HYPERSPECTRAL IMAGE CLASSIFICATION WITH PARTIAL LEAST SQUARE FOREST

Junshi Xia, Naoto Yokoya, Akira Iwasaki

Research Center for Advanced Science and Technology, The University of Tokyo,
153-0041 Tokyo, Japan

ABSTRACT

In the hyperspectral remote sensing community, decision forests combine the predictions of multiple decision trees (DTs) to achieve better prediction performance. Two well-known and powerful decision forests are Random Forest (RF) and Rotation Forest (RoF). In this work, a novel decision forest, called *Partial Least Square Forest* (PLSF), is proposed. In the PLSF, we adapt PLS to obtain the components for the hyperplane splitting. Moreover, the projection bootstrap technique is used to retain the full spectral bands for the selection of split in the projected space. Experimental results on three hyperspectral datasets indicated the effectiveness of the proposed PLSF because it enhances the diversity and accuracy within the ensemble when compared to RF and RoF.

Index Terms— Composite kernel learning, ensemble learning, classification, hyperspectral image.

1. INTRODUCTION

Hyperspectral sensors capture the images over narrows contiguous hundreds or thousands of bands of the electromagnetic spectrum. The detailed spectral information, as well as high spatial resolution, provides enhanced capability for thematic applications, i.e., land cover classification [1]. High dimensionality should be addressed to exploit the hyperspectral datasets. Several approaches have been proposed to tackle this issue [2, 3].

Multiple classifier systems (MCSs) or ensemble learning has proven to be effective in the classification of hyperspectral images among the currently available supervised learning methods. It is formed by combing the predictions of multiple base learners to achieve better performance than the single learner. In order to construct a strong MCS, the individual classifiers should be with high diversity and high accuracy [4].

Two decision tree (DT) ensemble methods, random forest (RF) [5] and rotation forest (RoF) [6], have attracted increasing attention in hyperspectral remote sensing community due to their great classification capability, fast out of sample prediction, and with slight parameter tuning [7, 8]. Two bagging strategies, i.e., each tree is trained on bootstrapped samples, and a subset of features is considered for the split of leaf, are used to inject the randomness into the construction of RF [5]. In the RoF, the accuracy and diversity are promoted by using principal component analysis (PCA) to extract features in several disjoint subsets for learning individual base DT classifiers [6]. The aforementioned techniques make RF and RoF widely used not only in hyperspectral data analysis but also in synthetic aperture radar (SAR) [9] and very high spatial resolution image analysis [10].

In order to further improve the accuracy and diversity within the ensemble, we propose *Partial Least Square Forest* (PLSF), which is constructed by several individual DTs that utilizes partial least square for the hyperplane splitting. Moreover, the projection bootstrap technique, which retains all spectral bands for split selection in the projected space, results in a significant increase in accuracy of diversity.

2. PARTIAL LEAST SQUARE FOREST

Let $\mathcal{S} \equiv \{1, ..., N\}$ denote a set of integers indexing the N pixels of a hyperspectral image; let $\mathcal{K} \equiv \{1, ..., K\}$ be a set of K labels; let $\mathbf{x} \equiv \{\mathbf{x}_1; ...; \mathbf{x}_N\} \in \mathbb{R}^{N \times D}$ denote an image of $D-$dimensional feature vectors; let $\mathbf{y} \equiv \{y_1, ..., y_N\} \in \mathcal{K}$ be a set of labels for the N pixels; and let $\mathcal{D} \equiv \{\mathbf{X}, \mathbf{Y}\} \equiv \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ be the training set, where n is the number of training samples. The objective of classification is to assign a label $y_i \in \mathcal{K}$ to each pixel $i \in \mathcal{S}$, based on the vector $\mathbf{x}_i$, resulting in an image of class label y_i.

As a new DT ensemble, PLSF combines individual oblique DTs, in which PLS is used to project the original data and select the best split in the projected space. In this work, we pay attention to Orthonormalized PLS (OPLS) [11, 12].

