Increasing the Effectiveness of Active Learning:
Introducing Artificial Data Generation in Active Learning for Land Use/Land Cover Classification

Joao Fonseca[1], Georgios Douzas[1], Fernando Bacao[1*]

[1]NOVA Information Management School, Universidade Nova de Lisboa

*Corresponding Author

Postal Address: NOVA Information Management School, Campus de Campolide, 1070-312 Lisboa, Portugal

Telephone: +351 21 382 8610

In remote sensing, Active Learning (AL) has become an important technique to collect informative ground truth data "on-demand" for supervised classification tasks. In spite of its effectiveness, it is still significantly reliant on user interaction, which makes it both expensive and time consuming to implement. Most of the current literature focuses on the optimization of AL by modifying the selection criteria, the chooser and/or predictors used. Although improvements in these areas will result in more effective data collection, the use of artificial data sources to reduce human-computer interaction remains unexplored. In this paper, we introduce a new component to the typical AL framework, the data generator, a source of artificial data to reduce the amount of user-labeled data required in AL. The implementation of the proposed AL framework is done using SMOTE and Geometric SMOTE as data generators. We compare the new AL framework to the original one using similar acquisition functions and predictors over three AL-specific performance metrics in seven benchmark datasets. We show that this modification to the AL framework significantly reduces cost and time requirements for a successful AL implementation in the context of remote sensing.

1 Introduction

The technological development of air and space borne sensors, as well as the increasing number of remote sensing missions have allowed the continuous collection of large amounts of high quality remotely sensed data. This data is often composed of multi and hyper spectral satellite imagery, essential for numerous applications, such as Land Use/Land Cover (LULC) change detection, ecosystem management [Nagai et al., 2020], agricultural management [Huang et al., 2018], water resource management [Wang and Xie, 2018], forest management, and urban monitoring [Khatami et al., 2016]. Despite LULC maps being essential for most of these applications, their production still a challenging task [Gavade and Rajpurohit, 2019, Wulder et al., 2018]. They can be updated using either one of the following strategies:

1. Photo-interpreted. Consists of evaluating a patch's LULC class based on orthophoto and satellite image interpretation [Costa et al., 2020]. This method guarantees a decent level of accuracy,

as it is dependent on the interpreter's expertise and human error. Typically, it is an expensive, time-consuming task that requires the expertise of a photo-interpreter. This task is also frequently applied to obtain ground-truth labels for training and/or validating Machine Learning (ML) algorithms for related tasks [Vermote et al., 2020, Costantino et al., 2020].

2. Automated mapping. It is based on the usage of a ML method or a combination of methods in order to obtain an updated LULC map. The development of a reliable automated method is still a challenge among the ML and remote sensing community, since the efficacy of existing methods vary across applications and geographical areas [Gavade and Rajpurohit, 2019]. Typically, this method requires the existence of ground-truth data, which is frequently outdated or nonexistent for the required time frame [Nagai et al., 2020]. On the other hand, employing a ML method provides readily available and relatively inexpensive LULC maps. The increasing quality of state-of-the-art classification methods have motivated the application and adaptation of these methods in this domain [Maxwell et al., 2018].

3. Hybrid approaches. They employ photo-interpreted data to augment the training dataset and improve the quality of automated mapping [Růžička et al., 2020]. It attempts to accelerate the photo-interpretation process by selecting a smaller sample of the study area to be interpreted. The goal is to minimize the inaccuracies found in the LULC map by supplying high-quality ground-truth data to the automated method. The final (photo-interpreted) dataset consists of only the most informative samples, i.e., patches that are typically difficult to classify for a traditional automated mapping method [Liu et al., 2020].

The latter method is best know as AL. It is especially useful whenever there is an absence of ground-truth data and/or the mapping region does not contain updated LULC maps [Su et al., 2020]. In a context of limited sample-collection budget, the collection of the most informative samples capable of optimally increasing the classification accuracy of a LULC map is of particular interest [Su et al., 2020]. AL attempts to minimize the human-computer interaction involved in photo-interpretation by selecting the data points to include into the annotation process. These data points are selected based on an uncertainty measure and represent the points close to the decision borders. Afterwards, they are passed on for photo-interpretation and added to the training dataset, while the points with the lowest uncertainty values are ignored for photo-interpretation and classification. This process is iterated until a convergence criterion is reached [Pasolli et al., 2016].

The relevant work developed within AL is described in detail in Section 2. The research attempts to address some of the challenges found in AL, mainly inherited from automated and photo-interpreted mapping: mapping inaccuracies and time consuming human-computer interactions. These challenges have different sources:

1. Human error. The involvement of photo-interpreters in the data labeling step carries an additional risk to the creation of LULC patches. The minimum mapping unit being considered, as well as the quality of the orthophotos and satellite images being used, are some of the factors that may lead to the overlooking of small-area LULC patches and label-noisy training data [Pelletier et al., 2017].

2. High-dimensional datasets. The amount of bands (i.e., features) present in multi and hyper spectral images introduce an increased level of complexity in the classification step [Stromann et al., 2020]. These datasets are often prone to the Hughes phenomenon, also known as the curse of dimensionality.

3. Class separability. Producing an LULC map considering classes with similar spectral signatures makes them difficult to separate [Alonso-Sarria et al., 2019]. A lower pixel resolution of the satellite

images may also imply mixed-class pixels, which may lead to both lower class separability as well as higher risk of human error.

4. Existence of rare land cover classes. The varying morphologies of different geographical regions naturally implies an uneven distribution of land cover classes [Feng et al., 2018]. This is particularly relevant in the context of AL: the data selection method is based on a given uncertainty measure over data points whose class label is unknown. Consequently, AL's iterative process of data selection may disregard wrongly classified land cover areas belonging to a minority class.

Research developed in the field of Active Learning typically focus on the reduction of human error by minimizing the human interaction with the process through the development of more efficient choosers and selection criteria within the generally accepted AL framework. Concurrently, the problem of rare land cover classes is rarely addressed. This is a frequent problem in the ML community, known as the Imbalanced Learning problem. This problem exists whenever there is an uneven between-class distribution in the dataset [Chawla et al., 2004]. Specifically, most classifiers attempt to optimize metrics such as overall accuracy, which are designed to work primarily with balanced datasets. Consequently, these metrics tend to introduce a bias towards the majority class by attributing an importance to each class proportional to its relative frequency [Maxwell et al., 2018]. As an example, such a classifier could achieve an overall accuracy of 99% on a binary dataset where the minority class represents 1% of the overall dataset and still be deemed useless. A number of methods have been developed to deal with this problem. They can be categorized into three different types of approaches [Fernández et al., 2013, Kaur et al., 2019]. Cost-sensitive solutions perform changes to the cost matrix in the learning phase. Algorithmic level solutions modify specific classifiers to reinforce learning on minority classes. Resampling solutions modify the dataset by removing majority samples and/or generating artificial minority samples. The latter is independent from the context and can be used alongside any classifier. We will focus on artificial data generation techniques, presented in Section 3.

In this paper, we propose a novel AL framework to address two limitations commonly found in the literature: minimize human-computer interaction and reduce the class imbalance bias. This is done with the introduction of an additional component in the iterative AL procedure (the generator), used to generate artificial data to both balance and augment the training dataset. The introduction of this component is expected to reduce the number of iterations required until convergence of the predictor's quality.

This paper is organized as follows: Section 1 exposes the problem and its context, Sections 2 and 3 describe the state of the art in AL and Oversampling techniques, Section 4 exposes the proposed method, Section 5 covers the datasets, evaluation metrics, ML classifiers and experimental procedure, Section 6 presents the results and statistical analyses and Section 7 reports the conclusions drawn from our findings.

2 Active Learning Approaches

As the amount of unlabeled data increases, the interest and practical usefulness of AL follows that trend [Kottke et al., 2017]. AL is used as the general definition of frameworks aiming to train a learning system in multiple steps, where a set of new data points are chosen and added to the training dataset each time [Růžička et al., 2020]. Typically, an AL framework is composed of 10 elements [Sverchkov and Craven, 2017, Su et al., 2020, Růžička et al., 2020]:

1. Data source. In the context of LULC classification, the data source is usually a hyper/multi-spectral image, a Synthetic-aperture radar (SAR) image, or a composite image.

2. Unlabeled dataset. Consists of a sample of the original data source. It is used in combination with the chooser and the selection criterion to retrieve uncertainty estimates on each iteration.

3. Initial training sample. It is a small sample of the unlabeled dataset, used to initiate the first AL iteration. The size of the initial training sample normally varies between no observations at all and 10% [Li and Guo, 2013].

4. Augmented training dataset. This dataset is the concatenation of the labeled initial training sample along with the datasets labeled by the oracle in past iterations.

5. Uncertainty map. The dataset containing the highest uncertainty points/patches to be labeled by the oracle.

6. Oracle. An external entity to which the uncertainty map is presented to. The oracle is responsible for annotating unlabeled samples to be added to the augmented dataset. In remote sensing, the oracle is typically a photo-interpreter, as is the case in [Li et al., 2020]. Some of the research also refers to the oracle as the *supervisor* [Su et al., 2020, Shrivastava and Pradhan, 2021].

7. Chooser. Produces the class probabilities for each unlabeled sample. This is a classifier trained using the augmented dataset. It is used to estimate the class probabilities for each sample over the unlabeled dataset.

8. Selection criterion. It quantifies the chooser's uncertainty level for each sample belonging to the unlabeled dataset. It is typically based on the class probabilities assigned by the chooser. In some situations, the chooser and the selection criterion are grouped together under the concept *acquisition function* [Růžička et al., 2020] or *query function* [Su et al., 2020]. Some of the literature refers to the selection criterion by using the concept *sampling scheme* [Liu et al., 2020].

9. Predictor. The classifier used to infer the land cover classes for the final output map. Once a stopping criterion is met, the classifier is trained using the augmented dataset and the LULC classes are inferred from the data source.

