

Article

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 (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal; gdouzas@novaims.unl.pt (G.D.); bacao@novaims.unl.pt (F.B.)

* Correspondence: jpfonseca@novaims.unl.pt (J.F.)

* Correspondence: jpfonseca@novaims.unl.pt

1 Abstract: 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 and the classifiers 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 Geometric SMOTE as data generator. We compare the new AL framework to the original one using similar acquisition functions and classifiers over three AL-specific performance metrics in seven benchmark datasets. We show that this modification of the AL framework significantly reduces cost and time requirements for a successful AL implementation in all of the datasets used in the experiment.

15 Keywords: Active Learning; Artificial Data Generation; Land Use/Land Cover Classification;
16 Oversampling; SMOTE

17 1. Introduction

18 The technological development of air and spaceborne sensors, as well as the increasing number of remote sensing missions have allowed the continuous collection
19 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
20 as Land Use/Land Cover (LULC) change detection, ecosystem management [1], agricultural management [2], water resource management [3], forest management, and
21 urban monitoring [4]. Despite LULC maps being essential for most of these applications,
22 their production is still a challenging task [5,6]. They can be updated using one of the
23 following strategies:

- 24 1. Photo-interpretation. This approach consists of evaluating a patch’s LULC class by
25 a human operator based on orthophoto and satellite image interpretation [7]. This
26 method guarantees a decent level of accuracy, as it is dependent on the interpreter’s
27 expertise and human error. Typically, it is an expensive, time-consuming task that
28 requires the expertise of a photo-interpreter. This task is also frequently applied to
29 obtain ground-truth labels for training and/or validating Machine Learning (ML)
30 algorithms for related tasks [8,9].

Citation: Fonseca, J.; Douzas, G.; Bacao, F. Increasing the Effectiveness of Active Learning: Introducing Artificial Data Generation in Active Learning for Land Use/Land Cover Classification. *Remote Sens.* **2021**, *1*, 0.
<https://doi.org/>

Received:

Accepted:

Published:

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Submitted to *Remote Sens.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

- 35 2. Automated mapping. This approach is based on the usage of a ML method or a
36 combination of methods in order to obtain an updated LULC map. The develop-
37 ment of a reliable automated method is still a challenge among the ML and remote
38 sensing community, since the effectiveness of existing methods varies across applica-
39 tions and geographical areas [5]. Typically, this method requires the existence of
40 ground-truth data, which is frequently outdated or nonexistent for the required
41 time frame [1]. On the other hand, employing a ML method provides readily
42 available and relatively inexpensive LULC maps. The increasing quality of state-of-
43 the-art classification methods have motivated the application and adaptation of
44 these methods in this domain [10].
- 45 3. Hybrid approaches. These approaches employ photo-interpreted data to augment
46 the training dataset and improve the quality of automated mapping [11]. It at-
47 tempts to accelerate the photo-interpretation process by selecting a smaller sample
48 of the study area to be interpreted. The goal is to minimize the inaccuracies found
49 in the LULC map by supplying high-quality ground-truth data to the automated
50 method. The final (photo-interpreted) dataset consists of only the most informa-
51 tive samples, *i.e.*, patches that are typically difficult to classify for a traditional
52 automated mapping method [12].

53 The latter method is best known as AL. It is especially useful whenever there is a
54 shortage or even absence of ground-truth data and/or the mapping region does not
55 contain updated LULC maps [13]. In a context of limited sample-collection budget,
56 the collection of the most informative samples capable of optimally increasing the
57 classification accuracy of a LULC map is of particular interest [13]. AL attempts to
58 minimize the human-computer interaction involved in photo-interpretation by selecting
59 the data points to include in the annotation process. These data points are selected
60 based on an uncertainty measure and represent the points close to the decision borders.
61 Afterwards, they are passed on for photo-interpretation and added to the training dataset,
62 while the points with the lowest uncertainty values are ignored for photo-interpretation
63 and classification. This process is repeated until a convergence criterion is reached [14].

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

- 68 1. Human error. The involvement of photo-interpreters in the data labeling step
69 carries an additional risk to the creation of LULC patches. The minimum mapping
70 unit being considered, as well as the quality of the orthophotos and satellite images
71 being used, are some of the factors that may lead to the overlooking of small-area
72 LULC patches and label-noisy training data [15].
- 73 2. High-dimensional datasets. Although the amount of bands (*i.e.*, features) present in
74 multi and hyper spectral images contain useful information for automated classifi-
75 cation, they also introduce an increased level of complexity and redundancy in the
76 classification step [16]. These datasets are often prone to the Hughes phenomenon,
77 also known as the curse of dimensionality.
- 78 3. Class separability. Producing an LULC map considering classes with similar
79 spectral signatures makes them difficult to separate [17]. A lower pixel resolution
80 of the satellite images may also imply mixed-class pixels, which may lead to both
81 lower class separability as well as higher risk of human error.
- 82 4. Existence of rare land cover classes. The varying morphologies of different geo-
83 graphical regions naturally implies an uneven distribution of land cover classes [18].
84 This is particularly relevant in the context of AL since the data selection method
85 is based on a given uncertainty measure over data points whose class label is
86 unknown. Consequently, AL's iterative process of data selection may disregard
87 wrongly classified land cover areas belonging to a minority class.

88 Research developed in the field of AL typically focus on the reduction of human
89 error by minimizing the human interaction with the process through the development
90 of more efficient classifiers choosers and selection criteria within the generally accepted
91 AL framework. Concurrently, the problem of rare land cover classes is rarely addressed.
92 This is a frequent problem in the ML community, known as the Imbalanced Learning
93 problem. This problem exists whenever there is an uneven between-class distribution in
94 the dataset [19]. Specifically, most classifiers are optimized and evaluated using accuracy-
95 like metrics, which are designed to work primarily with balanced datasets. Consequently,
96 these metrics tend to introduce a bias towards the majority class by attributing an
97 importance to each class proportional to its relative frequency [10]. As an example, such a
98 classifier could achieve an overall accuracy of 99% on a binary dataset where the minority
99 class represents 1% of the overall dataset and still be useless. A number of methods
100 have been developed to deal with this problem. They can be categorized into three
101 different types of approaches [20,21]. Cost-sensitive solutions perform changes to the
102 cost matrix in the learning phase. Algorithmic level solutions modify specific classifiers
103 to reinforce learning on minority classes. Resampling solutions modify the training data
104 set by removing majority samples and/or generating artificial minority samples. The
105 latter is independent from the context and can be used alongside any classifier. Since
106 we are interested in the introduction of artificial data generation in AL, we will analyze
107 the state-of-the-art on resampling techniques (specifically oversampling) in Section
108 3. Because of this we will focus on artificial data generation techniques, presented in
109 Section 3.

110 In this paper, we propose a novel AL framework to address two limitations com-
111 monly found in the literature: minimize human-computer interaction and reduce the
112 class imbalance bias. This is done with the introduction of an additional component
113 in the iterative AL procedure (the generator) that is used to generate artificial data to
114 both balance and augment the training dataset. The introduction of this component
115 is expected to reduce the number of iterations required until the classifier reaches a
116 satisfactory performance convergence of the classifier's quality.

