remote sensing

MDPI

Article

Optimizing Multiple Kernel Learning for the Classification of UAV Data

Caroline M. Gevaert *, Claudio Persello and George Vosselman

Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, 7500 AE Enschede, The Netherlands; c.persello@utwente.nl (C.P.); george.vosselman@utwente.nl (G.V.)

* Correspondence: c.m.gevaert@utwente.nl; Tel.: +31-53487-4578

Academic Editors: Farid Melgani, Xiaofeng Li and Prasad S. Thenkabail

Received: 26 October 2016; Accepted: 9 December 2016; Published: 16 December 2016

Abstract: Unmanned Aerial Vehicles (UAVs) are capable of providing high-quality orthoimagery and 3D information in the form of point clouds at a relatively low cost. Their increasing popularity stresses the necessity of understanding which algorithms are especially suited for processing the data obtained from UAVs. The features that are extracted from the point cloud and imagery have different statistical characteristics and can be considered as heterogeneous, which motivates the use of Multiple Kernel Learning (MKL) for classification problems. In this paper, we illustrate the utility of applying MKL for the classification of heterogeneous features obtained from UAV data through a case study of an informal settlement in Kigali, Rwanda. Results indicate that MKL can achieve a classification accuracy of 90.6%, a 5.2% increase over a standard single-kernel Support Vector Machine (SVM). A comparison of seven MKL methods indicates that linearly-weighted kernel combinations based on simple heuristics are competitive with respect to computationally-complex, non-linear kernel combination methods. We further underline the importance of utilizing appropriate feature grouping strategies for MKL, which has not been directly addressed in the literature, and we propose a novel, automated feature grouping method that achieves a high classification accuracy for various MKL methods.

Keywords: Unmanned Aerial Vehicles (UAVs); Support Vector Machines (SVMs); Multiple Kernel Learning (MKL); informal settlements; image classification

1. Introduction

Unmanned Aerial Vehicles (UAVs) are gaining enormous popularity due to their ability of providing high-quality spatial information in a very flexible manner and at a relatively low cost. Another considerable advantage is the simultaneous acquisition of a photogrammetric point cloud (i.e., a 3D model consisting of a collection of points with X, Y, Z coordinates) and very high-resolution imagery. Due to these reasons, the use of UAVs for a wide range of applications is being analyzed, such as agriculture [1,2], forestry [3], geomorphology [4], cultural heritage [5] and damage assessment [6]. Furthermore, the potential cost savings, improved safety and prospect of enhanced analytics they provide are being increasingly recognized as a competitive advantage from a business perspective [7].

Similar to traditional aerial photogrammetry, UAV imagery is processed to obtain a dense point cloud, Digital Surface Model (DSM) and orthomosaic. In [8], the general workflow of utilizing UAVs for mapping applications is described. UAVs are generally mounted with a camera and fly over the study area to obtain individual overlapping images. Flights are planned according to the camera parameters, UAV platform characteristics and user-defined specifications regarding the desired ground sampling distance and image overlap. The acquired images are then processed using photogrammetric methods, for which semi-automatic workflows are currently implemented in various software [9]. Key tie points are identified in multiple images, and a bundle-block adjustment is applied to simultaneously identify

the camera parameters of each image, as well as the location of these tie points in 3D space. Note that this step usually requires the inclusion of external ground control points for an accurate georeferencing. Dense matching algorithms, such as patch-based [10] or semi-global [11] approaches, are then applied to obtain a more detailed point cloud. The point cloud is filtered and interpolated to obtain a DSM that provides the height information for the orthomosaic derived from the UAV images. Thus, geospatial applications making use of UAV imagery have access to the information in a point cloud, DSM and orthomosaic for subsequent classification tasks.

Much research regarding the classification of urban areas from aerial imagery still relies on features from either only the imagery or the imagery and DSM. For example, Moranduzzo et al. [12] divide the orthomosaic into tiles and use Linear Binary Pattern (LBP) texture features to propose class labels that are present in that area. Tokarczk et al. [13] use Randomized Quasi-Exhaustive (RQE) feature banks to describe texture in UAV orthomosaics for the purpose of classifying impervious surfaces. Feng et al. [14] use radiometric and Gray-Level Co-Occurrence Matrix (GLCM) texture features to identify inundated areas. The inclusion of elevation data greatly improves image classification results in urban areas [15,16]. Indeed, a comparison of building extraction methods using aerial imagery indicates that the integration of image- and DSM-based features obtains high accuracies for (large) buildings [17]. However, combining the features derived from both the imagery and the point cloud directly (rather than the DSM) has been shown to prove beneficial for classification problems in the fields of damage assessment [18] and informal settlement mapping [19].

Combining features from multiple sources or from different feature subsets pertains to the field of multi-view learning [20]. For example, in this case, point-cloud-based and image-based features could be considered as different views of a study area. Although both are obtained from the same data source (UAV images), the point-cloud represents the geometrical properties of the objects in the scene, whereas the orthoimagery contains reflectance information (i.e., color). Xu et al. [20] distinguish three types of multi-view learning: co-training, sub-space learning and Multiple Kernel Learning (MKL). Co-training generally consists of training individual models on the different views and then enforcing the consistency of model predictions of unlabeled data. However, such methods require sufficiency, i.e., that each model is independently capable of recognizing all classes. This is not always the case, for example different roof types may be differentiated based on textural features from the imagery, but not geometrically distinguishable in the point cloud. Sub-space learning uses techniques, such as Canonical Correlation Analysis (CCA), to recover the latent subspace behind multiple views. The representation of samples in this subspace can be used for applications such as dimensionality reduction and clustering. MKL can be used in combination with kernel-based analysis and classification methods. Support Vector Machine (SVM) is a successful classification algorithm that utilizes a kernel function to map training sample feature vectors into a higher dimensional space in which the data are linearly separated. As a single mapping function may not be adequate to describe features with different statistical characteristics, MKL defines multiple mapping functions (either on different groups of features or the same group of features, but using different kernel parameters). A number of studies show that MKL achieves higher classification accuracies than single-kernel SVMs [21]. For example, Gu et al. [22] demonstrate an MKL algorithm for the integration of heterogeneous features from LiDAR and multispectral satellite for an urban classification problem. Although, as opposed to the integration of LiDAR and satellite imagery, the UAV point cloud and orthoimagery are obtained from a single set of cameras and sensors, we could expect MKL to have a similar beneficial effect.

In this paper, we illustrate how the utilization of classification algorithms that are specifically tailored to the integration of heterogeneous features is more appropriate for exploiting the complementary 2D and 3D information captured by UAVs for challenging classification tasks. The objective of this paper is two-fold. Firstly, we demonstrate the importance of using classification algorithms, such as MKL, which support the integration of heterogeneous features for the classification of UAV data. Secondly, we describe various feature grouping strategies, including a novel automatic grouping strategy, and compare their performances using a number of state-of-the-art MKL algorithms.

The methods are compared through a multi-class classification task using UAV imagery of an informal settlement in Kigali, Rwanda.

2. Background

Support Vector Machines (SVMs) are robust classifiers that are particularly suited to high dimensional feature spaces and have proven to obtain high classification accuracies in remote sensing applications [23]. These discriminative classifiers identify the linear discriminant function that separates a set of n training samples $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$ representing two classes $y_i \in \{-1, +1\}$ based on their respective feature vectors $\mathbf{x}_i$ in a non-linear feature space obtained by a mapping function $\phi(\mathbf{x})$:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b, \tag{1}$$

where b is a bias term and $\mathbf{w}$ is the vector of weight coefficients, which can be obtained by solving a quadratic optimization problem defined as:

$$\begin{array}{l} \min \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i, \\ \textit{with respect to} : \mathbf{w} \in \mathbb{R}^q,\ \boldsymbol{\xi} \in \mathbb{R}_+^n,\ b \in \mathbb{R} \\ \textit{subject to} : y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i\ \forall i = 1, \ldots, n. \end{array} \tag{2}$$

where C is a regularization parameter representing the relative cost of misclassification, ξ_i represent the slack variables associated with training samples, and q is the dimensionality of the feature space obtained by $\phi(\mathbf{x})$. Rather than calculating the mapping function $\phi(\mathbf{x})$, the kernel trick can be employed to directly obtain a non-linear similarity measure between each pair of samples $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. The optimization function is then solved using the Lagrangian dual formulation as follows:

$$\begin{array}{l} \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \\ \textit{subject to } \sum_{i=1}^{n} y_i \alpha_i = 0 \textit{ and } 0 \leq \alpha_i \leq C\ \forall\, i = 1, \ldots, n \end{array} \tag{3}$$

where α_i are the Lagrangian multipliers. Various kernel functions are described in the literature, such as the common (Gaussian) Radial Basis Function (RBF) kernel:

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right) \tag{4}$$

The RBF kernel function has one parameter, σ (often replaced by $\gamma = 1/2\sigma^2$), which represents the bandwidth of the Gaussian function. The bandwidth parameter can be determined by heuristics, such as the median distance between samples [24] or cross-validation [22,25].

