






# A Comprehensive Survey of Imbalance Correction Techniques for Hyperspectral Data Classification

Mercedes E. Paoletti , *Senior Member, IEEE*, Oscar Mogollon-Gutierrez ,  
Sergio Moreno-Álvarez , *Graduate Student Member, IEEE*, Jose Carlos Sancho ,  
and Juan M. Haut , *Senior Member, IEEE*

**Abstract**—Land-cover classification is an important topic for remotely sensed hyperspectral (HS) data exploitation. In this regard, HS classifiers have to face important challenges, such as the high spectral redundancy, as well as noise, present in the data, and the fact that obtaining accurate labeled training data for supervised classification is expensive and time-consuming. As a result, the availability of large amounts of training samples, needed to alleviate the so-called Hughes phenomenon, is often unfeasible in practice. The class-imbalance problem, which results from the uneven distribution of labeled samples per class, is also a very challenging factor for HS classifiers. In this article, a comprehensive review of oversampling techniques is provided, which mitigate the aforementioned issues by generating new samples for the minority classes. More specifically, this article pursues a twofold objective. First, it reviews the most relevant oversampling methods that can be adopted according to the nature of HS data. Second, it provides a comprehensive experimental study and comparison, which are useful to derive practical conclusions about the performance of oversampling techniques in different HS image-based applications.

**Index Terms**—Hyperspectral (HS), imbalance, machine learning, oversampling.

## I. INTRODUCTION

**H**YPERSPECTRAL (HS) imagery plays a fundamental role in many important remote sensing applications, in which the spectral-spatial resolution of the acquisition instrument becomes particularly relevant [1]. In this regard, extensive research work has been conducted on different HS-related areas, including image fusion [2], [3], spectral unmixing [4], [5], [6], target detection [7], [8], and fast processing [9], [10], [11], [12].

Manuscript received 4 February 2023; revised 4 May 2023; accepted 21 May 2023. Date of publication 24 May 2023; date of current version 16 June 2023. This work was supported in part by Consejería de Economía, Ciencia y Agenda Digital of the Junta de Extremadura and by the European Regional Development Fund of the European Union (FEDER) under Grant GR21040 and Grant GR21099, and in part by the Spanish Ministerio de Ciencia e Innovación under Grant PID2019-110315RB-I00 (APRISA) with identifier MCIN/AEI/10.13039/501100011033. (Corresponding author: Juan M. Haut.)

Mercedes E. Paoletti and Juan M. Haut are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Universidad de Extremadura, 10003 Caceres, Spain (e-mail: mpaoletti@unex.es; juanmariohaut@unex.es).

Oscar Mogollon-Gutierrez, Sergio Moreno-Álvarez, and Jose Carlos Sancho are with the Media Engineering Laboratory, Department of Computer and Telematics Systems Engineering, Universidad de Extremadura, 10140 Guadalupe, Spain (e-mail: oscarmg@unex.es; smoreno@unex.es; jcsanchon@unex.es).

The source will be made publicly available at <https://github.com/mhaut/imbalance-review>.

Digital Object Identifier 10.1109/JSTARS.2023.3279506

In particular, HS image classification and segmentation are some of the most active research domains within the remote sensing community, mainly because they steadily work for providing accurate earth's surface estimates and land-cover predictions, which eventually allow modern society to deal with current and future technological challenges and needs [13], [14], [15].

HS imaging combines conventional spectroscopy techniques with digital imaging to collect detailed spectral and spatial information from an observation area, producing a large data cube ( $\mathbf{X}$ ) for each recorded scene. In this context, HS classification/segmentation consists of assigning to each pixel (spectral vector) in the data, a single classification label [16], i.e., given the HS scene  $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$ , defined by its height ( $H$ ) and width ( $W$ ) dimensions, together with the number of spectral bands ( $D$ ), respectively, and the set of  $C$  possible land-cover classes, the purpose is obtaining the data-label pair  $\{x_i, y_i\}$  for all the pixel of the scene ( $i \in [1, N]$  defining  $N = WH$ ), with  $y_i = 1, \dots, C$  denoting the corresponding label. The improved performance of HS classifiers comes from the capacity of HS instruments to capture images using hundreds of narrow and contiguous spectral bands, which are recorded from different wavelengths of the electromagnetic spectrum, providing detailed spectral-spatial information of the target scene. For instance, one of the most popular sensors within the remote sensing community is the airborne visible infrared imaging sensor spectrometer (AVIRIS) [17], which collects 224 bands in the spectral range 400–2500 nm with 20 m of spatial resolution. Other popular HS instruments are the reflective optics system imaging spectrometer (ROSIS) [18], or the compact airborne spectrographic imager (CASI) [19], which have also been used to collect multiple remotely sensed data benchmarks [20]. Regardless of the acquisition instrument, HS processing techniques take advantage of the detailed spectral and spatial resolution available to provide a precise material identification over specific earth surface areas of interest [16], [21].

In this context, from kernel-based classification methods [22], [23], through statistical models [24], to the most recent deep learning approaches [25], [26], [27], [28], different paradigms have been successfully proposed and applied to process and classify remotely sensed HS data. Following the natural evolution of artificial intelligence and automatic processing algorithms, from traditional machine learning (ML), which tend to hand-crafted feature processing, to the current state-of-the-art dominated by deep learning (DL) models, characterized by their

unparalleled ability to automatically extract deep and abstract features, the scientific community has provided interesting HS classifiers based on supervised, unsupervised, and semisupervised approaches [29], [30]. In spite of their technical differences, all these technologies share a common requirement: they should all cope with the intrinsic complexity of the HS image domain. On the one hand, HS data often contains a high level of spectral redundancy, as narrow contiguous bands tend to be highly correlated and prone to produce spectral leakage on the radiance acquisition process [31]. Moreover, HS data cubes contain noise due to uncontrolled changes in atmospheric conditions and instrumental limitations, which are coupled with significant spectral mixing due to the tradeoff between spectral and spatial resolutions, results in high data variability. On the other hand, obtaining accurate labeled training data is expensive as well as time consuming. This eventually introduces an important limitation on the availability of ground-truth (labeled) HS data, and also contrasts with the requirement of large amounts of training samples needed to alleviate the so-called Hughes phenomenon [32]. The lack of training samples prevents the classifier from covering both the variability of the data and the complexity introduced by the high dimension of the features. As a result, the model does not fit properly and its behavior degrades rapidly. Similarly, this shortcoming poses a major challenge in semantic segmentation, given the requirement of pixel-level annotations, which are expensive and time-consuming to obtain. Indeed, training datasets for image semantic segmentation are often small and do not cover the full range of variations that can occur in real-world images.

#### A. Class Imbalance

In addition to these problems, there is also an important aspect that significantly affects remotely sensed HS data collections: the large *class-imbalance problem* [33]. This is characterized for having a training set with highly irregular distribution in terms of the number of samples per class, which may eventually introduce an important bias to the classes with more samples during the training process. Indeed, processing methods tend to fit more closely to the majority classes, producing large classification errors for minority classes. Regardless of whether these class differences are naturally present over the earth's surface or artificially generated by some external factors, the class-imbalance problem is a challenging factor when it comes to remotely sensed HS image classification [34], [35] and semantic segmentation [36]. With the ongoing developments in HS imaging acquisition and processing technologies, the earth's surface is being characterized in an unprecedented level of detail, providing rich information for multiple purposes, such as fine-grained land-cover classification (with an inherent class asymmetry). As more ambitious land-cover classification taxonomies are proposed, the class-imbalance problem is more likely to occur, since the class heterogeneity in the earth's surface is naturally diverse [37]. Different approaches have been developed to properly tackle the class-imbalance problem, such as cost-sensitive methods, kernel-based methods, and active learning methods [38], [39], [40]. Notwithstanding their positive

impact on final accuracy, these approaches have a number of limitations that hinder their performance in real HS scenes, e.g., kernel-based models and active learning have a high computational burden, while cost-sensitive methods have to define misclassification costs that are not usually available in HS datasets. Generally, patch-based processing methods have enhanced the classification of HS data by capturing fine-grained details and reducing the impact of within-class variability (the so-called salt and pepper classification noise) using small patches of the images, such as U-Net models. Novel works include hyperspectral change detection [41] and generative adversarial minority oversampling (3-D-HyperGAMO) strategies to increase the accuracy [35]. Nonetheless, these approaches could exacerbate data imbalance issues when certain classes are underrepresented in the patches, which can be solved by balancing the classes within the patches. Furthermore, patch-based approaches are usually computationally intensive, especially when dealing with high-resolution images or large datasets, and requiring from preprocessing steps to improve the performance, such as normalization or data augmentation. Traditionally, HS data dimensionality reduction methods, such as spectral band selection or the popular principal component analysis (PCA), have been used to simplify the feature space, reducing the complexity of the processing models while better separating samples belonging to different classes. Notwithstanding the improved results, these methods do not address the imbalance problem.

Over the past years, extensive research work has been conducted to address the class-imbalance problem [42], [43], [44]. From a general perspective, there are two main trends to deal with imbalanced datasets: 1) preprocessing; and 2) cost-sensitive techniques. Whereas the preprocessing approach is focused on modifying the original data collection to relieve the class-imbalance effect, the cost-sensitive solution proposes to manage these deviations in the classifier itself. Although both frameworks have shown to obtain competitive results in many different application domains, many works in the literature adopt the preprocessing scheme because of its simplicity and more generic design, as it does not affect the classification process itself [45]. The preprocessing strategy, is separated into two primal alternatives: 1) *oversampling* and; 2) *under-sampling*. Focusing on the former, oversampling techniques aim at generating new samples for the minority classes. On the contrary, under-sampling methods are focused on eliminating samples from the majority classes, thus, alleviating the class imbalance effect. From a general perspective, both resampling methods have been studied over standard data collections, where the oversampling scheme has become the predominant approach [46]. Nonetheless, the special complexity of HS images makes it difficult to extrapolate general purpose oversampling results to the remotely sensed HS image domain.

#### B. Contributions and Article Structure

Given the aforementioned issues, this article pursues a twofold objective. First, it reviews the most relevant oversampling methods that can be adopted according to the nature of HS data. Second, it provides an experimental study and comparison

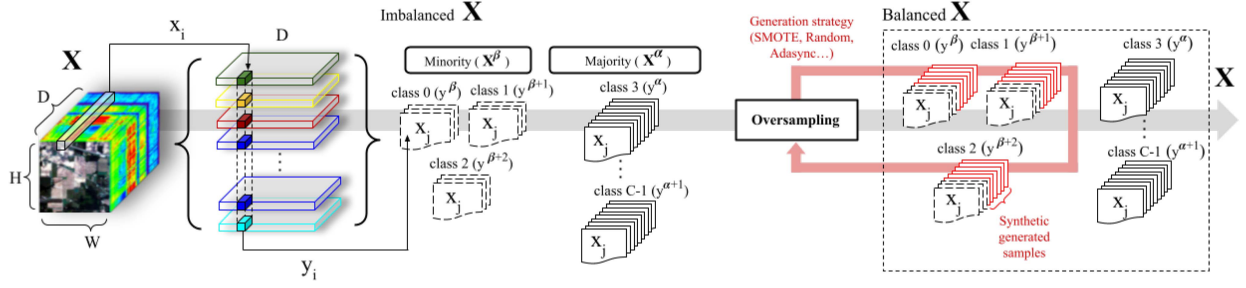


Fig. 1. Oversampling procedure for imbalanced HS scenes. From the imbalanced dataset  $\mathbf{X}$ , minority  $\mathbf{X}^\beta$  and majority  $\mathbf{X}^\alpha$  subsets are extracted. Consequently, each pixel-label pair  $(x_i, y_i)$  within  $\mathbf{X}^\beta$  and  $\mathbf{X}^\alpha$  has its corresponding class. Oversampling strategy is applied to generate a new balanced dataset  $\mathbf{X}$ .

useful to derive practical conclusions about the performance of oversampling techniques in different HS image-based applications. Moreover, this article presents a comprehensive analysis of state-of-the-art oversampling techniques used to improve the classification accuracy of underrepresented classes in HS scenes. In addition, the investigation extends to alternative class balancing methods for semantic segmentation. The contributions of this work provide valuable insights for improving the performance of HS image classification and related tasks, which can have significant implications in various domains, such as remote sensing.

The rest of this article is organized as follows. Section II provides a detailed discussion about some popular oversampling techniques, which have been widely-adopted for HS image classification. Section IV presents an experimental comparison of these techniques, using different widely-used classifiers and detailing the evaluation metrics that have been considered in Section IV-B, and the set of benchmark HS scenes in Section IV-A. This section also provides best practice recommendations for the selection of oversampling techniques in different application domains. Finally, Section V concludes this article with some remarks and hints at plausible future research lines.

## II. OVERSAMPLING FOR HS IMAGE CLASSIFICATION

In recent years, several efforts have been made toward developing novel techniques to effectively classify remotely sensed HS data [16], [21], [47], [48], [49]. Despite all the conducted research, there are still significant challenges to deal with, as the air-borne and space-borne image acquisition technologies are continuously improved [14]. In general, the increase of the spectral-spatial resolution of modern HS sensors makes the task of identifying pixel and subpixel components more complex, since more detailed information is available for the study and classification of the earth's surface. In addition, the class-imbalance problem also has a considerable impact on the final land-cover classification performance, mainly because minority classes may not have enough samples to be properly represented and generalized [45]. To this extent, oversampling techniques have shown to be excellent tools to balance the class distribution in the dataset, while guaranteeing a detailed earth surface characterization. In this context, the study conducted by [50] aimed to tackle the challenge of landslide classification

in remote sensing images through the use of oversampling techniques. The results of this research highlight the effectiveness of oversampling methods for imbalanced data issues and demonstrate the potential of these techniques in the scope of remote sensing data analysis.

This section describes the most popular oversampling methods in the remotely sensed HS image classification domain, providing their technical details. The overall procedure is shown in Fig. 1.

In this research, multiple oversampling techniques are reviewed, which can be divided into three groups—*random*-based, *SMOTE*-based, and *adaptive synthetic sampling* (ADASYN). First, random oversampling is a naive technique for class balancing based on the replication of existing training samples. Second, SMOTE consists of generating synthetic samples for minority classes. Due to the fact of showing good results in several applications, some variations have been proposed to improve its effectiveness. Finally, ADASYN applies adaptive learning to reduce class imbalance by adjusting the corresponding decision boundaries to those minority samples that are harder to learn.

