

Article

# Optimal Decision Fusion for Urban Land-Use/Land-Cover Classification Based on Adaptive Differential Evolution Using Hyperspectral and LiDAR Data

Yanfei Zhong <sup>1,2,\*</sup> , Qiong Cao <sup>1,2,\*</sup>, Ji Zhao <sup>3</sup>, Ailong Ma <sup>1,2,\*</sup>, Bei Zhao <sup>4</sup> and Liangpei Zhang <sup>1,2</sup>

<sup>1</sup> The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; zlp62@whu.edu.cn

<sup>2</sup> Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China

<sup>3</sup> College of Computer Science, China University of Geosciences, Wuhan 430074, China; zhaoji2015@gmail.com

<sup>4</sup> Department of Geography and Resource Management, The Chinese University of Hong Kong, Hong Kong, China; zhaoy@whu.edu.cn

\* Correspondence: zhongyanfei@whu.edu.cn (Y.Z.); cq0506@foxmail.com (Q.C.); maailong007@whu.edu.cn (A.M.); Tel.: +86-27-6877-9969 (Y.Z.)

Received: 26 June 2017; Accepted: 18 August 2017; Published: 22 August 2017

**Abstract:** Hyperspectral images and light detection and ranging (LiDAR) data have, respectively, the high spectral resolution and accurate elevation information required for urban land-use/land-cover (LULC) classification. To combine the respective advantages of hyperspectral and LiDAR data, this paper proposes an optimal decision fusion method based on adaptive differential evolution, namely ODF-ADE, for urban LULC classification. In the ODF-ADE framework the normalized difference vegetation index (NDVI), gray-level co-occurrence matrix (GLCM) and digital surface model (DSM) are extracted to form the feature map. The three different classifiers of the maximum likelihood classifier (MLC), support vector machine (SVM) and multinomial logistic regression (MLR) are used to classify the extracted features. To find the optimal weights for the different classification maps, weighted voting is used to obtain the classification result and the weights of each classification map are optimized by the differential evolution algorithm which uses a self-adaptive strategy to obtain the parameter adaptively. The final classification map is obtained after post-processing based on conditional random fields (CRF). The experimental results confirm that the proposed algorithm is very effective in urban LULC classification.

**Keywords:** data fusion; hyperspectral; LiDAR; differential evolution; decision fusion; land-use/land-cover classification

## 1. Introduction

Urban land-use/land-cover (LULC) classification plays an important role in various applications, including urban change studies and urban planning [1]. With the continuous development of Earth observation technology, there is now a variety of remote sensing sensors with different functions. These multiple sensors provide us with ample data for urban LULC classification. However, the recent studies of urban LULC classification have mainly used a specific source of remote sensing data [2]. Hyperspectral images can provide both detailed structural and spectral information about urban scenes [3]. Therefore, many researchers have used hyperspectral images in urban LULC classification [3–8]. However, with the influence of urbanization, urban classes are becoming more and more diversified and classification using a single sensor has some drawbacks. Hyperspectral images have abundant

spectral information, allowing the spectral characteristics of ground objects to be characterized well [9,10]. However, different objects possess similar spectral characteristics [11]. As such, it is difficult to distinguish the objects with similar spectral characteristics. Unlike hyperspectral sensors, light detection and ranging (LiDAR) has the advantage of acquiring dense, discrete, detailed, and accurate 3D point coverage over both the objects and ground surfaces [12]. Therefore, LiDAR can provide elevation information for urban LULC classification to distinguish objects with similar spectral characteristics but different elevations. It is possible to greatly improve the accuracy of the classification by fusing the two types of data [13].

In recent years, a number of researchers have fused hyperspectral images with LiDAR data. The classification accuracy of urban LULC has been greatly improved as a result of synthesizing the spectral, spatial, and elevation information. In Liao et al. [14], the urban LULC was acquired by classifying the spatial features, elevation features, spectral features and fusion features, respectively, and the final result was obtained by majority voting. In Ghamisi et al. [15], the attribute profile was considered to model the spatial information of the LiDAR and hyperspectral data. Two classification techniques have been considered to build the final classification map, i.e., random forest (RF) and support vector machine (SVM). In Wang et al. [16], both maximum likelihood and SVM classifiers were used to classify the combined synthesized waveform/hyperspectral image features. In [17–20], RF was used to classify the features extracted from the hyperspectral images and LiDAR data to generate the classification map of the urban area. In general, hyperspectral and LiDAR data fusion mostly uses different feature extraction methods and multiple classifiers or RF (which is also a multi-classifier ensemble system) to complete the classification. Previous studies have focused on voting for different classifiers with equal weights. However, due to the different abilities of the different classifiers to distinguish different types of objects, the voting approach using equal weights is unreasonable.

To solve the problem, this paper proposes optimal decision fusion for urban LULC classification based on adaptive differential evolution (ODF-ADE) to optimize the weights of the different classifiers for hyperspectral remote sensing imagery and LiDAR data. In ODF-ADE, the differential evolution (DE) algorithm—a powerful population-based stochastic search and global optimization technique [21,22]—is used to find the optimal weights of the different classification maps. DE uses genetic operators such as crossover, mutation and selection to guarantee strong global convergence ability and robustness, and is suitable for complex optimization problems. The DE algorithm has been widely applied for many real applications such as numerical optimization [22–25], mechanical engineering [26], feedforward neural network training [27], digital filter design [28], image processing [29,30] and pattern recognition [31,32]. Furthermore, it has also been used in a number of applications in remote sensing such as clustering [33,34], endmember extraction [35] and subpixel mapping [36].

The contributions of this paper are as follows:

(1) **The optimal decision fusion framework.** ODF-ADE is built for use with hyperspectral imagery and LiDAR data. Before the voting operation, the classification maps are generated by the support vector machine (SVM) [37], the maximum likelihood classifier (MLC) [38] and multinomial logistic regression (MLR) [39], which have different advantages in dealing with samples of different distributions. In line with this strategy, the weight optimization problem is transformed into an optimization problem in the feature space by maximizing the objective function, which is constructed using the minimum Euclidean distance between each pixel and the corresponding predicted class in the training samples. Due to the population-based stochastic search and global optimization technique of the DE algorithm, it is used to optimize the constructed objective function. By initializing a set of weights and using crossover and mutation operations for optimization, the performance of the objective function can be improved.

(2) **Adaptive differential evolution.** There are two control parameters involved in DE: the scaling factor  $F$  and the crossover rate  $CR$ . These parameters are often kept fixed throughout the optimization process and can significantly influence the optimization performance of DE [40,41]. An adaptive DE method is proposed to solve the optimal decision fusion problem, in which an adaptive strategy

is utilized to determine the scaling factor  $F$  and crossover rate  $CR$ . The parameters that need to be determined are encoded into an individual, i.e., an individual has a set of parameters and uses genetic operators such as crossover, mutation and selection for the evolution process. The better individuals with better parameters are more likely to survive and produce offspring. This method reduces the time required for finding the appropriate parameters and can produce flexible DE for optimal decision fusion.

(3) **Post-processing based on conditional random fields (CRF).** The commonly used classifiers do not consider the correlations between neighboring pixels, leading to the presence of much low-level noise in the classification map. As an improved model for Markov random fields (MRF), conditional random fields (CRF) has the ability to consider the spatial contextual information in both the labels and observed image data. In order to consider the spatial contextual information and preserve the spatial details in classification, pairwise CRF with an 8-neighborhood is used to smooth the final classification map. The pairwise potential uses the spatial smoothing and local class label cost terms to favor spatial smoothing in the local neighborhood and to take the spatial contextual information into account.

The experimental results obtained in this study demonstrate the efficiency of the proposed ODF-ADE fusion algorithm with the datasets provided by the Data Fusion Technical Committee (DFTC) of the IEEE Geoscience and Remote Sensing Society.

The rest of this paper is organized as follows: Section 2 briefly introduces the basics of the DE algorithm. Section 3 describes the proposed ODF-ADE approach for the fusion of hyperspectral and LiDAR data. The experimental results and analysis are given in Section 4. Section 5 discusses the main properties of ODF-ADE in theoretical and empirical terms. Finally, the conclusions are provided in Section 6.