Intuitively, one can understand that not all features may be best represented by the same kernel parameters. Instead, Multiple Kernel Learning (MKL) utilizes P independent input kernels, which allow nonlinear relations between training samples to be described by differing kernel parameters and/or differing input feature combinations. The calculation of the similarity between each pair of training samples using different kernel functions results in P different kernel matrices $\boldsymbol{K}_m$ that are then linearly or non-linearly combined into a single kernel $\boldsymbol{K}_\eta$ for the SVM classification:

$$\boldsymbol{K}_\eta(\mathbf{x}_i, \mathbf{x}_j) = f_\eta\left(\left\{\boldsymbol{K}_m\left(\mathbf{x}_i^m, \mathbf{x}_j^m\right)\right\}_{m=1}^{P} \middle| \boldsymbol{\eta}\right) \tag{5}$$

There are a number of advantages of MKL compared to standard SVM methods. Firstly, as it allows kernel parameters to be adapted towards specific feature groups, it may enhance the class separability. Secondly, the combined kernel $\boldsymbol{K}_\eta$ can be constructed by assigning various weights to the

input kernels, thus emphasizing more relevant features. In extreme cases, certain feature kernels may be assigned a weight of zero, thus causing the MKL to act as a feature selection method. Due to these characteristics, MKL is an appropriate classification method for combining features from heterogeneous data sources.

Much of the research regarding MKL for classification focusses on the strategies that are used to combine the input kernels. For example, a fixed rule can be adopted, where each kernel is given an equal weight [26]. Alternatively, the individual kernel weights could then be determined based on similarity measures between the combined kernel and, for example, an optimal kernel ($\boldsymbol{K}_y = \mathbf{y}\mathbf{y}^T$), which perfectly partitions the classes. Niazmardi et al. [27] refer to these as two-stage algorithms, as opposed to single-stage algorithms that optimize the kernel weighting and SVM parameters simultaneously. Although the latter group of methods, including SimpleMKL [28] and Generalized MKL [29], are more sophisticated and may potentially achieve higher classification accuracies, they often imply a higher computational complexity.

In fact, a review of MKL methods [21] suggests that although MKL leads to higher classification accuracies than single-kernel methods, more complex kernel combination strategies do not always lead to better results. Rather, simple linear combination strategies seem to work well for non-linear kernels, such as RBF kernels. Gehler and Nowozin [30] reached similar conclusions, stating that "baseline methods", such as averaging or multiplying kernels, reach similar accuracies as more complex algorithms, but at a much lower computational complexity.

Considering that one of the main motivations behind MKL is the ability to adapt kernel parameters for the various features, it is surprising that little work has been done regarding how to divide features into groups so they optimally benefit from the tailored feature mappings. Various MKL studies report different grouping strategies, but none of them seem to compare a wide range of grouping strategies and compare the influence of the grouping strategies on the classification accuracy. Intuitively, such an optimal feature grouping should: (i) group features that are optimally represented by the same kernel parameters; and (ii) group features in a way that allows less or non-relevant features to be suppressed by the kernel weighting strategy. The main difficulty is that measures used to determine the latter, i.e., feature relevance, through non-linear similarity measures often depend on the former, i.e., the chosen kernel parameters. At the same time, the optimal values for these kernel parameters depend on which features are included in the group.

In practice, some studies assign each feature to a unique kernel [25]. This allows the optimal kernel parameters to be defined per feature and a feature selection to be introduced through MKL methods promoting sparsity. Alternatively, Gu et al. [22] and Yeh et al. [31] adopt a multi-scale approach, defining a range of r bandwidth parameters for each feature f out of a total of n_f input features to create a total of $r \cdot n_f$ kernels as the input for the MKL. However, such approaches may fail to describe the complex relations between features and suffer an increased computational complexity due to the presence of more kernels. Another grouping strategy depends on the origins of the image features. In this case, separate kernels are defined for spectral or spatial features, or multi-spectral and radar imagery, and have been shown to outperform assigning features to individual kernels [25]. In an effort to define the groups automatically, one could consult the literature on view construction for co-training, the first multi-view learning technique. In general, co-training seems to work well when the different views provide complementary information [20]. This is similar to the observations of Di and Crawford [32], who found that using the "maximum disagreement" approach to spectral bands performed better than uniform or random sampling for hyperspectral image classification.

However, a direct comparison of different grouping strategies for MKL using heterogeneous features is still lacking. This paper addresses this issue and proposes an automatic algorithm that extracts potential kernel parameters from the training data and performs a backward feature-selection strategy to determine which features should be included in each kernel. It thus simultaneously functions as both a feature grouping and feature selection method.

3. Materials and Methods

Multiple kernel learning can be applied to UAV data through the workflow presented in Figure 1. Based on the input data, such as point clouds and imagery, the first step consists of extracting the relevant features from the input data. This may be supplemented by a feature selection strategy if desired. The next step is to divide the features into groups to define the input kernels, which are then combined into a single kernel that is used to define the SVM classifier. Depending on which MKL method is employed, the parameters for kernel weighting and SVM may be optimized jointly or separately. In the following section, we describe the heterogeneous features utilized in this study for the classification (Section 3.1), various feature grouping strategies (Section 3.2) and the multiple kernel learning algorithms utilized to classify the data (Section 3.3).

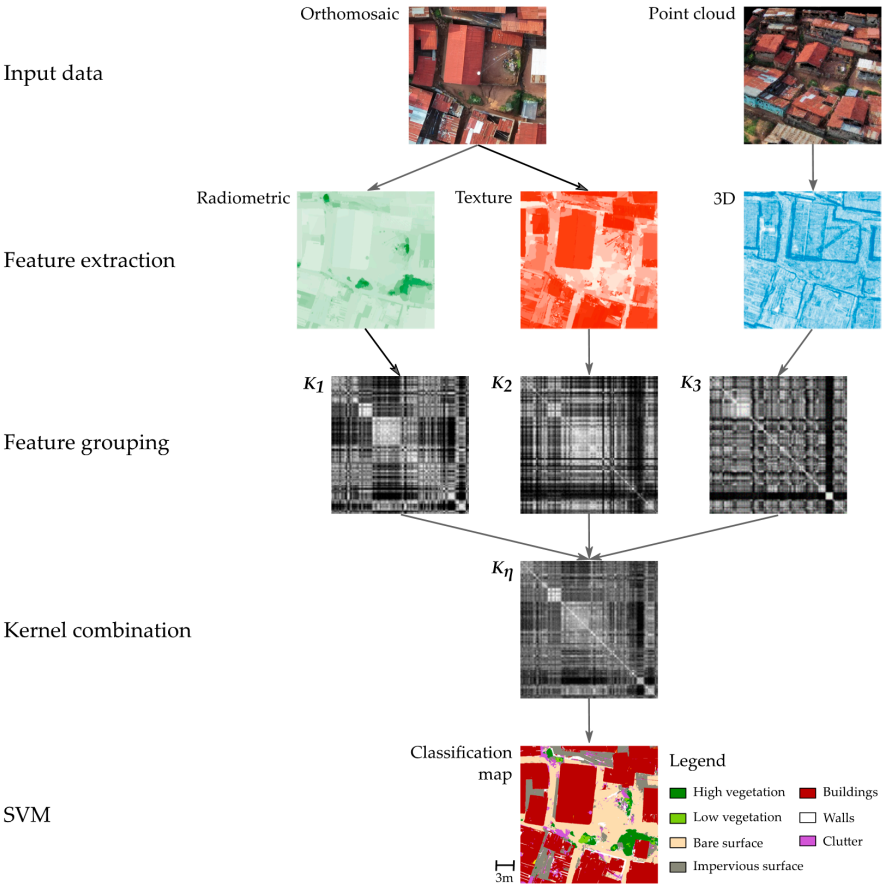


Figure 1. An illustrative example of the multiple kernel learning workflow for UAVs: first, features must be extracted from the orthomosaic and the point cloud; then, the features are grouped, and the $\boldsymbol{K}_m$ input kernels are constructed. MKL techniques are used to combine the different input kernels into the combined kernel $\boldsymbol{K}_\eta$, which is used to construct the SVM and perform the classification.

3.1. Feature Extraction from UAV Data

Four types of features were derived from the orthomosaic and point cloud: 14 image-based radiometric features, 54 image-based texture features, 22 3D features per pixel and 22 3D features averaged over image segments (Table 1). The image-based radiometric features consist of the original R, G, B color channels of the orthomosaic, their normalized values (r, g, b) and the ExG(2) vegetation index: $\mathrm{ExG}(2) = 2\mathrm{g} - \mathrm{r} - \mathrm{b}$ [33], at the pixel-level and averaged over image segments. Here, the segments were obtained through a mean shift segmentation [34] with a spatial bandwidth of 20 pixels and a spectral bandwidth of five gray values.

The image-based texture features are represented by Local Binary Pattern (LBP) features [35]. These rotationally-invariant texture features identify uniform patterns, such as edges and corners, based on a defined number of neighboring pixels (N) at a distance (R) from the center pixel. The relative presence of each $N + 2$ texture pattern in the local neighborhood can be summarized by constructing a normalized histogram for each mean shift segment, where the frequency of each bin is used as a feature.

Table 1. A list of the features extracted from the point cloud and orthomosaic in the current study. N refers to the number of features in the group.