### A. Random Oversampling

The random oversampling method (RANDOM) is the most basic technique to balance a data collection. In particular, this method randomly duplicates samples of the minority class until the classes are balanced. From a mathematical point of view,  $\mathbf{X}$  can be considered as an imbalance dataset comprising  $N$  samples with their corresponding labels,  $\{x_i, y_i\}_{i=1}^N$ , i.e.,

$$\mathbf{X} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

and assuming there are two classes, one majority ( $\mathbf{X}^\alpha$ ) and one minority ( $\mathbf{X}^\beta$ ), the entire dataset could be divided into two subsets, by splitting those samples that belong to the majority class (identified as  $x^\alpha$ ), and those that belong to the minority class (identified as  $x^\beta$ ). According to (1), there must be more samples from the majority class than from the minority class, i.e.,  $N_\alpha \gg N_\beta$

$$\mathbf{X}^\alpha = \{x_j^\alpha, y_j^\alpha\}_{j=1}^{N_\alpha} = \{(x_i, y_i) \in \mathbf{X} | y_i = \alpha\}_{i=1}^N \quad (1a)$$

$$\mathbf{X}^\beta = \{x_j^\beta, y_j^\beta\}_{j=1}^{N_\beta} = \{(x_i, y_i) \in \mathbf{X} | y_i = \beta\}_{i=1}^N. \quad (1b)$$

In this context, the RANDOM method samples new minority class instances by randomly replicating the original samples of

$\mathbf{X}^\beta$  until getting  $N_\alpha = N'_{\min}$ , where  $N'_{\min}$  is the new number of samples in the minority class after the oversampling. From a geometric perspective, the newly generated samples for the minority classes are always placed on the coordinates of an existing sample. As a consequence, it is noteworthy that this method only works for increasing the number of samples in the minority classes, without introducing any data variety in the newly generated samples.

Despite the simplicity of the random oversampling approach, this method is quite prone to over-fitting given that the same information is replicated multiple times within the minority classes. In this regard, some alternatives have been proposed to mitigate the issue of over-fitting in classification tasks. For instance, one of the most popular solutions involves addressing the noise problem that can arise during data oversampling, where generating synthetic data points that are too similar to existing data points can lead to over-fitting. To address this issue, researchers proposed the application of noise-robust oversampling (NROMM) [51] technique in the minority classes.

### B. Synthetic Minority Oversampling Technique (SMOTE)

The synthetic minority oversampling technique (SMOTE) [52] is one of the most popular oversampling methods. In particular, this algorithm is based on generating synthetic samples of the minority classes by using their corresponding nearest neighbors. Assuming the mathematical notation described in Section II-A, where  $x^\alpha$  denotes any sample from the majority class and  $x^\beta$  any sample from the minority class (for simplicity, the index  $j$  can be omitted), the SMOTE method defines two predefined input parameters:  $K$  and  $\omega$ . Parameter  $K$  refers to the number of neighbors that are computed for each minority sample  $x^\beta$ , while parameter  $\omega$  indicates the amount of oversampling to be performed.

The first step is to compute  $K$  nearest neighbors for each sample belonging to the minority class  $\mathbf{X}^\beta$ , producing a subset of minority nearest neighbors  $\mathcal{K}^\beta$ . Note that only minority samples are considered when computing its nearest neighbors  $x_k^\beta$ . This is expressed by the following:

$$\mathcal{K}^\beta = \{x_k^\beta \in \mathbf{X}^\beta : x_k^\beta \in \text{KNN}(x^\beta)\} \quad \forall k \in [1, K]. \quad (2)$$

Once the  $K$  neighbors are computed for each minority sample  $x^\beta$ , the next step generates new synthetic samples according to an oversampling parameter,  $\omega$ . Let  $S$  be the set of synthetic samples. The number of samples to be generated,  $|S|$ , is obtained as shown in the following:

$$|S| = |\mathbf{X}^\beta| \cdot \omega = N_\beta \cdot \omega. \quad (3)$$

The generation of synthetic samples is performed by drawing a segment between the selected minority sample,  $x^\beta$  and a random minority neighbor,  $x_k^\beta$ . Then, the distance between these two samples is multiplied by a random scalar,  $\lambda \in [0, 1]$ . In this way, if  $\lambda$  is lower than 0.5, the new sample will be created closer to the processed minority sample,  $x^\beta$ . Analogously, if  $\lambda$  is greater than 0.5, the synthetic sample will be located closer to the corresponding neighbor,  $x_k^\beta$ . Equation (4) represents the generation of a synthetic sample  $s$ . The number of repetitions

will depend on parameter  $\omega$

$$s = x^\beta + \lambda \cdot (x_k^\beta - x^\beta). \quad (4)$$

SMOTE seeks to address class imbalance by generating synthetic samples in underrepresented regions of the feature space. By augmenting the density of minority class instances in these regions, SMOTE aims to improve the identification of the underrepresented class. Various extensions of this algorithm have been developed to enhance its consistency and robustness, particularly in challenging scenarios. Four distinct variants of the SMOTE algorithm are discussed below, each with its unique characteristics: BORDERLINE-1, BORDERLINE-2, SVM-SMOTE, and K-Means SMOTE.

1) *SMOTE BORDERLINE-1 (SMOTE<sub>BD1</sub>)*: Many existing classification algorithms estimate accurate decision boundaries among classes in order to obtain high precision and reliability when classifying the input data. Nevertheless, the samples located in boundary regions (known as *borderline samples*) are more likely to be misclassified, as the level of uncertainty is higher due to the spectral mixture. Precisely, oversampling techniques can take advantage of this to produce more relevant synthetic data.

Contrary to the regular SMOTE version, which does not consider class boundaries, the BORDERLINE-1 [53] focuses on the *borderline samples* of the minority classes to generate more consistent class separability with the newly generated synthetic data. Specifically, the SMOTE BORDERLINE-1 works as follows, requiring the  $K$  and  $\omega$  parameters, too.

The first step of Borderline-1 is to calculate the  $\tilde{K}$  nearest neighbors for each minority sample from  $\mathbf{X}^\beta$ . Considering  $\mathbf{X}_k^\alpha$  as the subset of neighbors of  $x^\beta \in \mathbf{X}^\beta$  belonging to the majority class,  $\mathbf{X}^\alpha$ , Borderline-1 considers three kinds of samples: *noisy*, *dangerous* and *safe*. Based on the number of majority samples,  $|\mathbf{X}_k^\alpha|$  in the neighborhood of  $x^\beta$ , the algorithm follows the next considerations.

- 1) If  $|\mathbf{X}_k^\alpha| = \tilde{K}$ , the  $x^\beta$  sample is *noisy* since its whole neighborhood belongs to the majority class.
- 2) If  $\tilde{K}/2 \leq |\mathbf{X}_k^\alpha| < \tilde{K}$ ,  $x^\beta$  is *dangerous* because most of its neighboring samples are within the majority class.
- 3) If  $0 \leq |\mathbf{X}_k^\alpha| < \tilde{K}/2$ ,  $x^\beta$  is *safe* or *secure* as most of its neighborhood belongs to the minority class.

Let consider  $\mathbf{X}_D^\beta \subset \mathbf{X}^\beta$  as the set of *dangerous* minority samples. For each sample  $x^\beta \in \mathbf{X}_D^\beta$ ,  $K$  minority neighbors are computed to obtain the desired  $\mathcal{K}^\beta$ . Once minority neighbors are computed, the number of samples to be generated,  $|S|$  is calculated following:

$$|S| = |\mathbf{X}_D^\beta| \cdot \omega. \quad (5)$$

Then, a random number,  $1 \leq \theta \leq K$  of minority neighbors from  $\mathcal{K}^\beta$  is selected for each danger sample until  $|S|$  is reached. As in SMOTE, the generation of synthetic samples is performed according to (4).

As a result, the oversampling process can be conducted only by considering the borderline samples, labeled as *danger*, to increase the density on the minority class boundaries.

2) *SMOTE BORDERLINE-2 (SMOTE BD2)*: Inspired by BORDERLINE-1, the BORDERLINE-2 algorithm [53] considers a wider data diversity when generating the new synthetic samples. In order to achieve this goal, the SMOTE BORDERLINE-2 algorithm not only considers elements of the minority class ( $\mathbf{X}^\beta$ ) when computing the neighborhood of the borderline or danger samples,  $\mathbf{X}_D^\beta$ , but also elements of the majority class ( $\mathbf{X}^\alpha$ ) in order to produce a higher data variation in the minority class. Precisely, this higher variability introduces diversity into the training, which helps to reduce over-fitting.

Whereas the SMOTE BORDERLINE-1 algorithm is designed to produce new samples from the boundaries of minority classes, the SMOTE BORDERLINE-2 extension relaxes this constraint by introducing synthetic samples closer to majority class samples. Generation of new samples is performed as shown in (4) by introducing the random scalar,  $\lambda \in [0, 0.5]$  to calculate the location of the new synthetic sample around the minority observations.

3) *SVM-SMOTE*: Support vector machines (SVMs) have shown a huge potential to identify class boundaries in many different application domains effectively [12]. Nevertheless, the class-imbalance problem within remote sensing HS domain is not an exception [23]. Some authors are focused on modifying the SVM classification process to manage the class-imbalance problem [54]. SVMs also provide a robust framework to generate new synthetic samples. For instance, Japkowicz and Stephen [55] demonstrated that SVMs are excellent tools to deal with such imbalance issues, since class boundaries are typically based on a small number of *support vectors*.

In this context, the SVM-SMOTE method [56] is a popular oversampling technique, which exploits the robustness of SVM when dealing with high-dimensional data to generate new synthetic samples of minority classes. Likewise, BORDERLINE-1 and BORDERLINE-2, SVM-SMOTE also increases the minority class density in those feature space areas with a high uncertainty level.

Considering the imbalanced dataset described in (1), the first step to apply SVM-SMOTE is training an SVM classifier with all the available training data, i.e.,  $\mathbf{X}$ . Thus, the optimal hyperplane that best divides classes  $\mathbf{X}^\alpha$  and  $\mathbf{X}^\beta$  is found. The location of the sample  $x$  on/under the hyperplane is described as follows:

$$\begin{aligned} w \cdot x + b &= 1 \\ w \cdot x + b &= -1. \end{aligned} \quad (6)$$

To generate the weights,  $w$ , such that only the support vectors determine the borderline regions between classes, an optimization algorithm is necessary. Consequently, the support vectors, which are minority samples located in the vicinity of the class boundary, are used to determine the weights. Let  $\mathbf{X}_b^\beta \in \mathbf{X}^\beta$ , a set of minority support vectors and  $x_b^\beta \in \mathbf{X}_b^\beta$ , a support vector. The computation of  $K$  nearest neighbors to form  $\mathcal{K}$  is shown in the following:

$$\mathcal{K} = \{x_k \in (\mathbf{X}^\alpha \cup \mathbf{X}^\beta) : x_k \in \text{KNN}(x_b^\beta)\} \quad \forall k \in [1, K]. \quad (7)$$

In contrast to previous methods, only the borderline minority instances that are approximated by support vectors are over-sampled. Consequently, original SMOTE (3) is redefined into the following:

$$|S| = |\mathbf{X}_b^\beta| \cdot \omega. \quad (8)$$

Nonetheless, it must be noted that, when dealing with large imbalance problems, the decision hyperplane that best maximizes the margin between samples of different classes may be biased toward the majority class [57]. This produces two main issues: 1) minority instances lie far from the optimal decision hyperplane; and 2) SVMs bias majority instances when majority and minority observations overlap in feature space. In this regard, an *interpolation* procedure generates a new sample between two points, as depicted in (9a) when most of the points in  $\mathcal{K}$  belong to the majority class. Otherwise, *extrapolation* is conducted as represented by (9b)

$$s = x_b^\beta + \lambda \cdot (x_k - x_b^\beta) \quad (9a)$$

$$s = x_b^\beta + \lambda \cdot (x_b^\beta - x_k). \quad (9b)$$

One important difference between SVM-SMOTE and SMOTE that should be noted is the fact that new instances are generated in order, i.e., SVM-SMOTE iterates the neighborhood from the closest sample to the further one.

4) *K-Means SMOTE*: When addressing data sparsity, it is advisable to carefully deliberate before implementing oversampling. For certain problems, it may be the case that samples within the same class do not adhere to any discernible pattern, thus necessitating the establishment of suitable criteria for partitioning the data. To address this issue, the initial step involves employing the K-means algorithm to partition the data into  $n$  clusters, wherein each observation is assigned to the cluster the centroid of which is closest. This process is inspired by the unsupervised classifier, Douzas et al. [58], which implements the K-means SMOTE. Three stages are performed, i.e., *clustering*, *filtering*, and *oversampling*:

K-Means SMOTE is applied to an imbalanced dataset as described in (1). The first step consists in clustering data using K-Means algorithm. Let  $\mathbf{C}$  be a set of  $n$  clusters as specified in the following:

$$\mathbf{C} = \{C_1, C_2, \dots, C_n | C_i \in \text{K-Means}(\mathbf{X})\}. \quad (10)$$

In comparison with SMOTE, K-Means algorithms requires an additional parameter, the imbalanced ratio threshold (*IRT*). This parameter determines the necessity of applying oversampling for a specific cluster  $C_i$ . The following provides the calculation of the imbalance ratio given a cluster,  $IR(C_i)$ :

$$IR(C_i) = \frac{|C_i^\alpha|}{|C_i^\beta|}. \quad (11)$$

Then, the set of  $m$  clusters,  $m \leq n$ , to be oversampled is defined in the following:

$$\mathbf{C}' = \{C'_1, C'_2, \dots, C'_m | IR(C'_i) < IRT\}. \quad (12)$$

Finally, to determine the amount of oversampling to be performed in each cluster, sampling weight,  $SW_m$  is calculated

according to the density of minority samples in the feature space for each cluster. In this regard, high sampling weights yields more synthetic samples. The total number of synthetic samples to be generated is given by the following:

$$|S| = \sum_{i=1}^m SW_i \cdot \omega. \quad (13)$$

### C. Adaptive Synthetic (ADASYN) Oversampling

The adaptive synthetic sampling approach (ADASYN) is a popular oversampling approach implemented by He et al. [59]. Specifically, this technique addresses the class imbalance problem by gradually adapting the corresponding decision boundaries to the minority classes.