117 This paper is organized as follows: Section 1 explains the problem and its context,
118 Sections 2 and 3 describe the state of the art in AL and Oversampling techniques, Section
119 4 explains the proposed method, Section 5 covers the datasets, evaluation metrics, ML
120 classifiers and experimental procedure, Section 6 presents the experiment's results and
121 discussion and Section 7 presents the conclusions drawn from our findings.

122 2. Active Learning Approaches

123
124 As the amount of unlabeled data increases, the interest and practical usefulness of
125 AL follows that trend [22]. AL is used as the general definition of frameworks aiming to
126 train a learning system in multiple steps, where a set of new data points are chosen and
127 added to the training dataset each time [11]. Typically, an AL framework is composed of
128 the following elements [11,13,23]:

- 129 1. Unlabeled dataset. Consists of the original data source (or a sample thereof). It
130 is used in combination with the chooser and the selection criterion to expand the
131 training dataset in regions where the classification uncertainty is higher. Therefore,
132 the unlabeled dataset is used for both producing the initial training dataset by
133 selecting a set of instances for the supervisor to annotate (discussed in point 3) and
134 calculating the uncertainty map to augment the training dataset.
- 135 2. Supervisor. A human annotator (or team of human annotators) An external entity
136 to which the uncertainty map is presented to. The supervisor is responsible for
137 annotating unlabeled instances to be added to the augmented dataset. In remote
138 sensing, the supervisor is typically a photo-interpreter, as is the case in [24]. Some
139 of the research also refers to the supervisor as the oracle [11,25–27].

- 140 3. Initial training dataset. It is a small, **labeled** sample of **the original data source** used
 141 to initiate the first AL iteration. The size of the initial training sample normally
 142 varies between no instances at all and 10% **of the unlabeled dataset** [28].
 143 4. Current and expanded training dataset. It is the concatenation of the initial training
 144 **dataset** and the datasets labeled by the supervisor in past iterations (discussed in
 145 point 2).
 146 5. Chooser (classifier). Produces the class probabilities for each unlabeled instance.
 147 6. Selection criterion. It quantifies the chooser's uncertainty level for each instance
 148 belonging to the unlabeled dataset. It is typically based on the class probabilities
 149 assigned by the chooser. In some situations, the chooser and the selection criterion
 150 are grouped together under the concept *acquisition function* [11] or *query function* [13].
 151 Some of the literature refers to the selection criterion by using the concept *sampling
 152 scheme* [12].

153 Figure 1 schematizes the steps involved in a complete AL iteration. For a better
 154 context within the remote sensing domain, the prediction output **can be is** identified as
 155 the LULC map. This framework starts by collecting unlabeled data from the original
 156 data source. It is used to generate a random initial training sample and is labeled by
 157 the supervisor. In practical applications, the supervisor is frequently a group of photo-
 158 interpreters [22]. The chooser is trained on the resulting dataset and is used to predict the
 159 class probabilities on the unlabeled dataset. The class probabilities are fed into a selection
 160 criterion to estimate the prediction's uncertainty, out of which the instances with the
 161 highest uncertainty will be selected. This calculation is motivated by the absence of
 162 labels in the uncertainty dataset. Therefore, it is impossible to estimate the prediction's
 163 accuracy in the unlabeled dataset in a real case scenario. The iteration is completed when
 164 the selected points are tagged by the supervisor and added to the training dataset (*i.e.*,
 165 the augmented dataset).

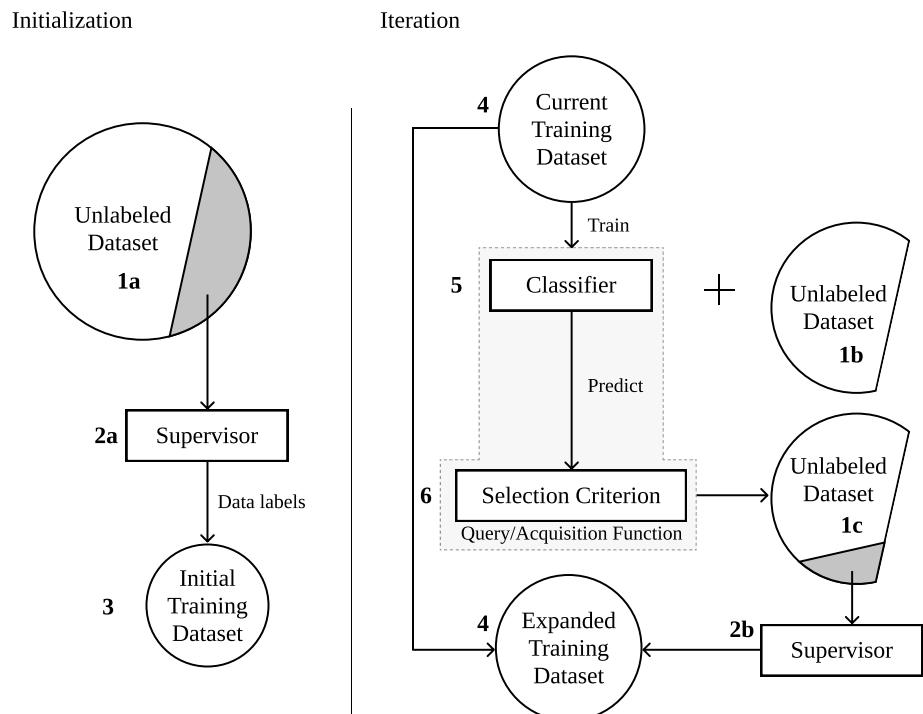


Figure 1. Diagram depicting the typical AL framework.

166 A common challenge found in AL tasks is ensuring the consistency of AL over
 167 different initializations [22]. There are two factors involved in this phenomenon. On one

hand, the implementation of the same method over different initializations may result in significantly different initial training samples, amounts to varying accuracy curves. On the other hand, the lack of a robust selection criterion and/or classifier chooser may also result in inconsistencies across AL experiments with different initializations. This phenomenon was observed and documented in a LULC classification context in [29].

The classification method plays a central role in the efficacy of AL. The classifier used should be able to generalise with a relatively small training dataset. Specifically, deep learning models are used in image classification due to its capability of producing high quality predictions. Although, to make such models generalizable the training set must be large enough, making its suitability for AL applications an open challenge [30–32]. Some studies in the Remote Sensing domain were developed to address this gap. In [30,32], the authors propose a deep learning-based AL approach by training the same Convolutional Neural Network incrementally across iterations and smoothen the decision boundaries of the model using the Markov Random Field model and a Best-versus-Second Best labelling approach. This allows the introduction of additional data variability in the final training dataset. Another study [31] combined transfer learning, active classification and segmentation techniques for vehicle detection. By combining different techniques, they were able to produce a classification mechanism that performed well when the amount of training data is limited.

Selecting an efficient selection criterion is particularly important to find the instances closest to the decision border (*i.e.*, instances difficult to classify) [33]. Therefore, many AL related studies focus on the design of the query/acquisition function [13].