Type of Feature	N	Source	Description	
Radiometric	14	Image	Pixel-based	Color (R, G, B) Normalized color (r, g, b) Vegetation index (ExG(2))
			Segment-based	Color (R, G, B) Normalized color (r, g, b) Vegetation index (ExG(2))
Texture	54	Image	Local Binary Patterns	$LBP_{R=1,N=8}$ $LBP_{R=2,N=16}$ $LBP_{R=3,N=24}$
3D features	22	Point cloud	Spatial binning	Points per pixel Max. height difference Height standard deviation
			Planar segments	Number of points Average residual Inclination angle Max height difference
			Local neighborhood	Linearity, planarity, planarity (2), scattering, omnivariance, anisotropy, eigenentropy, sum of eigenvalues, curvature, maximum height, range of height values, standard deviation of height values, inclination angle, sum of 2D eigenvalues, ratio of 2D eigenvalues
3D features per image segment	22	Both	Same as point cloud features, but averaged over image segments	

The third type of features consist of 3D features extracted from the point cloud: spatial binning features, planar segment features and local neighborhood features. Spatial binning features describe the number of 3D points corresponding to each 2D image pixel, as well as the maximal height difference and height standard deviation of these points. Planar segment features are obtained by applying a surface growing algorithm to the point cloud [36]. The algorithm calculates the planarity of the 10 nearest neighbors for each seed point and adds points within a radius of 1.0 m, which are within a 0.30-m threshold from the detected plane. The latter threshold is relatively high compared to the spatial resolution as in the informal settlement, there are often objects, such as rocks or other clutter, on top of roofs. This could result in non-planar objects, such as low vegetation, being considered as planar. However, such class ambiguities may be rectified through the other features included in our feature set. From each planar segment, four features were extracted: the number of points per segment, average residual to the plane, inclination angle and maximal height difference to the surrounding points. The local neighborhood features are based on the observation that the ratio between the eigenvalues of the covariance matrix of the XYZ coordinates of a point's nearest neighbors can represent the shape of the local neighborhood [37]. For example, the relative proportions between these eigenvalues may describe the local neighborhood as being planar, linear or scattered. More specifically, we consider an optimal neighborhood around each 3D point to define the covariance

matrix and extract the 3D features described in the framework presented by Weinmann et al. [38]. To assign 3D features calculated in the point cloud to 2D space, the attributes of the highest point for each pixel in the orthomosaic were assigned to the pixel in question. We thus obtain 3D features (spatial binning, planar segment and neighborhood shape) for each pixel.

The fourth and final type of features consist of averaging these pixel-based 3D features over the image segments. For a more detailed description of how the various features were extracted, the reader is referred to Gevaert et al. [19]. All feature values are normalized to a scale between 0 and 1 before feature grouping and classification.

3.2. Feature Grouping Strategies

3.2.1. Reference Grouping Strategies

After calculating the input features, each feature f in the complete set of features S ($f \in S$), where n_f indicates the total number of features, must be assigned to a group G_m, $m = 1, \ldots, P$, where each group will form an individual input kernel $\boldsymbol{K}_m$. Seven MKL grouping strategies are compared based on: (i) individual kernels; (ii) prior knowledge; (iii) random selection; (iv) feature similarity; (v) feature diversity; (vi) the kernel-based distance between samples; and (vii) a novel multi-scale Markov blanket selection scheme. In Case (i), each feature is assigned to an individual input kernel, so 112 features result in 112 input kernels $\boldsymbol{K}_m$. The prior knowledge strategy of Case (ii) consists of four kernel groups according to feature provenance: image-based radiometric features, image-based texture features, 3D features per pixel and 3D features averaged over image segments (i.e., the four types of features listed in Table 1). In Case (iii), the random selection strategy divides the features arbitrarily into a user-defined number of features groups. For Case (iv) the similarity strategy is represented by a kernel k-means clustering [39] over the feature vectors, thus grouping them into clusters that expose similar patterns in the input data. Di and Crawford [32] found that such an approach worked better than uniform or random feature grouping for multi-view active learning in hyperspectral image classification tasks.

For Case (v), diverse kernels are obtained by solving the Maximally-Diverse Grouping Problem (MDGP) through a greedy construction approach [40]. The basic idea of the approach is to iteratively select one unassigned feature, calculate the value of a disparity function considering the assignation of this feature to each feature group G_m and appoint the feature the group to which its membership would maximize the disparity. To do this, the user first defines the desired number of groups P, as well as the minimum (a) and maximum (b) number of features per group. The population of each group G_m is started by randomly selecting one of the features in the feature set S. The remaining features in the set of variables not yet assigned to a group are iteratively assigned to one of the groups G_m. One feature f_i is selected at random, and the disparity function D_{f,G_m} (6) is calculated considering its inclusion into each group, which has not yet reached the minimal number of features (i.e., $|G_m| < a$). Once each group has reached the minimal number of features, each group that has not yet reached the maximum (i.e., $|G_m| < b$) is considered. Here, the disparity function describes the normalized sum of the distances between features:

$$D_{f_i,G_m} = \frac{\Sigma_{j \in G_m} d_{ij}}{|G_m|} \tag{6}$$

where $|G_m|$ is the number of elements in group G_m and the distance d_{fi} is obtained from the Sample Distance Matrix (SDM). In this case, the SDM is an $n_f \times n_f$ matrix where the element $\mathrm{SDM}_{i,j}$ gives the ℓ_2-norm of the difference between features f_i and f_j. In other words, the disparity function is defined as the sum of the Euclidean distance between all of the features within a group over all of the samples divided by the number of features within the group.

Allowing kernel parameters to be set differently for various groups of features has been mentioned as one of the benefits of MKL. Furthermore, the mean distance between training samples is sometimes used as a heuristic for the bandwidth parameter of an RBF kernel [24]. Therefore, we also analyze

the utility of grouping features based on the distance between samples. For Case (vi), we consider using the median (a) within-class vs. (b) between-class distances, as well as (c) a combination of both distances to group the features. To implement this, an SDM is constructed for each single feature. Note that here, the SDM is a n_s *x* n_s matrix representing the distance between the samples as opposed to the feature distance matrix described in the previous paragraph. The median within-class and between-class distances for each class is obtained by finding the median of the relevant SDM entries and using this median as a feature attribute. For example, the within-class distance of class u is the median of the SDM entries of all rows and columns representing samples belonging to class u. Similarly, the between-class distance of class u is the median of all entries corresponding to rows of samples labeled as u and columns of all samples not labelled as u. Note that the median is used instead of the mean to reduce the effect of possible outliers. A classification problem with Q classes will thus result in Q feature attributes representing within-class distances, and Q attributes representing between-class distances. These are simply concatenated to Q + Q attributes for the third approach (i.e., within- and between-class distances). These feature attributes are then used as the input for a kernel k-means clustering. Thus, features that have similar (within- or between-class) sample distances and that may thus be best represented by the same bandwidth parameter will be grouped together.

3.2.2. Proposed Feature Grouping Strategy

For the final method (vii), we propose an automatic grouping strategy. Remember that the benefits of applying MKL rather than single-kernel SVM models include feature weighting and the use of different kernel parameters for various feature groups. Regarding the former, MKL allows some feature groups to be given more emphasis; in some cases, it may even assign certain kernels a weight η_m of zero, thus suppressing noisy or irrelevant feature groups. However, MKL can only suppress certain input kernels and, therefore, can only function as a feature selector if each feature is indeed assigned to a unique kernel, as in Case (i) above. This may fail to account for non-linear relationships that would be identified if features are combined in the same kernel. The second potential benefit of MKL was to allow different kernel mappings for the different feature groups. Gu et al. [22] even recommended using different bandwidths for the same feature groups, allowing similarities between samples to be recognized along multiple scales. They used pre-defined bandwidth intervals from 0.05 to 2.0 in intervals of 0.05. Yeh et al. [31] also construct multiple input kernels for each feature by selecting different bandwidth parameters, defining a 'group' as the conjunction of different kernel mappings of a single feature. MKL is applied to ensure sparsity amongst features, thus functioning as a feature selection method. Unlike [22], they select the bandwidth parameters in a data-driven manner based on the standard deviation of the distance between all training samples. Although both methods enable a multi-scale approach to define optimal feature representations using multiple bandwidth parameters, both methods pre-define which features will be grouped together in the input kernels. This could potentially lose non-linear relations between different features.

A good feature grouping strategy should therefore remove irrelevant features before kernel construction and allow features to be grouped according to optimal kernel parameters. The novel feature grouping algorithm we propose here does this by first analyzing the dataset to identify candidate bandwidth parameters, performs a feature ranking for each candidate bandwidth and restricts the features within each group according to a pre-defined threshold. The latter can be based on either defining the number of features per kernel or by defining a limit to the cumulative feature relevance. This simultaneous feature grouping and feature selection workflow consists of three steps: (i) selecting candidate bandwidth parameters; (ii) ranking the features using each parameter; and (iii) defining the cut-off criterion that selects the number of features per group (Figure 2). An additional benefit of the method is that it provides a heuristic for choosing the bandwidth parameter for the RBF kernel.