In addition to the training dataset provided by (1), it is necessary to define some input parameters:  $IR_{th}$ ,  $\omega$  and  $K$ . The first parameter is used to manage synthetic sample generation.  $\omega$  refers to the desired oversampling ratio. Finally, parameter  $K$  refers to the number of neighbors that are computed for each minority sample  $x^\beta \in \mathbf{X}^\beta$  during the oversampling process.

As a first step, the imbalance ratio must be calculated,  $IR$ , between minority and majority classes,  $\mathbf{X}^\beta$  and  $\mathbf{X}^\alpha$ , respectively. Provided that the obtained value is lower than  $IR_{th}$ , the ADASYN algorithm proceeds to the next step. Conversely, if the value exceeds  $IR_{th}$ , oversampling is concluded. The calculation of this value is shown in the following:

$$IR = \frac{|\mathbf{X}^\beta|}{|\mathbf{X}^\alpha|} = \frac{N_\beta}{N_\alpha}. \quad (14)$$

The number of synthetic samples to be generated is calculated at this point as showed in the following:

$$|S| = (|\mathbf{X}^\alpha| - |\mathbf{X}^\beta|) \cdot \omega = (N_\alpha - N_\beta) \cdot \omega. \quad (15)$$

At this stage, the computation of  $K$  nearest neighbors is necessary for each minority sample in  $\mathbf{X}^\beta$ . In contrast to SMOTE, both majority,  $\mathbf{X}^\alpha$ , and minority,  $\mathbf{X}^\beta$ , samples are considered. This process can be formulated as shown in the following:

$$\mathcal{K} = \{x_k \in (\mathbf{X}^\alpha \cup \mathbf{X}^\beta) : x_k \in \text{KNN}(x^\beta)\} \quad \forall k \in [1, K]. \quad (16)$$

Following the neighbors calculation, ratio of majority samples,  $R^\alpha$  is computed for each minority sample in order to decide the amount of oversampling per minority sample, as shown in the following:

$$R^\alpha = \left\{ r_1, r_2, \dots, r_n \mid r_i = \frac{|\mathcal{K}_i^\alpha|}{|\mathcal{K}|} \right\}. \quad (17)$$

Each ratio  $r_i$  must be normalized using the following:

$$r'_i = \frac{r_i}{\sum_{i=0}^{N_\beta} r_i}. \quad (18)$$

Once this ratio is computed, the expected number of synthetic samples to be created per sample, denoted by  $|S'_i|$ , can be estimated using the following:

$$|S'_i| = \sum_{i=0}^{N_\beta} r'_i \cdot |S|. \quad (19)$$

The procedure to generate synthetic samples for each minority sample,  $x_i \in \mathbf{X}^\beta$ , is the same as in (4).

The idea behind the ADASYN algorithm is based on using the  $r'_i$  density ratio to determine the number of synthetic samples required for each minority class sample  $x_i$ . This differs from the behavior of other oversampling methods, which consider the sample position belonging to the minority class (BORDERLINE) or are based on a random criterion (SMOTE). In practice, the density ratio of the ADASYN represents a quantification of the weight distribution for each minority class sample according to its difficulty level in the corresponding learning process. In this context, ADASYN is focused on generating more synthetic samples in the most challenging areas of the dataset, in order to encourage learning features from minority class samples (which are more difficult to be detected).

### D. Comparative Summary

Table I provides a comparison of the different oversampling methods applied in the research using the following criteria.

- 1) *Based on SMOTE*: These methods generate new synthetic samples based on SMOTE algorithm. Therefore, the location of the new sample is calculated between the processed sample and its neighbors belonging to minority classes.
- 2) *Selection of generator samples*: This criterion identifies the selection method for the generator samples in the  $\text{Toy}$  dataset.
- 3) *Use of classifier*: These oversampling methods train a classifier to identify generator samples or clusters prior to the generation of synthetic samples.
- 4) *Sample generation method*: This criterion identifies the method used to create a new sample from an existing one.
- 5) *Location of new synthetic samples*: Location of the synthetic samples in the feature space after applying an oversampling technique.

To better understand how each of the reviewed methods works, a synthetic dataset has been created with three classes (one majority and two minority) as shown in Fig. 2. As previously discussed, when oversampling is performed using random oversampling, new minority class samples are always generated on the coordinates of an existing sample. SMOTE-based methods generate new samples with different patterns. It is interesting to discuss the differences between Borderline1 and Borderline2. It is visible how the former method generates new samples considering the majority class since new samples are generated in the center of the axes. However, Borderline1 is limited to the boundaries of the minority classes. Moving to SVM-SMOTE it can be seen how new samples are generated taking into account support vector samples. This can be seen because most new samples are generated along a few directions. In the case of K-Means SMOTE, one minority class (displayed in green) clearly show how two clusters were created and new samples are generated inside them. Finally, concluding ADASYN operational mode using only the plots is more difficult. Nevertheless, newly generated samples are surrounded by samples from other classes. This will force the classifier to learn boundaries between classes.

TABLE I  
 COMPARISON OF REVIEWED OVERSAMPLING ALGORITHMS

	RANDOM	SMOTE	SMOTE-BL1	SMOTE-BL2	SVM-SMOTE	K-Means SMOTE	ADASYN
Based on SMOTE	No	Yes	Yes	Yes	Yes	Yes	No
Selection of generator samples	Random	Random	Danger	Danger	Support vectors	Clustering + Random	Random
Use of classifier	None	None	None	None	SVM	K-Means	None
Sample generation method	Cloning	Interpolation	Interpolation	Interpolation	Interpolation	Interpolation	Interpolation
Location of new synthetic samples	Random	Random	Borderline	Borderline	Borderline	Cluster-based	Density-based

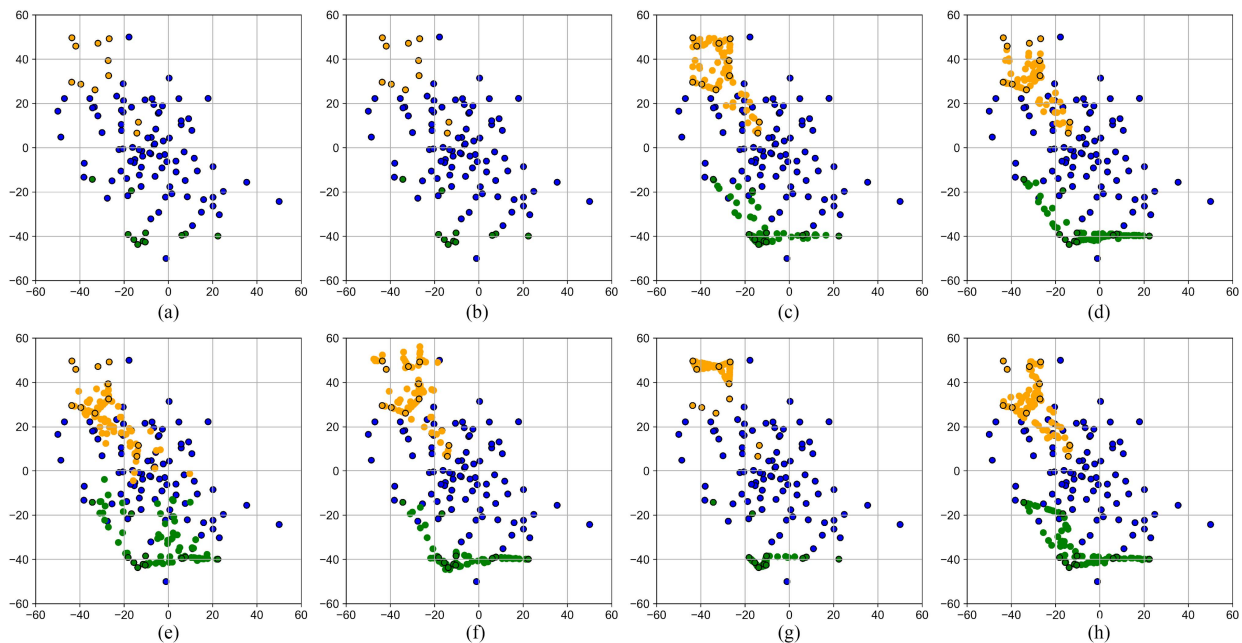


Fig. 2. Graphical comparison of oversampling methods reviewed over a synthetic dataset with three classes. To visualize results one class is considered as majority (blue) and the rest are minority classes (green and orange). The graphical plots depicting the oversampling techniques utilized in this study include ranges for both features, yet serve no actual analytical significance and are solely for the purpose of visualization. (a) Original. (b) RANDOM. (c) SMOTE. (d) SMOTE-BL1. (e) SMOTE-BL2. (f) SVM-SMOTE. (g) K-Means SMOTE. (h) ADASYN.

### III. TACKLING CLASS IMBALANCE BY MEANS OF LOSS FUNCTION

Oversampling techniques have been widely used to tackle the class imbalance problem, providing competent results. In this framework, there is another set of methods that require special attention in this work, due to their promising performance when facing class imbalance and their high impact on the design of the processing method. Indeed, great efforts have been invested to design more descriptive loss functions with the aim of facilitating the processing method to traverse the objective function surface toward the desired result. In this regard, with the aim of reducing the negative impact of class imbalance, multiple loss functions have been developed to increase the weight of underrepresented classes, playing a crucial role in the enhancement of the classification/segmentation performance for underrepresented classes. Most common functions are: 1) *multiclass cross-entropy loss*; 2) *focal loss*; 3) *cyclical focal loss*; 4) *asymmetric focal loss*.

The multiclass cross-entropy (CE) loss function assumes that all classes in a given dataset  $\mathbf{X}$  are equally represented, which is not often in real-world scenarios. The probability distribution generated by the model represents the likelihood of each pixel belonging to a particular class, i.e., considering  $C$  classes, each

$P(x_i|y_i = c)$  provides the probability that  $x_i$  belongs to the  $c$ th class  $\forall c \in [1, C]$ , i.e.,  $P(x_i|y_i = c) = 1$  if it is the correct classification label (i.e., the true label  $Y_c$ ), or  $P(x_i|y_i = c) = 0$  otherwise. The loss is minimized across all classes equally, without considering their distribution. The operation of cross-entropy is calculated by the following for a specific sample  $x_i$ , where  $P(x_i|y_i = c)$  is the class predicted probability:

$$\mathcal{L}_{CE} = - \sum_{c=1}^C Y_c \log(P(x_i|y_i = c)). \quad (20)$$

For the sake of simplicity,  $x_i$  and  $y_i$  can be obviated from (20), simplifying the expression to  $\mathcal{L}_{CE} = - \sum_{c=1}^C Y_c \log(P_c)$

Regarding the focal loss (FL) [60], it weights the loss calculation, assigning higher weights to misclassified samples, while reducing the importance of the correct classified ones (down weighting). Indeed, focusing on those samples where processing fails the most will ensure that the process improves its results on hard samples over time. This is demonstrated in (21a), where  $\gamma$  is the focusing parameter. Furthermore, an  $\alpha$ -balanced variant is used as described in (21b), where  $\alpha_c$  is the balancing factor

$$\mathcal{L}_{FL} = - \sum_{c=1}^C (1 - P_c)^\gamma \log(P_c) \quad (21a)$$

$$\mathcal{L}_{FL} = - \sum_{c=1}^C \alpha_c (1 - P_c)^\gamma \log(P_c). \quad (21b)$$

This model emphasizes on identifying and prioritizing samples that are challenging to classify, while mitigating the influence of easily classifiable samples. The selection of the most suitable loss function is contingent upon the specific objectives of the semantic segmentation task at hand, aimed at improving the performance of the model and addressing the challenge of class imbalance.

Cyclical focal loss (C-FL) [61] is a novel variant for the focal loss based on the learning-rate scheduler. The integration of a cyclical learning rate aids in improving model convergence by enabling the network to escape from suboptimal local minima, while simultaneously mitigating the impact of over-fitting and improving generalization. The C-FL functionality is based on a linear schedule, i.e.,  $\xi$ , defined in terms of the fraction between the current epoch  $e$  and the total number of epochs  $E$ , and a fixed cyclical factor  $f_c \geq 1$ , which provides the cycles of  $\xi$  (with  $f_c = 1$ ,  $\xi$  has one cycle over the epochs, from a value of 1 at the first epoch, to a value of 0 in the last epoch; with  $f_c = 2$ ,  $\xi$  has two cycles, from a value of 1 at the first epoch, to a value of 0 in the epoch  $E/2$ , and again rising to a value of 1 in the last epoch, and so on)

$$\xi = \begin{cases} 1 - f_c \frac{e}{E} & \text{if } f_c \times e \leq E \\ \frac{(f_c \frac{e}{E}) - 1}{f_c - 1} & \text{otherwise.} \end{cases}$$

Indeed,  $\xi$  controls the loss function at every epoch, which is expressed as a combination of FL and the CFL, each one controlled by the corresponding focusing parameters  $\gamma_1$  and  $\gamma_2$

$$\begin{aligned} \mathcal{L}_{C-FL} &= \xi CFL + (1 - \xi) FL \\ &= \xi \left( - \sum_{c=1}^C (1 + P_c)^{\gamma_1} \log(P_c) \right) \\ &\quad - (1 - \xi) \left( - \sum_{c=1}^C (1 - P_c)^{\gamma_2} \log(P_c) \right). \quad (22) \end{aligned}$$

Lastly, asymmetric focal loss (A-FL) [62] aims to prioritize the learning of harder-to-classify examples, which are typically the minority class examples in such datasets. A-FL loss achieves this by assigning different weights to the loss function for each class  $c$  based on the difficulty of the classification task. In this regard, the loss function assigns a higher weight to minority class samples that are more challenging to classify, while assigning a lower weight to majority class examples that are easier to classify. An approximation of this calculation is shown in the following:

$$\begin{aligned} \mathcal{L}_{A-FL} &= L_+ + L_- \\ &= \left( - \sum_{c=1}^C (1 - P_c)^{\gamma_1} + \log(P_c) \right) \\ &\quad + \left( - \sum_{c=1}^C (P_c)^{\gamma_2} - \log(1 - P_c) \right). \quad (23) \end{aligned}$$

## IV. EXPERIMENTAL RESULTS

A large set of experiments on different real and popular HS datasets, using different classifiers widely known by the scientific community, has been performed in order to evaluate the impact of the oversampling techniques reviewed above. In the following, the description of the HS datasets, the set of metrics used for the evaluation of the experiments, the setting and motivation of the experiments performed, and a detailed discussion of the results obtained are provided.