## 2. Differential Evolution (DE) Algorithm

DE was proposed in 1995 by Storn and Price [21]. Like the other evolutionary algorithms, DE is a stochastic model for the simulation of biological evolution through repeated iterations which preserves the individuals that adapt to the environment. However, compared to the other evolutionary algorithms, DE retains a global search strategy based on the population, with real encoding and simple mutation strategies to reduce the complexity of the genetic operations. The DE algorithm is mainly used for solving global optimization problems. The main steps are mutation, crossover and selection operations, to evolve from a randomly generated initial population to the final individual solution [42]. In the proposed method, we use classical DE [21,23] because this strategy is the most often used in practice. As shown in Figure 1, DE can be described as follows:

The minimization optimization problem in the continuous feature space can be represented as:

$$\min f(X_1, \dots, X_j, \dots, X_D) \text{ s.t. } X_j^{(L)} \leq X_j \leq X_j^{(U)}, j = 1, 2, \dots, D \quad (1)$$

where  $D$  indicates the dimension of the problem, and  $X_j^{(L)}$  and  $X_j^{(U)}$  indicate the minimum and maximum of the  $j$ th element of the individual vector  $X_j$ , respectively. The process of DE can be described as the following four steps:

- Step 1 Initialization: Initialize the population  $X$  randomly, where the size of the population is  $NP$ .
- Step 2 Mutation: With the difference vector of two individuals randomly chosen from the population as the source of random changes in the third individual, generate the mutant individual by obtaining the sum of the difference vector and the third individual according to a scaling factor  $F$ .
- Step 3 Crossover: Mix the parameters of a predetermined target individual  $X_i^t$  and the mutant vector  $V_i^t$  to produce a trial individual  $U_i^t$  by the crossover probability  $CR$ .
- Step 4 Selection: If the fitness value of the test individual is better than the fitness value of the target individual, the test individual replaces the target individual in the next generation; otherwise, the target body remains alive.

In the evolutionary process of each generation, each individual vector is considered as the target individual once. The algorithm retains the excellent individuals while eliminating the inferior individuals and guides the search process to the global optimum solution approximation through continuous iteration calculation.

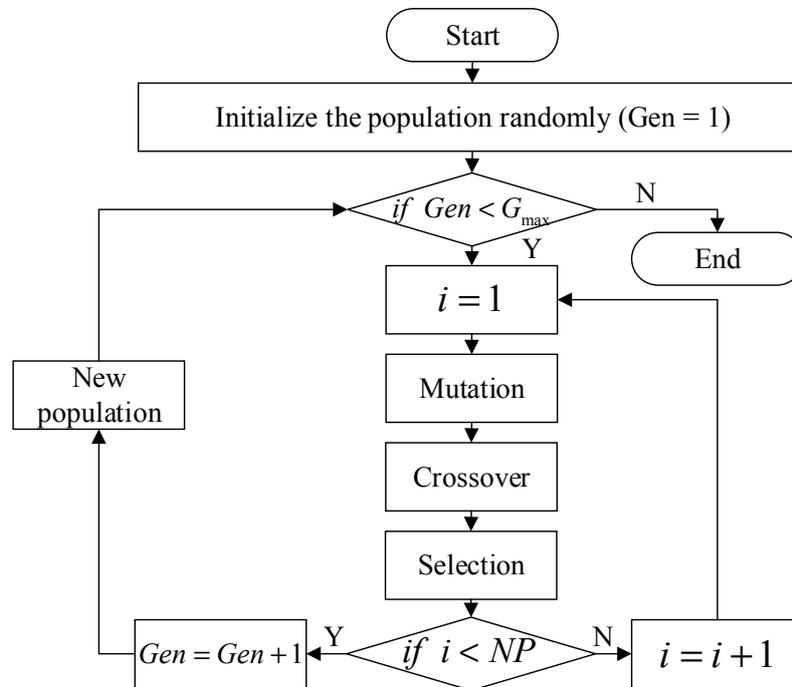


Figure 1. Framework of the differential evolution algorithm.

### 3. ODF-ADE Methodology

Before describing the proposed method, the notations used throughout this paper are defined (Table 1).

Table 1. The defined notations.

Notation	Description
$Map$	The classification maps used in the weighted voting.
$w$	The weight of each class for each classification map.
$C_i$	The $i$ th class of the classification map.
$D_{ai}$	Minimum distance between pixel $a$ and the training data $T_i$ , for which the label is class $i$ .
$P_i$	The population of the $i$ th generation.
$X_{i,j}$	The $j$ th individual of $P_i$ .
$NP$	The size of the population.
$F$ $CR$	The two parameters in DE (i.e., the mutation scale and the crossover probability).
$p_m$	The parameter in the self-adaptive strategy (i.e., the mutation ratio).

To solve the problem of the inadequate utilization of resources caused by equal voting, the proposed framework uses adaptive DE to optimize the weights of the different classification maps to achieve a better effect. Firstly, the normalized difference vegetation index (NDVI), the gray-level co-occurrence matrix (GLCM) textures and the digital surface model (DSM) elevation feature are added to the spectral features extracted by principal component analysis (PCA) or minimum noise fraction (MNF), to form the feature vector. Three classification algorithms (i.e., MLC, SVM, and MLR) are then used to obtain the initial classification maps. A more accurate classification map is generated

by weighted voting using the adaptive DE algorithm. The final classification map is generated after post-processing. The main procedure of the data fusion framework is shown in Figure 2 and is described as follows.

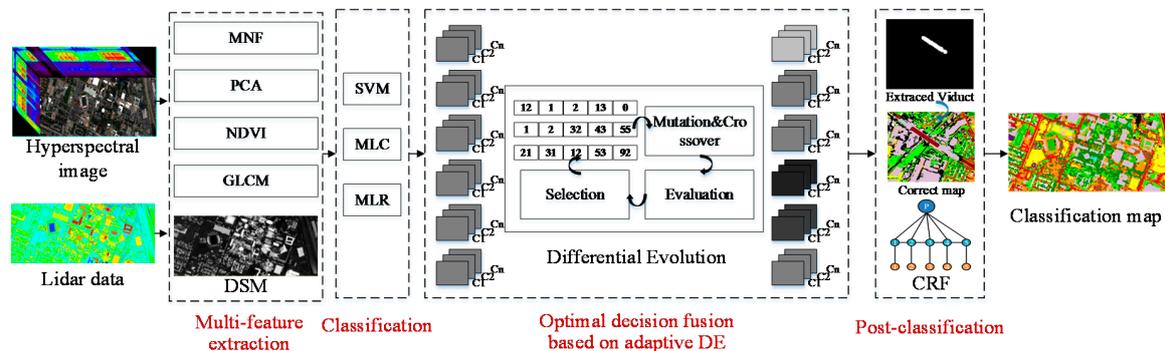


Figure 2. Framework of the proposed methodology.

### 3.1. Multi-Feature Extraction

In order to represent the features of objects from different angles, MNF or PCA are used to reduce the dimensionality of the hyperspectral image, and the NDVI is used to distinguish the vegetation. To utilize the spatial information, the GLCM is computed. Finally, the DSM is used to represent the elevation information. And the final feature maps are stacked by these features (i.e., MNF + NDVI + GLCM + DSM/PCA + NDVI + GLCM + DSM).

The NDVI is a simple ratio that can be used to analyze remote sensing measurements, to assess whether the target being observed contains live green vegetation or not. In general, if there is much more reflected radiation in the near-infrared wavelengths than in the red wavelengths, then the vegetation in that pixel is likely to be healthy.

A gray-level co-occurrence matrix or gray-level co-occurrence distribution is a matrix that is defined over an image as the distribution of the co-occurring pixel values (grayscale values) at a given offset. The gray-level co-occurrence matrices can measure the texture of the image and they are typically large and sparse, various metrics are used to obtain a more useful set of features. Therefore, the gray-level co-occurrence matrix can be utilized to increase the separability between classes. Homogeneity, also called inverse disparity, measures the local gray uniformity of an image. If the textures of the different regions are similar and the local gray-level of the image is uniform, then the homogeneity will be larger. Therefore the homogeneity of GLCM is used to describe the spatial texture feature.

The DSM refers to a ground elevation model which incorporates the ground surface, buildings, bridges and trees. In comparison, a digital elevation model (DEM) contains only the elevation information of the terrain and does not contain other surface information. The DSM contains the elevation information of any surface elements (soil, vegetation, artificial structures etc.). Therefore, the DSM data obtained from the LiDAR data are added to characterize the elevation information.

### 3.2. Urban LULC Classification by Different Classifiers

SVM is established based on the Vapnik-Chervonenkis (VC) dimension theory and risk minimization principle to obtain the best classification result, thereby finding the best balance between model complexity (i.e., learning accuracy of the specific training samples) and learning ability (i.e., the ability to identify any sample without error), according to the limited sample information. SVM has many unique advantages in solving small-sample, nonlinear and high-dimensional pattern recognition.