10. Prediction output. In the context of LULC classification, the prediction output is the estimated LULC map raster.

Figure 1 schematizes the steps involved in a complete AL iteration. For a better context within the remote sensing domain, the prediction output is identified as the LULC map. This framework starts by collecting unlabeled data from the original data source. It is used to generate a random initial training sample and is labeled by the oracle. In practical applications, the oracle is frequently a group of photo-interpreters [Kottke et al., 2017]. The chooser is trained on the resulting dataset and is used to predict the class probabilities on the unlabeled dataset. The class probabilities are fed into a selection criterion to estimate the prediction's uncertainty, out of which the samples with the highest uncertainty will be selected. This calculation is motivated by the absence of labels in the uncertainty dataset. Therefore, it is impossible to estimate the prediction's accuracy in a real case scenario. The iteration is completed when the selected points are tagged by the oracle and added to the training dataset (i.e., the augmented dataset).

A common challenge found in AL tasks is ensuring the consistency of AL over different initializations [Kottke et al., 2017]. There are two factors involved in this phenomenon. On the one hand, the implementation of the same method over different initializations may result in significantly different

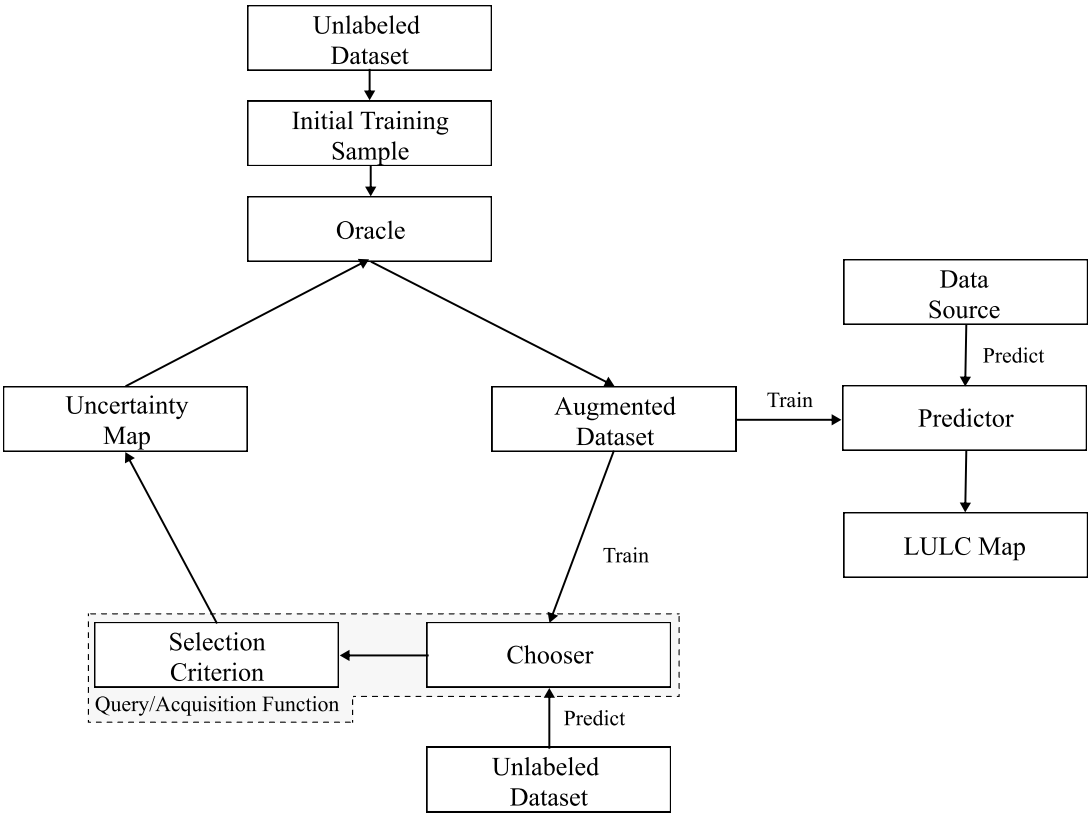

Figure 1: Typical AL framework.

initial training samples, amounts to varying accuracy curves. On the other hand, the lack of a robust selection criterion and/or chooser may also result in inconsistencies across AL experiments with different initializations. This phenomenon was observed and documented in a LULC classification context in [Tuia et al., 2011].

Selecting an efficient selection criterion is particularly important to find the samples closest to the decision border (i.e., samples difficult to classify) [Shrivastava and Pradhan, 2021]. Therefore, most of AL related studies focus on the design of the query/acquisition function [Su et al., 2020].

2.1 Non-informed selection criteria

Only one non-informed selection criterion was found. Random sampling selects unlabeled samples without considering any external information produced by the chooser. Since the method for selecting the unlabeled samples is random, this method disregards the usage of a chooser and is comparatively worse than any other selection criterion. Although, random sampling is still a powerful baseline method [Cawley, 2011]. Generally, different AL initializations return high performance variability [Kottke et al., 2017]. When this happens, the analysis of the mean performances over multiple repetitions is not of interest on its own. Instead, it is preferable to do pairwise comparison of different methods along with their corresponding variances.

2.2 Ensemble-based selection criteria

Ensemble disagreement is based on the class predictions of a set of classifiers. The disagreement between all the predictions for a given observation is a common measure for uncertainty, although computationally inefficient [Růžička et al., 2020, Pasolli et al., 2016]. It is calculated using the set of classifications over a single observation, given by the number of votes assigned to the most frequent class [Shrivastava and Pradhan, 2021]. This method was implemented successfully for complex applications such as deep active learning [Růžička et al., 2020].

Multiview [Muslea et al., 2006] consists on the training of multiple independent classifiers using different views, which correspond to the selection of subsets of features or observations in the dataset. Therefore, it can be seen as a bootstrap aggregation (bagging) ensemble disagreement method. It is represented by the maximum disagreement score out of set of disagreements calculated for each view [Shrivastava and Pradhan, 2021]. A lower value for this metric means a higher classification uncertainty. Multiview-based maximum disagreement has been successfully applied to hyper-spectral image classification in [Di and Crawford, 2012] and [Zhou et al., 2014].

An adapted disagreement criterion for an ensemble of k-nearest neighbors has been proposed in [Pasolli et al., 2016]. This method employs a k-nearest neighbors classifier and computes an instance's classification uncertainty based on the neighbors' class frequency using the maximum disagreement metric over varying values for k. As a result, this method is comparable to computing the dominant class' score over a weighted k-nearest neighbors classifier. This method was also used on a multimetric active learning framework [Zhang et al., 2016].

Another relevant ensemble-based selection criterion is the binary random forest-based query model [Su et al., 2020]. This method employs a one-versus-one ensemble method to demonstrate an efficient data selection method using the estimated probability of each binary random forest and determining the classification uncertainty based on the probabilities closest to 0.5 (i.e., the least separable pair of classes are used to determine the uncertainty value). Although, this study fails to compare the proposed method with other benchmark methods, such as random sampling.

2.3 Entropy-based criteria

A number of contributions have focused on entropy-based querying. The application of entropy is common among active deep learning applications [Aghdam et al., 2019], where the training of an ensemble of classifiers is often too expensive. The measure of entropy is formulated as follows:

$$H(x_i) = \sum_{\omega=1}^{N_i} p(y_i^* = \omega|x_i) \log_2[p(y_i^* = \omega|x_i)] \tag{1}$$

The measurement of entropy H is based on the observed probability $p(y_i^* = \omega|x_i)$ of obtaining class ω as the predicted class label y_i^*, where N_i is the number classes predicted for observation x_i.

Entropy query-by-bagging (EQB), also defined as maximum entropy [Liu et al., 2020], is an ensemble approach of the entropy selection criterion, originally proposed in [Tuia et al., 2009]. This strategy uses the set of predictions produced by the ensemble classifier to calculate those many entropy measurements. The estimated uncertainty measure for one sample is given by the maximum entropy within that set. EQB was observed to be an efficient selection criterion. Specifically, [Shrivastava and Pradhan, 2021] applied EQB on hyper-spectral remote sensing imagery using Support Vector Machines (SVM) and

Extreme Learning Machines (ELM) as choosers, achieving optimal results when combining EQB with ELM. Another study successfully implemented this method on an active deep learning application [Liu et al., 2020]. Another study improved over this method with a normalized EQB selection criterion [Copa et al., 2010].

2.4 Other relevant criteria

Margin Sampling is a SVM-specific criterion, based on the distance of a given point to the SVM's decision boundary [Shrivastava and Pradhan, 2021]. This method is less popular than the remaining methods because it is limited to one type of chooser (SVMs). One extension of this method is the multiclass level uncertainty [Shrivastava and Pradhan, 2021], calculated by subtracting the observation's distance to the decision boundaries of the two most probable classes [Demir et al., 2011].

The Mutual Information-based (MI) criterion selects the new training samples by maximizing the mutual information between the classifier and class labels in order to select samples from regions that are difficult to classify. Although this method is commonly used, it is frequently outperformed by the breaking ties selection criterion [Li et al., 2011, Liu et al., 2018].

The breaking ties (BT) selection criterion was originally introduced in [Luo et al., 2003]. It is formulated as follows:

$$BT(x_i) = \arg \min_{x_i, i \in S_u} \{\max_{\omega \in N} p(y_i^* = \omega | x_i) - \max_{\omega \in N \setminus \{\omega^+\}} p(y_i^* = \omega | x_i)\} \tag{2}$$

Which is the subtraction of the probabilities of the two most likely classes. Another related method is Modified Breaking Ties scheme (MBT), which aims at finding the samples containing the largest probabilities for the dominant class [Liu et al., 2018, Li et al., 2013]

Another type of selection criteria identified is the loss prediction method [Yoo and Kweon, 2019]. This method replaces the selection criterion with a predictor whose goal is to estimate the chooser's loss for a given prediction. This allows the new classifier to estimate the prediction loss on unlabeled observations and select the ones with the highest predicted loss.

Some of the literature fails to specify the strategy employed, although inferring it is generally intuitive. For example, [Ertekin et al., 2007] successfully used AL to address the imbalanced learning problem. They employed an ensemble of SVMs as the chooser and predictor, as well as an ensemble-based selection criterion. All of the research found related to this topic focused on the improvement of AL through modifications on the selection criterion, chooser or predictor. None of these publications proposed significant variations to the original AL framework.

3 Artificial Data Generation Approaches

The generation of artificial data is a common approach to address imbalanced learning tasks [Kaur et al., 2019], as well as improving the effectiveness of supervised learning tasks [DeVries and Taylor, 2017]. In recent years some sophisticated data generation approaches were found. Although, the scope of this work is to propose the integration of a generator within the AL framework. Due to the complexity and

computational cost of network-based approaches (e.g., Generative Adversarial Networks), we will only focus on heuristic data generation approaches, specifically, oversamplers.