### A. Datasets

Three widely used HS images, with different spatial and spectral characteristics and different numbers of labeled samples, have been used to conduct the experimental validation of oversampling methods: Indian Pines (IP), Botswana (BW), and Kennedy Space Centre (KSC) scenes. The IP and KSC scenes were collected by the AVIRIS, while BW was gathered by the UT center for space research purposes. Figs. 3–5 show a summary of the HS scenes, including the number of labeled samples per class, as well as the available ground-truth information. These datasets (along with the training and test sets) are all available from the IEEE Geoscience and Remote Sensing Society (GRSS) Data and Algorithm Standard Evaluation website (DASE) at: <http://dase.grss-ieee.org>.

- 1) The IP scene [see Fig. 3(a)] was captured in 1992 over the Indian Pines test site in NW Indiana, an agricultural area characterized by its crops of regular geometry and also forest regions. The scene consists of  $145 \times 145$  pixels with spatial resolution of 20 meters per pixel (mpp) and with 224 spectral bands, which have been collected in the wavelength range from 0.4 to 2.5 microns. From these bands, 24 were removed as they are null or water absorption bands (particularly, [104–108], [150–163] and 220), considering the remaining 200 bands for the experiments. The available ground truth comprises 16 mutually exclusive classes. In addition to the original scene, a spatially disjoint train-test scene (DIP) has been used to evaluate the behavior of certain spectral-spatial classifiers (see Fig. 4).
- 2) The KSC scene [see Fig. 3(b)] was also provided by AVIRIS during a flight campaign in 1996. The spectral information ranges from 400 to 2500 nm, with  $512 \times 614$  pixels and 176 spectral bands. Also, some low signal-to-noise ratio (SNR) bands have been removed. The ground-truth is divided into 13 mutually exclusive classes, pertaining to upland and wetland areas.
- 3) The BW dataset (see Fig. 5) was acquired over the Okavango Delta, Botswana, by the Hyperion sensor on the satellite EO-1. The scene contains  $1496 \times 256$  pixels characterized by 30 m of spatial resolution, and 242 bands in the spectral range 400–2500 nm. It must be noted that 97 uncalibrated and water-corrupted bands have been removed, keeping the remaining 145 spectral bands [35]. The ground truth comprises 14 different and mutually exclusive land-cover classes, including seasonal marshes, occasional swamps and drier woodlands located in the distal part of the Delta.



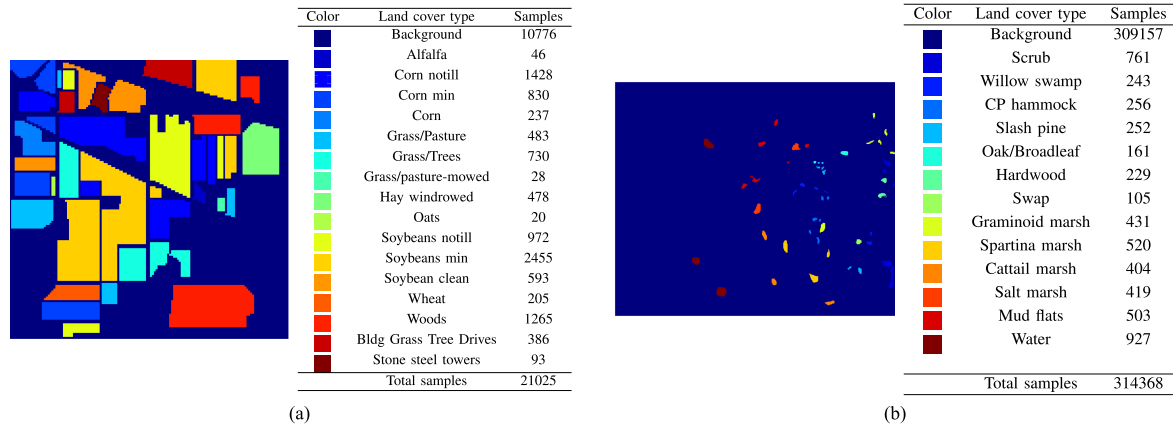


Fig. 3. Ground truth of Indian Pines (IP) and Kennedy Space Center (KSC) datasets. (a) Indian Pines (IP). (b) Kennedy Space Center (KSC).

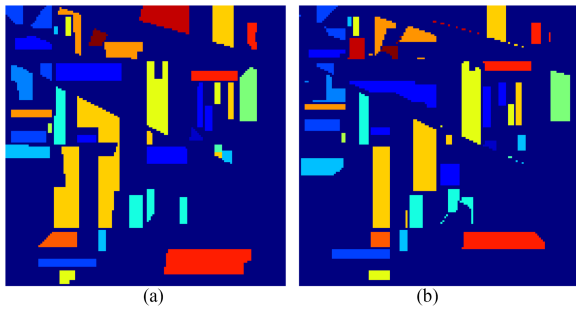


Fig. 4. Spatially disjoint training and test samples of Indian Pines (IP) scene. (a) Disjoint train. (b) Disjoint test.

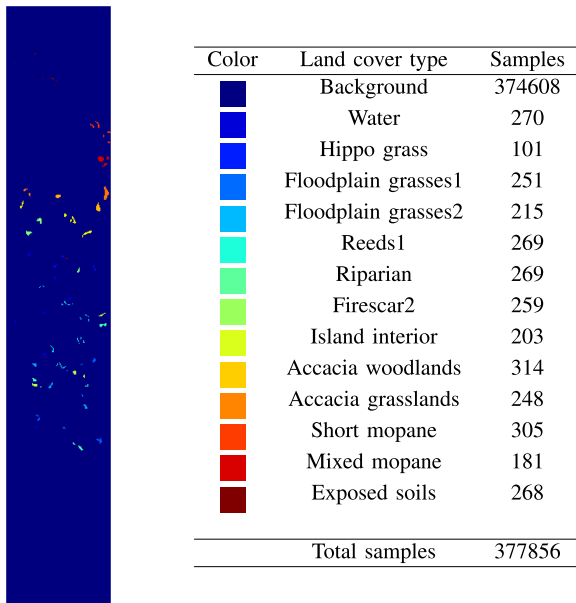


Fig. 5. Ground truth of Botswana (BW) dataset.

sensor, while the second had a VNIR hyperspectral Head-wall Photonics Micro Hyperspec E-Series CMOS sensor. The original resolution of AeroRIT is  $1973 \times 3975$  pixels, covering a broad spectral range from 397 nm to 1003 nm. In order to ensure the reliability of the data, ambiguous and inconsistent pixels were removed, resulting in a scene with a final resolution of  $1920 \times 3968$ . The processing hyperparameters and dataset configuration were extracted from the original study.

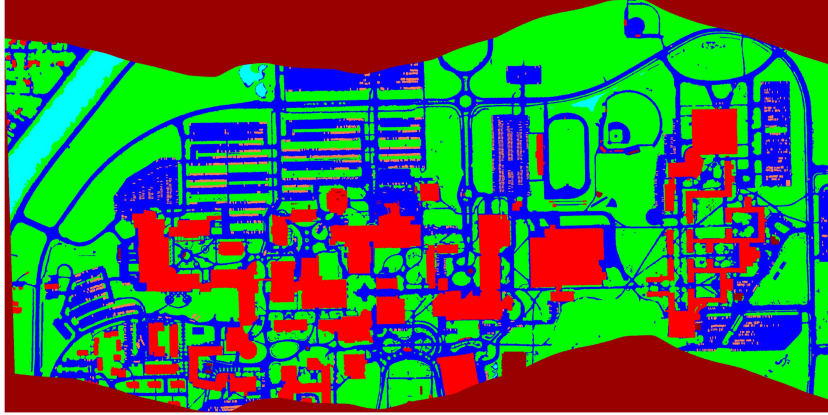
### B. Evaluation Metrics

When extracting knowledge for imbalanced data, it is necessary to implement evaluation metrics that assess model performance. In this regard, Table II provides the metrics considered in this study to evaluate the performance of the different oversampling methods. They are provided in terms of the confusion matrix of a binary classification problem, i.e., considering the two classes *Positive* and *Negative*. In this regard, from the distribution of classifier performance on the data, the measurements collected by Table II are expressed as a function of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), where the first two concepts refer to the results correctly predicted by the model, and the last two to the results incorrectly predicted by the model, for both the positive and negative classes, respectively.

### C. Experimental Settings

In this section, a detailed comparison between several oversampling algorithms, i.e., RANDOM sampling, SMOTE, SMOTE-BORDERLINE-1 (SMOTE BD1), SMOTE-BORDERLINE-2 (SMOTE BD2) and SVM-SMOTE, is performed on different classifiers, considering both traditional machine learning algorithms and state-of-the-art deep learning models, to evaluate the impact of oversampling on the final classification results. It must be noted that both K-means-SMOTE and ADASYN are not evaluated, as they impose severe restrictions on the minimum number of samples required for generating synthetical data properly. Indeed, these two methods have to run KNN algorithm to determine if a minority sample has to be

- 4) The AeroRIT (see Fig. 6) [63] scene was captured using a Cessna aircraft with two types of camera systems flown over the Rochester Institute of Technology’s university campus. The first system had an 80 MP RGB silicon



Color	Land cover type	Samples
Blue	Roads	1944506
Red	Buildings	917817
Green	Vegetation	3177633
Orange	Cars	132093
Cyan	Water	118664
Dark Red	Unspecified	1327847
Total samples		7618560

Fig. 6. Ground truth of AeroRIT dataset.

TABLE II  
EVALUATION METRICS

Metric	Description	Discussion	Equation
<b>Accuracy per class (Recall)</b>	Number of positive class samples correctly labeled as positive over all positive samples.	It is not affected by class distribution but it cannot assess samples that are incorrectly labeled as positive.	$\frac{TP}{TP+FN}$
<b>Average accuracy</b>	Average between the classification accuracy among all classes $k \in K$ .	This metric is more suitable than overall accuracy when dealing with imbalanced datasets since it estimates the classifier performance considering both minority and majority classes.	$\frac{\sum_k^K (Recall)}{k}$
<b>Overall accuracy</b>	Ratio between the classification hits and the total predictions.	Classification results can be evaluated easily using this metric. However class imbalance leads to an unreliable interpretation of this metric.	$\frac{TP+TN}{TP+TN+FP+FN}$
<b>Precision</b>	Number of positive samples identified correctly from all positive predictions.	Its calculation is affected by skewed class distribution and it cannot measure the amount of positive samples labelled incorrectly.	$\frac{TP}{TP+FP}$
<b>F1-Score</b>	F-measure is an evaluation index for a comprehensive evaluation of precision and recall ratio.	It measures the performance of a classifier by combining precision and recall, resulting in a very informative metric in an imbalanced domain.	$2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}$
<b>Geometric mean</b>	The G-mean value is the geometric mean of the recall rates of majority and minority classes.	The G-Mean measures the geometric mean of the data as a measure of the accuracies of the classes, aiming to maximize them while obtaining good balance.	$\sqrt{\frac{TP}{TP+TN} \cdot \frac{TN}{FP+TN}}$

oversampled. Conducted experiments require a minimum of five neighbors to compute distances in the data space. Consequently, when using 5% or lower amounts of labeled samples, some minority classes do not meet the required threshold. For instance, IP scene has a severe lack of samples for several land-cover classes (such as *Oats* and *Grass/pasture-mowed*). Hyperparameter optimization is conducted using GridSearch and tenfold cross-validation over the whole experimental pipeline, including the oversampling algorithm and the classifier. In this strategy, a wide range of hyperparameters is tested on the original training set over ten partitions for each conducted experiment. As a result, the optimal values for each hyperparameter in the pipeline are estimated. Thus, it has been decided to show their performance in Sections II-B4 and II-C, respectively, but not to evaluate them experimentally. To evaluate the classification results obtained after the application of the other oversampling methods, all the measures foreseen in Table II have been adopted. In order to assess the impact of class imbalance, the study employs a rigorous methodology consisting on five Monte Carlo runs. In each run, the same seed is utilized for all algorithms to ensure consistency and uniformity in the evaluation process. The experimentation

aims to analyze the robustness of the results to changes in the selected training data for imbalanced classes.

Regarding the classifiers, two different experiments have been conducted. The former performs a comparison between different standard and widely used pixelwise classifiers. Specifically, the following classifiers have been considered to evaluate the behavior of the oversampling methods on traditional machine learning models, i.e., multinomial logistic regression (MLR) [4], SVM [22], and shallow and deep multilayer perceptron (MLP and DMLP) [13]. The same procedure has been followed to fairly evaluate the oversampling methods. In particular, different amounts of randomly selected training data are selected from the HS scene (3%, 5%, 10%, 15%, and 20%). Then, the oversampling algorithms are applied to increase the number of samples within the training sets, producing an augmented set. Finally, the supervised classifiers are trained on the augmented training set and the obtained inference results provide the impact of the oversampling strategies. To further explore the impact of oversampling models, a detailed comparison is provided considering the 5% of training data, taking into account the oversampling technique with the highest G-mean score (OS), and comparing

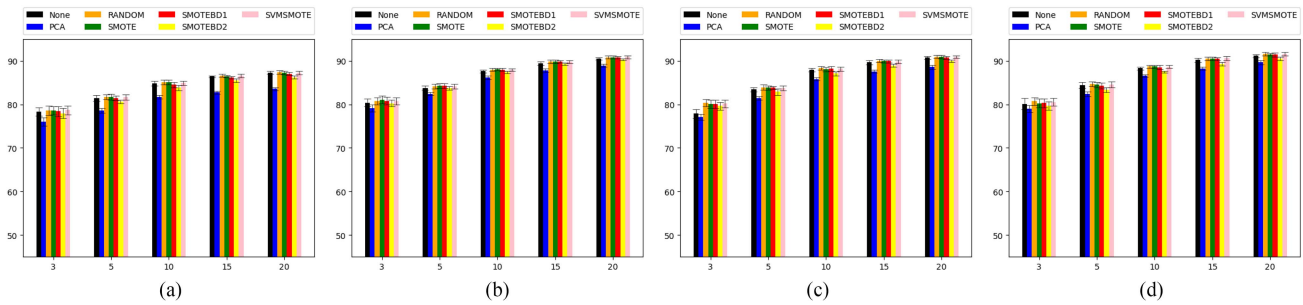


Fig. 7. Obtained G-Mean on Indian Pines (IP) scene. Included charts present the results of several oversampling algorithms, namely random oversampling (RANDOM), synthetic minority oversampling technique (SMOTE), SMOTE-BORDERLINE-1 (SMOTEBD1), SMOTE-BORDERLINE-2 (SMOTEBD2), support vector machine SMOTE (SVMSMOTE), and principal component analysis SMOTE (PCA). Multiple machine learning models were employed for the evaluation: (a) multiple linear regression (MLR); (b) support vector machine (SVM); (c) multilayer perceptron (MLP); (d) deep multilayer perceptron (DMLP).

its results with the ones obtained using techniques of no oversampling, i.e., training data without oversampling (RAW), and spectral reduced data based on principal components analysis (PCA) [32].