MLC is an image classification method based on statistical knowledge and computing probability. Firstly, the nonlinear discriminant function set is established according to Bayes' decision criterion.

It is then assumed that all kinds of distribution functions are normal distributions. Finally, the training area is selected to calculate the attribution probability of each sample area to obtain the classification map. When classifying, MLC not only considers the distance of the sample to the class center, but also takes into account the distribution characteristics.

MLR is a particular solution to the classification problem that assumes that a linear combination of the observed features and some problem-specific parameters can be used to determine the probability of each particular outcome of the dependent variable. The best values of the parameters for a given problem are usually determined from training data. The algorithm adopts an MLL prior to modeling the spatial information present in the class label images.

These three algorithms, which are all robust, can make full use of the prior information of the samples and are therefore suitable for the classification of complex objects in urban areas. The six classification images are obtained with the two sets of features—MNF + NDVI + GLCM + DSM, PCA + NDVI + GLCM + DSM—by these three classifiers.

### 3.3. Optimal Decision Fusion Based on Adaptive DE

After the classification maps are obtained by the classification step, they can be used to generate a more accurate classification map by decision fusion, e.g., majority voting. Different classifiers have different abilities to distinguish different objects. In order to avoid the unreasonable use of resources caused by majority voting, weighted voting is used for the decision-level fusion. The DE algorithm allows for global optimization and can be applied to optimize the weights. In addition, a self-adaptive parameter selection method is proposed to adaptively choose the appropriate parameters during the course of DE.

#### 3.3.1. Initial Population

After obtaining the classification maps of the different classifiers, the population can be initialized as  $P_g = \{X_{1,g}, \dots, X_{k,g}, \dots, X_{NP,g}\}$ , where  $X_{kG}$  represents the  $k$ th individual in the  $g$ th generation and NP is the size of the population. As shown in Figure 3, each individual  $X_{k,g} = \{x_{k,g}^1, \dots, x_{k,g}^t, \dots, x_{k,g}^D\}$  denotes the weight of each class for each classification map. The weights also need to be initialized. The two variables are defined as  $M$  and  $N$ , which represent the number of land-cover labels and classification maps, respectively.  $D$  equals  $M \times N$  and denotes the number of chromosomes that one individual  $x_{kG}$  contains. The initial population  $P_1$  is generated randomly from 0 to 1.

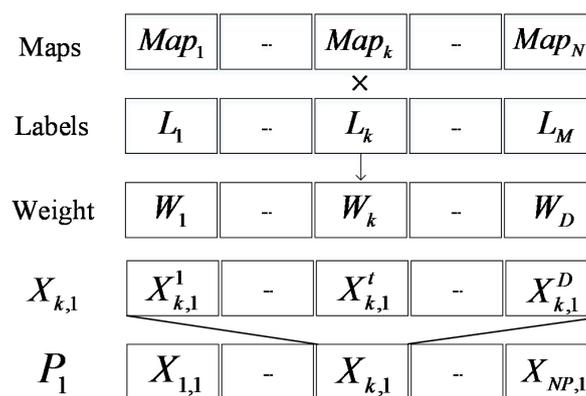


Figure 3. Initial population.

#### 3.3.2. Calculation of the Objective Function

In this paper, the objective function is constructed using the sum of the minimum Euclidean distances between each pixel and the corresponding predicted class in the training samples. In the proposed algorithm the purpose of DE is to obtain the maximum value of the objective function.

The classification map  $Map_{basis}$  is obtained by the usual majority voting. Using  $Map_{basis}$  as the basis, voting is undertaken with the weight of chromosome  $x_{k,g}^t$ . If the pixels in  $Map_1$  belong to class  $C_i$ , then the weight value of each classification map refers to the corresponding weight of class  $C_i$ , respectively. The new classification map  $Map_{new}$  is obtained by traversing the whole image.

If the predicted label of pixel  $a$  is class  $i$ , which is part of classification maps  $b_1, \dots, b_m$  ( $m$  denotes the number of classification maps that predict the label is  $i$ ), then:

$$j_a = \left( \frac{w_{b_1 i}}{D_{ai}} + \dots + \frac{w_{b_m i}}{D_{ai}} \right) / m \tag{2}$$

As shown in Figure 4,  $w_{b_m i}$  denotes the weight of class  $i$  of classification map  $b_m$  and  $D_{ai}$  denotes the minimum distance between pixel  $a$  and training data  $T_i$ , for which the label is class  $i$ :

$$D_{ai} = \min\{d(a, i_1), d(a, i_2), \dots, d(a, i_m), \dots, d(a, i_n)\} \tag{3}$$

$$d(a, i_m) = \|\mu_a - \mu_{i_m}\|_2^2 \tag{4}$$

where  $n$  represents the number of  $T_i$ ,  $i_m$  represents the  $m$ th pixel of  $T_i$ ,  $\mu_a$  represents the image vector of pixel  $a$  and  $\mu_{i_m}$  represents the image vector of pixel  $i_m$ .  $d(a, i_m)$  denotes the Euclidean distance between  $\mu_a$  and  $\mu_{i_m}$ . The smaller the value of  $D_{ai}$  and the greater the value of  $w_{b_m i}$ , the greater the value of  $j_a$ , which means a larger weight.

The fitness of the individual  $X_{k,g}$  is calculated as follows:

$$J_{k,g} = \sum_{a=1}^{num} j_a \tag{5}$$

where  $num$  denotes the total number of image pixels.

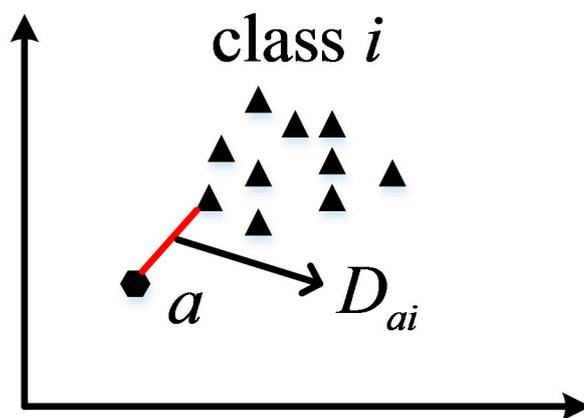


Figure 4. The objective function based on the minimum distance.

### 3.3.3. Adaptive Mutation and Crossover

DE generates the mutant individual by obtaining the sum of the difference vector and the third individual according to a scaling factor  $F$ . The trial individual  $U_i^t$  is produced by mixing the parameters of a predetermined target individual  $X_i^t$  and the mutant vector  $V_i^t$  using the crossover probability  $CR$ . Suitable control parameters are always different for different real problems. However, in some cases, the time for finding these appropriate parameters can be unacceptably long.

To solve the problem, a self-adaptive strategy for the control parameters is used. As shown in Figure 5, the control parameters  $F$  and  $CR$  are encoded to each individual. This means that each individual has its corresponding  $F$  and  $CR$  values, which can be adjusted during evolution [42].

The weight optimization solution is represented by the D-dimensional vector  $X_{k,g}$  and two control parameters  $F_{k,g}$  and  $CR_{k,g}$  in the  $g$ th generation, where  $k = 1, 2, \dots, NP$ .

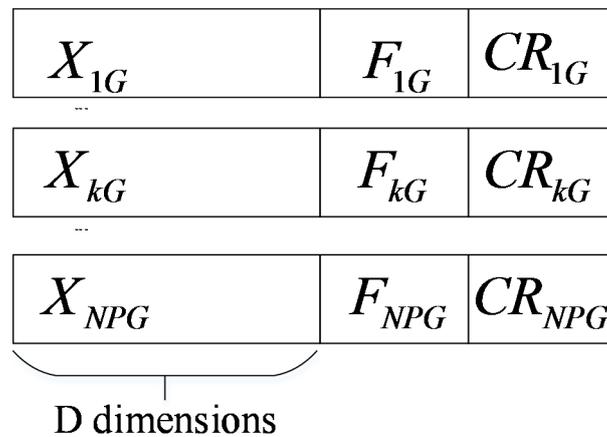


Figure 5. The self-adaptive encoding strategy.