Heuristic data resampling methods employ local and/or global information to generate new, relevant, non-duplicated instances. These methods are most commonly used to populate minority classes and balance the between-class distribution of a dataset. The Synthetic Minority Oversampling Technique (SMOTE) [Chawla et al., 2002] was the first heuristic oversampling algorithm to be proposed. The simplicity and effectiveness of this method contributes to its prevailing popularity. It generates a new instance $\overrightarrow{z}$ through a linear interpolation of a randomly selected minority-class observation $\overrightarrow{x}$ and one of its randomly selected k-nearest neighbors $\overrightarrow{y}$ such that $\overrightarrow{z} = \alpha\overrightarrow{x} + (1-\alpha)\overrightarrow{y}$ where α is a random float between 0 and 1, as shown in Figure 2.

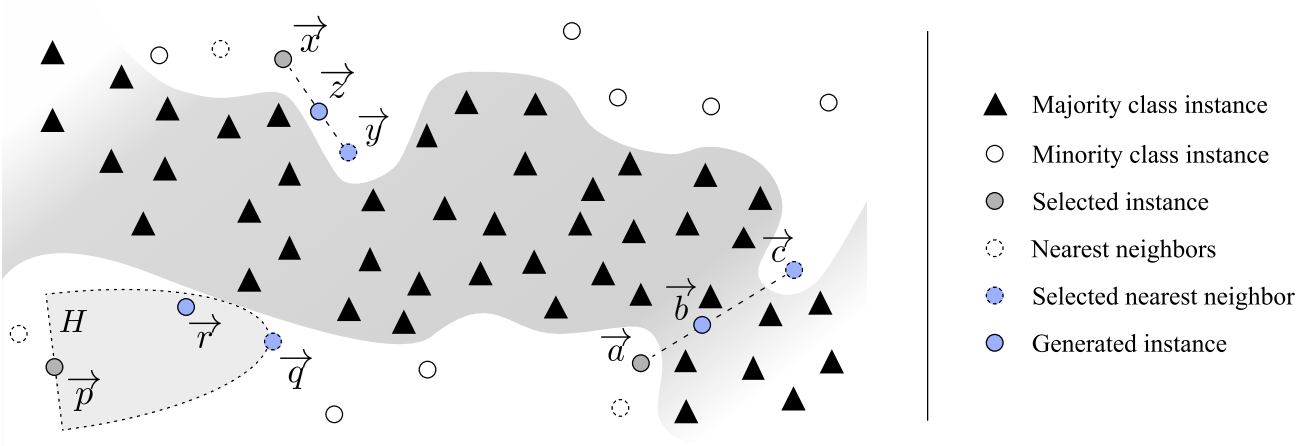

Figure 2: Examples of SMOTE and G-SMOTE generation process.

The implementation of SMOTE for LULC classification tasks has been found to improve the quality of the predictors used [Jozdani et al., 2019, Bogner et al., 2018]. Despite its popularity, its drawbacks motivated the development of other oversampling methods [Douzas and Bacao, 2019]:

1. Generation of noisy samples due to the selection of k-nearest neighbors and initial observation. The selection of a sample and/or neighboring sample located inside a majority class region may produce artificial samples within that region and amplify noisy data. Borderline-SMOTE [Han et al., 2005] is a modification of SMOTE in which only the minority examples near the borderline are over-sampled. This method avoids the generation of noisy samples by disregarding minority samples located in a majority class region as well as samples distant from the decision borders. The Adaptive Synthetic Sampling approach (ADASYN) [He et al., 2008] uses a density distribution ratio to address this limitation and focus the artificial data generation on minority class regions that are more difficult to classify.

2. Generation of noisy instances due to the use of observations from two different minority class clusters. Choosing a minority sample $\overrightarrow{a}$ and one of its nearest neighbors $\overrightarrow{b}$ belonging to a different minority cluster may lead to the generation of a sample $\overrightarrow{c}$ located within the two classes, as shown in Figure 2. K-means SMOTE [Douzas et al., 2018] and Self-Organizing map oversampling (SOMO) [Douzas and Bacao, 2017] reduce this effect by oversampling minority class samples within the same clusters.

3. Generation of nearly duplicated instances. The linear interpolation of parent samples that are close to each other produces an artificial sample with similar properties as its parents. Geometric SMOTE (G-SMOTE) [Douzas and Bacao, 2019] introduces a modification of the SMOTE algorithm

in the data generation mechanism to produce artificial samples with higher variability.

The G-SMOTE algorithm is introduced as a generalization of the vanilla SMOTE. Instead of generating artificial data as a linear combination of the parent samples, it is done within a deformed, truncated hyper-spheroid. G-SMOTE generates an artificial sample $\overrightarrow{r}$ within a hyper-spheroid H, formed by selecting a minority sample $\overrightarrow{p}$ and one of its nearest neighbors $\overrightarrow{q}$, as shown in Figure 2. The truncation and deformation parameters define the shape of the spheroid's geometry. The method also modifies the selection strategy for the k-nearest neighbors, accepting the generation of artificial samples using observations from different classes. G-SMOTE has shown superior performance when compared with other oversampling methods for LULC classification tasks, regardless of the classifier used [Douzas et al., 2019].

4 Proposed method

Within the literature identified, most of the work developed in the AL domain revolved around improving the quality of the chooser, predictor and/or selection criterion. Although these methods allow earlier convergence of the AL iterative process, the impact of these methods are only observed between iterations. Consequently, none of these contributions focused on the definition of decision borders within iterations. The method proposed in this paper modifies the AL framework by introducing an artificial data generation step within AL's iterative process. We define this component as the generator and is intended to be integrated into the AL framework as shown in Figure 3.

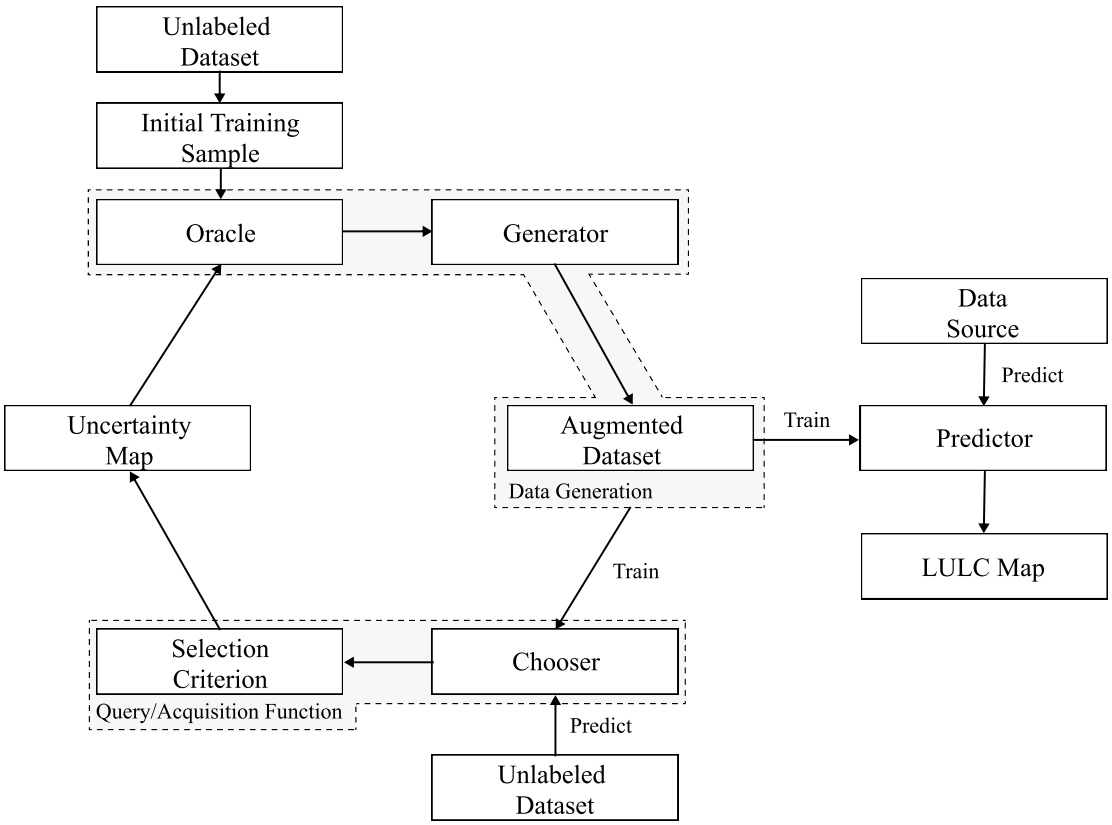

Figure 3: Proposed AL framework.

This method leverages the capacity of artificial data to introduce more data variability into the augmented dataset and facilitate the chooser's training phase with a more consistent definition of the decision boundaries at each iteration. Therefore, any algorithm capable of producing artificial data, be it domain agnostic or specific, can be employed. The artificial data is only used to train the classifiers involved in the process (chooser and predictor) and is discarded once the chooser's training phase is completed. The remaining steps in the AL framework remain unchanged. This method is addressed towards the limitations found in the previous sections:

1. The convergence of the predictor's performance should be anticipated with the clearer definition of the decision boundaries across iterations.

2. Annotation cost is expected to reduce as the need for labeled observations reduces along with the early convergence of the classification performance.

3. The class imbalance bias observed in typical classification tasks, as well as in AL is mitigated by balancing the class frequencies at each iteration.

Although the performance of this method is shown within a LULC classification context, the proposed framework is independent from the domain. The high dimensionality of remotely sensed imagery make its classification particularly challenging when the availability of labeled data is scarce and/or comes at a high cost, being subjected to the curse of dimensionality. Consequently, it is a relevant and appropriate domain to test this method.

5 Methodology

In this section we describe the datasets, evaluation metrics, oversamplers, classifiers, software used and the procedure developed. We demonstrate the proposed method's efficiency over 7 datasets, sampled from publicly available, well-known benchmark remote sensing landscapes frequently found in the literature. The datasets and sampling strategy are described in Subsection 5.1. On each of these datasets, we implement 3 different classifiers over the entire training set to estimate the optimal classification performance, the original AL framework as the baseline reference and the proposed method using two different generators, described in Subsection 5.2. The metrics used to estimate the performance of these algorithms are described in Subsection 5.3. Finally, the experimental procedure is described in Subsection 5.4.