The next experiment conducts a comparison between different state-of-the-art deep learning models to evaluate the impact of oversampling techniques. In this sense, an ablation study is performed using the convolutional neural network (CNN) as main structure [25]. Particularly, CNN3-D is used as the baseline classifier. Based on the CNN3-D, the CNN3-D + OV is built by introducing a convex 3-D hyperspectral patch generator unit to oversample the minority classes [35]. The comparison also includes the ssGAN3-D [64], a semisupervised classifier, and the 3-D-HyperGAMO [35], which is considered as a combination of the CNN3-D + OV and ssGAN3-D.

The last experiment, evaluates the impact of class imbalance on the performance of semantic segmentation models trained with different loss functions, i.e., cross-entropy (CE), focal loss (FL), asymmetric focal loss (A-FL), and cyclical focal loss (C-FL). To this end, the models were trained on an imbalanced dataset and tested using the mean intersection over union (mIoU) and overall accuracy (OA) metrics. A detailed comparison of these models is conducted for multiple image patches configurations.

### D. Evaluation on Standard Machine Learning Classifiers

For each HS dataset, the classification results obtained by standard machine learning algorithms when introducing different oversampling techniques is evaluated. In particular, Figs. 7–9 depict the evolution of the G-Mean obtained by the MLR, SVM, MLP, and DMLP in IP, KSC, and BW scenes when using raw data (no oversampling, none), PCA, random, SMOTE, SMOTEBD1, SMOTEBD2, and SVM-SMOTE techniques with different amounts of training data, i.e., 3%, 5%, 10%, 15%, and 20%. Furthermore, Tables III–V provide a detailed comparison in terms of F1, G-mean, OA, and AA, using a 5% of the labeled data to train the models, and focusing on the oversampling technique with the best G-mean (OS), raw data, and PCA-reduced data.

1) *Results on Indian Pines*: Fig. 7 provides the G-mean score of each classifier implementing the different oversampling

TABLE III  
CLASSIFICATION RESULTS FOR IP SCENE

Class	MLR			SVM			MLP			DMLP		
	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS
1	<b>66.88</b>	57.49	66.60	66	62.67	<b>68.22</b>	<b>67.02</b>	61.79	65.33	<b>69.07</b>	62.42	67.24
2	47.66	45.33	<b>53.46</b>	53.35	52.27	<b>57.40</b>	51.15	50.34	<b>55.19</b>	54.14	51.66	<b>57.18</b>
3	29.42	30.80	<b>40.58</b>	34.58	33.87	<b>37.60</b>	35.51	34	<b>40.71</b>	39.6	35.07	<b>42.04</b>
4	76.56	64.77	<b>81.24</b>	84.81	81.22	<b>86.36</b>	78.17	71.42	<b>80.13</b>	80.17	74.66	<b>82.64</b>
5	<b>92.87</b>	90.29	90.65	<b>94.28</b>	93.7	93.52	<b>92.81</b>	91.35	92.61	92.67	92.18	<b>93.24</b>
6	97.29	96.89	<b>97.31</b>	98.48	<b>98.83</b>	98.59	<b>98.57</b>	98.04	98.13	98.08	<b>98.61</b>	98.28
7	56.47	53.41	<b>65.19</b>	66.34	64.08	<b>69.24</b>	68.2	63.97	<b>70.29</b>	69.59	66.49	<b>71.85</b>
8	<b>76.59</b>	74.91	64.31	<b>80.17</b>	79.33	75.84	<b>77.4</b>	75.67	75.61	<b>77.65</b>	76.51	76.21
9	41.51	32.02	<b>52.27</b>	48.77	42.5	<b>53.84</b>	49.36	43.69	<b>56.11</b>	54.26	47.98	<b>57.51</b>
10	95.59	94.97	<b>97.69</b>	95.9	95.44	<b>96.10</b>	97.54	95.74	<b>98.10</b>	98.1	96.51	<b>98.10</b>
11	<b>93.07</b>	91.61	89.22	<b>95.17</b>	95.04	92.97	<b>93.62</b>	91.73	91.51	<b>93.99</b>	92.17	92.25
12	49.40	40.63	<b>56.21</b>	42.34	34.93	<b>49.21</b>	49.54	43.49	<b>53.68</b>	50.57	45.97	<b>53.08</b>
13	84.20	83.98	<b>88.30</b>	81.02	81.82	<b>82.16</b>	86.14	85.11	<b>87.16</b>	85.8	83.75	<b>87.27</b>
F1	<b>70.79</b>	66.60	70.63	73.98	71.96	<b>74.59</b>	73.4	70.38	<b>74.01</b>	74.77	71.76	<b>75.07</b>
G-Mean	81.46	78.62	<b>81.72</b>	83.77	82.43	<b>84.31</b>	83.41	81.41	<b>83.92</b>	84.38	82.36	<b>84.65</b>
OA	<b>71.40</b>	67.38	70.52	74.62	72.77	<b>74.88</b>	73.89	70.94	<b>74.24</b>	75.15	72.25	<b>75.32</b>
AA	69.81	65.93	<b>72.54</b>	72.4	70.44	<b>73.93</b>	72.69	69.72	<b>74.20</b>	74.13	71.07	<b>75.15</b>

For each classifier, best performance metric is highlighted in bold. Results in red are the higher metrics for IP scene. Train set is set to 5%.

techniques. Furthermore, the respective standard deviation are shown after five Monte Carlo runs. In general, the classifiers improve their results with increasing training data, with the results obtained by the DMLP, MLP, and SVM being superior to the MLR. Indeed, it is important to note that the MLR algorithm requires at least a 20% of labeled samples to obtain similar results (slightly inferior) to those produced by the other classifiers when training with 10% of labeled samples. In this sense, the DMLP obtains the best G-mean score (84.65%), while classifiers with PCA obtain the lowest results. Regarding the standard deviation, the SVM exhibits the most reliable/stable behavior.

It is interesting to note how, for different classifiers, the final results of classification with oversampling methods vary slightly. In fact, the results obtained show that the effectiveness of an oversampling strategy does not vary too much depending on the classifier. The percentage of samples constituting the training set also influences the final results, albeit to a lesser extent. Thus, for instance, MLR achieves better results with SVM-SMOTE technique when using 3% of labeled data, with SMOTE when using 5%–10%, and with RANDOM oversampling when considering 15%–20%, although the results between these three techniques are quite similar, with slight variations in the variance (SMOTE is more stable in general); similar behavior can be observed in SVM between SVM-SMOTE, SMOTE, and RANDOM methods, with SMOTEBD1 achieving

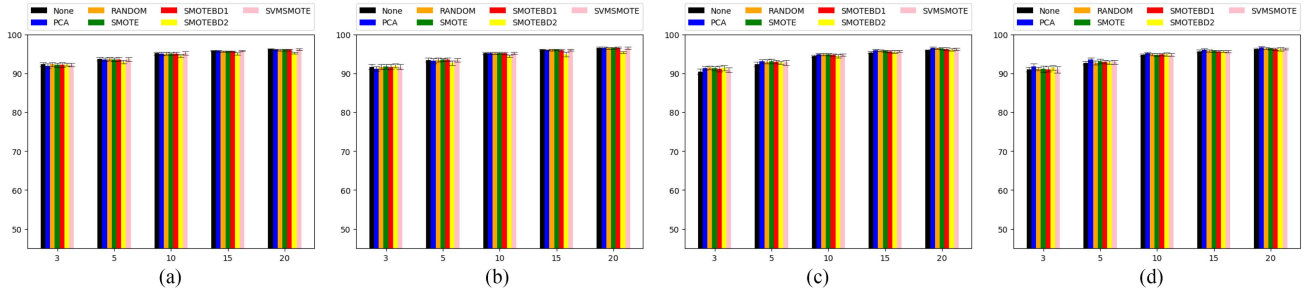


Fig. 8. Obtained G-Mean on Kennedy Space Center (KSC) scene. Included charts present the results of several oversampling algorithms, namely random oversampling (RANDOM), synthetic minority oversampling technique (SMOTE), SMOTE-BORDERLINE-1 (SMOTEBD1), SMOTE-BORDERLINE-2 (SMOTEBD2), support vector machine SMOTE (SVMSMOTE), and principal component analysis SMOTE (PCA). Multiple machine learning models were employed for the evaluation: (a) multiple linear regression (MLR); (b) support vector machine (SVM); (c) multilayer perceptron (MLP); (d) deep multilayer perceptron (DMLP).

the best results when using 5% (closely followed by SMOTE); however, for shallow MLP, the RANDOM technique provides the best classification results with few data, closely followed by SVM-SMOTE and SMOTE when using 15%–20% of training data, and finally for DMLP, both RANDOM and SVM-SMOTE impact favorably on the final results. For all cases, classifiers with PCA obtain the worst G-mean scores, while SVM-SMOTE, SMOTE, and RANDOM obtain the best results.

To further elaborate on these results, Table III provides the classification results for IP in terms of accuracy per class, F1-Score, G-Mean, OA, and AA. Moreover, classifiers have been trained with 5% of labeled samples, considering RAW data, PCA-based data and the best oversampling strategy (OS). The last one was determined by the G-mean value, thus, the MLR includes SMOTE, the SVM implements SMOTEBD1, and both the MLP and the DMLP take RANDOM oversampling strategy. Once more PCA-based classifiers obtains the lowest accuracy, while OS-based classifiers generally obtain the best values, with certain exceptions in the MLR algorithm. Indeed, oversampling enhances the classification performance in terms of F1, G-mean, OA, and AA in the SVM, MLP, and DMLP classifiers, but only in terms of G-mean and AA in the MLR. The DMLP algorithm outperforms the other classification methods, with F1 (75.07%), G-Mean (84.65%), OA (75.32%), and AA (75.15%). Focusing on the minority classes 7-*Oats* and 9-*Grass/pasture-mowed*, all classifiers significantly enhance the identification and classification of samples of these land-cover types by means of oversampling techniques.

2) *Results on Kennedy Space Center*: In this section, RANDOM, SMOTE, SMOTEBD1, SMOTEBD2, and SVM-SMOTE oversampling methods are evaluated against the KSC dataset. Once more, results obtained over RAW data and PCA-based data are included. Classifiers have been trained with 3%, 5%, 10%, 15%, 20% of randomly selected data. The rest of the data was used for testing.

Obtained G-mean score is depicted by Fig. 8, coupled with the respective standard deviation after five Monte Carlo runs. At first glance, it can be seen that KSC requires few samples to estimate the overall scene. Indeed, with 3% of training samples, the G-mean exceeds 85% for all classifiers, while in IP, they need almost 5%–10% of training data. Once more, the value of G-Mean is improved as the training set increases. Also, the

DMLP obtains the best classification result, closely followed by the SVM when there are few training samples (3%–5%). Indeed, the differences between DMLP and SVM are practically negligible.

Similar to IP scene, the G-mean scores prove that the effectiveness of an oversampling strategy does not vary too much depending on the classifier. Once more, the percentage of samples constituting the training set influences the final results, albeit to a lesser extent as the obtained results change proportionally. Nevertheless, the spectral nature of the image does play an important role. While IP is known for its large spectral mixture, KSC is challenging due to scarcity of labeled samples. In this scene, it can be seen that the PCA-based classifiers obtained better results than RAW and that even the oversampling methods, such as in the MLR (with 15% of labeled data), MLP (with 5%, 15%, and 20% of the training data) and DMLP (for all amounts of training data). Moreover, RAW-based results are sometimes very close to those obtained by the oversampling methods, especially in the MLR, it is pretty close to the best oversampling method (SVM-SMOTE) and even superior with 3% of training data. Focusing on MLR: with 3%, RANDOM and RAW-based techniques obtain the best G-mean; with 5%, SVM-SMOTE and RAW-based methods achieve the best results; with 10% of labeled data, SVM-SMOTE and RAW-based provide the best score, and with 15%–20%, the SVM-SMOTE outperforms the other strategies. Regarding SVM: with 3%, SMOTEBD2 clearly outperforms the other techniques, however, its performance decreases by 5%, where SMOTEBD1 and SMOTE are the best oversampling methods; with 10%, all the strategies obtain very similar results, with the exception of SMOTEBD2; with 15%, SVM-SMOTE, SMOTE, and RANDOM achieve the best G-mean scores, and finally, with 20% of labeled data, SMOTEBD1, SMOTE, and RANDOM outperform the results of the other sampling strategies. Related to MLP: with 3% of training data, SMOTEBD2 provides the best G-mean, followed by RANDOM oversampling; with 5%, PCA and SMOTE offer the best accuracy; with 10%, RANDOM, PCA, and SMOTE obtain the best G-mean values; with 15%, PCA, SVM-SMOTE and SMOTE are the best oversampling techniques, and finally, with 20% of the training data, PCA, RANDOM, and SMOTE reach the best results. Similar behavior is exhibited by the DMLP, where PCA and SMOTEBD2 stand out with 3% of labeled

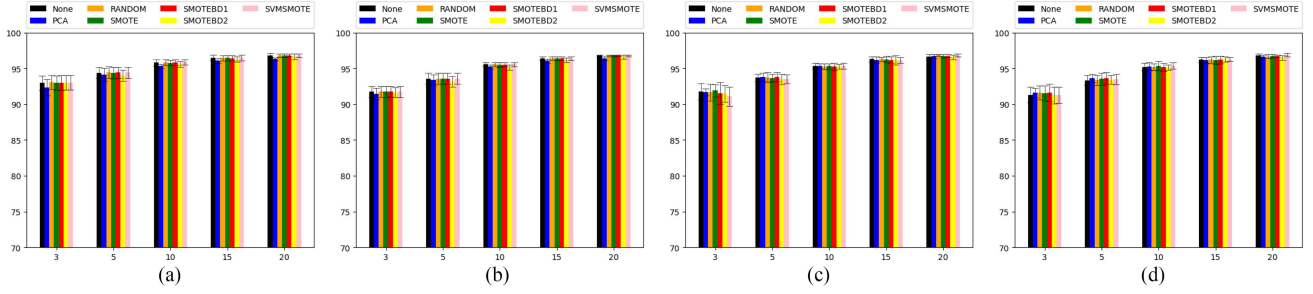


Fig. 9. Obtained G-Mean on Botswana (BW) scene. Included charts present the results of several oversampling algorithms, namely random oversampling (RANDOM), synthetic minority oversampling technique (SMOTE), SMOTE-BORDERLINE-1 (SMOTEBD1), SMOTE-BORDERLINE-2 (SMOTEBD2), support vector machine SMOTE (SVMSMOTE), and principal component analysis SMOTE (PCA). Multiple machine learning models were employed for the evaluation: (a) multiple linear regression (MLR); (b) support vector machine (SVM); (c) multilayer perceptron (MLP); (d) deep multilayer perceptron (DMLP).