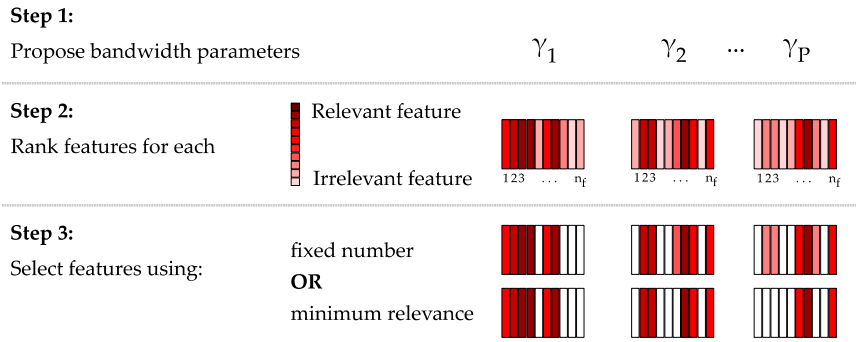
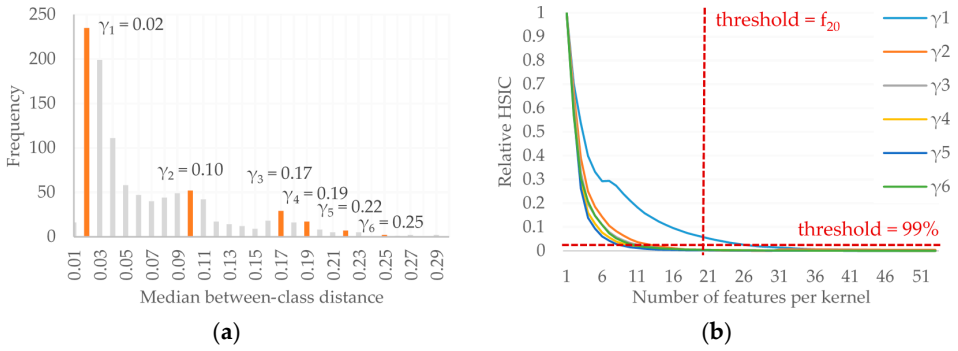


Figure 2. A graphical illustration indicating how the automatic feature grouping strategy works. Step 1 consists of proposing a number of bandwidth parameters for the RBF kernel; in Step 2, a feature ranking is done using backwards-elimination and a kernel-class separability measure with the assigned γ to determine the relative relevance of each n_f features; in Step 3, a feature set is selected for each kernel based on (i) using a fixed number of features per kernel, e.g., six in the illustrated example; or (ii) a minimum cumulative feature relevance level, which may result in different numbers of features per kernel.

In the first step, potential bandwidth parameters are identified by selecting the median between-class distances for each feature. These between-class distances are obtained by selecting all entries of the SDM that correspond to two samples from different classes. A histogram of these between-class distances is constructed, from which automatic methods can be used to select the potential bandwidth parameters (Figure 3a). Here, we simply select histogram bins associated with local maxima, i.e., which have a higher frequency than the two neighboring bins. Each of the bins corresponds to a potential bandwidth parameter; thus, different bandwidths are used for the various kernel groups and capturing data patterns at multiple scales. If the histogram does not present local peaks, other strategies could be considered, such as taking regular intervals over the possible between-class distances.



(**a**) (**b**)

Figure 3. The proposed feature selection method, first using peaks in the between-class distance histogram to identify candidate bandwidth parameters (**a**); and then using the feature ranking to determine which features to include in each group (**b**). The dashed red lines indicate the cut-off thresholds according to either a maximal number of features per kernel (f_{20}) or relative HSIC value (99%). Note that the graphs represented here do not reflect the exact data from the experiments, but have been slightly altered for illustrative purposes.

In the second step, a feature selection method based on a kernel-based class separability measure and backwards-elimination [41] is employed to determine which features to include in each kernel. The idea is to use a supervised strategy to identify the Markov blanket of the class labels [41]. That is to say, we attempt to identify which features are conditionally independent of the class labels given the remaining features. These conditionally independent features therefore do not influence the class

labels and may be removed. By using kernel class separability measures, we can identify non-linear class dependencies in the reproducing kernel Hilbert space. This is implemented by constructing a kernel using all of the features and the candidate bandwidth in question and calculating the class separability measure for an ideal kernel. One by one, the features are removed, and the measure is calculated again. The feature whose removal results in the lowest decrease in class separability is considered to be the least relevant and is removed from the set. The process is repeated until all features are ranked from most to least relevant for each candidate bandwidth. The method would work with any kernel-based class separability measure.

Finally, the user must define the cut-off metric of which features to select in each kernel based on the provided feature ranking for each bandwidth. In this case, the user can choose to either define the maximal number of features per kernel or to use a cumulative relevance metric, such as selecting the number of features that first obtain 99.9% of the maximum cumulative similarity measure provided by the feature ranking (Figure 3b). It should be noted that this feature grouping strategy also allows for a single feature to be included in various kernels. In theory, this could result in two groups containing identical features, but represented by different bandwidth parameters. The proposed methodology also potentially functions as a feature selection method, as irrelevant or redundant features are likely to be at the bottom of the feature ranking and may therefore not be included in any of the input kernels.

3.3. Kernel Weighting Strategies

3.3.1. Class Separability Measures and Ideal Kernel Definition

Various studies report that there are no large differences in different multiple kernel learning methods in terms of accuracy [21]. Furthermore, two-stage algorithms that update the combination function parameters independently from the classifier have a lower computational complexity [27]. Therefore, we hypothesize that the use of kernel class separability measures, or kernel alignment measures, between the individual kernels and an ideal kernel will provide an advantageous trade-off between computational complexity and classification accuracy. The ideal target kernel represents a case of perfect class separability, where kernel values for samples from the same class maximal and samples from different classes have minimal kernel values. Therefore, the similarity of an input kernel $\boldsymbol{K}_m$ to a target ideal kernel $\boldsymbol{K}_y$ provides an indication of the class separability given the input kernel. Such measures may be used to optimize kernel parameters or to define the proportional weights of the various feature kernels in the weighted summation. In this case, named class-separability-based MKL (CSMKSVM), the class separability measure $\mathcal{R}$ of each individual kernel $\boldsymbol{K}_m$ and an ideal kernel $\boldsymbol{K}_y$ is calculated, and then, a proportional weighting is applied as follows:

$$\eta_m = \frac{\mathcal{R}(\boldsymbol{K}_m, \boldsymbol{K}_y)}{\sum_{h=1}^{P} \mathcal{R}(\boldsymbol{K}_h, \boldsymbol{K}_y)} \; \forall m \tag{7}$$

Qiu and Lane [42] used a similar heuristic based on the kernel alignment measure [43]. Here, we compare four class separability measures found in the literature: the square Hilbert-Schmidt norm of the cross-covariance matrix (HSIC) [44,45] (8); (ii) Kernel Alignment (KA) [43] (9); (iii) Centered-Kernel Alignment (CKA) [46] (10); and (iv) Kernel Class Separability (KCS) [47] (11).

$$HSIC(\boldsymbol{K}_x, \boldsymbol{K}_y) = \frac{1}{n^2} Tr(\boldsymbol{K}_x \boldsymbol{H} \boldsymbol{K}_y \boldsymbol{H}) \tag{8}$$

$$KA(\boldsymbol{K}_x, \boldsymbol{K}_y) = \frac{\langle \boldsymbol{K}_x, \boldsymbol{K}_y \rangle_F}{\sqrt{\langle \boldsymbol{K}_x, \boldsymbol{K}_x \rangle_F \langle \boldsymbol{K}_y, \boldsymbol{K}_y \rangle_F}} \tag{9}$$

$$\begin{aligned} &CKA(\boldsymbol{K}_x, \boldsymbol{K}_y) = \frac{\langle \boldsymbol{K}_x^c, \boldsymbol{K}_y^c \rangle_F}{\sqrt{\langle \boldsymbol{K}_x^c, \boldsymbol{K}_x^c \rangle_F \langle \boldsymbol{K}_y^c, \boldsymbol{K}_y^c \rangle_F}} \\ &\text{where } \boldsymbol{K}^c = \boldsymbol{K} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{K} - \tfrac{1}{n}\boldsymbol{K}\mathbf{1}\mathbf{1}^T + \tfrac{1}{n^2}(\mathbf{1}^T\boldsymbol{K}\mathbf{1})\mathbf{1}\mathbf{1}^T \end{aligned} \tag{10}$$

$$KCS(\boldsymbol{K}_x, \boldsymbol{K}_y) = \frac{\Sigma \boldsymbol{W} - \frac{1}{n}\Sigma \boldsymbol{K}_x}{tr(\boldsymbol{K}_x) - \Sigma \boldsymbol{W}}$$
$$\text{where } \boldsymbol{W} = \frac{1}{n}\begin{pmatrix} \frac{1}{n_1}\boldsymbol{K}_{11} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{n_Q}\boldsymbol{K}_{QQ} \end{pmatrix} \tag{11}$$

where $\boldsymbol{H}_{ij} = \delta_{ij} - \left(\frac{1}{n}\right)$, δ_{ij} being the Kronecker delta and having a value of 1 if i and j adhere to the same class and 0 if they have different class labels, $Tr(.)$ indicates the trace function, n is the total number of samples, n_1 is the number of samples in the first class, n_Q is the number of samples in class Q, $\langle \boldsymbol{K}_x, \boldsymbol{K}_y \rangle_F = \sum_{i,j=1}^{n_s} \boldsymbol{K}_x(\boldsymbol{x}_i, \boldsymbol{x}_j)\boldsymbol{K}_y(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathbf{1}$ is an $n \times 1$ vector of ones. $\boldsymbol{K}_x$ is any input kernel and could therefore corresponds to either $\boldsymbol{K}_m$ or $\boldsymbol{K}_\eta$ depending on whether the class separability measure is being calculated for the input kernel or combined kernel, respectively.