OPLS exploits the correlation between the features and the target data by combining the merits of canonical variate analysis and PLS [11,12]. Let $\{\mathbf{X}, \mathbf{Y}\} \equiv \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ be the training set, where n is the number of training samples. Let $\mathbf{Y}$ is converted to 1-of-K labels $\mathcal{Y} \in \mathbb{I}^{n \times K}$, where $\mathcal{Y}_{ik} = 1$ implies the pixel i belongs to class k and $\mathbf{C_{xy}} = \frac{1}{n}\mathbf{X}^\top \mathcal{Y}$ represent the covariance between $\mathbf{X}$ and $\mathbf{Y}$, whereas the covariance matrix of $\mathbf{X}$ is given by $\mathbf{C_{xx}} = \frac{1}{n}\mathbf{X}^\top \mathbf{X}$. $\mathbf{W} \in \mathbb{R}^{D \times d}$ is referred as the projection matrix, thus the extracted features is achieved by $\mathbf{X}' = \mathbf{XW}$. It should be noted that OPLS only extracts the projection matrix from the input data $\mathbf{X}$.

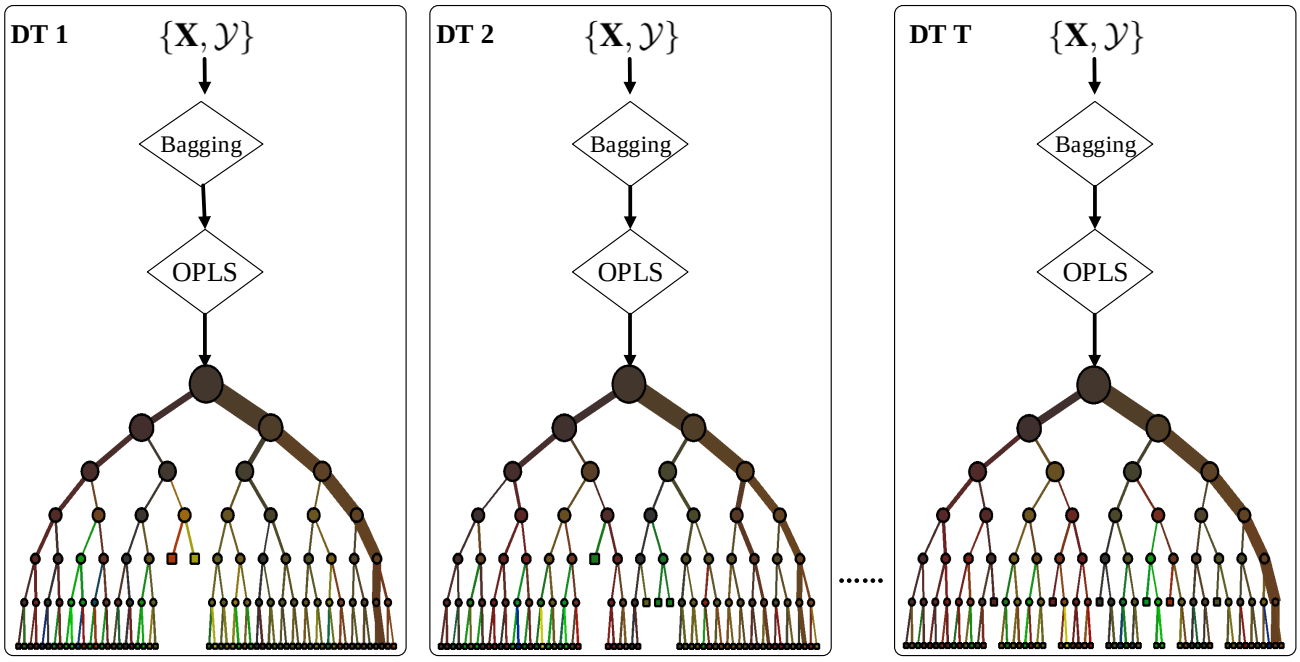


Fig. 1. PLSF

The objective of OPLS is formulated by

$$\textbf{OPLS:} \quad \begin{aligned} &\text{maximize:} && \text{Tr}\left\{\mathbf{W}^\top \mathbf{C}_{\mathbf{xy}} \mathbf{C}_{\mathbf{xy}}^\top \mathbf{W}\right\} \\ &\text{subject to:} && \mathbf{W}^\top \mathbf{C}_{\mathbf{xx}} \mathbf{W} = \mathbf{I} \end{aligned} \tag{1}$$

