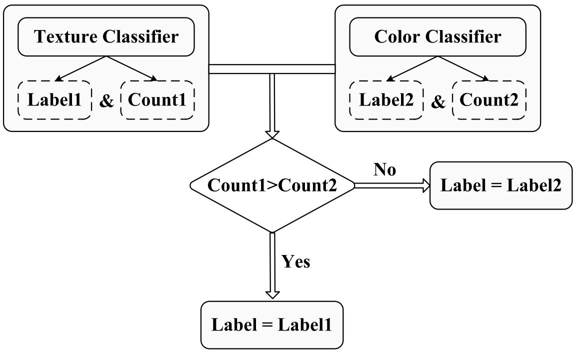

Figure 6. Fusion of color and texture descriptors in classifier level. Label and Count are the final label and agreed vote count of each classifier respectively.

6 Results and discussion

As Figure 7 shows, in spite of our limitations for gathering an efficient texture database, LBP-HF enhanced the segmentation results. In order to display segmentation result graphically buildings are shown in white, road in gray, water in blue, grass in green and trees in dark green.

Figure 8 shows a number of sample results. The results obtained from testing the algorithm over 25 aerial images are summarized in Table 5.

An important aspect of automatic site selection for UAV landing application is that some classes are safer to be misclassified than other classes. For example, it is very dangerous to classify buildings as grass. This is because grass is a safe area for landing while buildings are unsafe areas. As a result, a more meaningful classification rate is computed as shown in the right half of Table 5. This classification is based on a two class labeling: safe (Water, Grass, and Tree) and unsafe (Building, Road) areas.

One major difficulty during experiments was the collection of ground truth to evaluate segmentation results. We used manually segmented images as ground truth. Manual segmentation of high resolution images is tiresome and inevitably involves imprecision. Nev-

Table 4. Comparison between classification error rates for different versions of LBP with 4 distance metrics using k-NN classifier.

Texture Feature	**Feature Vector Length**	**L1-norm**	**L2-norm**	**Chi-square**	**Histogram Intersection**
$LBP^{riu}_{(8,1)}$	10	20.4%	19.6%	18.5%	20.4%
$LBP^{riu}_{(8,1)+(16,2)}$ (Multi scale)	28	14.9%	14.9%	13.8%	14.9%
$LBP^{riu}_{(8,1)+(16,2)}$ (RGB channels)	84	14.2%	13.1%	13.1%	14.2%
$LBP^{riu}_{(8,1)+(16,2)}$ (HS channels)	56	21.1%	21.4%	19.6%	21.1%
LBPV	302	22.5%	31.3%	22.2%	22.5%
LBP-HF	176	8%	**6.9%**	7.6%	33.4%