For each vector  $X_{kG}$  at generation  $G$ , its associated mutant vector  $V_{k,g} = \{V_{k,g}^1, V_{k,g}^2, \dots, V_{k,g}^D\}$  can be generated via the strategy DE/rand/1/bin (rand refers to the mutation strategy, which uses a random selection of individuals to prevent the population from getting into the deadlock of local searching, 1 represents the number of differential vectors and bin refers to the binomial crossover strategy to expand the search space), which is the strategy most often used in practice [30,43,44]. The mutation operators are as follows:

$$V_{k,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}), k = 1, \dots, NP \tag{6}$$

where the indices  $r_1, r_2$  and  $r_3$  are mutually exclusive integers randomly generated within the range  $(1, NP), r_1 \neq r_2 \neq r_3 \neq k$ .

The higher the objective function in Equation (5), the more likely the individual is to survive and produce offspring which results in better individuals and increases the probability of finding the optimal solution. To adaptively determine the mutation rate  $p_m$  according to the derivative of the objective value of each individual, the process is as follows:

$$p_m = \frac{J'(X_{k,g}) - \min(J'(X_{k,g}))}{\max(J'(X_{k,g})) - \min(J'(X_{k,g}))} \tag{7}$$

$$J'(X_{k,g}) = 1/J(X_{k,g}) \tag{8}$$

The new control parameters in the  $G + 1$ th generation  $F_{kG+1}$  and  $CR_{kG+1}$  are updated as follows, with probability  $p_m$ :

$$F_{k,g+1} = \begin{cases} 1 - rand_1^{(1 - \frac{g}{G_{max}})^b}, & \text{if } rand_2 < p_m \\ F_{k,g}, & \text{otherwise} \end{cases} \tag{9}$$

where  $rand_t, t \in \{1, 2\}$ , denotes the uniform random values within the range  $(0, 1)$ ,  $G_{max}$  is the maximum iteration number,  $g$  is the iteration number and  $b$  is a parameter to decide the nonconforming degree, for which the exponential value is set to three [45].

After the mutation phase, a crossover operation is applied to generate a trial vector  $U_{k,g} = \{U_{k,g}^1, U_{k,g}^2, \dots, U_{k,g}^D\}$  for the mutant vector  $V_{kG}$  as follows:

$$u_{k,g}^t = \begin{cases} v_{k,g}^t, & \text{if } (rand_t[0,1] < CR) \text{ or } (t = t_{rand}) \\ x_{k,g}^t, & \text{otherwise} \end{cases} \quad (10)$$

$CR_{kG+1}$  is updated using the following [38]:

$$CR_{k,g+1} = \begin{cases} rand_4, & \text{if } rand_3 < p_m \\ CR_{k,g}, & \text{otherwise} \end{cases} \quad (11)$$

where  $rand_t$ ,  $t \in \{3,4\}$  denotes the uniform random values within the range (0,1).  $F_{k,g+1}$  and  $CR_{k,g+1}$  are obtained before the mutation is performed. Therefore, they influence the mutation, crossover and selection operations of the new vector  $X_{k,g+1}$ .

### 3.3.4. Selection

After the calculation of the objective function using Equation (5), a selection operation is performed. The objective function value of each trial vector  $J(U_{k,g})$  is compared with that of its corresponding target vector  $J(X_{k,g})$  in the current population. If the weight vector, which is obtained in this generation, has a higher or equal objective function value compared with the corresponding target vector, the trial vector will replace the target vector and form the new population of the next generation. Otherwise, the target vector will remain in the population for the next generation. The selection operation can be expressed as follows:

$$X_{k,g+1} = \begin{cases} U_{k,g}, & \text{if } J(U_{k,g}) \geq J(X_{k,g}), k = 1, 2, \dots, NP \\ X_{k,g}, & \text{otherwise} \end{cases} \quad (12)$$

### 3.3.5. Stopping Condition

If generation  $g$  does not meet the maximum generation number  $G_{max}$ , go to Step 2. Otherwise, output the best individuals as the weight of each class for each classification map. Finally, obtain the final classification result using the optimized weights.

## 3.4. Post-Classification

**(1) Decision fusion for viaducts.** Viaducts are common in urban areas. However, due to the similarity of the construction materials, they can be easily confused with tall buildings. Therefore, the proposed framework employs an object-based method to extract and classify the viaducts in the urban area, so as to improve the overall classification accuracy. Viaducts are easy to extract due to the gradually changing characteristic of the viaducts in elevation. Region growing is a simple region-based image segmentation method. This approach to segmentation examines the neighboring pixels of the initial seed points (which are selected manually) and determines whether the pixel neighbors should be added to the region or not. The process is iterated in the same manner as the general data clustering algorithms. As a result, the region growing method performed in the DSM image is used to extract the viaducts to complete the operation.

**(2) Post-classification by CRF.** The spatial contextual information of remote sensing imagery is very important for the classification task [46,47]. Those prior operations which do not consider the correlations between neighboring pixels lead to the presence of much low-level noise in the classification map. As an improved model of MRF, CRF has the ability to consider the spatial contextual information in both the labels and observed image data. In order to consider the spatial contextual information and preserve the spatial details in the classification, pairwise CRF with an 8-neighborhood

is used to smooth the final classification map. The pairwise potential uses the spatial smoothing and local class label cost terms to favor spatial smoothing in the local neighborhood and to take the spatial contextual information into account. The local class label cost term also has the ability to alleviate an oversmooth classification result since it considers the different label information of the neighboring pixels at each iterative step in the classification.

## 4. Experiments

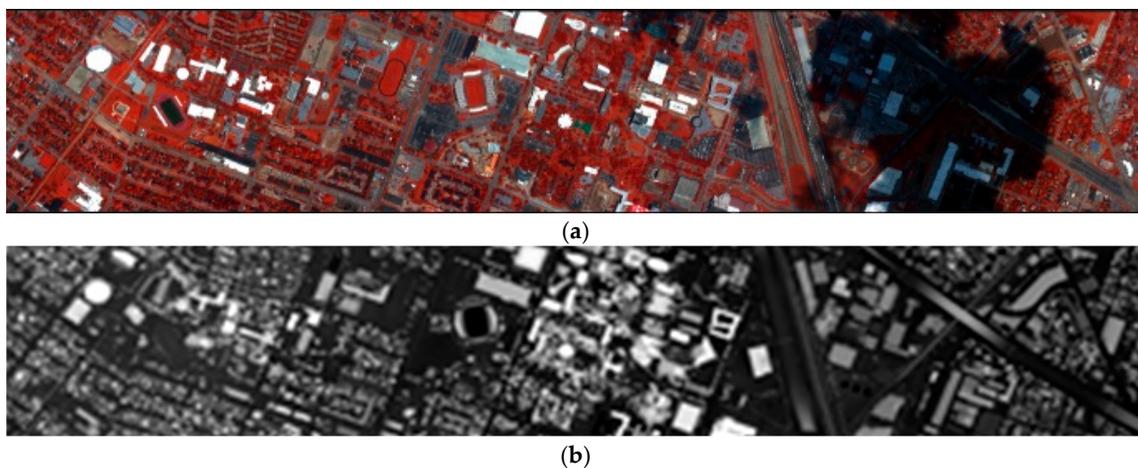
### 4.1. Experimental Data

#### 4.1.1. Hyperspectral Data

The hyperspectral imagery was acquired on 23 June 2012 between the times of 17:37:10 UTC and 17:39:50 UTC. The hyperspectral sensor used was the CASI visible near-infrared (VNIR) sensor and the average height of the sensor above ground was 1676.4 m. The hyperspectral imagery consists of 144 spectral bands in the 380–1050 nm region. The spatial and spectral resolutions are 2.5 m and 4.8 nm, respectively. And the image pixel number is  $1905 \times 349 \times 144$ .

#### 4.1.2. LiDAR Data

The LiDAR data were acquired on 22 June 2012 between the times of 14:37:55 UTC and 15:38:10 UTC. The LiDAR point cloud data were obtained from the National Science Foundation Funded Center for Airborne Laser Mapping (NCALM). The sensor recorded five returns and intensity at a platform altitude of 609.6 m above ground, with an average point spacing of 0.74 m. The LiDAR data is rasterized with a spatial resolution of 2.5 m which is identical to the spatial resolution of the hyperspectral image. In this study, the scan angle and atmospheric effects were not taken into account. The hyperspectral image and the DSM are shown in the Figure 6.