3.3.2. Comparison to Other MKL Methods

Once the most adequate kernel class separability measure and ideal kernel definition have been selected, the following experiments serve to compare the proposed method to benchmark MKL methods and the influence of the various feature grouping strategies. Six benchmark Multiple Kernel SVM (MKSVM) methods are selected from the MATLAB code provided by Gönen and Alpaydin [21] (https://users.ics.aalto.fi/gonen/) and compared to the kernel Class-Separability method (CSMKSVM) described previously. They consist of methods using a Rule-Based linearly-weighted combination of kernels (RBMKSVM), Alignment-Based methods based on the similarity of the weighted summation kernel and an ideal kernel (ABMKSVM) and methods that initiate a linearly (Group Lasso-based MKL (GLMKSVM) and SimpleMKL) or nonlinearly (Generalized MKL (GMKSVM) and Non-Linear MKL (NLMKSVM)) combined kernel and use the dual formation parameters to iteratively update the weight.

The first, RBMKSVM, is a fixed-rule method in which each kernel is given an equal weight $1/P$, and the resulting combined kernel is therefore simply the mean of the input kernels. This is followed by CSMKSVM, where the weights are defined by the proportional class separability measure as described in the previous section. The second reference method, ABMKSVM, forgoes the use of a class separability measure, but rather optimizes the difference between the combined kernel and ideal kernel directly. This optimization problem can be solved as follows:

$$\begin{aligned} &minimize \sum_{m=1}^{P}\sum_{h=1}^{P} \eta_m \eta_h \langle \boldsymbol{K}_m, \boldsymbol{K}_h \rangle_F - 2\sum_{m=1}^{P} \eta_m \langle \boldsymbol{K}_m, \boldsymbol{K}_y \rangle_F, \\ &with\ respect\ to\ \eta \in \mathbb{R}_+^P \quad subject\ to\ \Sigma_{m=1}^{P}\eta_m = 1. \end{aligned} \tag{12}$$

Other methods use the SVM cost term, rather than the distance to an ideal kernel, to update the weights. This can be done by initiating the kernel weights $\boldsymbol{\eta}$ to obtain a single combined kernel and performing the SVM on this kernel. The results of the SVM are then used to update the kernel weights. For example, recognizing the similarity between the MKL formulation and group lasso [48], Xu et al. [49] update the kernel weights according to the ℓ_p-norm. For GLMKSVM, we use the ℓ_i-norm, which results in using (13) to update the kernel weights.

$$\eta_m = \frac{\|\mathbf{w}_m\|_2}{\Sigma_{h=1}^{P}\|\mathbf{w}_h\|_2} \tag{13}$$

$$\|\mathbf{w}\|_2^2 = \eta_m^2 \Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{K}_m(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

Similarly, SimpleMKL [28] uses a gradient decent on the SVM objective value to iteratively update the kernel weights. The combined kernel is initiated as a linear summation where the weight of each

kernel is defined as $1/P$. The dual formulation of the MKL SVM is solved (15), and the weights $\boldsymbol{\eta}$ are optimized using the gradient function provided in (16).

$$maximize\ J(n) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{K}_\eta \left(\mathbf{x}_i, \mathbf{x}_j \right) \tag{15}$$

$$\frac{\delta J(\eta)}{\delta \eta_m} = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{K}_{\boldsymbol{\eta}} \left(\mathbf{x}_i, \mathbf{x}_j \right) \ \forall m \tag{16}$$

Varma and Babu [29] use a similar gradient descent method for updating the weights (17), but perform a nonlinear combination of kernels (18), rather than a weighted summation of kernels, as in SimpleMKL.

$$\frac{\delta J(\eta)}{\delta \eta_m} = \frac{\delta r(\eta)}{\delta \eta_m} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \frac{\delta \boldsymbol{K}_\eta \left(\mathbf{x}_i, \mathbf{x}_j \right)}{\delta \eta_m} \ \forall m \tag{17}$$

$$\boldsymbol{K}_\eta^P \left(\mathbf{x}_i, \mathbf{x}_j \right) = \exp \left(\sum_{m=1}^{D} -\eta_m \left(\mathbf{x}_i[m] - \mathbf{x}_j[m] \right)^2 \right) \tag{18}$$

Here, the regularization function r(·) is defined as $1/2 \left(\boldsymbol{\eta} - \frac{1}{P} \right)^T \left(\boldsymbol{\eta} - \frac{1}{P} \right)$. NLMKSVM also presents a non-linear combined kernel, namely the quadratic kernel presented in (19); where the weight optimization is defined as a min-max problem [50] (20) and the weights defined as a ℓ_1-norm bounded set $\mathcal{M}$ (21) with $\eta_0 = 0$ and $\Lambda = 1$ in the present implementation.

$$\boldsymbol{K}_\eta \left(\mathbf{x}_i, \mathbf{x}_j \right) = \sum_{m=1}^{P} \sum_{h=1}^{P} \eta_m \eta_h \boldsymbol{K}_m \left(\mathbf{x}_i^m, \mathbf{x}_j^m \right) \boldsymbol{K}_h \left(\mathbf{x}_i^h, \mathbf{x}_j^h \right) \tag{19}$$

$$\min_{\eta \in \mathcal{M}} \max_{\alpha \in \mathbb{R}^n} -\alpha^T \left(\boldsymbol{K}_\eta + \lambda \boldsymbol{I} \right) \alpha + 2 \mathbf{y}^T \alpha \tag{20}$$

$$\mathcal{M} = \left\{ \boldsymbol{\eta} : \boldsymbol{\eta} \in \mathbb{R}_+^P,\ \| \boldsymbol{\eta} - \eta_0 \|_1 \leq \Lambda \right\} \tag{21}$$

3.4. Experimental Set-Up

Remote sensing is a valuable tool for providing information regarding the physical status of informal settlements, or slums. Although many studies make use of satellite imagery, even sub-meter imagery may not be sufficient to distinguish between different objects, such as buildings, and to identify their attributes [51]. This motivates the use of UAVs, which are capable of providing images at a higher spatial resolution, thus enabling improved detection and characterization of objects, as well as more detailed elevation information than is available from satellite imagery. The flexible acquisition capabilities also facilitate the acquisition of recurrent imagery to monitor project implementation, especially in the context of slum upgrading projects. These are some of the incentives that motivate the use of UAVs for informal settlement mapping.

The UAV dataset used for the experiments consists of a point-cloud and RGB orthomosaic of an informal settlement in Kigali, Rwanda, which was acquired using a DJI Phantom 2 Vision + quadcopter in 2015. The UAV acquired images with the standard 14 megapixel fish-eye camera (110° FOV) at an approximate 90% forward- and 70% side-lap. The images were processed using the commercial software Pix4D Mapper (Version 2.0.104). The point cloud densification was performed using the 'Low (Fast)' processing option, a matching window size of 7 × 7 pixels and only matching points that are present in at least four images. The DSM was constructed using the Inverse Distance Weighting (IDW) interpolation, with the noise filtering and surface smoothing options enabled. The resulting 8-bit orthomosaic has a spatial resolution of 3 cm. The average density of the

utilized point clouds is 1031 points per m^2. This density depends on the data processing parameters, as well as the characteristics of the land cover type. For this application, the point density ranges between 796 and 1843 points per m^2 according to the land cover (see Table 2).

Table 2. Number of labelled pixels and point cloud density for each thematic class.

	Average Point Cloud Density (Points per m^2)
Roof Type I (R1)	1843
Roof Type II (R2)	994
Roof Type III (R3)	796
High Vegetation (HV)	1367
Low Vegetation (LV)	951
Bare Surface (BS)	946
Impervious Surface (IS)	970
Walls (W)	1164
Lamp posts (L)	1561
Clutter (C)	1157
Total	1031

The study area itself is characterized by small, irregular buildings, narrow footpaths and a steep topography. Ten thematic classes are defined for the classification problem: three different types of building roofs (corrugated iron sheets, galvanized iron with a tile pattern and galvanized iron with a trapezoidal pattern), high vegetation, low vegetation, bare surfaces, impervious surface, lamp posts, free-standing walls and clutter. The latter class may consist of, for example, laundry hung out to dry, the accumulation of solid waste on the streets, passing cars and pedestrians. Reference data were defined by visual interpretation and manually labelling pixels in the orthomosaic (based on the results of the over-segmentation and manually adjusting segment boundaries if necessary).

Ten sets of training data (n = 2000) were extracted from the Kigali dataset. Five sets followed an equal sampling strategy (n_c = 200), and five sets followed a stratified sampling strategy, which allows an analysis to be made regarding the sensitivity of kernel class separability measures to unequal class sizes. A set of 5000 samples was extracted for testing. The first set of experiments (Experiment I.A. and Experiment I.B.) compared the class separability measures and ideal kernel definitions using the prior knowledge (Case (ii)) feature grouping. The average Overall Accuracy (OA) for each of the folds, along with the standard deviation, is provided for the equal and stratified sampling training sets separately. In Experiment I.A. the class separability measures are used to define the optimal bandwidth parameter for each input RBF kernel $\boldsymbol{K_m}$. Experiment I.B., on the other hand, uses the class separability measure both to optimize the bandwidth parameter of $\boldsymbol{K_m}$ and to perform the proportional kernel weighting in (7) to obtain the kernel weights $\boldsymbol{\eta}$. In both cases, the search space of the bandwidth parameter was defined by first defining the bandwidth parameter as the mean intra-class ℓ_2-norm and defining a range of γ as 2^{-5}- to 2^5-times this mean bandwidth. For these experiments, three different ideal kernel definitions are compared: assigning values of 1, $1/n_c$ and $1/{n_c}^2$ to samples belonging to the same class, where n_c represents the number of samples within that specific class.