190 2.1. Non-informed selection criteria

191 Only one non-informed (*i.e.*, random) selection criterion was found in the literature.
192 Random sampling selects unlabeled instances without considering any external information
193 produced by the chooser. Since the method for selecting the unlabeled instances is
194 random, this method disregards the usage of a chooser and is comparatively worse than
195 any other selection criterion. However, random sampling is still a powerful baseline
196 method [27].

197 2.2. Ensemble-based selection criteria

198 Ensemble disagreement is based on the class predictions of a set of classifiers. The
199 disagreement between all the predictions for a given instance is a common measure for
200 uncertainty, although computationally inefficient [11,14]. It is calculated using the set of
201 classifications over a single instance, given by the number of votes assigned to the most
202 frequent class [33]. This method was implemented successfully for complex applications
203 such as deep active learning [11].

204 Multiview [34] consists on the training of multiple independent classifiers using
205 different views, which correspond to the selection of subsets of features or instances
206 in the dataset. Therefore, it can be seen as a bootstrap aggregation (bagging) ensemble
207 disagreement method. It is represented by the maximum disagreement score out of set
208 of disagreements calculated for each view [33]. A lower value for this metric means a
209 higher classification uncertainty. Multiview-based maximum disagreement has been
210 successfully applied to hyper-spectral image classification in [35] and [36].

211 An adapted disagreement criterion for an ensemble of k -nearest neighbors has been
212 proposed in [14]. This method employs a k -nearest neighbors classifier and computes
213 an instance's classification uncertainty based on the neighbors' class frequency using
214 the maximum disagreement metric over varying values for k . As a result, this method is
215 comparable to computing the dominant class' score over a weighted k -nearest neighbors
216 classifier. This method was also used on a multimetric active learning framework [37].

217 Another relevant ensemble-based selection criterion is the binary random forest-
218 based query model [13]. This method employs a one-versus-one ensemble method
219 to demonstrate an efficient data selection method using the estimated probability of

220 each binary random forest and determining the classification uncertainty based on the
221 probabilities closest to 0.5 (*i.e.*, the least separable pair of classes are used to determine
222 the uncertainty value). However, this study fails to compare the proposed method with
223 other benchmark methods, such as random sampling.

224 2.3. *Entropy-based criteria*

225 A number of contributions have focused on entropy-based querying. The applica-
226 tion of entropy is common among active deep learning applications [26], where the
227 training of an ensemble of classifiers is often too expensive.

228 Entropy query-by-bagging (EQB), also defined as maximum entropy [12], is an
229 ensemble approach of the entropy selection criterion, originally proposed in [38]. This
230 strategy uses the set of predictions produced by the ensemble classifier to calculate those
231 many entropy measurements. The estimated uncertainty measure for one instance is
232 given by the maximum entropy within that set. EQB was observed to be an efficient
233 selection criterion. Specifically, [33] applied EQB on hyper-spectral remote sensing im-
234 agery using Support Vector Machines (SVM) and Extreme Learning Machines (ELM) as
235 choosers, achieving optimal results when combining EQB with ELM. Another study suc-
236 cessfully implemented this method on an active deep learning application [12]. Another
237 study improved over this method with a normalized EQB selection criterion [39].

238 2.4. *Other relevant criteria*

239 Margin Sampling is a SVM-specific criterion, based on the distance of a given point
240 to the SVM's decision boundary [33]. This method is less popular than the remaining
241 methods because it is limited to one type of chooser (SVMs). One extension of this
242 method is the multiclass level uncertainty [33], calculated by subtracting the instance's
243 distance to the decision boundaries of the two most probable classes [40].

244 The Mutual Information-based (MI) criterion selects the new training instances
245 by maximizing the mutual information between the classifier and class labels in order
246 to select instances from regions that are difficult to classify. Although this method is
247 commonly used, it is frequently outperformed by the breaking ties selection criterion [41,
248 42].

249 The breaking ties (BT) selection criterion was originally introduced in [43]. It
250 consists of the subtraction between the probabilities of the two most likely classes.
251 Another related method is Modified Breaking Ties scheme (MBT), which aims at finding
252 the instances containing the largest probabilities for the dominant class [42,44].

253 Another type of selection criteria identified is the loss prediction method [25]. This
254 method replaces the selection criterion with a predictor whose goal is to estimate the
255 chooser's loss for a given prediction. This allows the new classifier to estimate the
256 prediction loss on unlabeled instances and select the ones with the highest predicted
257 loss.

258 Some of the literature fails to specify the strategy employed, although inferring it is
259 generally intuitive. For example, [45] successfully used AL to address the imbalanced
260 learning problem. They employed an ensemble of SVMs as the chooser, as well as
261 an ensemble-based selection criterion. All of the research found related to this topic
262 focused on the improvement of AL through modifications on the selection criterion
263 and classifiers used. None of these publications proposed significant variations to the
264 original AL framework.

265 3. Artificial Data Generation Approaches

266 The generation of artificial data is a common approach to address imbalanced learn-
267 ing tasks [21], as well as improving the effectiveness of supervised learning tasks [46]. In
268 recent years some sophisticated data generation approaches were developed. However,
269 the scope of this work is to propose the integration of a generator within the AL frame-

work. To do this, we will focus on heuristic data generation approaches, specifically, oversamplers.

Heuristic data resampling methods employ local and/or global information to generate new, relevant, non-duplicate 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) [47] is a popular heuristic oversampling algorithm, proposed in 2002. The simplicity and effectiveness of this method contributes to its prevailing popularity. It generates a new instance through a linear interpolation of a randomly selected minority-class instance and one of its randomly selected k -nearest neighbors. The implementation of SMOTE for LULC classification tasks has been found to improve the quality of the predictors used [48,49]. Despite its popularity, its drawbacks motivated the development of other oversampling methods [50].

Geometric SMOTE (G-SMOTE) [50] introduces a modification of the SMOTE algorithm in the data generation mechanism to produce artificial instances with higher variability. Instead of generating artificial data as a linear combination of the parent instances, it is done within a deformed, truncated hyper-spheroid. G-SMOTE generates an artificial instance \vec{z} within a hyper-spheroid, formed by selecting a minority instance \vec{x} and one of its nearest neighbors \vec{y} , 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 instances using instances from different classes, as shown in Figure 2d. The modification of both selection and generation mechanisms addresses the main drawbacks found in SMOTE, the generation of both noisy data (*i.e.*, generate minority class instances within majority class regions) and near-duplicate minority class instances [50]. G-SMOTE has shown superior performance when compared with other oversampling methods for LULC classification tasks, regardless of the classifier used [51].