Let $\mathcal{T} = \{t_1, ..., t_T\}$ denote a PLSF composed of T DTs t_i. Each DT is composed of a set of split nodes $\mathcal{S} = \{s_j\}_{j \in \mathcal{J}/\partial\mathcal{J}}$ and a set of leaf nodes $\mathcal{L} = \{l_j\}_{j \in \partial\mathcal{J}}$, where $\mathcal{J}$ is a set of node indices and $\partial\mathcal{J} \subseteq \mathcal{J}$ is the subset of leaf node indices. Each split node is defined $\left\{s_j^1, s_j^2, \phi_j, z_j\right\}$, where $\left\{s_j^1, s_j^2\right\} \subseteq \mathcal{J}/j$ are the *ids* of two child nodes, ϕ_j is a weight vector used to project the input features and z_j is the point at which the splitting occurs in the projected space $\mathbf{X}\phi_j$.

As shown in Fig. 1 and Algorithm 1, the detailed training steps are summarized as follows:

- $\mathbf{X}$ is centered with zero mean and unit variance.
- the new training set $\left\{\mathbf{X}', \mathcal{Y}'\right\}$ is randomly selected from $\{\mathbf{X}, \mathcal{Y}\}$ using Bagging technique.
- the new training set is used as the input of the GROWTREE algorithm (seen in Algorithm 2).

In the GROWTREE algorithm, the new training set $\left\{\mathbf{X}', \mathcal{Y}'\right\}$ is selected from $\left\{\mathbf{X}^j_{(:,\lambda)}, \mathcal{Y}\right\}$ by using Bagging technique, and then is input to the OPLS. The projection matrix $\mathbf{W}$ obtained from OPLS is used to produce the new features. At node j, we randomly select $\lambda = \min(M, |\mathbb{F}^j|)$ features without replacement from the available feature set $\mathbb{F}^j$. The split projection vector $\mathbf{W}_j$ is set as the column of $\mathbf{W}$ in which the best split existed in the training phrase at node j. z_j is the corresponding split point at $\mathbf{X}\mathbf{W}_j$. Furthermore, the index of the

Algorithm 1 PLSF

Training phase

Input: $\{\mathbf{X}, \mathcal{Y}\}$: training samples.$D$: number of spectral bands. T: ensemble size. M: number of sampling features.

Output: CCF $\mathcal{T} = \{t_i\}_{i=1,...,T}$

1: Centering $\mathbf{X}$ with zero mean and unit variance
2: $\left\{\mathbf{X}', \mathcal{Y}'\right\} \longleftarrow$ bootstrap sampling from $\{\mathbf{X}, \mathcal{Y}\}$
3: $[., \mathcal{S}, \mathcal{L}] = \text{GROWTREE}\left(\mathbf{X}', \mathcal{Y}', D, M\right)$
4: $t_i = \{\mathcal{S}, \mathcal{L}\}$
5: **return** $\mathcal{T} = \{t_i\}_{i=1,...,T}$

Prediction phase

Input: The ensemble $\mathcal{T} = \{t_i\}_{i=1,...,T}$. A new sample $\mathbf{x}^*$.

Output: class label y^*

1: get the output ensemble using $\mathcal{T} = \{t_i\}_{i=1,...,T}$.
2: $p(y^*|\mathbf{x}^*) = \frac{1}{T}\sum_{j=1}^{T} p(y^*|\mathbf{x}^* : t_j)$
3: $y^* = \underset{k \in \{1,2,...,K\}}{\text{argmax}} \sum_{j:t_j(\mathbf{x}^*)=k}^{T} 1$

Algorithm 2 GROWTREE

Input: $\{\mathbf{X}^j, \mathcal{Y}^j\}$: training samples at node j, M: number of sampling features. $\mathbb{F}^j$: available features.