Our methodology focuses on two objectives: (1) Comparison of optimal classification performance among active learners and traditional supervised learning and (2) Comparison of classification convergence efficiency across AL frameworks.

5.1 Datasets

The datasets used were extracted from publicly available hyperspectral scenes. Additionally, all datasets were collected using the same sampling procedure. The description of the hyperspectral scenes used in this study is provided in Table 1. These scenes were chosen because of their popularity in the research

community and their high baseline classification scores. Consequently, demonstrating an outperforming method in this context is particularly challenging and valuable.

Dataset	Sensor	Location	Dimension	Bands	Res. (m)	Classes
Botswana	Hyperion	Okavango Delta	1476 x 256	145	30	14
Salinas A	AVIRIS	California, USA	86 x 83	224	3.7	6
Kennedy Space Center	AVIRIS	Florida, USA	512 x 614	176	18	16
Indian Pines	AVIRIS	NW Indiana, USA	145 x 145	220	20	16
Salinas	AVIRIS	California, USA	512 x 217	224	3.7	16
Pavia University	ROSIS	Pavia, Italy	610 x 610	103	1.3	9
Pavia Centre	ROSIS	Pavia, Italy	1096 x 1096	102	1.3	9

Table 1: Description of the hyperspectral scenes used for this experiment.

The Indian Pines scene [Baumgardner et al., 2015] is composed of agriculture fields in approximately two thirds of its coverage, low density buildup areas and natural perennial vegetation in the remainder of its area (see Figure 4a). The Pavia Centre and University scenes are hyperspectral, high-resolution images containing ground truth data composed of urban-related coverage (see Figures 4b and 4c). The Salinas and Salinas A scenes contain at-sensor radiance data. As subset of Salinas, the Salinas A scene contains contains the vegetables fields present in Salinas and the latter is also composed of bare soils and vineyard fields (see Figures 4d and 4e). The Botswana scene contains ground truth data composed of seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta (see Figure 4f). The Kennedy Space Center scene contains a ground truth composed of both vegetation and urban-related coverage (see Figure 4g)

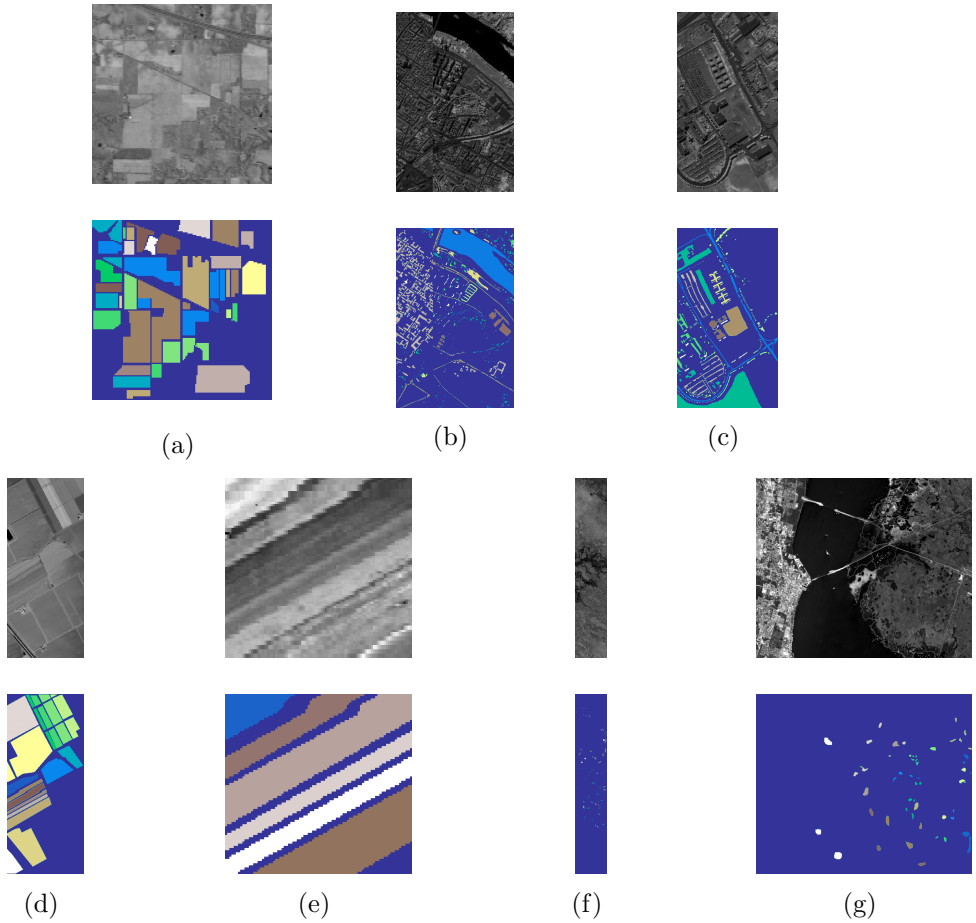

Figure 4: Gray scale visualization of a band (top row) and ground truth (bottom row) of each scene used in this study. (a) Indian Pines, (b) Pavia Centre, (c) Pavia University, (d) Salinas, (e) Salinas A, (f) Botswana, (g) Kennedy Space Center

The sampling strategy is similar to all datasets. The pixels without a ground truth label are first discarded. All the classes with cardinality lower than 150 are also discarded. This is done to maintain feasible Imbalance Ratios (IR) across datasets (where $IR = \frac{count(C_{maj})}{count(C_{min})}$). Finally, a stratified sample of 1500 observations are selected for the experiment. The resulting datasets are described in Table 2. The motivation for this strategy is three fold: (1) reduce the datasets to a manageable size and allow the experimental procedure to be completed within a feasible time frame, (2) ensure the relative class frequencies in the scenes are preserved and (3) ensure equivalent analyses across datasets and AL frameworks. In this context, a fixed number of observations per dataset is especially important to standardize the AL-related performance metrics.

Dataset	Features	Instances	Min. Instances	Maj. Instances	IR	Classes
Botswana	145	1500	89	154	1.73	12
Salinas A	224	1500	109	428	3.93	6
Kennedy Space Center	176	1500	47	272	5.79	12
Indian Pines	220	1500	31	366	11.81	12
Salinas	224	1500	25	312	12.48	16
Pavia University	103	1500	33	654	19.82	9
Pavia Centre	102	1500	27	668	24.74	9

Table 2: Description of the datasets used for this experiment.

5.2 Machine Learning Algorithms

We use two different types of ML algorithms. Data generation algorithms, used to form the generator, and classification algorithms, used to form the chooser and predictor. In order to maintain simplicity and a common approach to most of the literature in the topic, the classifiers used to play the chooser and predictor are the same.

Although any method capable of generating artificial data can be used as a generator, the ones used in this experiment are oversamplers, originally developed to deal with imbalanced learning problems. Specifically, we chose SMOTE for its popularity and simplicity. We also chose G-SMOTE as a better performing generalization of the former method.

Three classification algorithms are used as the chooser and predictor. We use different types of classifiers to test the framework's performance under varying situations: neighbors-based, linear and ensemble models. The neighbors-based classifier chosen was K-nearest neighbors (KNN) [Cover and Hart, 1967], a logistic regression (LR) [Nelder and Wedderburn, 1972] is used as the linear model and a random forest classifier (RFC) [Ho, 1995] was used as the ensemble model.

The acquisition function is completed by testing three different selection criteria. Random selection is used as a baseline selection criterion, whereas entropy (see Formula 1) and breaking ties (see Formula 2) are used due to their popularity and classifier independence.

5.3 Evaluation Metrics

According to [Gavade and Rajpurohit, 2019], nearly 80% of the satellite-based LULC studies employ the *Overall Accuracy* (OA) and *Kappa coefficient* performance metrics. However, these metrics are frequently insufficient to accurately depict classification performance [Olofsson et al., 2013, Pontius and Millones, 2011]. Metrics such as Producer's Accuracy (or *Recall*) and User's Accuracy (or *Precision*) are also commonly used. Since they consist of ratios based on True/False Positives (TP and FP) and Negatives (TN and FN), formulated as $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, they provide per class information regarding the classifier's classification performance. However, in this experiment, the meaning and number of classes available in each dataset varies, making these metrics difficult to synthesize.

While OA and Kappa tend to overestimate a classifier's performance on datasets with high IR, other metrics such as *F-score* and *Geometric mean* (G-mean) are less sensitive to the data imbalance bias [Jeni et al., 2013, Kubat et al., 1997]. Therefore, we employ 3 performance metrics:

1. The G-mean scorer is the geometric mean of $Specificity = \frac{TN}{TN+FP}$ and *Sensitivity* (also known as *Recall*) [Kubat et al., 1997]. Both metrics are calculated in a multiclass context considering a one-versus-all approach. For multiclass problems, the *G-mean* scorer is calculated as its average per class values:

$$G\text{-}mean = \sqrt{\overline{Sensitivity}_i \times \overline{Specificity}_i} \tag{3}$$

2. F-score is the harmonic mean of *Precision* and *Recall*. The two metrics are also calculated considering a one-versus-all approach. The *F-score* for the multi-class case can be calculated using its

average per class values [He and Garcia, 2009]:

$$F\text{-}score = 2\frac{\overline{Precision} \times \overline{Recall}}{\overline{Precision} + \overline{Recall}} \tag{4}$$

3. OA consists of the ratio between the number of correctly classified observations and the total number of observations. This metric, because of its popularity and easy interpretability, is kept for discussion purposes. Considering C as the set of classes within a dataset, it is expressed as:

$$OA = \frac{\sum_i^C \mathrm{TP}_i}{\sum_i^C (\mathrm{TP}_i + \mathrm{FP}_i)} \tag{5}$$

The comparison of classification convergence across AL frameworks and selection criteria is done using 3 AL-specific performance metrics. Particularly, we follow the recommendations found in [Kottke et al., 2017]. Each AL configuration is evaluated using the *Area Under the Learning Curve* (AULC) performance metric. It is the sum of the classification performance values of all iterations. To facilitate the analysis of the results, we fix the range of this metric between $[0, 1]$ by dividing it with the total amount of iterations (i.e., the maximum performance area). The metric *Data Utilization Rate* (DUR) [Reitmaier and Sick, 2013] consists of the ratio between the minimum number of observations necessary to reach a given performance threshold by an AL strategy and an equivalent baseline strategy. The deficiency score [Yanık and Sezgin, 2015] is used to compare the performance between two active learners. The deficiency of algorithm A with respect to algorithm B is calculated with the areas between the respective learning curves and the maximum performance line MP:

$$deficiency = \frac{MP - AULC_A}{2MP - AULC_A - AULC_B} \tag{6}$$

This metric varies between $[0, 1]$, where values 0 and 1 are achieved by algorithms A and B, capable of achieving maximal performance from the first iteration onwards, respectively. A deficiency score of 0.5 means that active learners A and B are equivalent.