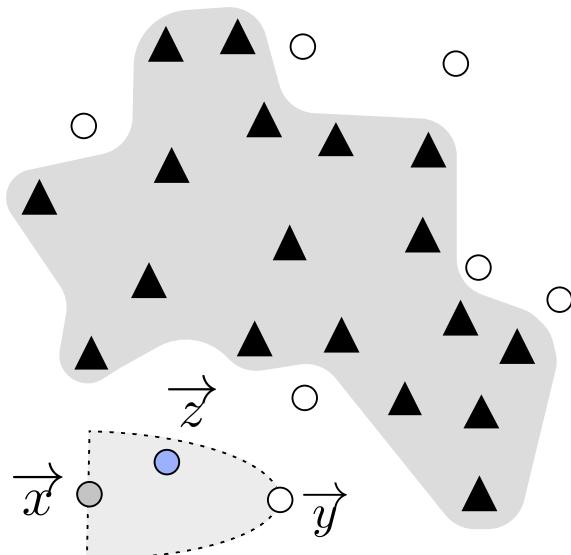


Figure 2. Example of G-SMOTE's generation process. G-SMOTE randomly selects instance \vec{x} and one of its nearest neighbors \vec{y} to produce instance \vec{z} .

4. Proposed method

Within the literature identified, most of the work developed in the AL domain revolved around improving the quality of classification algorithms and/or selection criteria. Although these methods allow earlier convergence of the AL iterative process,

303 the impact of these methods are only observed between iterations. Consequently, none
 304 of these contributions focused on the definition of decision borders within iterations. The
 305 method proposed in this paper modifies the AL framework by introducing an artificial
 306 data generation step within AL's iterative process. We define this component as the
 307 generator and is intended to be integrated into the AL framework as shown in Figure 3.

308 This modification, by using a new source of data to augment the training set,
 309 leverages the data annotation work conducted by the human operator. The artificial
 310 data that is generated between iterations reduces the amount of labeled data required
 311 to reach optimal performance and lower the amount of human labor required to train
 312 a classifier to its optimal performance. This process lowers the annotation and overall
 313 training costs by translating some of the annotation cost into computational cost.

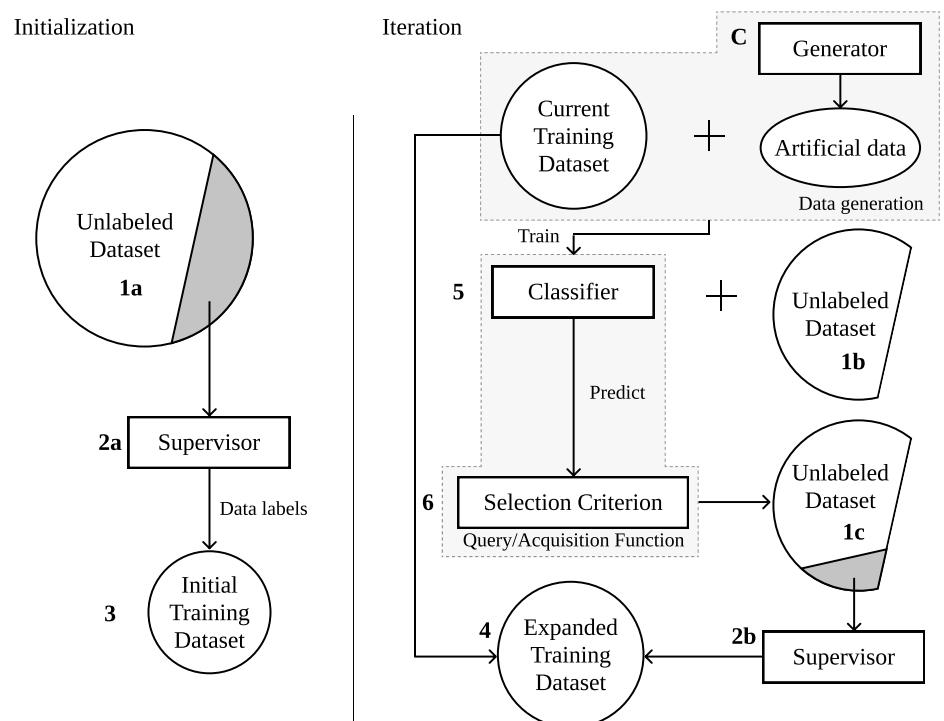


Figure 3. Proposed AL framework. **This paper's contribution, comprises a change in the AL framework through the introduction of a data generation mechanism, the data generation mechanism, is represented as the generator (marked with C), which is used to add artificial instances to the training dataset. data generation phase. The remaining steps are left unchanged.**

314 This method leverages the capability of artificial data to introduce more data vari-
 315 ability into the augmented dataset and facilitate the chooser's training phase with a
 316 more consistent definition of the decision boundaries at each iteration. Therefore, any
 317 algorithm capable of producing artificial data, be it agnostic or specific to the domain,
 318 can be employed. The artificial data is only used to train the classifiers involved in the
 319 process and is discarded once the training phase is completed. The remaining steps in
 320 the AL framework remain unchanged. This method addresses the limitations found in
 321 the previous sections:

- 322 1. The convergence of classification performance should be anticipated with the
 323 clearer definition of the decision boundaries across iterations.
- 324 2. Annotation cost is expected to reduce as the need for labeled instances reduces
 325 along with the early convergence of the classification performance.
- 326 3. The class imbalance bias observed in typical classification tasks, as well as in AL is
 327 mitigated by balancing the class frequencies at each iteration.

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

334 5. Methodology

335
 336 In this section we describe the datasets, evaluation metrics, oversampler, classifiers,
 337 software used and the procedure developed. We demonstrate the proposed method's
 338 efficiency over 7 datasets, sampled from publicly available, well-known remote sensing
 339 hyperspectral scenes frequently found in remote sensing literature. The datasets and
 340 sampling strategy are described in Subsection 5.1. On each of these datasets, we apply
 341 3 different classifiers over the entire training set to estimate the optimal classification
 342 performance, the original AL framework as the baseline reference and the proposed
 343 method using G-SMOTE as a generator, described in Subsection 5.2. The metrics used to
 344 estimate the performance of these algorithms are described in Subsection 5.3. Finally,
 345 the experimental procedure is described in Subsection 5.4.

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

349 5.1. Datasets

350
 351 The datasets used were extracted from publicly available repositories containing
 352 hyperspectral images and ground truth data. Additionally, all datasets were collected
 353 using the same sampling procedure. The description of the hyperspectral scenes used in
 354 this study is provided in Table 2. These scenes were chosen because of their popularity
 355 in the research community and their high baseline classification scores. Consequently,
 356 demonstrating an outperforming method in this context is particularly challenging and
 357 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 2: Description of the hyperspectral scenes used in this experiment. The column “Res. (m)” refers to the resolution of the sensors (in meters) that captured each of the scenes.

358 The Indian Pines scene [52] is composed of agriculture fields in approximately
 359 two thirds of its coverage, low density buildup areas and natural perennial vegetation
 360 in the remainder of its area (see Figure 4a). The Pavia Centre and University scenes
 361 are hyperspectral, high-resolution images containing ground truth data composed of
 362 urban-related coverage (see Figures 4b and 4c). The Salinas and Salinas A scenes contain
 363 at-sensor radiance data. As subset of Salinas, the Salinas A scene contains contains the
 364 vegetables fields present in Salinas and the latter is also composed of bare soils and
 365 vineyard fields (see Figures 4d and 4e). The Botswana scene contains ground truth data

³⁶⁶ composed of seasonal swamps, occasional swamps, and drier woodlands located in the
³⁶⁷ distal portion of the Delta (see Figure 4f). The Kennedy Space Center scene contains a
³⁶⁸ ground truth composed of both vegetation and urban-related coverage (see Figure 4g).