The second set of experiments analyzed both the influence of feature grouping and MKL methodology. Regarding the feature grouping strategy, the random-, similarity-, diversity- and class-difference-based methods require the user to define the number of desired kernels. For these experiments, six kernels were defined, as this is the number of kernels identified by the automatic feature grouping method. For the novel feature grouping strategy, we use the results of Experiment I to select the best class separability measure (the HSIC). Furthermore, we report the results of using two different cut-off metrics to define how many features to include in each kernel: we report the results when defining a maximum of 45 features per kernel (HSIC-f$_{45}$) and when using the 99.9% cumulative relevance cut-off per kernel (HSIC-99.9%). These thresholds were selected based on the results of the feature ranking (e.g., Figure 3b). The minimum (a) and maximum (b) number of features per group using the diverse kernel strategy were set to a = 5 and b = 70 based on experimental analyses.

MKL was performed on these feature kernels using the seven algorithms described above. Note that the grouping strategy was applied separately for each fold, so the feature groups will not by definition be the same for each training set. Once the groups were identified, the same feature kernels were used as input for each MKL method.

The methods were again compared by computing the mean overall accuracy over each of the 10 folds with reference to the same 5000 sample test set. The error matrix of the CSMKSVM method using the HSIC-f_{45} feature grouping strategy is also presented, as well as the correctness (22) and completeness (23) for each of the 10 thematic classes.

$$Correctness = TP/(TP + FP) \tag{22}$$

$$Completeness = TP/(TP + FN) \tag{23}$$

where TP indicates the number of true positives per class, FP is the number of false positives and FN is the number of false negatives.

Furthermore, the MKL methods were compared to two baseline classifiers: a standard SVM classifier, implemented in LibSVM [52], where all features are combined in a single kernel, and a random forest classifier. For the SVM, RBF bandwidth parameter γ was defined as described previously, and the regularization parameter C was optimized through a 5-fold cross-validation between 2^{-5} and 2^{15}. Regarding the random forest classifier, the number of trees was optimized between 100 and 1500 in steps of 100.

In a final step, we provide classification maps of 30 $\times$ 30 m subsets of the Kigali dataset. Similar to the other experiments, 2000 labelled pixels were extracted from ten tiles representing the different characteristics of the study area through stratified sampling. These pixels were used to construct a single-kernel SVM, and CSMKSVM using the HSIC-f_{45} feature grouping strategy. Classification maps of three of the tiles are provided to illustrate the results.

4. Results and Discussion

4.1. Class Separability Measures and Ideal Kernel Definition

Kernel class separability measures require the definition of a target kernel. If each class has a similar number of samples, i.e., equal sampling, then the value assigned to the ideal kernel for samples adhering to the same class ($y_i = y_j$) does not have to be adjusted for the class size (Tables 3 and 4). However, this is not the case when there are large differences in the number of training samples per class, which may occur in stratified sampling, which is common to the processing of remotely-sensed images. The class separability measure used to define kernel weights will target the most common class, and therefore, results can be improved when the value is normalized by the number of samples per class (Table 3).

Table 3. The overall accuracy obtained for Experiment I.A.: optimizing the bandwidth parameters γ_m for each input kernel $\boldsymbol{K_m}$ using various kernel class separability measures and ideal kernel definitions. n_c indicates the number of samples for a specified class. CKA, Centered-Kernel Alignment; KCS, Kernel Class Separability.

Value $y_i = y_j$	HSIC	KA	CKA	KCS
Equal sampling (5 folds)				
1	**89.5 ± 0.46**	89.3 ± 0.43	89.1 ± 0.66	88.8 ± 1.13
$1/n_c$	**89.5 ± 0.46**	89.3 ± 0.43	89.1 ± 0.66	88.8 ± 1.13
$1/n_c^2$	**89.5 ± 0.46**	89.3 ± 0.43	89.1 ± 0.66	88.8 ± 1.13
Stratified sampling (5 folds)				
1	86.6 ± 0.59	86.8 ± 0.53	86.9 ± 0.45	86.8 ± 0.53
$1/n_c$	86.9 ± 0.38	86.8 ± 0.54	86.7 ± 0.42	86.8 ± 0.53
$1/n_c^2$	**87.2 ± 0.45**	86.9 ± 0.54	87.1 ± 0.44	86.8 ± 0.53

When the class separability measure is used both to define the bandwidth of the RBF kernel, as well as to define the relative kernel weights, this effect is minimized, and simply assigning a value of '1' to samples of the same class appears to be adequate (Table 4). Regarding the comparison between the various class separability measures, the HSIC outperformed KA, CKA and KCS through both a higher and more stable OA for the stratified samples, although KA performed slightly better in the case of equal sampling. Due to these observations, the subsequent analyses were carried out using the HSIC class separability measure and an ideal kernel where samples adhering to the same class are assigned a value of '1' and '0' otherwise, which is used both to optimize the kernel parameters and for the proportional kernel weighting. The HSIC is therefore also used as the kernel-based class separability measure for the proposed feature grouping strategy in the next experiments.

Table 4. The overall accuracy obtained for Experiment I.B.: optimizing both the bandwidth parameters γ_m for each input kernel $\boldsymbol{K_m}$ and the relative kernel weights $\boldsymbol{\eta}$ using various kernel class separability measures and ideal kernel definitions. n_c indicates the number of samples for a specified class.

Value $y_i = y_j$	HSIC	KA	CKA	KCS
Equal sampling (5 folds)				
1	90.3 ± 0.40	**90.6 ± 0.53**	89.2 ± 0.81	86.7 ± 0.67
$1/n_c$	90.3 ± 0.41	**90.6 ± 0.53**	89.2 ± 0.81	86.7 ± 0.67
$1/n_c^2$	90.3 ± 0.41	**90.6 ± 0.53**	89.2 ± 0.81	86.7 ± 0.67
Stratified sampling (5 folds)				
1	**87.2 ± 0.38**	82.7 ± 1.06	86.1 ± 0.48	80.8 ± 0.71
$1/n_c$	87.0 ± 0.81	84.1 ± 0.75	85.3 ± 0.52	80.8 ± 0.71
$1/n_c^2$	87.0 ± 0.57	86.0 ± 0.71	84.2 ± 0.86	80.8 ± 0.71

4.2. Comparison of Feature Grouping and Kernel Weighting Strategies

The first observation from the results of the various feature grouping and kernel weighting strategies (Table 5) is that almost all MKL methods perform better than a standard SVM where all features are described by a single kernel, which achieves an accuracy of 85.4%. Only some of the rule-based mean kernel weighting (ABMKSVM) classification results and GLMKSVM using individual features perform worse than the standard SVM. Furthermore, we see that most of the MKL implementations perform better than the random forest classifier, which has an average accuracy of 86.5%. A McNemar test with the continuity correction [53] indicates that the improved classification accuracy of the HSIC-f_{45} CSMKSVM method is significant compared to both the results of the single-kernel SVM (p-value of 0.0021) and random forest classification (p-value of 0.0032).

Regarding feature grouping strategies, the HSIC grouping strategy proposed in this paper obtains the highest accuracy for most MKL algorithms, where all methods except ABMKSVM achieve

an accuracy above 90%. Furthermore, the results suggest that the high accuracy is more stable than other grouping methods. Both stopping criteria (either by selecting a fixed number of features per kernel or thresholding the cumulative cost function) have a similar performance. The OA is also quite robust to the cut-off metrics. Selecting 45 features (HSIC-f_{45}) obtains the highest accuracy of 90.6% when combined with the CSMKSVM method, though using 30 or 60 features only lowered this accuracy by 0.1% and 0.2%, respectively. Similarly, the 90.5% accuracy achieved with HSIC-99.9% was only 0.2% higher than the accuracy obtained by HSIC-99.7%. Further analysis of the two methods indicated that a feature selection was indeed performed. HSIC-f_{45} selected an average of 78 features out of the 107 per fold, and HSIC-99.9% selected an average of 70 features per fold. This could advocate thresholding the cumulative feature relevance rather than fixing the number of features per kernel, as it is more suited in automatic workflows and uses a lower number of features while achieving a similar accuracy. The grouping strategy utilizing prior knowledge (i.e., feature provenance) also performs well for the CSMKSVM and NLMKSVM methods. The individual kernel grouping strategy works well for the NLMKSVM method. This is not entirely surprising, as NLMKSVM is a nonlinear kernel combination method and may therefore mimic the nonlinear similarity, which is achieved when various features are grouped into an input kernel through a non-linear mapping function.

Table 5. Overall accuracy (averaged over the 10 folds) using various feature grouping and kernel weighting strategies. p indicates the number of feature groups form which the $\boldsymbol{K_m}$ input kernels are obtained. All combinations resulting in an OA above 90% are marked in bold. "HSIC grouping" refers to the proposed feature grouping strategy.