5.4 Experimental Procedure

A common practice in methodological evaluations is the implementation of an offline experiment [Kagy et al., 2019]. It consists of using an existing set of labeled data as a proxy for the population of unlabeled samples. Because the dataset is already fully labeled, the oracle's typical annotation process involved in each iteration is done at zero cost. Each AL and classifier configuration is tested using a stratified 5-fold cross validation testing scheme. For each round, the larger partition is split in a stratified fashion to form a training and validation set (containing 20% of the original partition). The validation set is used to evaluate the convergence efficiency of active learners; the chooser's classification performance metrics and amount of data points used at each iteration are used to compute the AULC and DUR. Additionally, within the AL iterative process, the classifier with optimal performance on the validation set is evaluated using the test set. In order to further reduce possible initialization biases, this procedure is repeated 3 times with different seeds and the results of all runs are averaged (i.e., each configuration is trained and evaluated 15 times). Finally, the maximum performance lines are calculated using the same approach. In those cases, the validation set is not used. The experimental procedure is depicted in Figure 5.

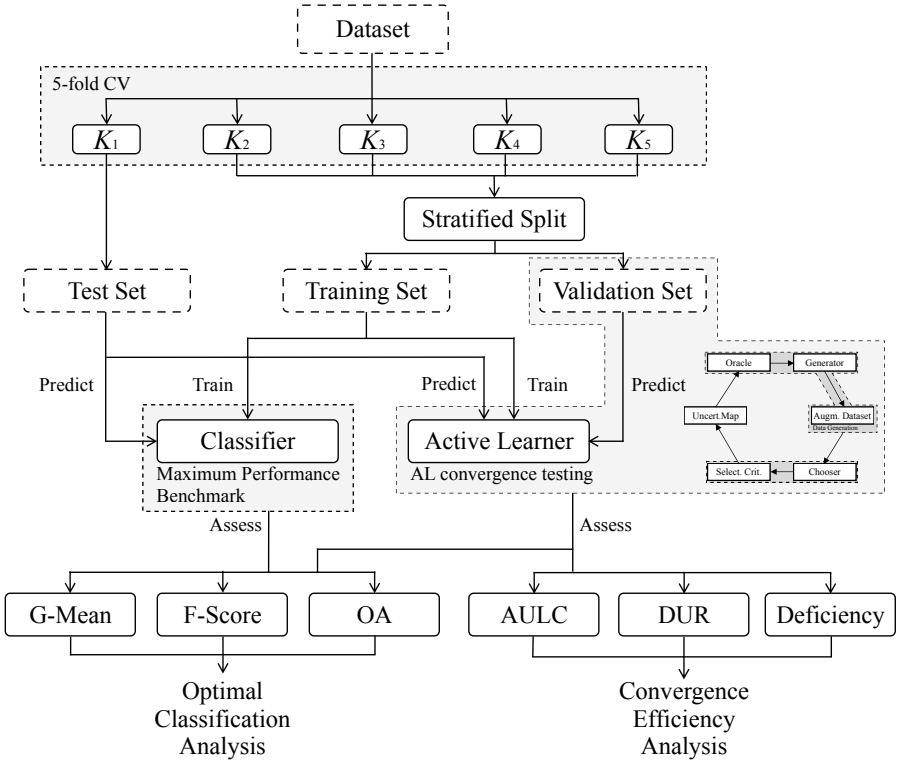

Figure 5: Experimental procedure. The performance metrics are averaged over the 5 folds across each of the 3 different initializations of this procedure for a given combination of generator, chooser/predictor and selection criterion.

To make the convergence metrics comparable across active learners, the configurations of the different frameworks must be similar. For each dataset, the number of observations is constant to facilitate the analysis of the same metrics.

In most practical AL applications it is assumed that the number of observations in the initial training sample is too small to perform hyperparameter tuning. Consequently, in order to ensure realistic results, our experimental procedure does not include hyperparameter optimization. The predefined hyperparameters are shown in Table 3. They were set up based on general recommendations and default settings for the classifiers and generators used.

The AL iterative process is set up with a randomly selected initial training sample with 15 initial samples. At each iteration, an additional 15 samples are added to the training set. This process is stopped after 49 iterations, once 50% of the dataset is added to the augmented dataset.

Classifier	Hyperparameters	Values
LR	maximum iterations	10000
	solver	sag
	penalty	None
KNN	# neighbors	5
	weights	uniform
	metric	euclidean
RF	maximum tree depth	None
	# estimators	100
	criterion	gini
Generator		
SMOTE	# neighbors	5
G-SMOTE	# neighbors	5
	deformation factor	0.5
	truncation factor	0.5

Table 3: Hyper-parameters grid.

5.5 Software Implementation

The experiment was implemented using the Python programming language, along with the Python libraries Scikit-Learn [Pedregosa et al., 2011], Imbalanced-Learn [Lemaître et al., 2017], Geometric-SMOTE, Cluster-Over-Sampling and Research-Learn libraries. All functions, algorithms, experiments and results are provided in the GitHub repository of the project.

6 Results

The evaluation of the different AL frameworks in a multiple dataset context should not rely uniquely on the mean of the performance metrics across datasets. [Demšar, 2006] recommends the usage of mean ranking scores, since the performance levels of the different frameworks varies according to the data it is being used on. Consequently, evaluating these performance metrics solely based on their mean values might lead to inaccurate analyses. Accordingly, the results of this experiment is analysed using both the mean ranking and absolute scores for each model. Finally, these results are used to perform a statistical significance analysis, presented in Subsection 6.1. The rank values are assigned based on the mean scores resulting from three different initializations of 5-fold cross validation for each classifier and active learner.

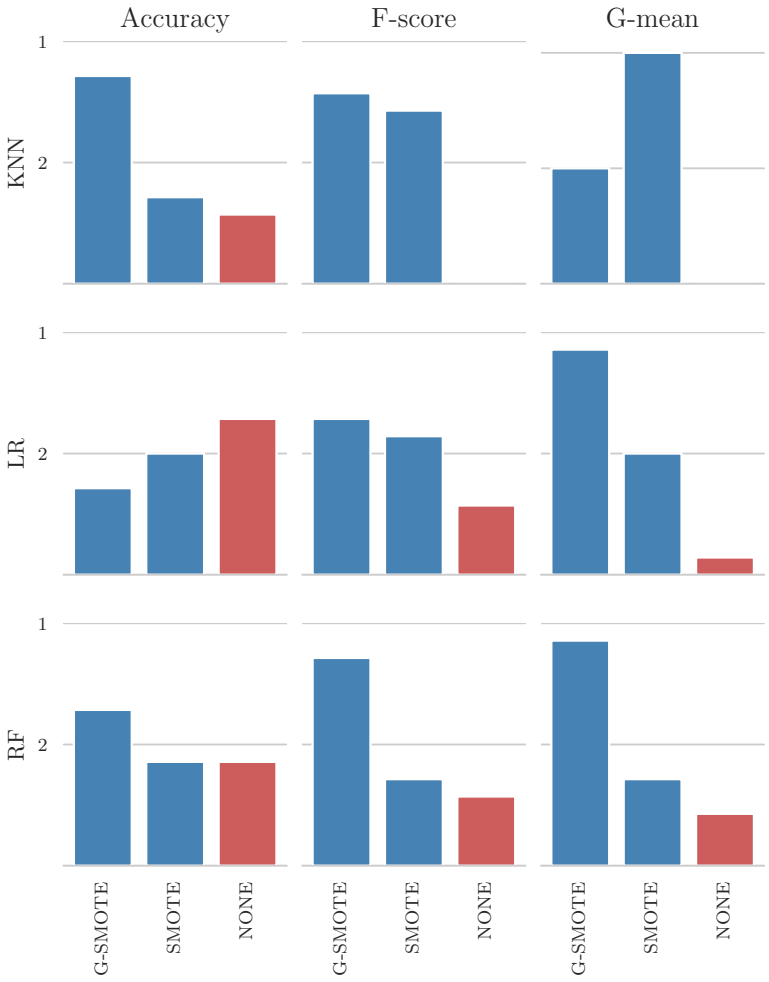

Figure 6: Mean AULC ranking of AL frameworks across datasets.

The AULC mean ranking scores show that the convergence efficiency of the proposed method (i.e., using generators G-SMOTE and SMOTE) is consistently higher than the traditional AL framework (NONE). With the exception of two scenarios, the proposed framework using G-SMOTE was able to outperform the remaining methods. The only scenario where the baseline active learner was able to outperform the remaining methods was using the LR classifier and OA as the optimization goal. Table 4 shows the quantitative results of average AULC ranking scores and standard deviations across datasets for each active learner.

Classifier	Evaluation Metric	NONE	SMOTE	G-SMOTE
KNN	Accuracy	2.43 ± 0.9	2.29 ± 0.45	**1.29 ± 0.45**
KNN	F-score	3.00 ± 0.0	1.57 ± 0.49	**1.43 ± 0.49**
KNN	G-mean	3.00 ± 0.0	**1.00 ± 0.0**	2.00 ± 0.0
LR	Accuracy	**1.71 ± 0.88**	2.00 ± 0.76	2.29 ± 0.7
LR	F-score	2.43 ± 0.9	1.86 ± 0.64	**1.71 ± 0.7**
LR	G-mean	2.86 ± 0.35	2.00 ± 0.53	**1.14 ± 0.35**
RF	Accuracy	2.14 ± 0.64	2.14 ± 0.83	**1.71 ± 0.88**
RF	F-score	2.43 ± 0.73	2.29 ± 0.7	**1.29 ± 0.45**
RF	G-mean	2.57 ± 0.49	2.29 ± 0.7	**1.14 ± 0.35**

Table 4: Mean AULC ranking scores.