Output: sub-tree root node identifier j, sub-tree discriminant nodes Ψ, sub-tree leaf nodes Θ

set current node index j to an unique node identifier
randomly selecting λ features by taking $\min(M, |\mathbb{F}^j|)$ samples without replacement from $\mathbb{F}^j$
$\{\mathbf{X}', \mathcal{Y}'\} \longleftarrow$ bootstrap sampling from $\left\{\mathbf{X}^j_{(:,\lambda)}, \mathcal{Y}^j\right\}$
$[\mathbf{W}, \cdot] = \text{OPLS}\left(\mathbf{X}', \mathcal{Y}'\right)$
$\mathbf{W}_{(\lambda:\text{end},\lambda:\text{end})} \leftarrow \mathbf{0}$
$\mathbf{U} = \mathbf{XW}$
$[\xi, z_j, gain] = \text{FINDBESTSPLIT}(\mathbf{U})$
$\phi_j = \mathbf{W}_{(:,\xi)}$
If $\exists \mathbf{X}_{(i,:)}\phi_j \leq z_j$ **then** τ_l **else** τ_r **EndIf**
$\left[s^1_j, \mathcal{S}_l, \mathcal{L}_l\right] = \text{GROWTREE}\left(\mathbf{X}^j_{(\tau_l,:)}, \mathcal{Y}^j_{(\tau_l,:)}, |\mathbb{F}^j|, M\right)$
$\left[s^2_j, \mathcal{S}_r, \mathcal{L}_r\right] = \text{GROWTREE}\left(\mathbf{X}^j_{(\tau_r,:)}, \mathcal{Y}^j_{(\tau_r,:)}, |\mathbb{F}^j|, M\right)$
obtain $\left\{s^1_j, s^2_j, \phi_j, z_j\right\}$
return $\left[j, \{\mathcal{S}_l \cup \mathcal{S}_r\}, \{\mathcal{L}_l \cup \mathcal{L}_r\}\right]$

projection giving the best split ξ is obtained from the function FINDBESTSPLIT that is calculated by maximizing the information gain [13]. The samples are then split into left and right nodes according to the rule $\exists \mathbf{X}_{(i,:)}\phi_j \leq z_j$. The tree is grown until some stopping criterion, such as a maximum tree depth, is reached. It should be noticed that OPLS is only performed at the training stage while the split rule is directly used in the test stage.

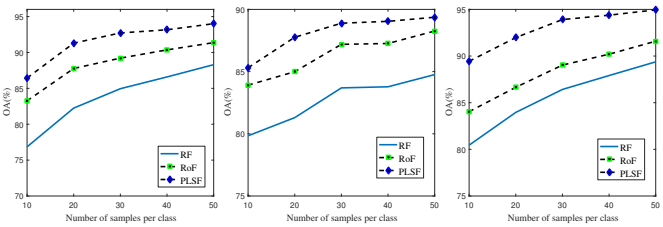
In the prediction phase, for a new sample $\mathbf{x}^*$, the final result is generated by combining the results from individual DT in the ensemble using a majority voting rule.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. Hyperspectral datasets

In this work, three hyperspectral datasets are used to assess the performance of the proposed PLSF and compared with RF and RoF classifiers.

- Kennedy space center (KSC): This dataset was acquired by the AVIRIS sensor with a ground sampling distance (GSD) of 18 m. We keep 176 bands after removing water absorption and low SNR bands. 13 classes are used in this scene.
- Salinas: This scene was also collected by the AVIRIS sensor over Salinas Valley, California with a GSD of 3.7 m. The area covered comprises 512 lines by 217 samples with 16 classes and 224 bands.
- Botswana. This dataset was obtained by the Hyperion sensor on the NASA EO-1 satellite over the Okavango Delta, Botswana. The scene consists of 14 identified classes and 145 bands.



(a) (b) (c)

Fig. 2. Overall accuracies with different sizes of training set. (a) KSC. (b) Salinas. (c) Botswana.

3.2. Results

The ensemble size (T) is set to be 20. The number of features in a subset is set to be the square root of the number of the used features and 10 for the RF and RoF, respectively. For the PLSF, $M = 9$. We randomly select the training set from the ground truth for ten Monte Carlo runs and report the mean values.