TABLE IV  
CLASSIFICATION RESULTS FOR KSC SCENE

Class	MLR			SVM			MLP			DMLP		
	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS
1	<b>94.07</b>	93.61	92.03	92.61	91.26	<b>92.63</b>	<b>93.69</b>	93.39	91.52	<b>93.65</b>	93.03	92.31
2	<b>87.97</b>	87.32	86.67	<b>85.41</b>	84.98	85.37	<b>86.45</b>	86.45	84.89	85.50	<b>86.75</b>	85.28
3	83.00	<b>84.98</b>	81.36	83.50	<b>86.05</b>	83.17	<b>87.49</b>	87.08	85.06	<b>86.42</b>	86.13	85.97
4	<b>64.39</b>	61.97	62.89	64.39	63.39	<b>64.98</b>	60.25	52.38	<b>61.05</b>	58.54	57.41	<b>63.85</b>
5	49.61	46.67	<b>59.02</b>	53.59	<b>53.99</b>	53.4	36.54	39.48	<b>56.86</b>	44.71	50.85	<b>54.12</b>
6	64.15	64.19	<b>66.45</b>	62.07	58.53	<b>63.13</b>	52.53	<b>65.85</b>	62.12	56.36	<b>67.37</b>	61.98
7	79.70	80.40	<b>83.2</b>	71.30	70.20	<b>71.8</b>	72.20	69.90	<b>81.3</b>	72.40	72.00	<b>77.20</b>
8	<b>88.97</b>	88.61	88.53	<b>88.66</b>	87.24	88.63	86.97	<b>89.32</b>	88.88	88.44	<b>89.29</b>	88.85
9	94.13	<b>94.41</b>	93.36	95.57	<b>95.63</b>	95.57	95.30	<b>95.38</b>	94.82	95.73	<b>96.72</b>	94.68
10	89.92	<b>89.97</b>	89.74	90.86	90.76	<b>90.89</b>	87.99	<b>91.77</b>	88.2	88.02	<b>91.25</b>	87.79
11	95.55	95.58	<b>95.7</b>	<b>96.33</b>	96.31	96.31	95.15	<b>97.64</b>	94.85	94.47	<b>96.48</b>	94.50
12	<b>91.74</b>	91.44	91.65	89.25	89.31	<b>89.31</b>	86.09	<b>88.81</b>	85.54	86.92	<b>88.56</b>	86.61
13	<b>99.97</b>	99.97	99.94	100.00	100.00	<b>100</b>	<b>100.00</b>	99.73	99.95	100.00	99.75	<b>100.00</b>
F1	<b>89.02</b>	88.81	88.97	88.61	88.20	<b>88.69</b>	86.85	<b>88.03</b>	87.96	87.36	<b>88.74</b>	88.07
G-Mean	<b>93.70</b>	93.56	93.67	93.46	93.20	<b>93.51</b>	92.37	<b>93.08</b>	93.05	92.70	<b>93.54</b>	93.15
OA	<b>89.04</b>	88.82	88.86	88.61	88.19	<b>88.68</b>	87.10	<b>88.19</b>	87.9	87.53	<b>88.79</b>	88.07
AA	83.32	83.01	<b>83.89</b>	82.58	82.12	<b>82.71</b>	80.05	81.32	<b>82.7</b>	80.86	<b>82.74</b>	82.55

For each classifier, best performance metric is highlighted in bold. Results in red are the higher metrics for IP scene. Train set is set to 5%.

data; PCA is undoubtedly the best technique, followed by far by SMOTE with a training percentage of 5%; again PCA is the best with 10% of labeled data, followed by SMOTEBD1 and SMOTEBD2, and finally, the SMOTE technique is only surpassed by PCA with 15% and 20% of training.

To further explore these results, Table IV provides the classification measurements obtained over the KSC scene with 5% of the training data. Consistent with Fig. 8, the best F1, G-mean, and OA values are provided by the MLR with no oversampling method. This is because KSC samples are very sparse and oversampling techniques based on interpolations may introduce too much variability/noise in the new samples, while RANDOM oversampling makes information redundant. Also, MLP and DMLP improve their classification by PCA, which alleviates the overfitting caused by the large spectral dimension, although the SMOTE oversampling outperforms the results achieved by RAW-data. Focusing on SVM, the generation of new samples to balance the training set improves the classification results in comparison with the RAW and PCA-based data. Focusing on the minority class 7-Swap, all classifiers with oversampling techniques improve its identification and classification.

3) Results on Botswana: Fig. 9 provides the obtained G-mean score and the respective standard deviation after 5 Monte Carlo runs of the spectral classifiers using RAW, RANDOM, SMOTE, SMOTEBD1, SMOTEBD2, and SVM-SMOTE strategies. In this case, MLR obtains the best G-mean score when few labeled data are available (3%). Nevertheless, this behavior

TABLE V  
CLASSIFICATION RESULTS FOR BW SCENE

Class	MLR			SVM			MLP			DMLP		
	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS	RAW	PCA	OS
1	100	100	<b>100</b>	99.73	99.73	<b>99.77</b>	100	100	<b>100</b>	100	100	<b>100</b>
2	94.79	94.58	<b>96.67</b>	94.58	<b>94.69</b>	94.17	90.21	<b>92.29</b>	90.73	88.23	90	<b>91.98</b>
3	<b>97.94</b>	97.56	97.86	96.81	96.68	<b>96.81</b>	96.64	<b>97.1</b>	96.76	96.13	96.09	<b>96.39</b>
4	93.38	92.75	<b>93.73</b>	<b>91.57</b>	90.93	91.52	92.79	<b>93.33</b>	93.09	92.4	91.62	<b>93.09</b>
5	<b>78.11</b>	76.7	77.72	79.48	78.86	<b>79.95</b>	80.7	<b>82.58</b>	80.66	78.66	<b>83.75</b>	81.87
6	70.47	69.53	<b>70.74</b>	68.66	66.71	<b>68.82</b>	72.03	72.03	<b>75.28</b>	75.2	73.79	74.1
7	96.79	96.42	<b>96.91</b>	95.77	95.65	<b>95.89</b>	95.53	95.93	<b>96.26</b>	95.41	96.06	<b>96.42</b>
8	97.31	97.25	<b>97.62</b>	93.32	93.01	<b>93.42</b>	<b>96.37</b>	96.01	95.96	95.18	<b>96.42</b>	95.49
9	<b>86.74</b>	86.31	86.07	82.05	81.98	<b>82.05</b>	<b>87.38</b>	86.91	85.00	84.6	<b>85.84</b>	84.23
10	88.01	87.37	<b>88.35</b>	87.5	<b>87.63</b>	87.54	77.8	<b>81.65</b>	79.36	77.42	<b>80.93</b>	79.96
11	<b>91.59</b>	91.41	91.38	<b>90.45</b>	90.00	<b>90.31</b>	<b>92.97</b>	91.9	91.86	90.59	<b>90.62</b>	89.62
12	92.5	92.33	<b>93.02</b>	91.8	91.86	<b>92.03</b>	92.79	91.28	<b>93.02</b>	91.34	90.93	<b>91.8</b>
13	<b>90.59</b>	90.04	90.16	<b>87.49</b>	87.29	87.33	<b>85.8</b>	83.14	85.73	<b>86.67</b>	83.29	86.04
14	92.89	92.44	<b>94.33</b>	<b>94.67</b>	94.67	94.44	85.89	86.44	<b>87.33</b>	<b>86.56</b>	85.78	86.33
F1	90.08	89.6	<b>90.15</b>	88.64	88.28	<b>88.68</b>	88.8	88.97	<b>89.01</b>	88.24	<b>88.71</b>	88.7
G-Mean	94.41	94.13	<b>94.45</b>	93.58	93.36	<b>93.6</b>	93.7	93.79	<b>93.82</b>	93.36	<b>93.64</b>	93.63
OA	90.17	89.69	<b>90.24</b>	88.72	88.38	<b>88.76</b>	88.89	89.06	<b>89.07</b>	88.27	<b>88.77</b>	88.76
AA	90.79	90.34	<b>91.04</b>	89.56	89.26	<b>89.58</b>	89.06	89.33	<b>89.36</b>	88.46	<b>88.94</b>	89.09

For each classifier, best performance metric is highlighted in bold. Results in red are the higher metrics for IP scene. Train set is set to 5%.

changes when the training set increases. As a result, the classifiers offer a similar performance when training with 20% of labeled samples.

Unlike the other HS datasets, BW is quite balanced, so in many cases, RAW-data are the best option to train the classifiers. Nevertheless, it is noted that when the number of labeled samples of BW is limited, it is highly recommended to apply preprocessing techniques (PCA or oversampling methods) to reduce the overfitting and/or increase the number of minority class samples. Focusing on MLR, the best oversampling strategies per training are: RANDOM (3%); SMOTEBD1 and SVM-SMOTE (5%); SVM-SMOTE (10%), and RAW, RANDOM, and SMOTE (15%–20%). Regarding SVM, the best strategies are: SMOTE (3%); RAW followed by RANDOM and SMOTE (5%–10%), and RANDOM and SMOTE (15%–20%). Related to MLP: SMOTE (3%); PCA followed by RANDOM (5%); SMOTE(10%); RAW closely followed by RANDOM (15%), and RANDOM oversampling (20%). Finally, MLP achieves the best results in terms of G-mean when using: PCA and RANDOM (3%); PCA and SMOTEBD1 (5%); SMOTE and SVM-SMOTE (10%); SMOTEBD2 (15%), and SVM-SMOTE (20%).

Finally, Table V provides the classification metrics obtained by the spectral classifiers over BW scene, considering 5% of training data. In general, the application of oversampling methods improves the classification results in comparison with RAW

TABLE VI  
CLASSIFICATION RESULTS OF CNN3-D, CNN3-D + OV, ssGAN3-D, AND 3-D-HYPERGAMO USING DISJOINT TRAIN-TEST IP DATASET AND BY RANDOMLY SELECTING 5% TRAINING SAMPLES FROM KSC AND BW DATASETS

Class	Disjoint Indian Pines				Kennedy Space Center				Botswana			
	CNN3-D	CNN3-D+OV	ssGAN3D	3-D-HyperGAMO	CNN3-D	CNN3-D+OV	ssGAN3D	3-D-HyperGAMO	CNN3-D	CNN3-D+OV	ssGAN3D	3-D-HyperGAMO
1	56.0	44.0	<b>94.67</b>	70.0	<b>98.8</b>	98.11	97.83	97.33	99.35	<b>99.74</b>	99.48	99.31
2	72.2	70.18	<b>83.11</b>	33.33	77.78	<b>83.4</b>	80.38	71.68	98.26	<b>100.0</b>	90.28	97.04
3	45.05	49.67	69.14	<b>97.33</b>	88.61	<b>94.51</b>	82.72	88.84	97.9	98.32	95.1	<b>98.45</b>
4	42.76	42.42	36.7	<b>74.24</b>	57.88	53.84	67.78	<b>89.9</b>	91.67	91.5	<b>100.0</b>	91.02
5	72.38	84.55	65.33	<b>96.67</b>	66.45	69.93	52.72	<b>82.99</b>	82.37	81.12	79.95	<b>98.04</b>
6	96.05	<b>99.25</b>	89.83	37.71	88.48	<b>92.78</b>	82.11	77.82	88.01	87.71	93.49	<b>100.0</b>
7	16.67	0.0	0.0	<b>76.16</b>	86.67	96.0	81.0	<b>99.74</b>	99.73	<b>99.86</b>	96.34	99.72
8	93.87	91.47	<b>99.47</b>	98.53	90.3	87.2	88.02	<b>99.58</b>	<b>96.72</b>	91.71	93.78	88.8
9	<b>90.0</b>	73.33	80.0	62.29	<b>97.71</b>	96.49	83.87	94.95	92.51	93.29	<b>96.98</b>	93.23
10	75.48	62.89	<b>88.54</b>	86.29	93.75	94.01	97.83	<b>99.37</b>	94.77	97.88	99.15	<b>99.19</b>
11	75.46	74.4	91.05	<b>96.52</b>	<b>99.5</b>	95.98	97.74	98.85	98.74	97.82	99.31	<b>100.0</b>
12	56.03	49.17	<b>91.49</b>	61.63	96.37	95.96	95.12	<b>98.3</b>	97.87	99.22	96.51	<b>100.0</b>
13	<b>98.75</b>	92.92	96.25	89.13	100.0	100.0	100.0	<b>100.0</b>	99.87	99.74	98.69	<b>99.89</b>
14	91.07	88.99	95.84	<b>98.23</b>	—	—	—	—	95.19	91.11	91.85	<b>98.99</b>
15	78.79	70.37	36.7	<b>89.48</b>	—	—	—	—	—	—	—	—
16	87.88	78.79	48.48	<b>91.92</b>	—	—	—	—	—	—	—	—
OA	75.17	73.39	84.57	<b>86.96</b>	92.48	92.54	90.31	<b>95.31</b>	94.99	94.83	95.43	<b>97.43</b>
AA	71.78	67.03	72.91	<b>78.72</b>	87.87	89.09	85.16	<b>92.26</b>	95.21	94.93	95.06	<b>97.4</b>

Best performance metric is highlighted in bold.

and PCA-data. For instance, the classification measurements obtained by MLR, SVM, and MLP are noticeably improved when using augmented training. Focusing on F1 and G-Mean, the MLR algorithm outperforms the rest of the classifiers. Nevertheless, it is quite interesting that, in the minority class *2-Hippo grass*, PCA is more beneficial for some classifiers.