The mean absolute scores are provided in Table 5. In some situations, there is a significant performance superiority across active learners. The high AULC values are owed to the naturally high classification performance of baseline methods, which makes the performance variability among frameworks particularly meaningful.

Classifier	Evaluation Metric	NONE	SMOTE	G-SMOTE
KNN	Accuracy	0.811 ± 0.115	0.806 ± 0.141	**0.820 ± 0.123**
KNN	F-score	0.762 ± 0.131	**0.796 ± 0.123**	0.794 ± 0.123
KNN	G-mean	0.864 ± 0.079	**0.892 ± 0.068**	0.886 ± 0.073
LR	Accuracy	**0.868 ± 0.114**	**0.868 ± 0.113**	0.867 ± 0.115
LR	F-score	0.839 ± 0.119	**0.843 ± 0.117**	**0.843 ± 0.116**
LR	G-mean	0.907 ± 0.074	0.910 ± 0.071	**0.911 ± 0.071**
RF	Accuracy	**0.851 ± 0.09**	0.850 ± 0.09	**0.851 ± 0.092**
RF	F-score	0.810 ± 0.109	0.816 ± 0.097	**0.819 ± 0.1**
RF	G-mean	0.890 ± 0.068	0.896 ± 0.058	**0.901 ± 0.059**

Table 5: Mean absolute AULC scores.

The mean deficiency scores of the proposed framework is shown in Table 6. This metric is calculated using the proposed framework using G-SMOTE and SMOTE as algorithms A and the baseline active learner as algorithm B. The proposed method shows consistent a improvement over the baseline method, except in three situations, where the difference among active learners is marginal.

Classifier	Evaluation Metric	SMOTE	G-SMOTE
KNN	Accuracy	**0.454 ± 0.129**	**0.373 ± 0.183**
KNN	F-score	**0.320 ± 0.11**	**0.318 ± 0.146**
KNN	G-mean	**0.004 ± 0.673**	**0.072 ± 0.651**
LR	Accuracy	0.507 ± 0.038	0.509 ± 0.039
LR	F-score	**0.484 ± 0.025**	**0.483 ± 0.025**
LR	G-mean	**0.462 ± 0.037**	**0.453 ± 0.039**
RF	Accuracy	0.526 ± 0.063	**0.487 ± 0.028**
RF	F-score	**0.490 ± 0.057**	**0.452 ± 0.041**
RF	G-mean	**0.471 ± 0.068**	**0.385 ± 0.095**

Table 6: Mean deficiency scores. The scores were calculated by estimating the deficiency of the proposed framework with respect to the baseline method.

The average DURs are shown in Figure 7. They were calculated for various threshold levels, varying at a step of 5% between 60% and 95%. The DURs shown in the figure use the typical AL framework as the baseline strategy. The results show a generalized decrease of data required to reach the performance thresholds in the various scenarios. For higher performance thresholds, the gap between the proposed and baseline methods tend to be reduced, since the amount of data required is larger and the benefits of data generation is less apparent.

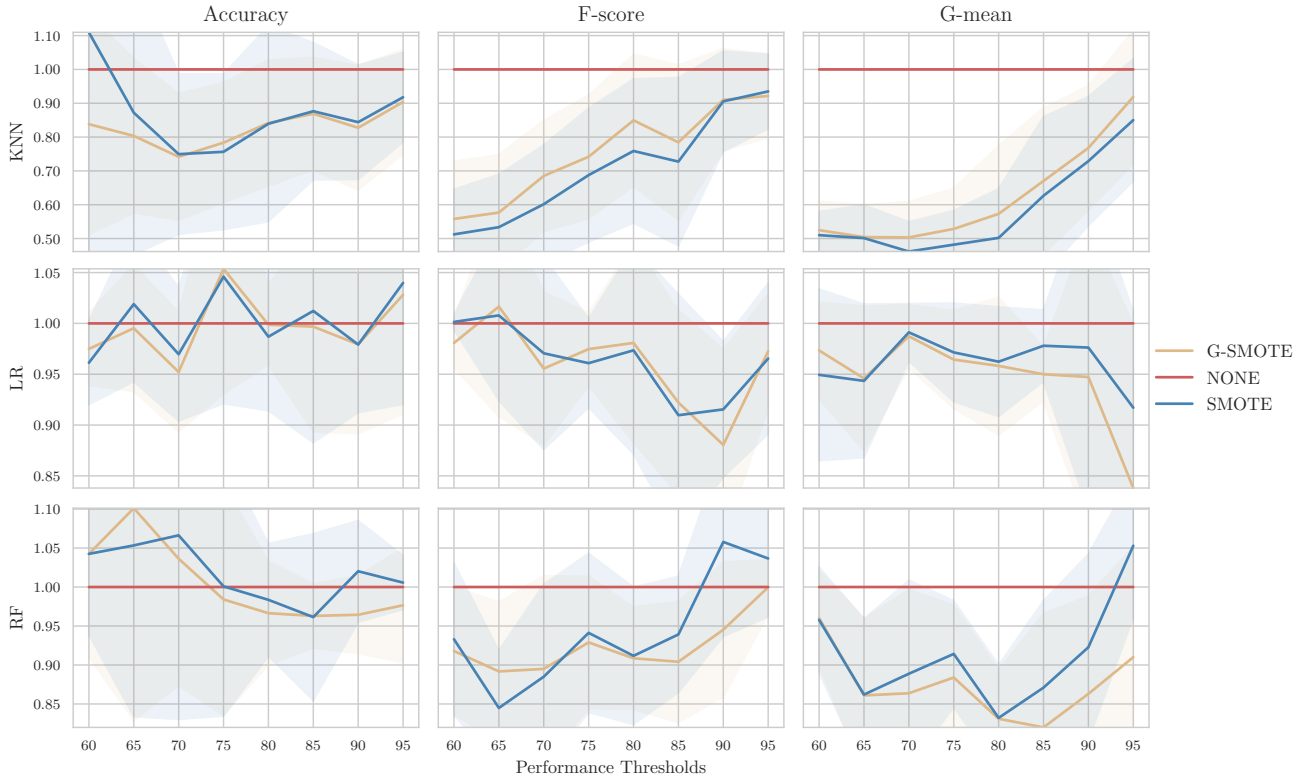

Figure 7: Mean data utilization rates.

The mean optimal classification scores is shown in Table 7. One of the goals of this study is ensuring that the classification performance of the predictors resulting from the proposed framework are not worse than the predictors produced using the typical AL framework. Using a generator in AL's iterative process was capable of outperforming the baseline framework in most cases and the maximum performance classifier in 6 of those.

Classifier	Evaluation Metric	MP	NONE	SMOTE	G-SMOTE
KNN	Accuracy	**0.864 ± 0.096**	0.861 ± 0.101	0.847 ± 0.123	0.860 ± 0.102
KNN	F-score	0.838 ± 0.106	0.835 ± 0.115	0.841 ± 0.111	**0.843 ± 0.105**
KNN	G-mean	0.907 ± 0.063	0.904 ± 0.069	**0.918 ± 0.056**	0.912 ± 0.061
LR	Accuracy	**0.902 ± 0.088**	0.899 ± 0.094	0.897 ± 0.096	0.897 ± 0.097
LR	F-score	**0.890 ± 0.084**	0.883 ± 0.096	0.888 ± 0.095	0.887 ± 0.097
LR	G-mean	0.935 ± 0.052	0.931 ± 0.059	0.936 ± 0.057	**0.938 ± 0.055**
RF	Accuracy	0.884 ± 0.072	0.889 ± 0.07	**0.890 ± 0.068**	**0.890 ± 0.071**
RF	F-score	0.859 ± 0.083	0.866 ± 0.081	**0.870 ± 0.074**	0.869 ± 0.08
RF	G-mean	0.918 ± 0.051	0.921 ± 0.051	0.928 ± 0.042	**0.930 ± 0.043**

Table 7: Optimal classification scores.

The optimal AULC results of each method are reported in Table 8. These results depict in higher detail the findings drawn from this experiment.

Dataset	Classifier	Evaluation Metric	NONE	SMOTE	G-SMOTE
Botswana	KNN	Accuracy	0.823 ± 0.027	0.851 ± 0.017	**0.852 ± 0.018**
Botswana	KNN	F-score	0.814 ± 0.02	**0.856 ± 0.022**	0.850 ± 0.022