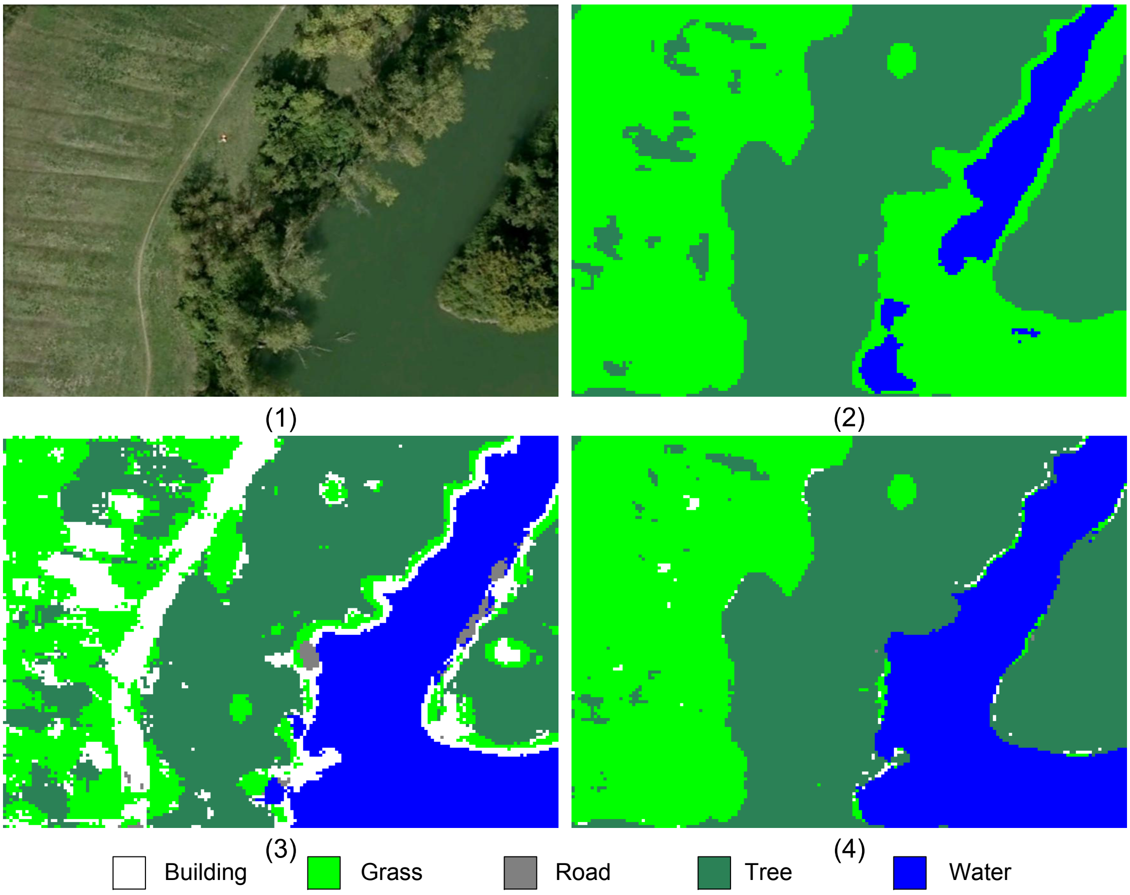

Figure 7. (1) Original image (2) Segmentation result using RGB histograms (3) Segmentation result using LBP-HF (4) Segmentation result using color and texture fusion. The texture classifier compensates the color classifier weakness in classifying water and the color classifier corrects texture classifier failure in classification of grass and tree classes.

ertheless, we tested the algorithm over a database containing 25 high resolution images which is considerably bigger than test sets used in previous works outlined in Table 1.

The comparison between total error rates in Table 3 and 5 demonstrates that adding texture information to semantic segmentation has improved average segmentation accuracy by 2.87%. Intra-class misclassification accounts for 37.7% of the total error. Regarding the fact that there is no any meaningful distinction between safe classes or unsafe classes, this part of error can be discarded. Using the proposed method we could detect unsafe sites with 95.3% and safe sites with 96.1% accuracy (Note that we compute the error through pixel by pixel matching). The specificity