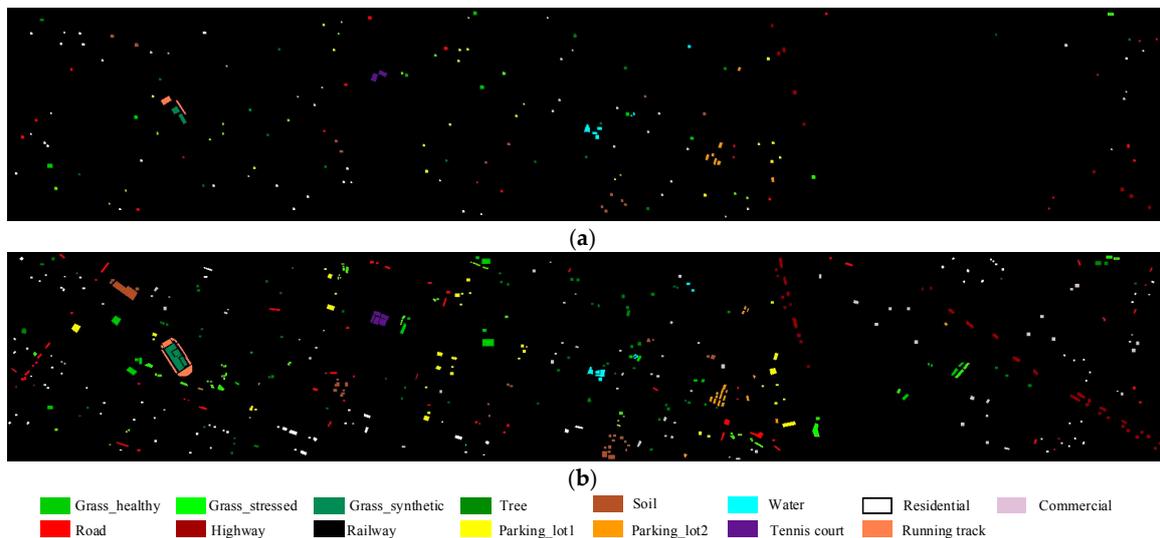
**Figure 6.** Experimental data. (a) False-color image of the hyperspectral image. (b) LiDAR-derived DSM.

#### 4.1.3. Training Samples and Validation Samples

Each pixel in the image was mapped to one of 15 classes, namely, healthy grass, stressed grass, synthetic grass, trees, soil, water, residential, commercial, road, highway, railway, parking lot 1 (there are cars in the parking lot), parking lot 2 (there is no car in the parking lot), tennis court and running track. The numbers of training and validation samples are shown in Table 2. The location and distribution of the training and validation samples are shown in Figure 7.

**Table 2.** Number of training and validation samples.

	Grass_healthy	Grass_stressed	Grass_synthetic	Tree	Soil	Water	Residential	Commercial
Training	198	190	192	188	186	182	196	191
validation	1053	1064	505	1056	1056	143	1072	1053
	Road	Highway	Railway	Parking_lot1	Parking_lot2	Tennis court	Running track	
Training	193	191	181	192	184	181	187	
validation	1059	1036	1054	1041	285	247	473	

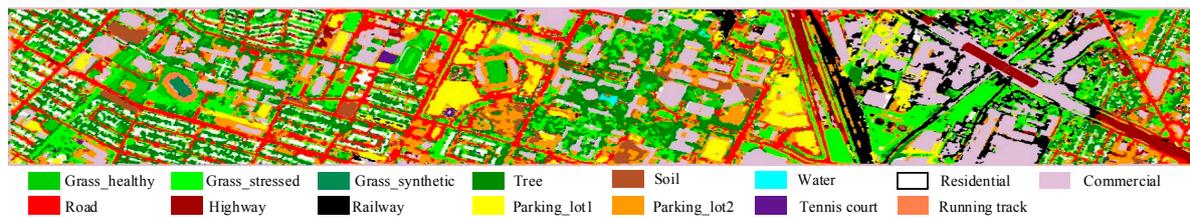
**Figure 7.** Location and distribution of the training and validation samples. (a) Location and distribution of the training samples. (b) Location and distribution of the validation samples.

## 4.2. Experimental Results and Analysis

### 4.2.1. Experimental Results

MNF and PCA were the two methods used to extract the spectral features. The 22 features containing the most information for the hyperspectral image were kept for both MNF and PCA. The vegetation was characterized by the NDVI (band 69 is the red band, band 82 is the infrared band). In order to increase the class separability, the GLCM was added to characterize the texture information. The GLCM texture was produced by the homogeneity measure with a window size of nine using the first three principal components obtained by PCA. Finally, the DSM data generated by the LiDAR data were added to form the final feature image. The two feature maps (PCA + NDVI + GLCM + DSM/MNF + NDVI + GLCM + DSM) were classified by SVM, MLC and MLR, using DE to optimize the weights of each classification map. The final classification map was obtained by a post-classification operation on the weighted voting result. The features (i.e., MNF, PCA, NDVI, GLCM) were extracted by ENVI. The classifiers SVM, MLC, MLR were operated by Visual C++ 6.0, ENVI and Matlab R2014a. The DE and CRF algorithms were both operated using Visual C++ 6.0. The final classification map is shown in Figure 8. The final overall classification accuracy is 93.5% and the Kappa coefficient is 0.9299.

The classification accuracy of each category is shown in Table 3. The data show that the algorithm achieves a good effect in most of the classes, especially the grass\_stressed, grass\_synthetic, tree, soil, tennis court and running track classes, where the accuracy reaches or almost reaches 100%. However, the spectral, texture and elevation information of the residential and commercial classes are very similar, which leads to the classification results of these categories not being ideal. There is also some confusion between highway and railway due to the influence of the shadow area in the hyperspectral image.



**Figure 8.** Final classification map.

**Table 3.** Accuracy of the classes (%).

Grass_healthy	Grass_stressed	Grass_synthetic	Tree	Soil	Water	Residential	Commercial
83.1	99.6	99.4	99.2	99.8	95.8	82.8	94.9
Road	Highway	Railway	Parking_lot1	Parking_lot2	Tennis court	Running track	OA
96.5	81.3	95.5	96.8	87.4	100.0	99.6	93.5

In order to verify the effects of the proposed algorithm, a multi-group comparison experiment is carried out. In addition, McNemar's test [48] is used to determine the statistical significance of the differences between the classification results obtained by the varying algorithms, using the same test sample set. Given two classifiers  $C_1$  and  $C_2$ , the number of pixels misclassified by  $C_1$  but not by  $C_2$  is denoted as  $M_{12}$ , and  $M_{21}$  represents the number of cases misclassified by  $C_2$  but not by  $C_1$ . If  $M_{12} + M_{21} \geq 20$ , the  $X_2$  statistic can be considered as following a chi-squared distribution:

$$X^2 = \frac{(|M_{12} - M_{21}| - 1)^2}{M_{12} + M_{21}} \approx \chi_1^2 \quad (13)$$

This test can check whether the difference between varying classification results is meaningful. Given a significance level of 0.05, then  $\chi_{0.05,1}^2 = 3.841459$ . If  $X_2$  is greater than  $\chi_{0.05,1}^2$ , the results of the two classifiers  $C_1$  and  $C_2$  are significantly different.

#### 4.2.2. Effects of Adding LiDAR Data

In this paper, the DSM is used to form the two feature maps, i.e., PCA + NDVI + GLCM + DSM and MNF + NDVI + GLCM + DSM. Adding the LiDAR data to characterize the elevation information can improve the classification result. In order to verify the effects of adding the LiDAR data, the two feature maps (MNF + NDVI + GLCM \ PCA + NDVI + GLCM) extracted from the hyperspectral image were classified. The weights of the six classification maps obtained by the MLC \ MLR \ SVM classifiers were then optimized using the proposed ODF-ADE. The overall accuracy (OA) of the classification results and the accuracy of each category are shown in Table 4, where S means that the result was obtained only using the hyperspectral image (i.e., obtained by optimal weights only using hyperspectral image) and S + L means that the result was obtained by the proposed method (i.e., obtained by optimal weights using both hyperspectral and LiDAR data). Both the results are without post-processing.