Feature Grouping Strategy	p	MKL Method [1]						
		AB	**CS**	**GL**	**G**	**NL**	**RB**	**S**
Reference MKL feature grouping strategies								
Individual	107	81.2 ± 3.1	89.5 ± 1.4	80.7 ± 2.8	89.6 ± 1.4	**92.0** ± 1.0	89.3 ± 1.5	89.6 ± 1.4
Prior knowledge	4	88.2 ± 2.9	**90.2** ± 1.4	89.9 ± 1.7	89.9 ± 1.7	**90.7** ± 1.5	89.8 ± 1.6	89.9 ± 1.7
Random	6	83.6 ± 2.6	87.6 ± 2.0	87.3 ± 2.4	87.3 ± 2.5	87.5 ± 2.0	87.3 ± 2.2	87.3 ± 2.5
Similarity	6	77.8 ± 6.0	87.3 ± 1.9	87.5 ± 1.9	87.6 ± 1.9	86.9 ± 2.2	86.6 ± 2.1	87.5 ± 1.9
Diversity	6	82.2 ± 4.0	86.9 ± 2.0	86.9 ± 1.9	86.9 ± 1.9	87.1 ± 2.0	86.9 ± 1.9	86.9 ± 1.9
Between-class sample distance	6	83.7 ± 2.8	88.0 ± 1.7	87.9 ± 1.9	87.9 ± 1.8	88.5 ± 1.9	87.8 ± 1.8	87.9 ± 1.8
Within-class sample distance	6	81.5 ± 6.9	87.9 ± 2.2	87.5 ± 2.3	87.5 ± 2.3	88.1 ± 2.1	87.5 ± 2.2	87.4 ± 2.3
Between + within-class distance	6	81.6 ± 4.0	87.6 ± 2.0	87.5 ± 1.8	87.5 ± 1.9	88.2 ± 2.1	87.4 ± 2.0	87.5 ± 1.9
Proposed MKL grouping strategy								
HSIC-f_{30}	6	89.1 ± 2.0	**90.5** ± 1.5	**90.3** ± 1.6	**90.3** ± 1.6	**90.3** ± 1.6	**90.5** ± 1.5	**90.3** ± 1.6
HSIC-f_{45}	6	89.5 ± 1.6	**90.6** ± 1.5	**90.5** ± 1.6	**90.4** ± 1.7	**90.2** ± 1.7	**90.5** ± 1.5	**90.4** ± 1.7
HSIC-f_{60}	6	89.6 ± 1.7	**90.4** ± 1.7	**90.4** ± 1.7	**90.4** ± 1.7	**90.1** ± 1.7	**90.4** ± 1.6	**90.4** ± 1.7
HSIC-99.7%	6	88.8 ± 1.8	**90.3** ± 1.6	**90.0** ± 1.8	**90.1** ± 1.7	**90.1** ± 1.7	**90.1** ± 1.7	**90.2** ± 1.6
HSIC-99.9%	6	89.3 ± 1.8	**90.5** ± 1.5	**90.3** ± 1.7	**90.3** ± 1.7	**90.3** ± 1.7	**90.5** ± 1.6	**90.3** ± 1.7
Reference classification strategies								
Single-kernel SVM	1				85.4 ± 2.3			
Random forest	-				86.5 ± 1.9			

[1] AB = ABMKSVM, CS = CSMKSVM, GL = GLMKSVM, G = GMKSVM, NL = NLMKSVM, RB = RBMKSVM, S = SimpleMKL.

Similar to the results of previous studies (e.g., [21]), we observe a similar performance between the results obtained by the various MKL algorithms to combine the kernels. In this case, simply taking the mean of the input kernels (ABMKSVM) consistently performs worse than the other MKL algorithms. However, there is not one single algorithm that consistently outperforms the others. For the prior knowledge-based feature grouping strategy, the best OAs are achieved by the proposed proportional

HSIC-weighting measure (CSMKSVM) with 90.2% and the nonlinear NLMKSVM method with 90.7%. Regarding the proposed feature grouping strategy, CSMKSVM also obtains slightly better results than the other MKL methods at 90.6% when utilizing 45 features per kernel.

The ability of the selected features to distinguish between the different land cover classes will also depend on the study area. For example, vegetation will be more difficult to distinguish for study areas in which it is not always green (for example, the leaf-off season in temperate climates, ripening agricultural crops or arid climates). It is possible that some of the 3D features will capture the geometric traits of vegetation in these situations. However, the extent to which this is possible will greatly depend on the characteristics of the study area. Furthermore, the UAV flight parameters and data processing options will influence the suitability of the 3D features. Low texture, pixel saturation or mismatch in the scale of objects in the UAV images may cause artefacts in the point cloud, such as irregular elevation values. This, in turn, influences the quality of the 3D features, such as planar segments. Similarly, the (textural) characteristics of different land cover types will influence the point cloud density and affect the suitability of 3D features. For a more detailed analysis of the interaction between different features from UAV images and why these features were selected, the reader is referred to [19].

A visual analysis of the results is presented in Figure 4, which compares the classification maps of a standard SVM and the proposed HSIC-f_{45} CSMKSVM method. The results indicate the standard single-kernel SVM is noisier than the MKL methods. Although MKL performs better than the standard SVM method, there are still difficulties in distinguishing between bare versus impervious surfaces, surfaces versus clutter, building roofs versus clutter and building roofs versus walls (Table 6).

Figure 4. Sample classification results of three image tiles, with the input RGB tile in the first row (**a**–**c**); followed by the classification results using a standard single-kernel SVM (**d**–**f**); the classification results using the proposed CSMKSVM measure and the HSIC-f_{45} feature grouping strategy (**g**–**i**); and the reference classification data (**j**–**l**).

Table 6. Error matrix of the HSIC-f_{45} CSMKSVM method; numbers indicate the total number of pixels over the 10 folds. The final column provides the completeness (Comp.) of each class, and the final row provides the correctness (Corr.). R1, R2 and R3 correspond to 3 types of roof materials; HV = high vegetation, LV = low vegetation, BS = bare surface, IS = impervious surface, W = wall structures, L = lamp posts, C = clutter.

		Predicted Class Label										**Comp. (%)**
		R1	**R2**	**R3**	**HV**	**LV**	**BS**	**IS**	**W**	**L**	**C**	
Reference class label	**R1**	*2048*	55	0	0	0	0	0	3	0	24	96.2
	R2	215	*19,968*	7	29	3	293	299	321	0	405	92.7
	R3	0	21	*1759*	9	0	5	10	9	0	7	96.7
	HV	2	10	0	*3351*	192	16	2	17	0	30	92.6
	LV	2	8	0	67	*1595*	7	0	0	0	1	94.9
	BS	0	69	10	66	19	*7346*	403	144	1	182	89.2
	IS	9	92	26	14	17	453	*5738*	228	15	198	84.5
	W	1	27	1	11	0	69	32	*1086*	0	73	83.5
	L	0	0	0	0	0	0	0	0	*1020*	0	100
	C	16	69	0	24	6	139	105	129	1	*1371*	73.7
Corr. (%)		89.3	98.3	97.6	93.8	87.1	88.2	87.1	56.1	98.4	59.8	OA = 90.6

5. Conclusions

In this paper, we demonstrate the suitability of MKL as a classification method for integrating heterogeneous features obtained from UAV data. Utilizing a novel feature grouping strategy and a simple heuristic for weighting the individual input kernels (CSMKSVM), we are able to obtain a classification accuracy of 90.6%, an increase of 5.2% over a standard SVM implementation and 4.1% over a random forest classification model. These improvements are statistically significant with a p-value <0.005, which indicates strong evidence as standard tests use confidence levels of 0.05 or 0.01 to indicate significant differences. A series of experiments reinforces observations by other researchers that complex kernel weighting strategies do not seem to perform significantly better than simple heuristics, such as a proportional weighting based on the HSIC class separability measure.

Furthermore, we observe that much of the literature on MKL classification has focused on ways to weigh the kernels, but not how to group the features appropriately. Experiments demonstrate the importance of the latter to effectively apply MKL. In this application, satisfactory results are obtained when grouping features based on their provenance (i.e., radiometric, texture or 3D features). A novel, automated grouping strategy is also proposed, which consistently obtains high classification accuracies for all seven MKL methods that were tested here. Furthermore, for most MKL methods, the proposed feature grouping strategy performed better than when using individual kernels for each feature. This underlines the importance of proper feature grouping, which not only produces a high and stable overall accuracy, but also reduces the number of input kernels for the MKL and, thus, reduces the computational complexity. These observations support a deeper understanding of MKL for classification tasks. Future applications of classification tasks with heterogeneous features are recommended to start by grouping features according to the proposed automated method and to use CSMKSVM to weight the input kernels for the SVM classification. Finally, this manuscript demonstrates that features extracted from point clouds and orthoimagery derived from UAVs are suitable for land cover classification. Additional research would be needed to analyze to what degree the features are sensitive to the type of UAV, flight parameters and algorithms utilized to produce the point clouds and orthoimagery.

Acknowledgments: The authors would like to thank Mehmet Gönen for making the MATLAB implementation of the various MKL methods freely available for research purposes.

Author Contributions: Caroline Gevaert and Claudio Persello conceived of and designed the experiments and analyzed the data. Caroline Gevaert performed the experiments and wrote the majority of the paper.