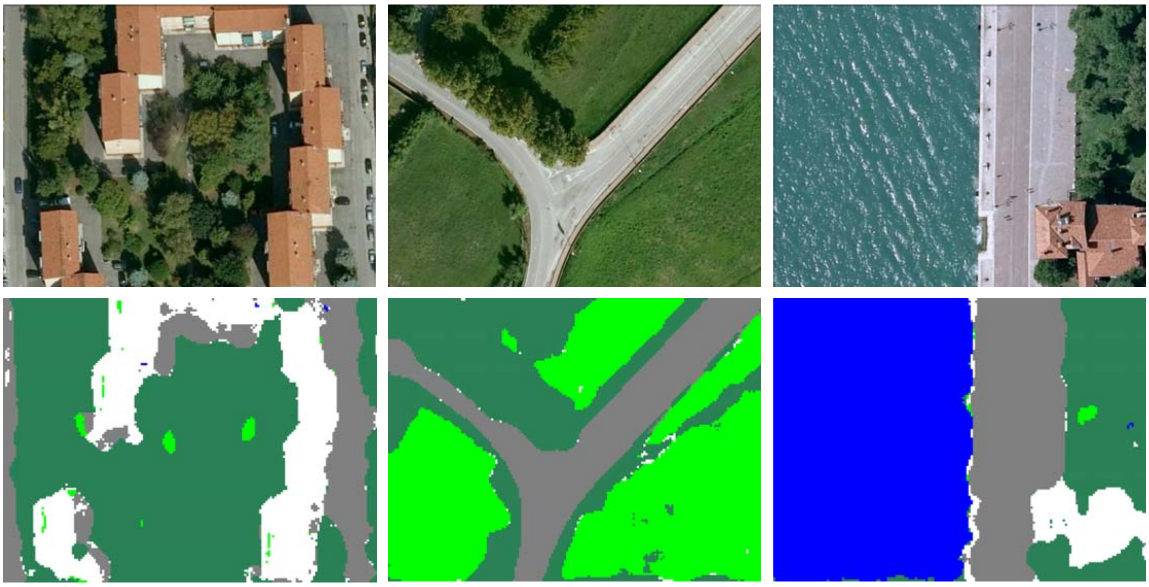

Figure 8. Input image (top row) and segmentation result (bottom row).

Table 5. Segmentation error rate over 25 aerial images using two k-NN classifiers for color and texture.

Confusion Matrix						**Binary Pixel-Level Labeling Error**			
						Inter-class		**Intra-class**	**Total**
						Unsafe to Safe	**Safe To Unsafe**		
	Building	**Grass**	**Road**	**Tree**	**Water**				
Building	83.4%	1.3%	6.7%	8.6%	0	4.74%	3.92%	5.24%	**13.89%**
Grass	0.7%	84.8%	5.9%	8.6%	0				
Road	2.7%	4.1%	82.1%	10.4%	0.7%				
Tree	3.9%	2.6%	3.8%	87.5%	2.2%				
Water	0	0	1.8%	4.9%	93.3%				

and sensitivity indices of the method for detection of unsafe sites are 91.13% and 94.28% respectively. Accordingly, the algorithm extracts unsafe zones with the accuracy of 93.05%. The results of detection of unsafe sites obtained by the proposed method are given on a Pie chart in Figure 9 to be compared graphically.

Our experimental results would not be comparable with previous works mainly because of basic differences existing between images in terms of resolution, lighting and scene complexity. More importantly, the number of images greatly affected classification accuracy. In addition, there are a variety of methods for evaluating results which makes direct comparisons difficult. The only conclusion from browsing similar case studies of automatic classification of aerial images is that our system performed well and confirms that the proposed system can be incorporated into industrial systems for the automated analysis of similar images.

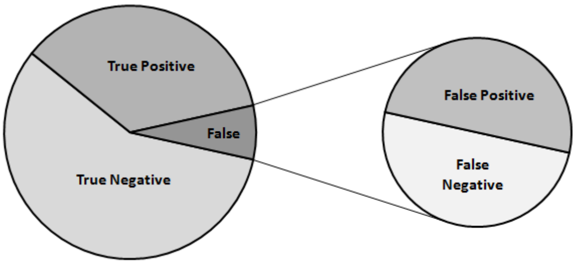

Figure 9. Unsafe regions detection results obtained by the proposed method over a database of 25 multi-class images

7 Conclusion and future directions

This paper presents a semantic segmentation algorithm for aerial image understanding. Regarding the nature of aerial images, features must be invariant to rotation and scale. In this paper two separate k-NN classifiers recognize image texture and color. Local texture characteristics of the gray image and local color histograms are calculated and classified individually. Final segmentation is obtained by evaluating the certainty with each classifier (color or texture) independently. Employing two classifiers is motivated by this observation that it is unlikely that both classifiers make mistake in the same case. Therefore, it is possible to correct errors made by each other. In aerial images, like most natural images neighboring pixels usually have similar characteristics. Based on this quality, calculations can be reduced and better segmentation can be obtained. Accordingly, classification result of each feature vector is assigned to a group of adjacent pixels (a patch) instead of labeling each pixel. The proposed method, which is simple and rapid, is applicable to any type of color-texture images. The other outcome of comparison between obtained results is that proposed method for fusing color and texture information in classifier level is more efficient than applying LBP operator on color channels. The main advantage of the proposed method is that it facilitates using different clues and fusing them. Although, the histogram of RGB channels was a distinctive descriptor for our database, we used a texture descriptor to show how different descriptors can be fused easily. The error rate of the proposed algorithm is small and segmentation results for aerial images are visually acceptable. The majority of the segmentation error pertains to intra-class misclassification such as when grass mixes up with tree. Because we do not consider specific distinction between the two risk classes or three safe classes, this part of segmentation error is not very important. In addition, the comparison of the obtained results with similar works confirms the effectiveness of the proposed method especially that the number of the images which the algorithm is tested on is considerably more. However, basic differences in terms of image databases and the methods for reporting results make direct comparisons difficult.