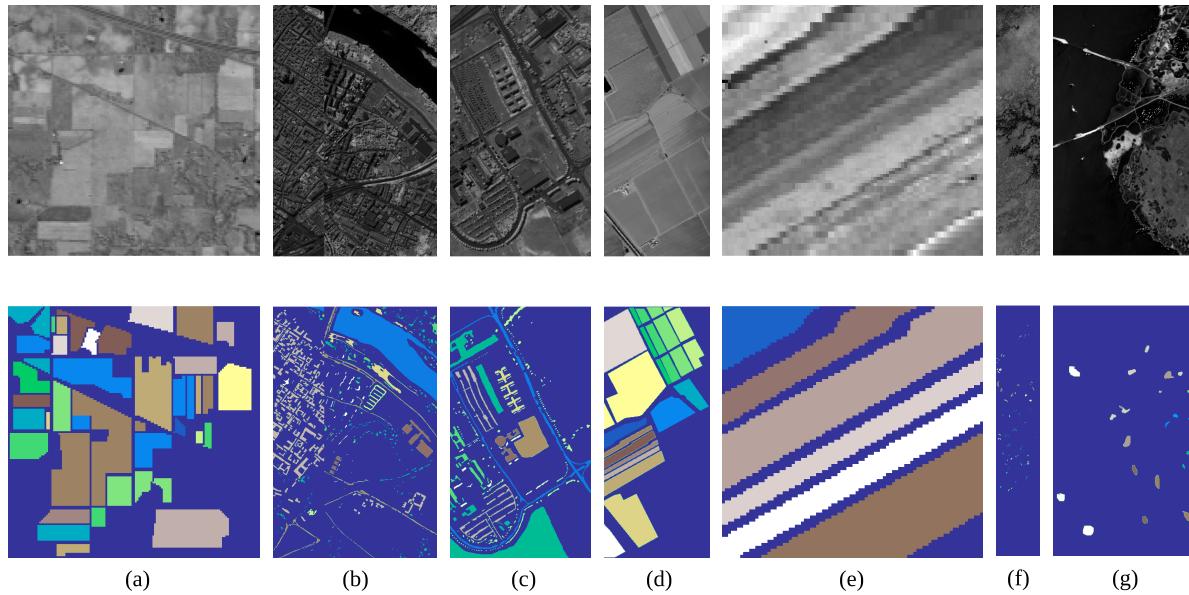


Figure 4. Gray scale visualization of a band (top row) and ground truth (bottom row) of each scene used in this study. (a) Indian Pines, (b) Pavia Centre, (c) Pavia University, (d) Salinas, (e) Salinas A, (f) Botswana, (g) Kennedy Space Center

³⁶⁹ The sampling strategy is similar to all datasets. The pixels without a ground
³⁷⁰ truth label are first discarded. All the classes with cardinality lower than 150 are also
³⁷¹ discarded. This is done to maintain feasible Imbalance Ratios (IR) across datasets
³⁷² (where $IR = \frac{count(C_{maj})}{count(C_{min})}$). Finally, a stratified sample of 1500 instances are selected for
³⁷³ the experiment. The resulting datasets are described in Table 3. The motivation for
³⁷⁴ this strategy is three fold: (1) reduce the datasets to a manageable size and allow the
³⁷⁵ experimental procedure to be completed within a feasible time frame, (2) ensure the
³⁷⁶ relative class frequencies in the scenes are preserved and (3) ensure equivalent analyses
³⁷⁷ across datasets and AL frameworks. In this context, a fixed number of instances per
³⁷⁸ 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 3: Description of the datasets collected from each corresponding scene. The sampling strategy is similar to all scenes.

³⁷⁹ 5.2. Machine Learning Algorithms

³⁸⁰
³⁸¹ We use two different types of ML algorithms. A data generation algorithm, used
³⁸² to form the generator, and classification algorithms, used to calculate the classification

383 uncertainties in the unlabeled dataset and predict the class labels in the validation and
 384 test sets.

385 Although any method capable of generating artificial data can be used as a generator,
 386 the one used in this experiment is an oversampler, originally developed to deal with
 387 imbalanced learning problems. Specifically, we chose G-SMOTE, a state-of-the-art
 388 oversampler.

389 Three classification algorithms are used. We use different types of classifiers to
 390 test the framework's performance under varying situations: neighbors-based, linear
 391 and ensemble models. The neighbors-based classifier chosen was K-nearest neighbors
 392 (KNN) [53], a logistic regression (LR) [54] is used as the linear model and a random
 393 forest classifier (RFC) [55] was used as the ensemble model.

394 The acquisition function is completed by testing three different selection criteria.
 395 Random selection is used as a baseline selection criterion, whereas entropy and breaking
 396 ties are used due to their popularity and independence of the classifier used.

397 5.3. Evaluation Metrics

398
 399 Since the datasets used in this experiment have an imbalanced distribution of
 400 class frequencies, metrics such as the *Overall Accuracy* (OA) and *Kappa coefficient* are
 401 insufficient to accurately depict classification performance [56,57]. Instead, metrics such
 402 as Producer's Accuracy (or *Recall*) and User's Accuracy (or *Precision*) can be used. Since
 403 they consist of ratios based on True/False Positives (TP and FP) and Negatives (TN
 404 and FN), they provide per class information regarding the classifier's classification
 405 performance. However, in this experiment, the meaning and number of classes available
 406 in each dataset varies, making these metrics difficult to synthesize.

407 The performance metric *Geometric mean* (G-mean) and *F-score* are less sensitive to
 408 the data imbalance bias [58,59]. Therefore, we employ both of these scorers. G-mean
 409 consists of the geometric mean of $Specificity = \frac{TN}{TN+FP}$ and $Sensitivity = \frac{TP}{TP+FN}$ (also
 410 known as *Recall*) [59]. Both metrics are calculated in a multiclass context considering a
 411 one-versus-all approach. For multiclass problems, the *G-mean* scorer is calculated as its
 412 average per class values:

$$G\text{-mean} = \sqrt{Sensitivity_i \times Specificity_i}$$

413 The *F-score* performance metric is the harmonic mean of *Precision* and *Recall*. The
 414 two metrics are also calculated considering a one-versus-all approach. The *F-score* for
 415 the multi-class case can be calculated using its average per class values [60]:

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

416 The comparison of classification convergence across AL frameworks and selection
 417 criteria is done using 2 AL-specific performance metrics. Particularly, we follow the
 418 recommendations found in [22]. Each AL configuration is evaluated using the *Area
 419 Under the Learning Curve* (AULC) performance metric. It is the sum of the classification
 420 performance values of all iterations. To facilitate the analysis of the results, we fix the
 421 range of this metric between [0, 1] by dividing it with the total amount of iterations (*i.e.*,
 422 the maximum performance area).