Claudio Persello and George Vosselman contributed conceptual ideas, supported the development of the paper and revised the paper over several rounds.

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

References

1. Zhang, C.; Kovacs, J.M. The application of small unmanned aerial systems for precision agriculture: A review. *Precis. Agric.* **2012**, *13*, 693–712. [CrossRef]
2. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of Spectral-Temporal Response Surfaces by Combining Multispectral Satellite and Hyperspectral UAV Imagery for Precision Agriculture Applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
3. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR system with application to forest inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [CrossRef]
4. Tarolli, P. High-resolution topography for understanding Earth surface processes: Opportunities and challenges. *Geomorphology* **2014**, *216*, 295–312. [CrossRef]
5. Remondino, F.; Campana, S. *3D Recording and Modelling in Archaeology and Cultural Heritage: Theory and Best Practices*; BAR International Series; British Archaeological Reports: Oxford, UK, 2014; Volume 2598.
6. Vetrivel, A.; Gerke, M.; Kerle, N.; Vosselman, G. Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 61–78. [CrossRef]
7. Thibault, G.; Aoude, G. Harvard Buisness Review. Available online: https://www.hbrsubscribe.org/?ploc=9001624 (accessed on 12 December 2016).
8. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
9. Sona, G.; Pinto, L.; Pagliari, D.; Passoni, D.; Gini, R. Experimental analysis of different software packages for orientation and digital surface modelling from UAV images. *Earth Sci. Inform.* **2014**, *7*, 97–107. [CrossRef]
10. Furukawa, Y.; Ponce, J. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1362–1376. [CrossRef] [PubMed]
11. Hirschmüller, H. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 328–341. [CrossRef] [PubMed]
12. Moranduzzo, T.; Melgani, F.; Mekhalfi, M.L.; Bazi, Y.; Alajlan, N. Multiclass Coarse Analysis for UAV Imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6394–6406. [CrossRef]
13. Tokarczyk, P.; Leitao, J.P.; Rieckermann, J.; Schindler, K.; Blumensaat, F. High-quality observation of surface imperviousness for urban runoff modelling using UAV imagery. *Hydrol. Earth Syst. Sci.* **2015**, *19*, 4215–4228. [CrossRef]
14. Feng, Q.; Liu, J.; Gong, J. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier-A case of yuyao, China. *Water* **2015**, *7*, 1437–1455. [CrossRef]
15. Hartfield, K.A.; Landau, K.I.; van Leeuwen, W.J. D. Fusion of High Resolution Aerial Multispectral and LiDAR Data: Land Cover in the Context of Urban Mosquito Habitat. *Remote Sens.* **2011**, *3*, 2364–2383. [CrossRef]
16. Longbotham, N.; Chaapel, C.; Bleiler, L.; Padwick, C.; Emery, W.J.; Pacifici, F. Very High Resolution Multiangle Urban Classification Analysis. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1155–1170. [CrossRef]
17. Rottensteiner, F.; Sohn, G.; Gerke, M.; Wegner, J.D.; Breitkopf, U.; Jung, J. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 256–271. [CrossRef]
18. Vetrivel, A.; Gerke, M.; Kerle, N.; Vosselman, G. Segmentation of UAV-based images incorporating 3D point cloud information. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *40*, 261–268. [CrossRef]
19. Gevaert, C.M.; Persello, C.; Sliuzas, R.; Vosselman, G. Informal settlement classification using point-cloud and image-based features from UAV data. *ISPRS J. Photogramm. Remote Sens.* **2016**, in press. [CrossRef]
20. Xu, C.; Tao, D.; Xu, C. A Survey on Multi-view Learning. *arXiv* **2013**, *36*, arXiv:1304.5634.
21. Gönen, M.; Alpaydın, E. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2211–2268.
22. Gu, Y.; Wang, Q.; Jia, X.; Benediktsson, J.A. A novel MKL model of integrating LIDAR data and MSI for urban area classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5312–5326.

23. Bruzzone, L.; Persello, C. Approaches based on Support Vector Machines to classification of remote sensing data. In *Handbook of Pattern Recognition and Computer Vision*; World Scientific: London, UK, 2010; pp. 329–352.
24. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A.J. A kernel method for the two-sample-problem. *Adv. Neural Inf. Process. Syst.* **2006**, *19*, 513–520.
25. Tuia, D.; Camps-Valls, G.; Matasci, G.; Kanevski, M. Learning relevant image features with multiple-kernel classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3780–3791. [CrossRef]
26. Pavlidis, P.; Weston, J.; Cai, J.; Grundy, W.N. Gene functional classification from heterogeneous data. In Proceedings of the Fifth Annual International Conferences on Compututational Molecular Biology (RECOMB01), Montreal, QC, Canada, 22–25 April 2001.
27. Niazmardi, S.; Demir, B.; Bruzzone, L.; Safari, A.; Homayouni, S. A Comparative Study on Multiple Kernel Learning for Remote Sensing Image Classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016.
28. Rakotomamonjy, A.; Bach, F.R.; Canu, S.; Grandvalet, Y. SimpleMKL. *J. Mach. Learn. Res.* **2008**, *9*, 2491–2521.
29. Varma, M.; Babu, B.R. More generality in efficient multiple kernel learning. In Proceedings of the 26th Annual Internationl Conference Machine Learning (ICML'09), Montreal, QC, Canada, 14–18 June 2009; pp. 1–8.
30. Gehler, P.; Nowozin, S. On feature combination for multiclass object classification. In Proceedings of the 12th IEEE International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 221–228.
31. Yeh, Y.; Lin, T.; Chung, Y.; Wang, Y.F. A Novel Multiple Kernel Learning Framework for Heterogeneous Feature Fusion and Variable Selection. *IEEE Trans. Multimed.* **2012**, *14*, 563–574.
32. Di, W.; Crawford, M.M. View generation for multiview maximum disagreement based active learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1942–1954. [CrossRef]
33. Woebbecke, D.M.; Meyer, G.E.; Von Bargen, K.; Mortensen, D.A. Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. *Trans. ASAE* **1995**, *38*, 259–269. [CrossRef]
34. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
35. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
36. Vosselman, G. Automated planimetric quality control in high accuracy airborne laser scanning surveys. *ISPRS J. Photogramm. Remote Sens.* **2012**, *74*, 90–100. [CrossRef]
37. Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality Based Scale Selection in 3D Lidar Point Clouds. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *XXXVIII*, 97–102. [CrossRef]
38. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [CrossRef]
39. Schölkopf, B.; Smola, A.; Müller, K.-R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comput.* **1998**, *10*, 1299–1319. [CrossRef]
40. Gallego, M.; Laguna, M.; Martí, R.; Duarte, A. Tabu search with strategic oscillation for the maximally diverse grouping problem. *J. Oper. Res. Soc.* **2013**, *64*, 724–734. [CrossRef]
41. Strobl, E.V.; Visweswaran, S. Markov Blanket Ranking using Kernel-based Conditional Dependence Measures. In Proceedings of the NIPS 2013 Workshop on Causality: Large-scale Experiment Design and Inference of Causal Mechanisms, Lake Tahoe, NV, USA, 9 December 2014.
42. Qiu, S.; Lane, T. A framework for multiple kernel support vector regression and its applications to siRNA efficacy prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2009**, *6*, 190–199. [PubMed]
43. Cristianini, N.; Shawe-Taylor, J.; Elisseef, A.; Kandola, J. On kernel-target alignment. *Adv. Neural Inf. Process. Syst.* **2002**, *14*, 205–256.
44. Gretton, A.; Bousquet, O.; Smola, A.; Schölkopf, B. *Algorithmic Learning Theory*; Jain, S., Simon, H.U., Tomita, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 63–77.
45. Persello, C.; Bruzzone, L. Kernel-Based Domain-Invariant Feature Selection in Hyperspectral Images for Transfer Learning. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2615–2626. [CrossRef]
46. Cortes, C.; Mohri, M.; Rostamizadeh, A. Two-stage learning kernel algorithms. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–25 June 2010; pp. 239–246.

47. Ramona, M.; Richard, G.; David, B. Multiclass feature selection with kernel gram-matrix-based criteria. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1611–1623. [CrossRef] [PubMed]
48. Bach, F. Consistency of the group Lasso and multiple kernel learning. *J. Mach. Learn. Res.* **2007**, *9*, 1179–1225.
49. Xu, Z.; Jin, R.; Yang, H.; King, I.; Lyu, M.R. Simple and efficient multiple kernel learning by group lasso. In Proceedings of the Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–25 June 2010; pp. 1191–1198.
50. Cortes, C.; Mohri, M.; Rostamizadeh, A. Learning non-linear combinations of kernels. In Proceedings of the International Conference on Neural Information Proceessing System, Vancouver, BC, Canada, 7–10 December 2009; pp. 1–9.
51. Kuffer, M.; Pfeffer, K.; Sliuzas, R. Slums from Space—15 Years of Slum Mapping Using Remote Sensing. *Remote Sens.* **2016**, *8*. [CrossRef]
52. Chang, C.-C.; Lin, C.-J. LIBSVM. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
53. Foody, G.M. Thematic Map Comparison: Evaluating the Statistical Significance of Differences in Classification Accuracy. *Photogramm. Eng. Remote Sens.* **2004**, *70*, 627–633. [CrossRef]