In our future work, we will develop the proposed algorithm for automatic landing site selection for UAV forced landing. In the test stage we would use a superpixel partitioning algorithm instead of rectangular windows.

References

[1] Thomas Blaschke. Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16, 2010.

[2] Giovanni Maria Farinella, Sebastiano Battiato, and Roberto Cipolla. *Advanced Topics in Computer Vision*. Springer, 2013.

[3] Timo Ojala and Matti Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999.

[4] Xiangyun Hu, C Vincent Tao, and Björn Prenzel. Automatic segmentation of high-resolution satellite imagery by integrating texture, intensity, and color features. *Photogrammetric engineering and remote sensing*, 71(12):1399, 2005.

[5] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[6] Yi Yang and Shawn Newsam. Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1852–1855. IEEE, 2008.

[7] Sheng Xu, Tao Fang, Deren Li, and Shiwei Wang. Object classification of aerial images with bag-of-visual words. *Geoscience and Remote Sensing Letters, IEEE*, 7(2):366–370, 2010.

[8] Stefan Kluckner, Thomas Mauthner, Peter M Roth, and Horst Bischof. Semantic classification in aerial imagery by integrating appearance and height information. In *Computer Vision–ACCV 2009*, pages 477–488. Springer, 2010.

[9] I Nitze, U Schulthess, and H Asche. Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification. *Proc. of the 4th GEOBIA*, pages 7–9, 2012.

[10] Jiangye Yuan, Shaun S Gleason, and Anil M Cheriyadat. Systematic benchmarking of aerial image segmentation. 2013.

[11] Yining Deng and BS Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001.

[12] Mark Nixon and Alberto S Aguado. *Feature extraction & image processing*. Academic Press, 2008.

[13] Sung-Hyuk Cha. *Use of distance measures in handwriting analysis*. PhD thesis, State University of New York, 2001.

[14] Brian D. Ripley. *Pattern recognition and neural*

networks. Cambridge university press, 2008.

[15] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer, 2008.

[16] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] Frank Moosmann, Eric Nowak, and Frederic Jurie. Randomized clustering forests for image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1632–1646, 2008.

[19] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[20] Abdolhossein Fathi and Ahmad Reza Naghsh-Nilchi. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recognition Letters*, 33(9):1093–1100, 2012.

[21] Topi Mäenpää. *The Local binary pattern approach to texture analysis: Extenxions and applications*. Oulun yliopisto, 2003.

[22] Zhenhua Guo, Lei Zhang, and David Zhang. Rotation invariant texture classification using lbp variance (lbpv) with global matching. *Pattern recognition*, 43(3):706–719, 2010.

[23] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Image Analysis*, pages 61–70. Springer, 2009.

Mahdie Rezaeian received her B.Sc. degree in electronics from University of Isfahan and M.Sc. in telecommunications from Isfahan University of Technology, Isfahan, Iran. Her research interest includes signal and image processing, computer vision and image interpretation.

Rasoul Amirfattahi received B.Sc. degree in Electrical Engineering from Isfahan University of technology, Isfahan, Iran in 1993, M.Sc. degree in Biomedical Engineering and PhD degree in Electrical Engineering both from Amirkabir University of technology (Tehran Polytechnic), Tehran, Iran in 1996 and 2002 respectively. He joined Isfahan University of Technology in 2003 and he is currently an associate professor and the director of digital signal processing research laboratory at department of Electrical and Computer Engineering. His research interests are signal and image processing and DSP algorithms. He is the author or coauthor of more than 150 technical papers, one book and two book chapters.

Saeed Sadri received B.Sc. and M.Sc. degree in electrical engineering from University of Tehran, Faculty of Engineering, Tehran, Iran, in 1977 and 1979 respectively. He received his PhD degree in telecommunications from Isfahan University of Technology, in 1997. He is currently an associate professor in Electrical and Computer Engineering Department, Isfahan University of technology, Isfahan, Iran.