Fig. 2 shows the overall accuracies with different sizes of training set. Here, 10, 20, 30, 40 and 50 samples per class are chosen to form the training set. It can be seen that PLSF shows the best classification results in all the cases. Table. 1 presents the global and class-individual accuracies produced by PLSF as well as RF and RoF. From this table, PLSF not only provides the best global accuracies but also yields the best performance for individual classes (11 out of 13, 14 out of 16, 13 out of 14 for KSC, Salinas and Botswana).

Table 2 presents the measures such as the "OA (%)", the percentage average overall accuracies of the individual DT classifier, "AOA (%)", and the *Coincident Failure Diversity (CFD)* [14]. A stronger diversity is represented by a higher value of *CFD*. It is apparent that the best performance of PLSF is attributed to the strongest diversities for all three datasets although RoF gives the best results of AOA. It proves that the diversity is of critical importance within the ensemble in a practical viewpoint.

Fig. 3 compares the performances of the three decision forests (i.e., RF, RoF, and PLSF) by using different values of the number of features in a subset (M). As in the detailed test, PLSF tends to have better performance in small values of M. In contrast, this value is data dependent for RoF.

4. CONCLUSION

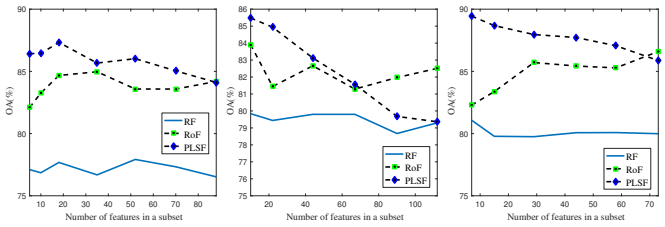
In this paper, we proposed a decision tree ensemble classifier (PLSF) for hyperspectral image classification. PLSF provides a natural way to enhance the diversity of an ensemble by using OPLS and the projection bagging technique. Experimental results support the theoretical basis as mentioned above.

Table 1. Overall, average and class-specific accuracies obtained for the Kennedy Space Center, Salinas and Botswana images when ten samples per class are used to form the training set.

Kennedy Space Center				Salinas				Botswana			
	RF	RoF	PLSF		RF	RoF	PLSF		RF	RoF	PLSF
OA	76.85	83.26	**86.44**	OA	79.84	83.91	**85.31**	OA	80.45	84.04	**89.44**
AA	72.71	79.87	**83.22**	AA	87.78	90.44	**92.47**	AA	82.34	85.62	**90.60**
κ	74.28	81.41	**84.95**	κ	77.72	82.15	**83.70**	κ	78.85	82.73	**88.57**
Scrub	80.78	**80.85**	79.84	Brocoli_green_weeds_1	98.18	96.49	**98.00**	Water	97.52	98.44	**100.00**
Willow swamp	73.87	85.43	**86.13**	Brocoli_green_weeds_2	97.60	94.65	**99.46**	Hippo grass	90.89	91.39	**98.61**
Cabbage palm hammock	84.57	83.75	**89.49**	Fallow	84.27	**92.67**	92.42	Floodplain grass1	86.65	88.29	**96.81**
Cabbage palm/oak hammock	46.11	48.29	**50.40**	Fallow_rough_plow	99.11	99.17	**99.45**	Floodplain grass2	83.72	91.95	**94.47**
Slash pine	59.50	63.17	**71.61**	Fallow_smooth	94.75	96.01	**97.56**	Reeds 1	74.72	79.18	**79.63**
oak/broadleaf hammock	46.90	**62.84**	61.22	Stubble	96.06	96.65	**99.59**	Riparian	50.33	55.92	**71.56**
Hardwood swamp	89.05	92.57	**92.67**	Celery	97.54	98.80	**99.69**	Firescar2	91.27	92.93	**97.26**
Graminoid marsh	48.84	72.62	**84.52**	Grapes_untrained	51.90	**63.97**	61.52	Island interior	87.44	89.80	**93.69**
Spartina marsh	77.92	83.37	**94.27**	Soil_vinyard_develop	93.22	97.18	**99.25**	Acacia woodlands	63.89	74.01	**84.52**
Cattail marsh	71.44	94.18	**93.22**	Corn_senesced_green_weeds	70.88	80.44	**84.30**	Acacia shrublands	75.16	79.31	**83.91**
Salt marsh	92.96	91.98	**92.20**	Lettuce_romaine_4wk	87.43	91.40	**94.70**	Acacia grasslands	86.55	86.89	**90.33**
Mud flats	75.19	79.64	**86.34**	Lettuce_romaine_5wk	94.81	98.69	**99.56**	Short mopane	89.67	93.48	**95.03**
Water	98.05	99.67	**100.00**	Lettuce_romaine_6wk	96.42	97.18	**98.25**	Mixed mopane	75.49	77.57	**84.48**
-	-	-	-	Lettuce_romaine_7wk	91.97	93.17	**93.74**	Exposed soils	99.47	**99.58**	98.11
-	-	-	-	Vinyard_untrained	62.82	63.10	**65.04**	-	-	-	
-	-	-	-	Vinyard_vertical_trellis	87.43	87.50	**97.03**	-	-	-	