Dataset	Classifier	Evaluation Metric	NONE	SMOTE	G-SMOTE
Botswana	KNN	G-mean	0.898 ± 0.016	**0.922 ± 0.011**	0.921 ± 0.011
Botswana	LR	Accuracy	0.929 ± 0.01	**0.935 ± 0.01**	**0.935 ± 0.011**
Botswana	LR	F-score	0.929 ± 0.01	**0.936 ± 0.008**	0.935 ± 0.009
Botswana	LR	G-mean	0.962 ± 0.005	**0.966 ± 0.005**	**0.966 ± 0.005**
Botswana	RF	Accuracy	0.855 ± 0.02	0.856 ± 0.022	**0.860 ± 0.02**
Botswana	RF	F-score	0.856 ± 0.021	0.857 ± 0.023	**0.861 ± 0.021**
Botswana	RF	G-mean	0.920 ± 0.012	0.921 ± 0.013	**0.924 ± 0.011**
IP	KNN	Accuracy	**0.595 ± 0.016**	0.524 ± 0.026	0.586 ± 0.023
IP	KNN	F-score	0.506 ± 0.02	0.550 ± 0.024	**0.552 ± 0.017**
IP	KNN	G-mean	0.711 ± 0.012	**0.755 ± 0.016**	0.740 ± 0.015
IP	LR	Accuracy	0.620 ± 0.017	**0.623 ± 0.016**	0.620 ± 0.017
IP	LR	F-score	0.584 ± 0.025	0.590 ± 0.024	**0.592 ± 0.025**
IP	LR	G-mean	0.747 ± 0.015	**0.754 ± 0.016**	**0.754 ± 0.015**
IP	RF	Accuracy	0.682 ± 0.023	**0.683 ± 0.023**	0.679 ± 0.024
IP	RF	F-score	0.611 ± 0.03	**0.641 ± 0.025**	0.632 ± 0.029
IP	RF	G-mean	0.768 ± 0.017	**0.791 ± 0.015**	0.787 ± 0.015
KSC	KNN	Accuracy	0.829 ± 0.016	0.840 ± 0.014	**0.846 ± 0.01**
KSC	KNN	F-score	0.774 ± 0.013	**0.789 ± 0.012**	**0.789 ± 0.014**
KSC	KNN	G-mean	0.871 ± 0.008	**0.885 ± 0.007**	0.884 ± 0.007
KSC	LR	Accuracy	**0.912 ± 0.017**	**0.912 ± 0.016**	0.911 ± 0.015
KSC	LR	F-score	**0.872 ± 0.018**	0.871 ± 0.019	0.870 ± 0.016
KSC	LR	G-mean	0.929 ± 0.01	**0.931 ± 0.01**	**0.931 ± 0.008**
KSC	RF	Accuracy	0.861 ± 0.012	0.863 ± 0.009	**0.866 ± 0.012**
KSC	RF	F-score	0.803 ± 0.014	0.807 ± 0.009	**0.811 ± 0.013**
KSC	RF	G-mean	0.890 ± 0.008	0.893 ± 0.007	**0.896 ± 0.008**
PC	KNN	Accuracy	0.948 ± 0.012	0.954 ± 0.011	**0.958 ± 0.011**
PC	KNN	F-score	0.834 ± 0.038	0.857 ± 0.026	**0.860 ± 0.029**
PC	KNN	G-mean	0.912 ± 0.017	**0.931 ± 0.011**	**0.931 ± 0.012**
PC	LR	Accuracy	**0.969 ± 0.011**	0.968 ± 0.011	0.968 ± 0.01
PC	LR	F-score	0.885 ± 0.038	0.885 ± 0.03	**0.886 ± 0.031**
PC	LR	G-mean	0.939 ± 0.018	0.940 ± 0.015	**0.941 ± 0.016**
PC	RF	Accuracy	0.955 ± 0.011	0.952 ± 0.009	**0.956 ± 0.01**
PC	RF	F-score	0.860 ± 0.03	0.852 ± 0.024	**0.865 ± 0.027**
PC	RF	G-mean	0.925 ± 0.016	0.922 ± 0.014	**0.932 ± 0.013**
PU	KNN	Accuracy	**0.749 ± 0.022**	0.720 ± 0.023	0.737 ± 0.02
PU	KNN	F-score	0.654 ± 0.03	**0.712 ± 0.012**	0.702 ± 0.008
PU	KNN	G-mean	0.791 ± 0.02	**0.847 ± 0.009**	0.833 ± 0.006
PU	LR	Accuracy	**0.850 ± 0.011**	0.841 ± 0.019	0.840 ± 0.014
PU	LR	F-score	0.774 ± 0.023	**0.784 ± 0.021**	0.783 ± 0.024
PU	LR	G-mean	0.861 ± 0.013	0.876 ± 0.01	**0.878 ± 0.01**
PU	RF	Accuracy	**0.794 ± 0.013**	0.787 ± 0.014	0.791 ± 0.015
PU	RF	F-score	0.702 ± 0.038	0.726 ± 0.02	**0.733 ± 0.014**
PU	RF	G-mean	0.817 ± 0.021	0.839 ± 0.011	**0.852 ± 0.009**
Salinas	KNN	Accuracy	0.778 ± 0.021	0.787 ± 0.018	**0.793 ± 0.02**
Salinas	KNN	F-score	0.808 ± 0.011	**0.843 ± 0.016**	0.838 ± 0.018
Salinas	KNN	G-mean	0.896 ± 0.006	**0.922 ± 0.007**	0.916 ± 0.007
Salinas	LR	Accuracy	0.821 ± 0.014	0.822 ± 0.016	**0.823 ± 0.016**
Salinas	LR	F-score	0.860 ± 0.023	0.862 ± 0.014	**0.863 ± 0.015**
Salinas	LR	G-mean	0.924 ± 0.013	0.924 ± 0.01	**0.927 ± 0.008**
Salinas	RF	Accuracy	0.839 ± 0.023	**0.841 ± 0.028**	0.836 ± 0.029
Salinas	RF	F-score	**0.871 ± 0.021**	0.864 ± 0.021	0.868 ± 0.023

Dataset	Classifier	Evaluation Metric	NONE	SMOTE	G-SMOTE
Salinas	RF	G-mean	0.932 ± 0.009	0.930 ± 0.009	**0.934 ± 0.011**
SA	KNN	Accuracy	0.956 ± 0.006	0.968 ± 0.011	**0.970 ± 0.012**
SA	KNN	F-score	0.944 ± 0.01	0.966 ± 0.009	**0.967 ± 0.01**
SA	KNN	G-mean	0.966 ± 0.005	**0.980 ± 0.006**	**0.980 ± 0.007**
SA	LR	Accuracy	**0.974 ± 0.005**	0.973 ± 0.005	0.973 ± 0.005
SA	LR	F-score	**0.973 ± 0.006**	0.972 ± 0.006	0.972 ± 0.006
SA	LR	G-mean	**0.983 ± 0.005**	**0.983 ± 0.005**	**0.983 ± 0.005**
SA	RF	Accuracy	**0.969 ± 0.012**	0.968 ± 0.012	**0.969 ± 0.01**
SA	RF	F-score	0.964 ± 0.021	0.963 ± 0.022	**0.965 ± 0.018**
SA	RF	G-mean	0.979 ± 0.012	0.979 ± 0.012	**0.980 ± 0.011**

Table 8: Mean cross-validation scores of AL algorithms for each dataset. Legend: IP Indian Pines, KSC Kennedy Space Center, PC Pavia Center, PU Pavia University, SA Salinas A.

6.1 Statistical Analysis

The methods used to test the experiment's results must be appropriate for a multi-dataset context. Therefore the statistical analysis is performed using the Friedman test [Friedman, 1937] and the Holm-Bonferroni method [Holm, 1979] for a post-hoc analysis. The variable used for this test is the AULC, considering the various performance metrics used.

Table 9 shows the results of the Friedman test. In most cases the null hypothesis is rejected, which indicates a statistically significant difference on the performance among AL frameworks.

Classifier	Evaluation Metric	p-value	Significance
KNN	Accuracy	6.6e-02	True
KNN	F-score	5.1e-03	True
KNN	G-mean	9.1e-04	True
LR	Accuracy	5.6e-01	False
LR	F-score	3.7e-01	False
LR	G-mean	5.8e-03	True
RF	Accuracy	6.5e-01	False
RF	F-score	6.6e-02	True
RF	G-mean	1.8e-02	True

Table 9: Results for Friedman test. Statistical significance is tested at a level of $\alpha = 0.15$. The null hypothesis is that there is no difference in the classification across AL frameworks.

The Holm-Bonferroni test results are shown in Table 10. The null hypothesis was rejected in most situations, with exception to the AULC using overall accuracy.

Classifier	Evaluation Metric	SMOTE	G-SMOTE
KNN	Accuracy	7.3e-01	3.1e-01
KNN	F-score	**9.7e-04**	**8.9e-04**
KNN	G-mean	**3.5e-03**	**2.0e-03**
LR	Accuracy	1.0e+00	1.0e+00

Classifier	Evaluation Metric	SMOTE	G-SMOTE
LR	F-score	**1.2e-01**	**1.2e-01**
LR	G-mean	**1.3e-01**	**1.3e-01**
RF	Accuracy	1.0e+00	1.0e+00
RF	F-score	3.2e-01	1.6e-01
RF	G-mean	2.1e-01	**1.4e-01**

Table 10: Adjusted p-values using the Holm-Bonferroni method. Bold values are statistically significant at a level of $\alpha = 0.15$. The null hypothesis is that the tested method does not perform better than the control method (benchmark AL framework).

7 Conclusion

The aim of this experiment was to test the effectiveness of a new AL framework, where a new element is introduced to improve the convergence rate of AL through the use of artificial data generation. The experiment was designed to test the proposed method under particularly challenging conditions, where the maximum performance line is naturally high in most datasets (with exception to the Indian Pines dataset). In order to test basic setups for this new framework, the elements that constitute the Generator component were set up in a plug-and-play scheme, without significant tuning of the data generators (SMOTE and G-SMOTE). The tests showed that this new framework is able to consistently outperform the original AL framework in most scenarios, as shown in Table 8. These results could be further improved through the modification and more intense tuning of the data generation strategy. In our experiment, during each iteration, the new artificial data is generated only to match each non-majority class frequency with the majority class frequency, thus strictly balancing the class distribution. Generating a larger amount of data for all classes (especially in early iterations) can further improve these results.

We also consider the fast convergence of AL on these datasets. The high performance scores for the baseline AL framework made the achievement of significant improvements over the traditional AL framework under these conditions particularly meaningful. The advantage of the proposed AL framework is shown in Figure 7. In most of the presented scenarios there is a substantial reduction of necessary data to reach each of the tested performance metric thresholds.

The results from this experiment show that the inclusion of a data generator in the AL framework will yield significant improvements in the convergence of the method. The proposed method successfully anticipated the predictor's optimal performance, as shown in Tables 4, 5 and 6. This means the annotation cost would have been reduced in a real application since the number of iterations and labeled samples necessary to reach near optimal classification performance is reduced, as shown in Figure 7. The class imbalance bias observed in AL tasks is reduced, as shown in Tables 6, where data imbalance appropriate metrics are always improved over the baseline scores.

References

[Aghdam et al., 2019] Aghdam, H. H., Gonzalez-Garcia, A., Lopez, A., and Weijer, J. (2019). Active learning for deep detection neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 3671–3679.

[Alonso-Sarria et al., 2019] Alonso-Sarria, F., Valdivieso-Ros, C., and Gomariz-Castillo, F. (2019). Isolation forests to evaluate class separability and the representativeness of training and validation areas in land cover classification. *Remote Sensing*, 11(24):3000.

[Baumgardner et al., 2015] Baumgardner, M. F., Biehl, L. L., and Landgrebe, D. A. (2015). 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3.

[Bogner et al., 2018] Bogner, C., Seo, B., Rohner, D., and Reineking, B. (2018). Classification of rare land cover types: Distinguishing annual and perennial crops in an agricultural catchment in South Korea. *PLoS ONE*, 13(1).

[Cawley, 2011] Cawley, G. (2011). Baseline Methods for Active Learning. *Proceedings of Active Learning and Experimental Design workshop In conjunction with AISTATS*, 16:47–57.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

[Chawla et al., 2004] Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6.