**Table 4.** Comparison of the classification accuracy after adding LiDAR data (%).

	Grass_healthy	Grass_stressed	Grass_synthetic	Tree	Soil	Water	Residential	Commercial
S	83.0	98.9	100.0	95.7	98.3	96.5	85.6	73.0
S + L	83.0	98.6	99.4	97.7	99.3	95.1	83.9	93.1
	Road	Highway	Railway	Parking_lot1	Parking_lot2	Tennis court	Running track	OA
S	85.5	61.1	88.6	93.9	86.6	100.0	98.3	87.8
S + L	93.3	61.7	94.2	93.9	87.7	100.0	98.3	90.8

From both the OA and the accuracy of the various categories, the method proposed in this paper achieves very good results. As can be seen from Table 4, the overall precision is increased by 3% after

adding the LiDAR data. For certain classes, such as commercial, road and railway, the classification accuracy is greatly improved. These classes have similar spectral information, but can be distinguished by the LiDAR data because of the different elevations. According to these data, we can clearly see that the method that fuses LiDAR and hyperspectral data achieves good results in the urban area land-use classification and achieves the expected goal. In addition, the McNemar's test value of these two approaches is given in Tables 5 and 6 to evaluate the statistical significance. It can be seen from Tables 5 and 6 that McNemar's values between S and S + L are greater than the critical value of  $\chi^2_{0.05,1}$  (3.841459), which means that the differences are significant.

**Table 5.** Comparison of the McNemar's test values after adding LiDAR data.

	S	S + L
S	NA	15.002
S + L		NA

**Table 6.** McNemar's test values of majority voting and weighted voting.

	MV	WV
MV	NA	3.9047
WV		NA

#### 4.2.3. Effect of the Weighted Voting

Due to the abilities of the different classifiers to distinguish different types of objects, a voting approach that integrates the classification results using equally weighted classifiers lacks scientific rigor. Therefore, the weights of each class for each classifier were optimized through weighted voting of the six classification images obtained by the different classifiers using different features. The weights of the 15 classes of the six classification maps were initialized randomly from the range (0,1). The initial population size was 30 and the maximum number of iterations was 500. Through optimizing the weight according to the distance between each pixel and the corresponding training sample data, the global optimum solution could be found by iteration. After the optimal decision fusion, the OA of the classification map reached 90.83%, which is much higher than any of the prior classification maps. The classification accuracy of each classifier and the voting result are shown in Table 7 where most of the class accuracies obtained by the weighted voting are better than any of the six initial classifications maps. It can also be seen from Table 8 that the McNemar's test values between voting and any other classifiers are greater than the critical value of  $\chi^2_{0.05,1}$  (3.841459), which means that the algorithms are significant.

In order to verify the effect of the weighted voting, the results of majority voting for which the weight was equal and weighted voting for which the weight was optimized by adaptive DE are compared. The input classification maps were the same six maps obtained before. The OA of the classification results and the accuracy of each category are shown in Table 9. MV means that the result was obtained by voting with equal weights using both the hyperspectral image and LiDAR data; WV means that the result was obtained by optimal weights using the same data. Both the results are without post-processing.

As can be seen from Table 9, the OA is improved after the weighted voting. The accuracy of certain classes, such as highway, railway, parking\_lot1 and parking\_lot2, is greatly improved. The results show that the weighted voting can fully utilize the differences among the different classifiers and improve the classification result. The result of McNemar's test is shown in Table 6. The value is greater than the critical value of  $\chi^2_{0.05,1}$  (3.841459), which means the proposed algorithm has a significant difference with the majority voting.

**Table 7.** Comparison of the different classification strategies (%).

	MNF_SVM	MNF_MLC	MNF_MLR	PCA_SVM	PCA_MLC	PCA_MLR	Voting
Grass_healthy	82.8	82.1	82.9	81.7	82.2	81.9	82.9
Grass_stressed	98.8	87.7	84.5	98.5	84.4	79.8	98.6
Grass_synthetic	100.0	99.2	100.0	97.0	99.2	100.0	99.4
Tree	90.9	99.4	92.5	91.4	96.7	95.0	97.7
Soil	98.2	97.4	99.2	97.2	94.1	97.5	99.3
Water	99.3	95.1	97.2	97.9	77.6	90.9	95.1
Residential	84.0	84.8	76.6	90.2	70.8	78.2	83.8
Commercial	62.9	97.6	66.5	61.4	99.9	44.1	93.1
Road	83.9	93.2	85.9	78.6	85.9	88.5	93.3
Highway	60.8	54.0	76.3	57.1	51.3	44.4	61.7
Railway	90.2	80.7	92.3	63.5	72.1	71.8	94.2
Parking_lot1	82.9	78.9	93.4	69.4	75.6	50.9	93.9
Parking_lot2	82.1	82.1	74.7	78.9	83.2	69.8	87.7
Tennis court	98.0	99.2	100.0	99.6	98.4	100.0	100.0
Running track	99.0	95.4	97.5	96.6	94.1	94.9	98.3
OA (%)	85.3	86.9	86.3	81.0	82.9	75.9	90.8

**Table 8.** McNemar's test values of the different classification strategies.

	MNF_SVM	MNF_MLC	MNF_MLR	PCA_SVM	PCA_MLC	PCA_MLR	Voting
MNF_MLC	NA	4.7907	3.2276	15.1764	6.4899	24.3791	21.729
MNF_MLR		NA	1.6839	15.9792	16.3601	28.3189	17.3028
MNF_SVM			NA	14.3217	9.4674	26.649	15.2761
PCA_MLC				NA	4.9738	12.6119	30.9685
PCA_MLR					NA	17.1309	27.7131
PCA_SVM						NA	39.4597
WV							NA

**Table 9.** Classification accuracy of majority voting and weighted voting (%).

	Grass_healthy	Grass_stressed	Grass_synthetic	Tree	Soil	Water	Residential	Commercial
MV	83.0	98.5	100.0	95.8	99.7	97.2	85.3	93.7
WV	83.0	98.6	99.4	97.7	99.3	95.1	83.9	93.1
	Road	Highway	Railway	Parking_lot1	Parking_lot2	Tennis court	Running track	OA
MV	96.2	59.8	90.4	89.3	83.3	100.0	98.3	90.2
WV	93.3	61.7	94.2	93.9	87.7	100.0	98.3	90.8

#### 4.2.4. Effect of Post-Classification

In order to solve the problem of the viaduct being confused with tall buildings, the DSM data were used to identify the viaduct by a region growing operation. The extracted viaduct was used to correct the classification of the highway class and is shown in Figure 9. The CRF-based smoothing approach was used in the experiment to generate the final classification map.

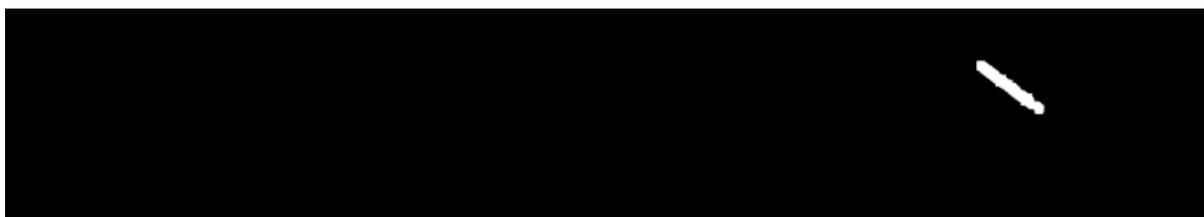
**Figure 9.** The extracted viaduct.

Table 10 shows the classification accuracy of the result obtained by weighted voting and post-classification. WV means the result of weighted voting and APC means the result after

post-classification. The accuracy of the highway class increases by 20% after the post-classification operation. It can also be clearly seen that the algorithm considering the spatial interaction shows an improvement of more than 3% over the result of weighted voting, in terms of the OA, which demonstrates the effectiveness of incorporating the spatial contextual information. The post-classification also has a great effect on the accuracy of each class.

**Table 10.** Classification accuracy of weighted voting and post-classification (%).

	Grass_healthy	Grass_stressed	Grass_synthetic	Tree	Soil	Water	Residential	Commercial
WV	83.0	98.6	99.4	97.7	99.3	95.1	83.9	93.1
APC	83.0	99.6	100.0	99.2	99.8	95.8	82.9	94.9
	Road	Highway	Railway	Parking_lot1	Parking_lot2	Tennis court	Running track	OA
WV	93.3	61.7	94.2	93.9	87.7	100.0	98.3	90.8
APC	96.5	81.2	95.5	96.8	87.0	100.0	99.6	93.5

## 5. Sensitivity Analysis

### 5.1. Sensitivity to Features

In the proposed framework, multiple features are extracted to characterize the experimental area. In order to verify the effect of adding the multiple features, we compared different combinations of different features. Twelve feature maps were classified by MLC and the OA is shown in Table 11. In addition, the McNemar's test is operated to verify whether there are significant differences between different features and the results are shown in Table 12.