Table 2. Comparison among RF, RoF and PLSF by using "OA (%)", "AOA (%)", and diversities.

Datasets		RF	RoF	PLSF
Salinas	OA (%)	79.84	83.91	**85.32**
	AOA (%)	70.83	**77.12**	69.97
	Diversity	67.37	69.93	**71.85**
Botswana	OA (%)	80.45	84.04	**89.44**
	AOA (%)	66.83	**73.38**	72.92
	Diversity	64.36	68.12	**69.77**
KSC	OA (%)	76.85	83.26	**86.44**
	AOA (%)	64.32	**73.26**	68.33
	Diversity	59.13	66.19	**69.02**



(a) (b) (c)

Fig. 3. Sensitivity to the change of the number of features in a subset. (a) KSC. (b) Salinas. (c) Botswana.

5. ACKNOWLEDGMENT

The authors would like to thank Prof D. Landgrebe from Purdue University for providing the image. The authors would also like to that KAKENHI Grant Number 24360347 and 16H04587, and JSPS KAKENHI Grant Number 16F16053 for supporting this work.

6. REFERENCES

[1] C. I. Chang, *Hyperspectral Data Exploitation: Theory and Applications*, Wiley-Interscience, Hoboken, NJ, 2007.

[2] P. Du, J. Xia, W. Zhang, K. Tan, Y. Liu, and S. Liu, "Multiple classifier system for remote sensing image classification: A review," *Sensors*, vol. 12, no. 4, pp. 4764–4792, 2012.

[3] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, no. S1, pp. 110 – 122, 2009.

[4] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.

[5] L. Breiman, "Random forest," *Machine Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[6] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method.," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 28, no. 10, pp. 1619–1630, 2006.

[7] J. Xia, P. Du, X. He, and J. Chanussot, "Hyperspectral remote sensing image classification based on rotation forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 239 – 243, 2014.

[8] J. Xia, J. Chanussot, P. Du, and X. He, "Spectral-spatial classification for hyperspectral data using rotation forests with local feature extraction and markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2532–2546, 2015.

[9] P. Du, A. Samat, B. Waske, S. Liu, and Z. Li, "Random forest and rotation forest for fully polarized sar image classification using polarimetric and spatial features," *ISPRS J. Photogramm Remote. Sens.*, vol. 105, pp. 38 – 53, 2015.

[10] T. Kavzoglu, I. Colkesen, and T. Yomralioglu, "Object-based classification with rotation forest ensemble learning algorithm using very-high-resolution worldview-2 image," *Remote Sensing Letters*, vol. 6, no. 11, pp. 834–843, 2015.

[11] J. Arenas-García, K. Petersen, G. Camps-Valls, and L. K. Hansen, "Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 16–29, 2013.

[12] J. Arenas-García and G. Camps-Valls, "Efficient kernel orthonormalized pls for remote sensing applications," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 10, pp. 2872–2881, 2008.

[13] J.R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[14] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach Learn.*, vol. 51, no. 2, pp. 181–207, 2003.