[Copa et al., 2010] Copa, L., Tuia, D., Volpi, M., and Kanevski, M. (2010). Unbiased query-by-bagging active learning for VHR image classification. In Bruzzone, L., editor, *Image and Signal Processing for Remote Sensing XVI*, volume 7830, page 78300K. SPIE.

[Costa et al., 2020] Costa, H., Benevides, P., Marcelino, F., and Caetano, M. (2020). Introducing automatic satellite image processing into land cover mapping by photo-interpretation of airborne data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:29–34.

[Costantino et al., 2020] Costantino, D., Pepe, M., Dardanelli, G., and Baiocchi, V. (2020). USING OPTICAL SATELLITE AND AERIAL IMAGERY FOR AUTOMATIC COASTLINE MAPPING. *Geographia Technica*, pages 171–190.

[Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

[Demir et al., 2011] Demir, B., Persello, C., and Bruzzone, L. (2011). Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1014–1031.

[Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

[DeVries and Taylor, 2017] DeVries, T. and Taylor, G. W. (2017). Dataset augmentation in feature space. In *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*. International Conference on Learning Representations, ICLR.

[Di and Crawford, 2012] Di, W. and Crawford, M. M. (2012). View generation for multiview maximum disagreement based active learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50(5 PART 2):1942–1954.

[Douzas and Bacao, 2017] Douzas, G. and Bacao, F. (2017). Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert Systems with Applications*, 82:40–52.

[Douzas and Bacao, 2019] Douzas, G. and Bacao, F. (2019). Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences*, 501:118–135.

[Douzas et al., 2019] Douzas, G., Bacao, F., Fonseca, J., and Khudinyan, M. (2019). Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric SMOTE algorithm. *Remote Sensing*, 11(24):3040.

[Douzas et al., 2018] Douzas, G., Bacao, F., and Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465:1–20.

[Ertekin et al., 2007] Ertekin, S., Huang, J., and Giles, C. L. (2007). Active learning for class imbalance problem. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07*, pages 823–824, New York, New York, USA. ACM Press.

[Feng et al., 2018] Feng, W., Huang, W., Ye, H., and Zhao, L. (2018). Synthetic minority over-sampling technique based rotation forest for the classification of unbalanced hyperspectral data. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 2018-July, pages 2651–2654. Institute of Electrical and Electronics Engineers Inc.

[Fernández et al., 2013] Fernández, A., López, V., Galar, M., del Jesus, M. J., and Herrera, F. (2013). Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97–110.

[Friedman, 1937] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

[Gavade and Rajpurohit, 2019] Gavade, A. B. and Rajpurohit, V. S. (2019). Systematic analysis of satellite image-based land cover classification techniques: literature review and challenges. *International Journal of Computers and Applications*, pages 1–10.

[Han et al., 2005] Han, H., Wang, W. Y., and Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Lecture Notes in Computer Science*, 3644(PART I):878–887.

[He et al., 2008] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1322–1328. IEEE.

[He and Garcia, 2009] He, H. and Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, page 278, USA. IEEE Computer Society.

[Holm, 1979] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

[Huang et al., 2018] Huang, Y., xin CHEN, Z., YU, T., zhi HUANG, X., and fa GU, X. (2018). Agri-

cultural remote sensing big data: Management and applications. *Journal of Integrative Agriculture*, 17(9):1915–1931.

[Jeni et al., 2013] Jeni, L. A., Cohn, J. F., and De La Torre, F. (2013). Facing imbalanced data - Recommendations for the use of performance metrics. In *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, pages 245–251.

[Jozdani et al., 2019] Jozdani, S. E., Johnson, B. A., and Chen, D. (2019). Comparing Deep Neural Networks, Ensemble Classifiers, and Support Vector Machine Algorithms for Object-Based Urban Land Use/Land Cover Classification. *Remote Sensing*, 11(14):1713.

[Kagy et al., 2019] Kagy, J.-F., Kayadelen, T., Ma, J., Rostamizadeh, A., and Strnadova, J. (2019). The practical challenges of active learning: Lessons learned from live experimentation.

[Kaur et al., 2019] Kaur, H., Pannu, H. S., and Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys*, 52(4):1–36.

[Khatami et al., 2016] Khatami, R., Mountrakis, G., and Stehman, S. V. (2016). A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. *Remote Sensing of Environment*, 177:89–100.

[Kottke et al., 2017] Kottke, D., Calma, A., Huseljic, D., Krempl, G., and Sick, B. (2017). Challenges of reliable, realistic and comparable active learning evaluation. In *CEUR Workshop Proceedings*, volume 1924, pages 2–14.

[Kubat et al., 1997] Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Citeseer.

[Lemaître et al., 2017] Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

[Li et al., 2011] Li, J., Bioucas-Dias, J. M., and Plaza, A. (2011). Hyperspectral image segmentation using a new bayesian approach with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10 PART 2):3947–3960.

[Li et al., 2013] Li, J., Bioucas-Dias, J. M., and Plaza, A. (2013). Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):844–856.

[Li et al., 2020] Li, J., Huang, X., and Chang, X. (2020). A label-noise robust active learning sample collection method for multi-temporal urban land-cover classification and change analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163(January):1–17.

[Li and Guo, 2013] Li, X. and Guo, Y. (2013). Active learning with multi-label svm classification. In *In IJCAI*, pages 1479–1485.

[Liu et al., 2020] Liu, S.-J., Luo, H., and Shi, Q. (2020). Active Ensemble Deep Learning for Polarimetric Synthetic Aperture Radar Image Classification. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5.

[Liu et al., 2018] Liu, W., Yang, J., Li, P., Han, Y., Zhao, J., and Shi, H. (2018). A novel object-based

supervised classification method with active learning and random forest for PolSAR imagery. *Remote Sensing*, 10(7).

[Luo et al., 2003] Luo, T., Kramer, K., Goldgof, D., Hall, L. O., Samson, S., Remsen, A., and Hopkins, T. (2003). Learning to recognize plankton. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 888–893.

[Maxwell et al., 2018] Maxwell, A. E., Warner, T. A., and Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817.

[Muslea et al., 2006] Muslea, I., Minton, S., and Knoblock, C. A. (2006). Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27:203–233.

[Nagai et al., 2020] Nagai, S., Nasahara, K. N., Akitsu, T. K., Saitoh, T. M., and Muraoka, H. (2020). Importance of the Collection of Abundant Ground-Truth Data for Accurate Detection of Spatial and Temporal Variability of Vegetation by Satellite Remote Sensing. In *Biogeochemical Cycles: Ecological Drivers and Environmental Impact*, pages 223–244. American Geophysical Union (AGU).

[Nelder and Wedderburn, 1972] Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

[Olofsson et al., 2013] Olofsson, P., Foody, G. M., Stehman, S. V., and Woodcock, C. E. (2013). Making better use of accuracy data in land change studies: Estimating accuracy and area and quantifying uncertainty using stratified estimation. *Remote Sensing of Environment*, 129:122–131.

[Pasolli et al., 2016] Pasolli, E., Yang, H. L., and Crawford, M. M. (2016). Active-metric learning for classification of remotely sensed hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(4):1925–1939.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

[Pelletier et al., 2017] Pelletier, C., Valero, S., Inglada, J., Champion, N., Sicre, C. M., and Dedieu, G. (2017). Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sensing*, 9(2):173.

[Pontius and Millones, 2011] Pontius, R. G. and Millones, M. (2011). Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15):4407–4429.

[Reitmaier and Sick, 2013] Reitmaier, T. and Sick, B. (2013). Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4ds. *Information Sciences*, 230:106–131.

[Růžička et al., 2020] Růžička, V., D'Aronco, S., Wegner, J. D., and Schindler, K. (2020). Deep active learning in remote sensing for data efficient change detection. *arXiv preprint arXiv:2008.11201*.

[Shrivastava and Pradhan, 2021] Shrivastava, V. K. and Pradhan, M. K. (2021). Hyperspectral Remote Sensing Image Classification Using Active Learning. In *Studies in Computational Intelligence*, volume 907, pages 133–152. Springer.

[Stromann et al., 2020] Stromann, O., Nascetti, A., Yousif, O., and Ban, Y. (2020). Dimensionality Reduction and Feature Selection for Object-Based Land Cover Classification based on Sentinel-1 and Sentinel-2 Time Series Using Google Earth Engine. *Remote Sensing*, 12(1):76.

[Su et al., 2020] Su, T., Zhang, S., and Liu, T. (2020). Multi-spectral image classification based on an object-based active learning approach. *Remote Sensing*, 12(3):504.

[Sverchkov and Craven, 2017] Sverchkov, Y. and Craven, M. (2017). A review of active learning approaches to experimental design for uncovering biological networks. *PLOS Computational Biology*, 13(6):e1005466.

[Tuia et al., 2011] Tuia, D., Pasolli, E., and Emery, W. J. (2011). Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242.

[Tuia et al., 2009] Tuia, D., Ratle, F., Pacifici, F., Kanevski, M. F., and Emery, W. J. (2009). Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2218–2232.

[Vermote et al., 2020] Vermote, E. F., Skakun, S., Becker-Reshef, I., and Saito, K. (2020). Remote sensing of coconut trees in tonga using very high spatial resolution worldview-3 data. *Remote Sensing*, 12(19):3113.

[Wang and Xie, 2018] Wang, X. and Xie, H. (2018). A review on applications of remote sensing and geographic information systems (GIS) in water resources and flood risk management. *Water (Switzerland)*, 10(5):608.

[Wulder et al., 2018] Wulder, M. A., Coops, N. C., Roy, D. P., White, J. C., and Hermosilla, T. (2018). Land cover 2.0. *International Journal of Remote Sensing*, 39(12):4254–4284.

[Yanık and Sezgin, 2015] Yanık, E. and Sezgin, T. M. (2015). Active learning for sketch recognition. *Computers & Graphics*, 52:93–105.

[Yoo and Kweon, 2019] Yoo, D. and Kweon, I. S. (2019). Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhang et al., 2016] Zhang, Z., Pasolli, E., Yang, H. L., and Crawford, M. M. (2016). Multimetric Active Learning for Classification of Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, 13(7):1007–1011.

[Zhou et al., 2014] Zhou, X., Prasad, S., and Crawford, M. (2014). Wavelet domain multi-view active learning for hyperspectral image analysis. In *Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing*, volume 2014-June. IEEE Computer Society.