As can be seen in Table 11, the combination of spectral, NDVI and spatial information is better than any other combination. The NDVI is added to highlight the vegetation, the texture operators based on the GLCM are utilized to increase the separability between classes and the DSM data obtained from the LiDAR data are added to characterize the elevation information. Therefore, two groups of feature maps, i.e., MNF + NDVI + GLCM + DSM/PCA + NDVI + GLCM + DSM are used in consideration of the OA and McNemar's test values.

As can be seen in Table 12, most of the McNemar's test values between the combination of spectral, NDVI and spatial information (i.e., MNF + NDVI + GLCM + DSM/PCA + NDVI + GLCM + DSM) and any other combination, are greater than the critical value of  $\chi^2_{0.05,1}$  (3.841459) which means the different combinations of features are significant.

**Table 11.** Classification accuracy of the different features.

	PCA	PCA + NDVI	PCA + GLCM	PCA + DSM	PCA + NDVI + GLCM	PCA + GLCM + NDVI + DSM
OA(%)	82.1	82.2	80.7	83.9	80.9	82.9
Kappa	0.805	0.807	0.791	0.825	0.793	0.815
	MNF	MNF + NDVI	MNF + GLCM	MNF + DSM	MNF + NDVI + GLCM	MNF + GLCM + NDVI + DSM
OA(%)	81.7	81.0	83.5	85.3	84.4	86.9
Kappa	0.802	0.794	0.820	0.840	0.830	0.857

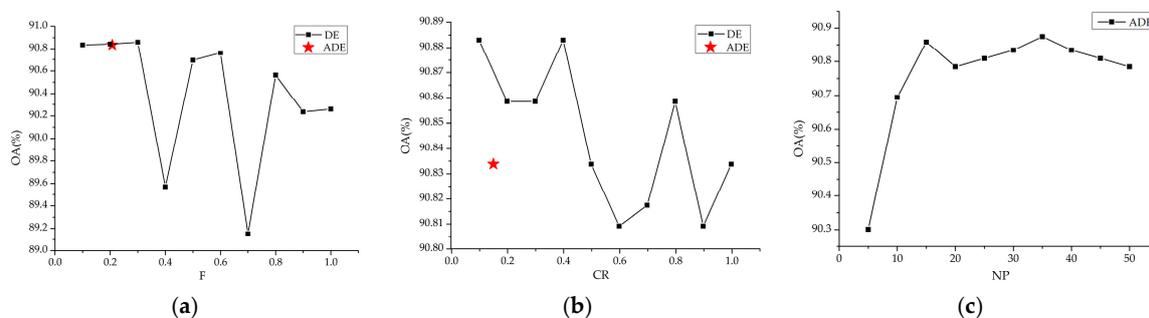


## 5.2. Sensitivity to the Parameter of ODF-DE

To compare the self-adaptive version of the ODF-DE algorithm, i.e., ODF-ADE, with the original ODF-DE algorithm, the best control parameters setting for ODF-DE are needed. For the ODF-DE algorithm, the control parameter values during the subpixel mapping process were not changed, except for the analyzed parameter. The experimental parameters of the original ODF-DE algorithm were set as follows: CR = 0.8, F = 0.3, and the maximum number of iterations was 500.

### 5.2.1. Sensitivity of Parameter F

According to the brief introduction to DE provided above, F is an important parameter for the ODF-DE algorithm. Hence, the impact of parameter F on the algorithm was tested. Parameter F was set from 0.1 to 1.0 with a step size of 0.1, and the other parameters were fixed as NP = 30, CR = 0.8, and the maximum number of iterations as 500. The experimental results are presented in Figure 10a. From this figure, the best adjusted OA of ODF-DE, i.e., 90.86% for the experimental image, is obtained when F is equal to 0.3. Although ODF-DE can obtain a higher OA than ODF-ADE, the ODF-ADE algorithm does not need any other prior knowledge.



**Figure 10.** Sensitivity to the parameters of ODF-DE. (a) Sensitivity of ODF-DE in relation to F. (b) Sensitivity of ODF-DE in relation to CR. (c) Sensitivity of ODF-DE in relation to NP.

### 5.2.2. Sensitivity of Parameter CR

For the experimental images, the ODF-ADE algorithm was performed with CR taken from (0.1, 1.0) with a step size of 0.1, while the other parameters were set as follows: NP = 30, F = 0.3, and the maximum number of iterations was 500. The experimental results are shown in Figure 10b. The best adjusted OA value of ODF-DE for the experimental image, i.e., 90.86%, is obtained by CR = 0.2 (NP = 30, F = 0.3, and the maximum number of iterations is 500). The values are slightly higher than ODF-ADE, i.e., 90.83%. Although ODF-DE can obtain satisfactory results by adjusting the value of parameter CR, ODF-ADE can adaptively provide similar or better decision fusion results, without prior knowledge or experience.

### 5.2.3. Sensitivity of Parameter NP

The number of the initial population NP is very important in maintaining the diversity of the population and extending the search range in the feature space. To analyze the sensitivity in relation to parameter NP, the other parameters, i.e., CR and F, were determined adaptively and NP assumed the following values for the experimental images: NP = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}. Figure 10c shows the sensitivity of ODF-ADE in relation to parameter NP by analyzing the relationship between OA and NP. There is an upward trend in the OA of the ODF-ADE algorithm when the value of NP is changed from 5 to 50. When NP is equal to 35, the highest OA value of ODF-ADE is 90.87%.

Based on the aforementioned sensitivity analyses, there is a disadvantage in the original ODF-DE, i.e., the best control parameter settings of ODF-DE are problem-dependent. The proposed ODF-ADE

overcomes this disadvantage and it is much more independent than the original ODF-DE. Therefore the conclusion is that ODF-ADE is an effective decision fusion algorithm.

## 6. Conclusions

Based on DE theory, this paper has proposed a new optimal decision fusion strategy for the fusion of hyperspectral images and LiDAR data, namely ODF-ADE. In line with this strategy, the optimal decision fusion problem is transformed into an optimization problem in the feature space by maximizing the objective value. The traditional voting algorithm always uses equal weights to fuse the classification maps, which results in the differences among the different classifiers not being fully utilized. In the proposed method, DE, which has the ability of global optimization, is used to obtain the weights of the different classification maps. In addition, in the traditional DE it is necessary to choose appropriate control parameters, employing the prior experience of the user, for population size NP, crossover rate CR and scaling factor F. This is quite a difficult task because the best settings for the control parameters are not easy to determine for complex problems. In the proposed method, a self-adaptive strategy is utilized to determine the parameters.

The data sets of the 2013 Data Fusion Contest were used to test the effectiveness of the proposed algorithm. The experimental results show that ODF-ADE cannot only make full use of the advantages of LiDAR data, but it can also obtain more reasonable classification maps using the weighted voting. ODF-ADE can overcome the shortcomings of classification using data from a single sensor and can achieve good results in urban LULC classification.

In our future work, the number of classification maps and the diversity of these maps will be increased to obtain a better result. Ensemble voting will also be considered to further improve the classification result and additional information about LiDAR may be considered in the future work.

**Acknowledgments:** This work was supported by National Key Research and Development Program of China under Grant No. 2017YFB0504202, National Natural Science Foundation of China under Grant Nos. 41771385, 41622107 and 41371344, and Natural Science Foundation of Hubei Province in China under Grant No. 2016CFA029. The authors would like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the data sets used in this study, and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest.

**Author Contributions:** All the authors made significant contributions to the work. Yanfei Zhong and Qiong Cao designed the research and analyzed the results. Ji Zhao, Ailong Ma, Bei Zhao and Liangpei Zhang provided advice for the preparation and revision of the paper.