423 The *Data Utilization Rate* (DUR) [61] metric consists of the ratio between the number
 424 of instances required to reach a given G-mean score threshold by an AL strategy and
 425 an equivalent baseline strategy. For easier interpretability, we simplify this metric by
 426 using the percentage of training data used by an AL strategy to reach the performance
 427 threshold, instead of presenting these values as a ratio of the baseline strategy. The DUR
 428 metric is measured at 9 different performance levels, between 0.6 and 0.95 G-mean scores
 429 at a 0.05 step.

430 5.4. Experimental Procedure

431

432 A common practice in methodological evaluations is the implementation of an
 433 offline experiment [62]. It consists of using an existing set of labeled data as a proxy for
 434 the population of unlabeled instances. Because the dataset is already fully labeled, the
 435 supervisor's typical annotation process involved in each iteration is done at zero cost.
 436 Each AL and classifier configuration is tested using a stratified 5-fold cross validation
 437 testing scheme. For each round, the larger partition is split in a stratified fashion to form a
 438 training and validation set (containing 20% of the original partition). The validation set is
 439 used to evaluate the convergence efficiency of active learners; the chooser's classification
 440 performance metrics and amount of data points used at each iteration are used to
 441 compute the AULC and DUR. Additionally, within the AL iterative process, the classifier
 442 with optimal performance on the validation set is evaluated using the test set. In
 443 order to further reduce possible initialization biases, this procedure is repeated 3 times
 444 with different initialization seeds and the results of all runs are averaged (*i.e.*, each
 445 configuration is trained and evaluated 15 times). Finally, the maximum performance
 446 lines are calculated using the same approach. In those cases, the validation set is not
 447 used. The experimental procedure is depicted in Figure 5.

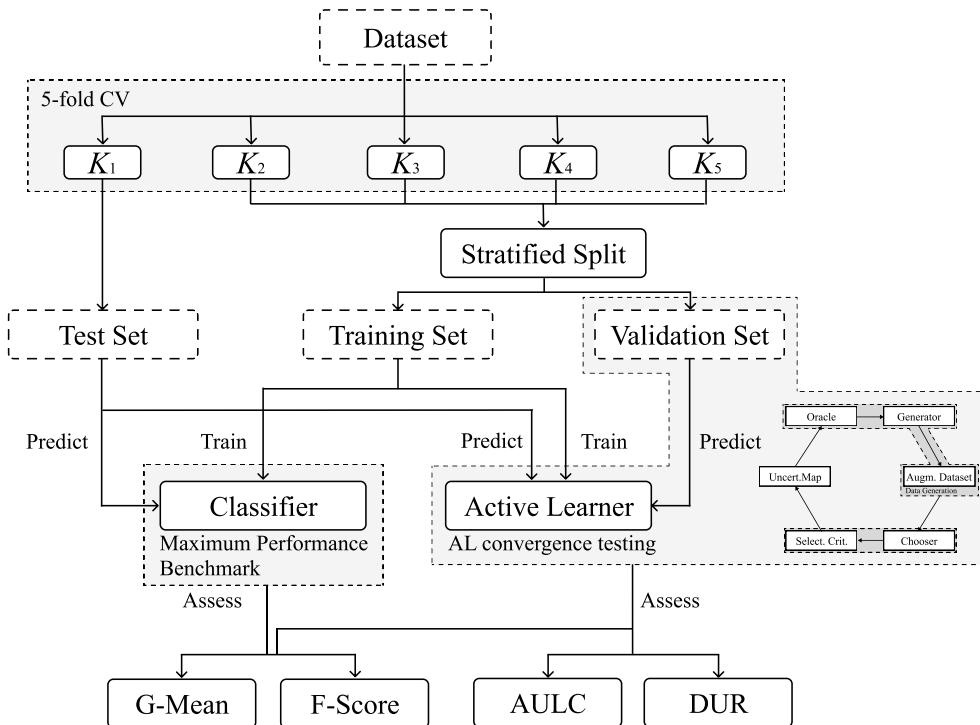


Figure 5. Experimental procedure. The datasets extracted from hyperspectral scenes are split in 5 folds. 1 of those (*e.g.*, K_1) is used to test the optimal performance of AL algorithms and the classification without AL. The training set is used to iterate AL algorithms and train classifiers. The validation set is used to test the convergence of AL algorithms. The results are averaged over the 5 folds across each of the 3 different initializations of this procedure.

448

449 To make the AL-specific metrics comparable among active learners, the configura-
 450 tions of the different frameworks must be similar. For each dataset, the number of
 451 instances is constant to facilitate the analysis of the same metrics.

451

452 In most practical AL applications it is assumed that the number of instances in the
 453 initial training sample is too small to perform hyperparameter tuning. Consequently,
 454 in order to ensure realistic results, our experimental procedure does not include hyper-
 455 parameter optimization. The predefined hyperparameters are shown in Table 4. They

⁴⁵⁵ were set up based on general recommendations and default settings for the classifiers
⁴⁵⁶ and generators used.

⁴⁵⁷ The AL iterative process is set up with a randomly selected initial training sample
⁴⁵⁸ with 15 initial samples. At each iteration, 15 additional samples are added to the training
⁴⁵⁹ set. This process is stopped after 49 iterations, once 50% of the entire dataset (*i.e.*, 78% of
⁴⁶⁰ the training set) is added to the augmented dataset.

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

Table 4: Hyper-parameter definition for the classifiers and generator used in the experiment.

⁴⁶¹ 5.5. Software Implementation

⁴⁶² The experiment was implemented using the Python programming language, along
⁴⁶³ with the Python libraries [Scikit-Learn](#) [63], [Imbalanced-Learn](#) [64], [Geometric-SMOTE](#),
⁴⁶⁴ [Cluster-Over-Sampling](#) and [Research-Learn](#) libraries. All functions, algorithms, experi-
⁴⁶⁵ ments and results are provided in the [GitHub repository of the project](#).

⁴⁶⁶ 6. Results & Discussion

⁴⁶⁷ The evaluation of the different AL frameworks in a multiple dataset context should
⁴⁶⁸ not rely uniquely on the mean of the performance metrics across datasets. [65] recom-
⁴⁶⁹ mends the use of mean ranking scores, since the performance levels of the different
⁴⁷⁰ frameworks varies according to the data it is being used on. Consequently, evaluating
⁴⁷¹ these performance metrics solely based on their mean values might lead to inaccurate
⁴⁷² analyses. Accordingly, the results of this experiment are analysed using both the mean
⁴⁷³ ranking and absolute scores for each model. The rank values are assigned based on the
⁴⁷⁴ mean scores resulting from three different initializations of 5-fold cross validation for
⁴⁷⁵ each classifier and active learner. The goal of this analysis is to understand whether the
⁴⁷⁶ proposed framework (AL with the integration of an artificial data generator) is capable
⁴⁷⁷ of using less data from the original dataset while simultaneously achieving better classi-
⁴⁷⁸ fication results than the standard AL framework, *i.e.*, guarantee a faster classification
⁴⁷⁹ convergence.