#### E. Experiment on Deep Learning Classifiers

Currently, deep learning models have established themselves as the current state of the art (SoTA) due to the unparalleled results achieved in automatic image processing. In particular, the CNN has stood out in recent years, thanks to its ability to automatically extract descriptive spatial-spectral features from the data. Notwithstanding the impressive classification result achieved by this architecture [30], their results are significantly degraded by the scarcity of training data and the high variability of the samples. In this sense, oversampling techniques are of great interest to improve the processing of deep networks. Some interesting efforts have been conducted to implement oversampling techniques for deep models. This experiment compares the performance of the baseline CNN3-D, the CNN3-D + OV (with oversampling), the ssGAN, and the 3-D-HyperGAMO models.

Table VI provides the obtained results, in terms of OA and AA. The highest values of the different evaluation metrics among classifiers are represented in bold. Focusing on the DIP dataset (see Fig. 4), the comparison ensures that there is no spatial overlap between both training and testing samples. It is interesting to note that, despite including an oversampling-mechanism (or precisely because of its inclusion), the CNN3-D + OV provides the poorest accuracy results. The complexity of the model, coupled with the sparsity of the data and the large spectral mixture (which increases intraclass variability), prevent the model from achieving better results. Furthermore, the comparison between CNN3-D and CNN3-D + OV highlights the weak performance of the latter for the disjoint IP dataset. On the contrary, 3-D-HyperGAMO model provides the best accuracy, as it extract useful information from those pixels adjacent to the minority classes. Focusing on minority classes, such as 16-*Stone*

*steel towers*, the generation of synthetic samples made by the 3-D-HyperGAMO model enhance effectively their classification in comparison with other models, such as the ssGAN3-D. In contrast, poor results are obtained for the *7-Grass/grass-stone* class with CNN3-D + OV and ssGAN3-D compared to 3-D-HyperGAMO. This is mainly because the methods (CNN3-D, CNN3-D + OV, and ssGAN3-D) fail to extract information for classes with a low number of training samples, as they do not properly cover the features of minority classes. Finally, the oversampling strategy applied to CNN3-D does not introduce any new information, and thus, its classification results are worse. This fact is aggravated by the high complexity of the IP training set. In addition, Fig. 10 depicts the classification maps produced by the CNN3-D, CNN3-D + OV, ssGAN3d, and 3-D-HyperGAMO models. The resulting maps tend to smooth the boundaries between different land cover types. Particularly, the 3-D-HyperGAMO attains a visually comprehensible classification map with a clear and distinguishable border zones, and the noise is very localized and reduced compared to the other deep models. On the contrary, the CNN3-D and CNN3-D + OV result in noisy classification maps with slight differences.

Focusing on the KSC scene, the classifiers have been trained with 5% of labeled samples randomly chosen from the available data. Note that the class imbalance ratio in this scene is lower than in the IP dataset. Nevertheless, the results obtained in Table VI indicate that better results are obtained for almost all land cover classes by alleviating the imbalance problem using oversampling-based models. In this context, the baseline model, i.e., the CNN3-D, achieves the highest accuracy values for classes 1-*Scrub*, 9-*Spartina marsh*, and 11-*Salt marsh*. Nonetheless, regarding the minority classes, such as the 7-*Swap*, the 3-D-HyperGAMO provides a huge improvement (+13.07%) over the baseline model. In contrast to the IP scene, the results obtained on KSC reveal that the random selection of labeled samples provides a significant improvement in classification performance. Once more, the 3-D-HyperGAMO classifier reports the highest overall metrics for OA (95.31%) and AA (92.26%). Moreover, Fig. 11 provides the graphical results of the CNN3-D, CNN3-D + OV, ssGAN3d, and 3-D-HyperGAMO models. The

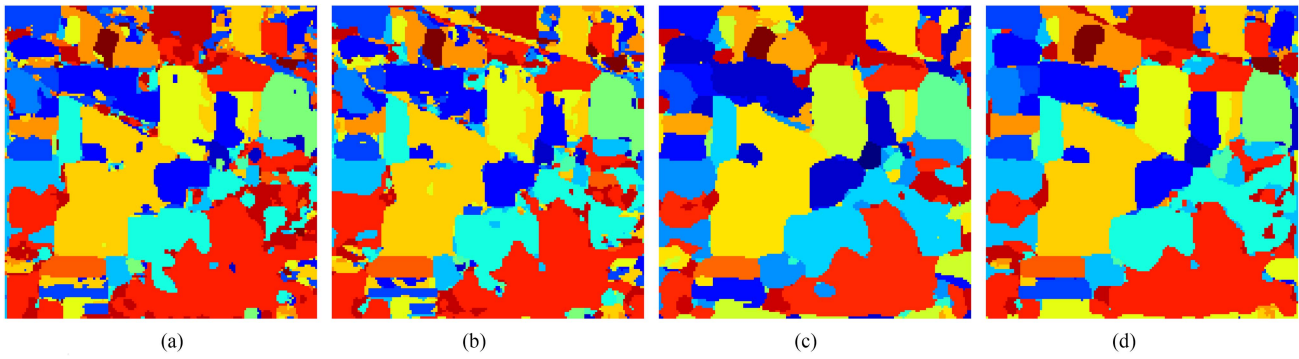


Fig. 10. Classification maps of Indian Pines (IP) obtained from: (a) 3-D convolutional neural network (CNN3-D); (b) CNN3-D with oversampling (CNN3D + OV); (c) semisupervised generative adversarial network (ssGAN); (d) 3-D hyperspectral generative adversarial minority oversampling (3-D-HyperGAMO) classifiers. (a) CNN3-D (75.17%). (b) CNN3-D + OV (73.39%). (c) ssGAN (84.57%). (d) 3-D-HyperGAMO (86.96%).

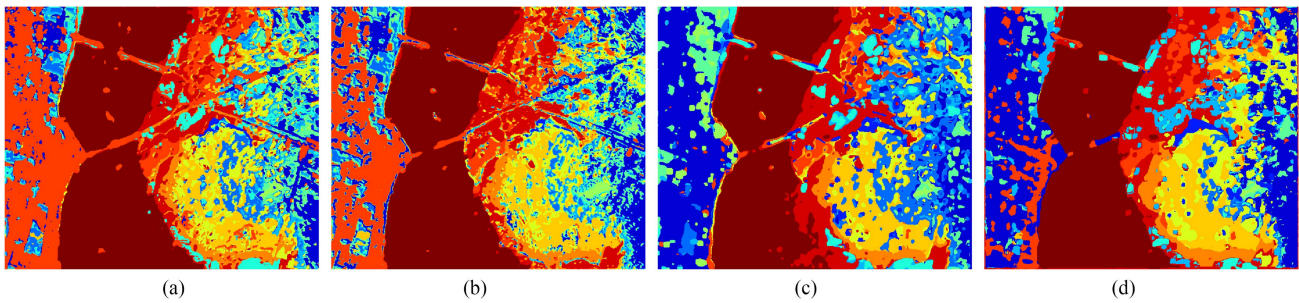


Fig. 11. Classification maps of Kennedy Space Center (KSC) obtained from: (a) 3-D convolutional neural network (CNN3-D); (b) CNN3-D with oversampling (CNN3D + OV); (c) semisupervised generative adversarial network (ssGAN); (d) 3-D hyperspectral generative adversarial minority oversampling (3-D-HyperGAMO) classifiers. (a) CNN3-D (92.48%). (b) CNN3-D + OV (92.54%). (c) ssGAN (90.31%). (d) 3-D-HyperGAMO (95.31%).

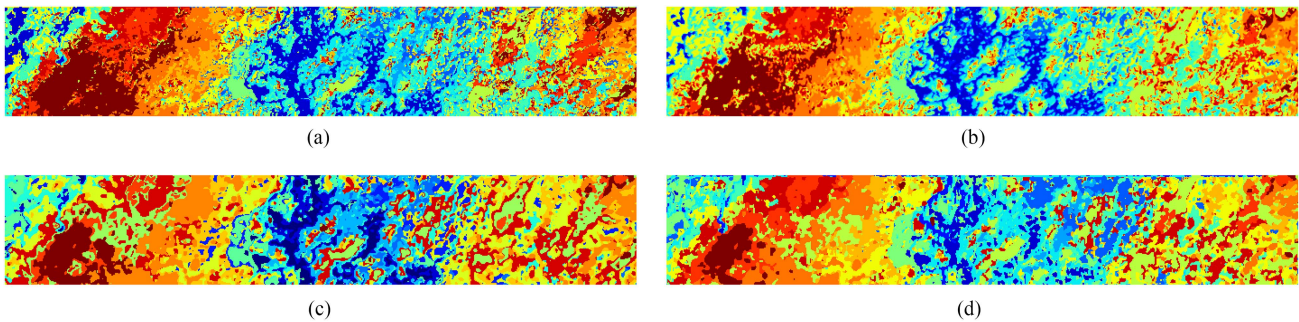


Fig. 12. Classification maps of Botswana (BW) obtained from: (a) 3-D convolutional neural network (CNN3-D); (b) CNN3-D with oversampling (CNN3D + OV); (c) semisupervised generative adversarial network (ssGAN); (d) 3-D hyperspectral generative adversarial minority oversampling (3-D-HyperGAMO) classifiers. (a) CNN-3D (94.99%). (b) CNN3-D + OV (94.83%). (c) ssGAN (95.43%). (d) 3-D-HyperGAMO (97.43%).

CNN3-D and CNN3-D + OV tend to classify the left zone as 11-*Salt marsh*, while the ssGAN3d and 3-D-HyperGAMO models identify the zone as 1-*Scrub*. Nonetheless, the labeled pixels in the test remain correctly classified overall, despite the scarcity of ground truth. In addition, CNN3-D and CNN3-D-OV classify the 12-*Mud flats* and 5-*Oak/Broadleaf* classes oppositely in the center and bottom of the image, although both achieve general improvements compared to ssGAN.

Finally, the performance of the spectral-spatial classifiers on the BW scene is evaluated by randomly selecting 5% of the data for training. It is noteworthy that BW suffers from the lowest class imbalance ratio compared to the IP and KSC scenes. Indeed, the minority and majority classes, i.e., 2-*Hippo grass* and

9-*Accacia woodlands*, contain 101 samples and 314 samples, respectively, with a difference of 213 samples. This indicates an imbalance ratio of approximately 3:1. Obtained results are reported in Fig. 5. Once again, the 3-D-HyperGAMO model outperforms the other classifiers in performance, achieving OA (97.43%) and AA (97.4%). Focusing on some minority class, such as the 2-*Hippo grass*, obtained results show a slight improvement when oversampling is conducted. As in the KSC experiments, the ability to classify minority classes benefits from standard oversampling due to the generation of training data. Fig. 12 depicts the classification maps obtained by the considered classifiers. Similar to previous experiments, the ssGAN3d and 3-D-HyperGAMO models produce quite similar

TABLE VII  
PERFORMANCE EVALUATION OF LOSS FUNCTIONS ON THE AERORIT DATASET

Class	CE-L	FL	A-FL	C-FL	SamplesTR	SamplesVAL	SamplesTE	ImbalanceTR
Roads	79.36	79.21	79.77	81.97	843770	319228	781508	30.93
Buildings	84.14	82.54	79.91	83.41	423605	141424	352788	15.53
Vegetation	96.08	95.59	95.94	96.21	1277105	349211	1551317	46.82
Cars	36.22	39.97	45.70	41.60	70313	19537	42243	2.58
Water	42.01	75.44	76.86	77.53	112946	0	5718	4.14
mIoU	67.56	74.55	75.64	76.14				
OA	93.34	93.41	93.56	94.19				

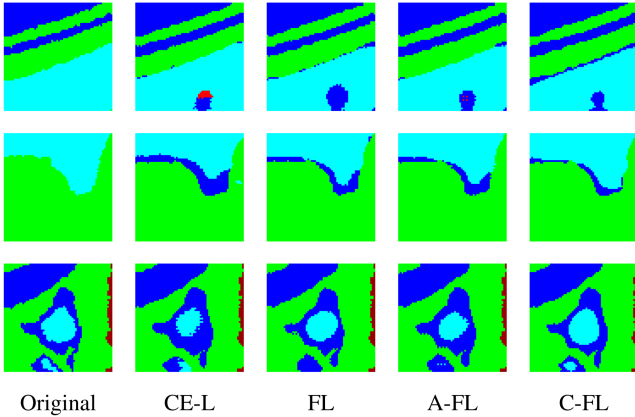


Fig. 13. AeroRIT image patches obtained for cross entropy loss (CE-L), focal loss (FL), asymmetric focal loss (A-FL), and cyclical focal loss (C-FL) experiments. The first row shows patches obtained using a patch size of 170, the second row shows patches obtained using 197, and the third row displays patches obtained using 703. Patches are extracted from the test samples.

results, particularly at the left and right areas of the HS image. In contrast, on the left side of the classification maps, the CNN3-D and CNN3-D-OV classifiers identify a large number of pixels belonging to classes 13-Exposed soils and 12-Mixed mopane.