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

## References

1. Momeni, R.; Aplin, P.; Boyd, D. Mapping Complex Urban Land Cover from Spaceborne Imagery: The Influence of Spatial Resolution, Spectral Band Set and Classification Approach. *Remote Sens.* **2016**, *8*, 88. [[CrossRef](#)]
2. He, C.; Gao, B.; Huang, Q.; Ma, Q.; Dou, Y. Environmental degradation in the urban areas of China: Evidence from multi-source remote sensing data. *Remote Sens. Environ.* **2017**, *193*, 65–75. [[CrossRef](#)]
3. Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [[CrossRef](#)]
4. Camps-Valls, G.; Tuia, D.; Bruzzone, L.; Benediktsson, J.A. Advances in hyperspectral image classification. *IEEE Signal. Proc. Mag.* **2014**, *31*, 45–54. [[CrossRef](#)]
5. Li, X.; Wu, T.; Liu, K.; Li, Y.; Zhang, L. Evaluation of the Chinese Fine Spatial Resolution Hyperspectral Satellite TianGong-1 in Urban Land-Cover Classification. *Remote Sens.* **2016**, *8*, 438. [[CrossRef](#)]
6. Liu, T.; Gu, Y.; Jia, X.; Benediktsson, J.A.; Chanussot, J. Class-Specific Sparse Multiple Kernel Learning for Spectral-Spatial Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7351–7365. [[CrossRef](#)]

7. Nagne, A.D.; Dhumal, R.K.; Vibhute, A.D.; Rajendra, Y.D.; Gaikwad, S.; Kale, K.V.; Mehrotra, S.C. Performance evaluation of urban areas Land Use classification from Hyperspectral data by using Mahalanobis classifier. In Proceedings of the IEEE International Conference on Intelligent Systems and Control (ISCO), Coimbatore, Indian, 5–6 January 2017; pp. 388–392.
8. Tuia, D.; Flamary, R.; Courty, N. Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *105*, 272–285. [[CrossRef](#)]
9. Xiong, X.; Zhong, Y.; Zhang, L. Sub-pixel mapping based on a Map model with multiple shifted hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 580–593.
10. Zhong, Y.; Wang, X.; Zhao, L.; Feng, R.; Zhang, L.; Xu, Y. Blind spectral unmixing based on sparse component analysis for hyperspectral remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *119*, 49–63. [[CrossRef](#)]
11. Kim, Y. Generation of Land Cover Maps through the Fusion of Aerial Images and Airborne LiDAR Data in Urban Areas. *Remote Sens.* **2016**, *8*, 521. [[CrossRef](#)]
12. Zhang, J.; Lin, X.; Ning, X. SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas. *Remote Sens.* **2013**, *5*, 3749–3775. [[CrossRef](#)]
13. Priem, F.; Canters, F. Synergistic Use of LiDAR and APEX Hyperspectral Data for High-Resolution Urban Land Cover Mapping. *Remote Sens.* **2016**, *8*, 787. [[CrossRef](#)]
14. Liao, W.; Bellens, R.; Pizurica, A.; Gautama, S.; Philips, W. Combining feature fusion and decision fusion for classification of hyperspectral and LiDAR data. *IEEE Geosci. Remote Sens.* **2014**, 546–549. [[CrossRef](#)]
15. Ghamisi, P.; Benediktsson, J.A.; Phinn, S. Land-cover classification using both hyperspectral and LiDAR data. *Int. J. Image Data Fusion* **2015**, *6*, 1–27. [[CrossRef](#)]
16. Wang, H.; Glennie, C. Fusion of waveform LiDAR data and hyperspectral imagery for land cover classification. *ISPRS J. Photogramm. Remote Sens.* **2015**, *108*, 1–11. [[CrossRef](#)]
17. Merentitis, A.; Debes, C.; Heremans, R.; Frangiadakis, N. Automatic fusion and classification of hyperspectral and LiDAR data using random forests. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Quebec City, QC, Canada, 13–18 July 2014; pp. 1245–1248.
18. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; Kasteren, T.V.; Liao, W.; Bellens, R.; Pižurica, A.; Gautama, S.; et al. Hyperspectral and LiDAR Data Fusion: Outcome of the 2013 GRSS Data Fusion Contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2405–2418. [[CrossRef](#)]
19. Ghamisi, P.; Benediktsson, J.A.; Phinn, S. Fusion of hyperspectral and LiDAR data in classification of urban areas. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Quebec City, QC, Canada, 13–18 July 2014; pp. 181–184.
20. Huang, R.; Zhu, J. Using Random Forest to integrate LiDAR data and hyperspectral imagery for land cover classification. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, Australia, 21–26 July 2013; pp. 3978–3981.
21. Storn, R.; Price, K.V. Differential Evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *4*, 341–359. [[CrossRef](#)]
22. Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **2009**, *2*, 398–417. [[CrossRef](#)]
23. Gong, W.; Cai, Z.; King, C.X.; Li, H. Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Trans. Syst. Man Cybern. B* **2011**, *41*, 397–413. [[CrossRef](#)] [[PubMed](#)]
24. Das, S.; Konar, A.; Chakraborty, U.K. Two improved differential evolution schemes for faster global search. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Washington, DC, USA, 25–29 June 2005; pp. 991–998.
25. Das, S.; Abraham, A.; Chakraborty, U.K.; Konar, A. Differential evolution using a neighborhood-based mutation operator. *IEEE Trans. Evol. Comput.* **2009**, *13*, 526–553. [[CrossRef](#)]
26. Joshi, R.; Sanderson, A.C. Minimal representation multisensor fusion using differential evolutions. *IEEE Trans. Man Cybern. A* **1999**, *29*, 63–76. [[CrossRef](#)]
27. Chen, C.H.; Lin, C.J.; Lin, C.T. Nonlinear system control using adaptive neural fuzzy networks based on a modified differential evolution. *IEEE Trans. Syst. Man Cybern. C* **2009**, *39*, 459–473. [[CrossRef](#)]
28. Storn, R. Designing digital filters with differential evolution. In *New Ideas in Optimization*; McGraw-Hill Ltd.: London, UK, 1999; pp. 109–126.

29. Das, S.; Sil, S. Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm. *Inf. Sci.* **2010**, *180*, 1237–1256. [[CrossRef](#)]
30. Das, S.; Konar, A. Automatic image pixel clustering with an improved differential evolution. *Appl. Soft Comput.* **2009**, *9*, 226–236. [[CrossRef](#)]
31. Das, S.; Abraham, A.; Konar, A. Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Man Cybern. A* **2008**, *38*, 218–237. [[CrossRef](#)]
32. Maulik, U.; Saha, I. Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3503–3510. [[CrossRef](#)]
33. Zhong, Y.; Ma, A.; Zhang, L. An Adaptive Memetic Fuzzy Clustering Algorithm with Spatial Information for Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1235–1248. [[CrossRef](#)]
34. Ma, A.; Zhong, Y.; Zhang, L. Adaptive Multiobjective Memetic Fuzzy Clustering Algorithm for Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4202–4217. [[CrossRef](#)]
35. Zhong, Y.; Zhao, L.; Zhang, L. An adaptive differential evolution endmember extraction algorithm for hyperspectral remote sensing imagery. *IEEE Geosci Remote Sens. Lett.* **2014**, *11*, 1061–1065. [[CrossRef](#)]
36. Zhong, Y.; Zhang, L. Remote sensing image subpixel mapping based on adaptive differential evolution. *IEEE Trans. Syst. Man Cybern. B* **2012**, *42*, 1306–1329. [[CrossRef](#)] [[PubMed](#)]
37. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
38. Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*; Emanuel, P., Kunio, T., Genshiro, K., Eds.; Springer: New York, NY, USA, 1998; pp. 199–213.
39. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Hyperspectral Image Segmentation Using a New Bayesian Approach with Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3947–3960. [[CrossRef](#)]
40. Yuan, X.; Su, A.; Nie, H.; Yuan, Y.; Wang, L. Application of enhanced discrete differential evolution approach to unit commitment problem. *Energy Convers. Manag.* **2009**, *50*, 2449–2456. [[CrossRef](#)]
41. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [[CrossRef](#)]
42. Wang, L.; Pan, Q.; Suganthan, P.N.; Wang, W.; Wang, Y. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Comput. Oper. Res.* **2010**, *37*, 509–520. [[CrossRef](#)]
43. Brest, J.; Greiner, S.; Boškovi, B.; Mernik, M.; Umer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [[CrossRef](#)]
44. Price, K.; Storn, R. *Differential Evolution: A Practical Approach to Global Optimization*; Springer: Berlin/Heidelberg, Germany, 2005.
45. Zhang, L.; Zhong, Y.; Huang, B.; Li, P. Dimensionality reduction based on clonal selection for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4172–4185. [[CrossRef](#)]
46. Zhao, J.; Zhong, Y.; Zhang, L. Detail-Preserving Smoothing Classifier Based on Conditional Random Fields for High Spatial Resolution Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2440–2452. [[CrossRef](#)]
47. Zhao, J.; Zhong, Y.; Shu, H.; Zhang, L. High-Resolution Image Classification Integrating Spectral-Spatial-Location Cues by Conditional Random Fields. *IEEE Trans. Image Process.* **2016**, *25*, 4033–4045. [[CrossRef](#)] [[PubMed](#)]
48. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [[CrossRef](#)] [[PubMed](#)]