⁴⁸¹ 6.1. Results

⁴⁸²
⁴⁸³ Table 5 shows the average rankings and standard deviations across datasets of the
⁴⁸⁴ AULC scores for each active learner.

Classifier	Evaluation Metric	Standard	Proposed
KNN	F-score	2.00 ± 0.0	1.00 ± 0.0
KNN	G-mean	2.00 ± 0.0	1.00 ± 0.0
LR	F-score	1.71 ± 0.45	1.29 ± 0.45
LR	G-mean	2.00 ± 0.0	1.00 ± 0.0
RF	F-score	1.86 ± 0.35	1.14 ± 0.35
RF	G-mean	2.00 ± 0.0	1.00 ± 0.0

Table 5: Mean rankings of the AULC metric over the different datasets (7), folds (5) and runs (3) used in the experiment. This means that the use of G-SMOTE **almost** always improves the results of the original framework.

485 The mean AULC absolute scores are provided in Table 6. These values are computed
 486 as the mean of the sum of the scores of a specific performance metric over all iterations
 487 (for an AL configuration). In other words, these values correspond to the average AULC
 488 over 7 datasets \times 5 folds \times 3 initializations.

Classifier	Evaluation Metric	Standard	Proposed
KNN	F-score	0.762 ± 0.131	0.794 ± 0.123
KNN	G-mean	0.864 ± 0.079	0.886 ± 0.073
LR	F-score	0.839 ± 0.119	0.843 ± 0.116
LR	G-mean	0.907 ± 0.074	0.911 ± 0.071
RF	F-score	0.810 ± 0.109	0.819 ± 0.1
RF	G-mean	0.890 ± 0.068	0.901 ± 0.059

Table 6: Average AULC of each AL configuration tested. Each AULC score is calculated using the G-mean scores of each iteration in the validation set. By the end of the iterative process, each AL configuration used a total of 750 instances of the 960 instances that compose the training set.

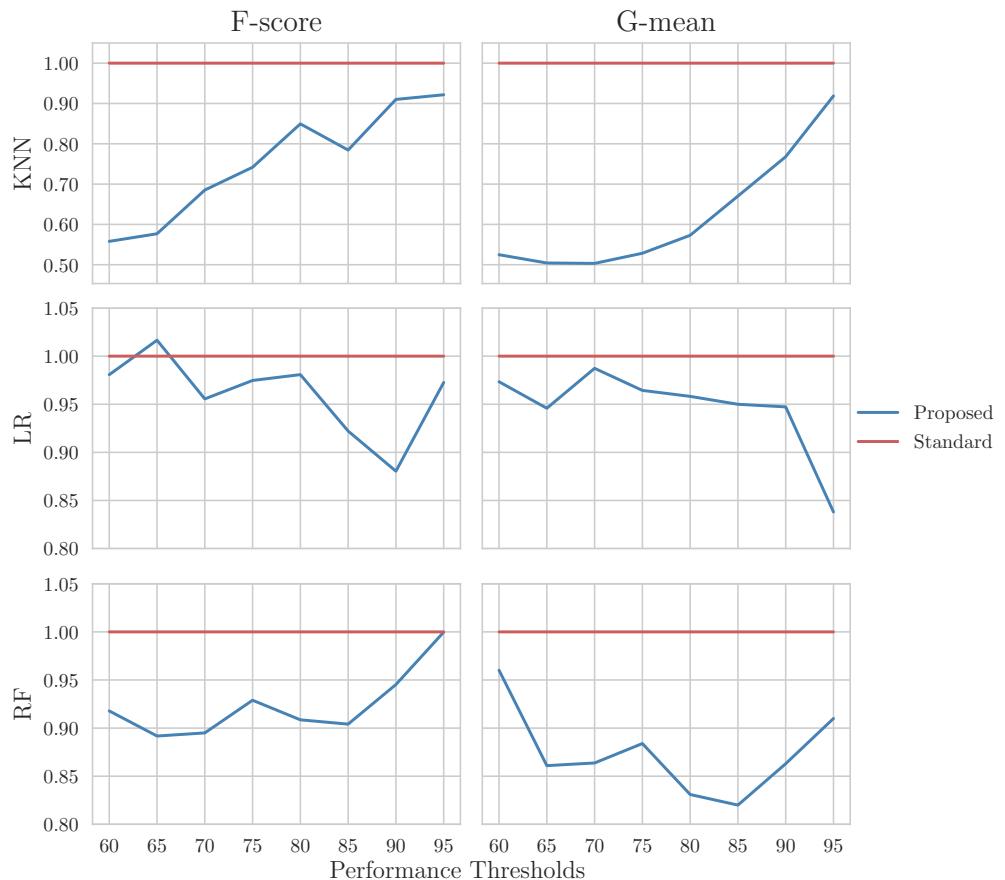
489 The average DURs are shown in Table 4. They were calculated for various G-mean
 490 scores thresholds, varying at a step of 5% between 60% and 95%. Each row shows the
 491 percentage of training data required by the different AL configurations to reach that
 492 specific G-mean score.

G-mean Score	Classifier	Standard	Proposed
0.60	KNN	4.0%	2.1%
0.60	LR	2.2%	2.1%
0.60	RF	2.2%	2.1%
0.65	KNN	5.6%	2.8%
0.65	LR	3.0%	2.7%
0.65	RF	3.1%	2.6%
0.70	KNN	7.9%	4.1%
0.70	LR	4.2%	4.1%
0.70	RF	4.5%	3.6%
0.75	KNN	13.5%	7.1%
0.75	LR	7.2%	6.6%
0.75	RF	6.6%	5.4%
0.80	KNN	24.4%	16.9%
0.80	LR	13.1%	11.7%
0.80	RF	11.6%	9.2%
0.85	KNN	29.8%	23.6%
0.85	LR	19.8%	18.8%
0.85	RF	23.1%	17.3%

G-mean Score	Classifier	Standard	Proposed
0.90	KNN	41.0%	36.1%
0.90	LR	28.1%	24.8%
0.90	RF	37.1%	30.3%
0.95	KNN	71.3%	69.1%
0.95	LR	45.8%	40.2%
0.95	RF	64.6%	62.2%

Table 4: Mean data utilization of AL algorithms, as a percentage of the training set.

493 The DUR of the proposed method relative to the baseline method is shown in
 494 Figure 6. A DUR below 1 means that the proposed framework requires less data to reach
 495 the same performance threshold (as a percentage, relative to the amount of data required
 496 by the baseline framework). For instance, in the upper left graphic we can see that the
 497 proposed framework achieves 90% classification using F-score while using 91% of the
 498 amount of data used by the traditional AL framework, in other words 9% less data.

**Figure 6.** Mean data utilization rates. The y-axis shows the percentage of data (relative to the baseline AL framework) required to reach the different performance thresholds.

499 The averaged optimal classification scores are shown in Table 5. The maximum
 500 performance (MP) classification scores are shown as a benchmark and represent the
 501 performance of the corresponding classifier using the entire training set.