#### F. Assessing Imbalance Methods for Semantic Segmentation

As stated before, semantic segmentation is a crucial task in computer vision and machine learning applications, where class imbalance poses a common challenge. Indeed, it is pretty common that certain classes are significantly underrepresented in the training data, which can lead to a poor performance of the segmentation model. To tackle this challenge, several imbalance methods have been evaluated, including loss functions such as focal loss (FL), cross entropy (CE), asymmetric focal loss (A-FL), and cyclical focal loss (C-FL). Next, an evaluation of the effectiveness of these loss functions in semantic segmentation tasks is conducted for the AeroRIT dataset.

The behavior of the aforementioned loss functions is presented in Table VII. The mean intersection over union (mIoU) metric is reported for each class along with the respective number of training (SamplesTR), validation (SamplesVAL), and test samples (SamplesTE). It can be observed that a notable imbalance training percentage (ImbalanceTR) is present for the majority of the classes. Specifically, classes *vegetation* and *roads* exhibit an imbalance percentage of 46.82% and 30.93%, respectively, while classes *cars* and *water* have significantly less training data (i.e., minority classes). It is pertinent to note that the overall accuracy (OA) metric exhibits similar results across all

models, thereby rendering it unsuitable for the comprehensive evaluation of a segmentation model performance. However, balance-aware methods, such as FL, A-FL, and C-FL, significantly improve the mIoU, whereas CE performs the worst among the evaluated models due to its inability to address imbalanced classes.

Finally, Fig. 13 presents the prediction patches for studied loss functions, where the aforementioned benefits through the mIoU are observable for the FL, A-FL, and C-FL models. These models shown a better representation of the original. Therefore, obtained findings suggest that balance-aware methods through loss functions should be considered in the development of semantic segmentation models for imbalanced datasets.

#### V. CONCLUSION

This article provides a review of different oversampling and class imbalance methods for the classification of remotely sensed hyperspectral scenes. Specifically, the goal of these methods is to alleviate the problem of class imbalance. Different oversampling algorithms have been reviewed, i.e., Random oversampling, SMOTE, SMOTE BORDERLINE-1, SMOTE BORDERLINE-2, SVM-SMOTE, K-Means SMOTE, and ADASYN. Moreover, comprehensive experiments have been conducted to empirically evaluate the random oversampling, SMOTE, SMOTE BORDERLINE-1, SMOTE BORDERLINE-2, and SVM-SMOTE oversampling methods over widely used machine learning classifiers, such as the MLR, SVM, shallow MLP, and deep MLP, using different amounts of training data. Also, three deep learning approaches have also been tested, i.e., CNN3-D + OV, ssGAN3-D, and 3-D-HyperGAMO. As a result, the impact of oversampling methods during HS data classification has been estimated.

The obtained results demonstrate that the exploitation of oversampling techniques enhances the training procedure, while improving the final classification performance without modifying the operational behaviour of the main classifier. Also, it has also demonstrated the limitations of some oversampling mechanisms, such as K-Means SMOTE and ADASYN, with restrictive constraints on the minimum number of samples per class. On the other hand, it highlights the need to generate new oversampling mechanisms for deep networks that allow a good tradeoff between the complexity of the architecture and the final results. Additionally, the evaluation of imbalance methods in semantic segmentation has revealed several insights. First, it was observed that the traditional cross-entropy loss function struggles with imbalanced datasets, resulting in poor performance for minority classes. This has highlighted the importance of using balance-aware loss functions for addressing class imbalance. Finally, the study has shown that overall accuracy is not a reliable metric for evaluating performance on imbalanced datasets, and mIoU should be preferred instead.

As future work, it is proposed to extend the study performed to new techniques of both oversampling and undersampling, the latter being of great interest, in order to test the classification capabilities after selecting a subset of samples from the original set.



## REFERENCES

- [1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
- [2] R. Fernandez-Beltran, J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "Multimodal probabilistic latent semantic analysis for sentinel-1 and sentinel-2 image fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1347–1351, Sep. 2018.
- [3] R. Fernandez-Beltran, J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "Remote sensing image fusion using hierarchical multimodal probabilistic latent semantic analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4982–4993, Dec. 2018.
- [4] J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [5] R. Fernandez-Beltran, A. Plaza, J. Plaza, and F. Pla, "Hyperspectral unmixing based on dual-depth sparse probabilistic latent semantic analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6344–6360, Nov. 2018.
- [6] X. Tao et al., "Fast orthogonal projection for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5523313.
- [7] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.
- [8] Z. Zou and Z. Shi, "Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1100–1111, Mar. 2018.
- [9] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing [applications corner]," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 119–126, May 2011.
- [10] C. A. Lee, S. D. Gasser, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.
- [11] Y. Ma et al., "Remote sensing Big Data computing: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 51, pp. 47–60, 2015.
- [12] M. E. Paoletti, J. M. Haut, X. Tao, J. P. Miguel, and A. Plaza, "A new GPU implementation of support vector machines for fast hyperspectral image classification," *Remote Sens.*, vol. 12, no. 8, 2020, Art. no. 1257.
- [13] J. A. Benediktsson, J. Chanussot, and W. M. Moon, "Very high-resolution remote sensing: Challenges and opportunities [point of view]," *Proc. IEEE*, vol. 100, no. 6, pp. 1907–1910, Jun. 2012.
- [14] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 11, pp. 2207–2219, Nov. 2016.
- [15] N. Pettorelli et al., "Satellite remote sensing of ecosystem functions: Opportunities, challenges and way forward," *Remote Sens. Ecol. Conserv.*, vol. 4, no. 2, pp. 71–93, 2018.
- [16] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.
- [17] R. O. Green et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, 1998.
- [18] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. van der Piepen, and M. Schroder, "ROSI (reflective optics system imaging spectrometer) - a candidate instrument for polar platform missions," *Proc. SPIE*, vol. 868, pp. 134–141, 1988.
- [19] P. Gong, R. Pu, and J. R. Miller, "Compact airborne spectrographic imager data," *Photogrammetric Eng. Remote Sens.*, vol. 61, pp. 1107–1117, 1995.
- [20] M. T. Eismann, *Hyperspectral Remote Sensing*. Bellingham, WA, USA: SPIE Bellingham, 2012.
- [21] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.
- [22] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.
- [23] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.
- [24] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.
- [25] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 120–147, 2018.
- [26] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.
- [27] M. E. Paoletti et al., "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.
- [28] M. E. Paoletti, S. Moreno-Álvarez, and J. M. Haut, "Multiple attention-guided capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5520420.
- [29] U. B. Gewali, S. T. Monteiro, and E. Saber, "Machine learning based hyperspectral image analysis: A survey," 2018, *arXiv:1802.08701*.
- [30] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 158, pp. 279–317, 2019.
- [31] C.-I. Chang, *Hyperspectral Data Exploitation: Theory and Applications*. New York, NY, USA: Wiley, 2007.
- [32] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.
- [33] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.
- [34] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.
- [35] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5500615.
- [36] R. Grewal, S. S. Kasana, and G. Kasana, "Hyperspectral image segmentation: A comprehensive survey," *Multimedia Tools Appl.*, vol. 82, pp. 20819–20872, 2023, doi: [10.1007/s11042-022-13959-w](https://doi.org/10.1007/s11042-022-13959-w).
- [37] D. Rocchini et al., "Remotely sensed spatial heterogeneity as an exploratory tool for taxonomic and functional diversity study," *Ecological Indicators*, vol. 85, pp. 983–990, 2018.
- [38] Y. Lu et al., "Hyperspectral imaging with cost-sensitive learning for high-throughput screening of loblolly pine (*Pinus taeda* L.) seedlings for freeze tolerance," *Trans. ASABE*, vol. 64, no. 6, pp. 2045–2059, 2021.
- [39] T. Sun, L. Jiao, J. Feng, F. Liu, and X. Zhang, "Imbalanced hyperspectral image classification based on maximum margin," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 522–526, Mar. 2015.
- [40] I. Khosravi and Y. Jouybari-Moghaddam, "Hyperspectral imbalanced datasets classification using filter-based forest methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4766–4772, Dec. 2019.
- [41] M. S. S. Moustafa, S. A. Mohamed, S. Ahmed, and A. H. Nasr, "Hyperspectral change detection based on modification of UNet neural networks," *J. Appl. Remote Sens.*, vol. 15, no. 2, 2021, Art. no. 028505. [Online]. Available: <https://doi.org/10.1117/1.JRS.15.028505>
- [42] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newslett.*, vol. 6, no. 1, pp. 1–6, 2004.
- [43] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, no. 9, pp. 1263–1284, Sep. 2009.
- [44] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
- [45] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017.
- [46] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. García-Borroto, "Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases," *Neurocomputing*, vol. 175, pp. 935–947, 2016.
- [47] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

- [48] P. S. Singh, V. P. Singh, M. K. Pandey, and S. Karthikeyan, "Enhanced classification of hyperspectral images using improvised oversampling and undersampling techniques," *Int. J. Inf. Technol.*, vol. 14, no. 1, pp. 389–396, 2022.
- [49] G. Chen, G. Pei, Y. Tang, T. Chen, and Z. Tang, "A novel multi-sample data augmentation method for oriented object detection in remote sensing images," in *Proc. IEEE 24th Int. Workshop Multimedia Signal Process.*, 2022, pp. 1–7.
- [50] S. Sreelakshmi and S. S. V. Chandra, "Landslide classification using deep convolutional neural network with synthetic minority oversampling technique," in *Distributed Computing and Intelligent Technology*, A. R. Molla, G. Sharma, P. Kumar, and S. Rawat, Eds. Cham, Switzerland: Springer Nature Switzerland, 2023, pp. 240–252.
- [51] Y. Liu, Y. Liu, B. X. Yu, S. Zhong, and Z. Hu, "Noise-robust oversampling for imbalanced data classification," *Pattern Recognit.*, vol. 133, 2023, Art. no. 109008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320322004885>
- [52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [53] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.
- [54] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proc. Int. Joint Conf. AI*, 1999, vol. 55, p. 60.
- [55] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.
- [56] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," in *Proc.: 5th Int. Workshop Comput. Intell. Appl.* 2009, vol. 2009, no. 1, pp. 24–29.
- [57] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *Int. J. Knowl. Eng. Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011. [Online]. Available: <https://doi.org/10.1504/ijkesdp.2011.039875>
- [58] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018. [Online]. Available: <https://doi.org/10.1016/j.ins.2018.06.056>
- [59] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, 2008, pp. 1322–1328.
- [60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
- [61] L. N. Smith, "Cyclical focal loss," 2022.
- [62] T. Ridnik et al., "Asymmetric loss for multi-label classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 82–91.
- [63] A. Rangnekar, N. Mokashi, E. J. Ientilucci, C. Kanan, and M. J. Hoffman, "AeroRIT: A new scene for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8116–8124, Nov. 2020.
- [64] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.



**Mercedes E. Paoletti** (Senior Member, IEEE) received the Ph.D. degree in information technology from the Department of Technology of Computers and Communications, University of Extremadura, Badajoz, Spain, in 2020, supported by a University Teacher Training Programme, Spanish Ministry of Education.

She is currently serving as a Researcher with the University of Extremadura. Her research interests include remote sensing analysis through DL models and high performance computing.

Dr. Paoletti has served as a Reviewer for IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and *IEEE Geoscience and Remote Sensing Letters*, in which she was recognized as a best reviewer in 2019 and 2020. She was also a recipient of the 2019 Outstanding Paper Award recognition in the IEEE WHISPERS 2019 conference, and a recipient of the Outstanding Ph.D. Award at the University of Extremadura in 2020.



**Oscar Mogollon-Gutierrez** received the M.Sc. degree in computer engineering in 2022 from the University of Extremadura, Badajoz, Spain, where he is currently working toward the Ph.D. degree in computer science.

He is a Researcher with the Department of Computer and Telematics Systems Engineering and Member of the Media Engineering Group, University of Extremadura. His research interests include imbalanced learning and machine learning.

Mr. Mogollon-Gutierrez has participated in several conferences.



**Sergio Moreno-Álvarez** (Graduate Student Member, IEEE) received the bachelor's and the master's degrees in computer engineering in 2017 and 2019, respectively, from the University of Extremadura, Badajoz, Spain, and the Ph.D. degree in information technology from the Department of Computer Systems Engineering and Telematics, University of Extremadura, in 2022.

He has participated in regional and national projects. He is currently a Researcher with the University of Extremadura. He has authored or coauthored

nine JCR papers in international journals and five presentations at international and national conferences. His research interests include heterogeneous systems and high performance computing.



**Jose Carlos Sancho** received the M.Sc. and Ph.D. degrees in computer science from the University of Extremadura, Badajoz, Spain, in 2014 and 2021, respectively.

He has been a Substitute Professor with the Department of Computer and Telematics Systems Engineering, University of Extremadura, since 2018. He has authored or coauthored four JCR papers and several presentations at international conferences. His research interests include software audit and software development.



**Juan M. Haut** (Senior Member, IEEE) received the Ph.D. degree in information technology in 2019, from the University of Extremadura, Caceres, Spain, supported by a University Teacher Training Programme, Spanish Ministry of Education.

He is currently a Professor with the Department of Computers and Communications, University of Extremadura. Also, he is a Member of the Hyperspectral Computing Laboratory (HyperComp), University of Extremadura. He has authored/coauthored more than 50 JCR journal articles (more than 30 in IEEE

journals) and more than 30 peer-reviewed conference proceeding papers. His research interests include remote sensing data processing and high dimensional data analysis, applying machine/deep learning and cloud computing approaches.

Dr. Haut was the recipient of the Outstanding Ph.D. Award at the University of Extremadura in 2019. Some of his contributions have been recognized as hot-topic publications for their impact on the scientific community. Also, he was the recipient of the Outstanding Paper Award in the 2019 and 2021 IEEE WHISPERS conferences. He is a reviewer of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, and *IEEE Geoscience and Remote Sensing Letters*, and he has been awarded with the Best Reviewer recognition of *IEEE Geoscience and Remote Sensing Letters*, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2018 and 2020, respectively. Furthermore, he has guest-edited three special issues on hyperspectral remote sensing for different journals. He is also an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *IEEE Geoscience and Remote Sensing Letters*, and *IEEE Journal on Miniaturization for Air and Space Systems*.