Classifier	Evaluation Metric	MP	Standard	Proposed
KNN	F-score	0.838 ± 0.106	0.835 ± 0.115	0.843 ± 0.105
KNN	G-mean	0.907 ± 0.063	0.904 ± 0.069	0.912 ± 0.061
LR	F-score	0.890 ± 0.084	0.883 ± 0.096	0.887 ± 0.097
LR	G-mean	0.935 ± 0.052	0.931 ± 0.059	0.938 ± 0.055
RF	F-score	0.859 ± 0.083	0.866 ± 0.081	0.869 ± 0.08
RF	G-mean	0.918 ± 0.051	0.921 ± 0.051	0.930 ± 0.043

Table 5: Optimal classification scores. The Maximum Performance (MP) classification scores are calculated using classifiers trained using the entire training set.

502 6.2. Statistical Analysis

503
 504 The methods used to test the experiment's results must be appropriate for a multi-
 505 dataset context. Therefore the statistical analysis is performed using the Wilcoxon signed-
 506 rank test [66] as a post-hoc analysis. The variable used for this test is the data utilization
 507 rate **based on the G-mean performance metric**, considering the various performance
 508 thresholds from Table 4.

509 The Wilcoxon signed-rank test results are shown in Table 6. We test as null hypoth-
 510 esis that the performance of the proposed framework is the same as the original AL
 511 framework. The null hypothesis was rejected in all datasets.

Dataset	p-value	Significance
Botswana	3.8e-03	True
Indian Pines	2.3e-04	True
Kennedy Space Center	1.3e-04	True
Pavia Centre	4.3e-03	True
Pavia University	4.6e-05	True
Salinas	4.6e-05	True
Salinas A	3.0e-03	True

Table 6: Adjusted p-values using the Wilcoxon signed-rank method. Bold values are statistically significant at a level of $\alpha = 0.05$. The null hypothesis is that the performance of the proposed framework is similar to that of the original framework.

512 6.3. Discussion

513 This paper expands the AL framework by adding an artificial data generator into its
 514 iterative process. This modification is done to accelerate the classification convergence
 515 of the standard AL procedure, which is reflected in the reduction of the amount of data
 516 necessary to reach better classification results.

517 The convergence efficiency of the proposed method is always higher than the
 518 baseline AL framework (**NONE**), **with the exception of one comparison**, as shown in
 519 Table 5 and Figure 6. **This means the proposed**The AL framework using data generation
 520 was able to outperform the baseline AL in **nearly** all scenarios.

521 The mean AULC scores in Table 6 show a significant improvement in the per-
 522 formance of AL when a generator is used. The mean performance of the proposed
 523 framework is always better than the baseline framework. This improvement is explained
 524 by:

- 525 1. Earlier convergence of AL, *i.e.*, requiring less data to achieve comparable perfor-
 526 mance levels. This effect is shown in Table 4, where we found that the proposed
 527 framework always uses less data for similar performance levels, regardless of the
 528 classifier used.

529 2. Higher optimal classification performance, *i.e.*, reaching higher performance levels
530 overall. This effect is shown in Table 5, where we found that using a generator in
531 AL led to a better classification performance and was capable of outperforming the
532 MP threshold.

533 Our results show statistical significance in every dataset. The proposed framework
534 had a superior performance with statistical significance on each dataset at a level of
535 $\alpha = 0.05$. This indicates that regardless of the context under which an AL algorithm is
536 used, the proposed framework reduces the amount of data necessary in the AL's iterative
537 process.

538 This paper introduces the concept of applying data a generation algorithm in the
539 AL framework. This was done with the implementation of a recent state of the art
540 generalization of a popular data generation algorithm. Although, since this algorithm
541 is based on heuristics, future work should focus on improving these results through
542 the design of new data generation mechanisms, at the cost of additional computational
543 power. In addition, we also noticed significant standard errors in our experimental
544 results (see Subsection 6.1). This indicates that AL procedures seem to be particularly
545 sensitive to the initialization method, which is still a limitation of AL, regardless of the
546 framework and configurations used. This is consistent with the findings in [22], which
547 future work should attempt to address. Although using a generator marginally reduced
548 this standard error, it is not sufficient to address this specific limitation.

549 7. Conclusion

550
551 The aim of this experiment was to test the effectiveness of a new AL framework
552 that introduces artificial data generation in its iterative process. The experiment was
553 designed to test the proposed method under particularly challenging conditions, where
554 the maximum performance line is naturally high in most datasets. The element that
555 constitute the Generator component was set up in a plug-and-play scheme, without
556 significant tuning of the G-SMOTE oversampler. Using a generator in AL improved
557 the original AL framework in all scenarios. These results could be further improved
558 through the modification and more intense tuning of the data generation strategy. In
559 our experiment, artificial data was generated only to match each non-majority class
560 frequency with the majority class frequency, strictly balancing the class distribution.
561 Generating a larger amount of data for all classes can further improve these results.

562 The high performance scores for the baseline AL framework made the achievement
563 of significant improvements over the traditional AL framework under these conditions
564 particularly meaningful. The advantage of the proposed AL framework is shown in
565 Table 4. In most of the presented scenarios there is a substantial reduction of data
566 necessary to reach a given performance threshold.

567 The results from this experiment show that using a data generator in the AL frame-
568 work will improve the convergence of the method. This framework successfully antici-
569 pate the predictor's optimal performance, as shown in Tables 5, 6 and 4. Therefore, in a
570 real application, the annotation cost would have been reduced since less iterations and
571 labeled instances are necessary to reach near optimal classification performance.

572 **Acknowledgments:** The authors would like to thank Professor Victor Lobo (NOVA IMS, Universi-
573 dade Nova de Lisboa, and CINAV, Escola Naval, CIDIUM) for reviewing this paper and providing
574 important feedback throughout its development.

575 **Author Contributions:** Conceptualization, F.B.; Methodology, J.F. and G.D.; Software, J.F. and
576 G.D.; Validation, F.B., G.D.; Formal Analysis, J.F.; Writing - Original Draft Preparation, J.F.; Writing
577 - Review & Editing, F.B., G.D., J.F.; Supervision, F.B.; Funding Acquisition, F.B.

578 **Funding:** This research was funded by “Fundação para a Ciência e a Tecnologia” (Portugal)
579 [grant numbers PCIF/SSI/0102/2017 - foRESTER, DSAIPA/AI/0100/2018 - IPSTERS].

580 **Data Availability Statement:** The data reported in this study is publicly available. It can
581 be retrieved and preprocessed using the Python source code provided at <https://github.com/joaopfonseca/research>. Alternatively the original data is available at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

584 **Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the
585 design of the study; in the collection, analyses, or interpretation of data; in the writing of the
586 manuscript, or in the decision to publish the results.

References

1. Nagai, S.; Nasahara, K.N.; Akitsu, T.K.; Saitoh, T.M.; Muraoka, H. 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*; American Geophysical Union (AGU), 2020; pp. 223–244. doi:10.1002/9781119413332.ch11.
2. Huang, Y.; xin CHEN, Z.; YU, T.; zhi HUANG, X.; fa GU, X. Agricultural remote sensing big data: Management and applications. *Journal of Integrative Agriculture* **2018**, *17*, 1915–1931. doi:10.1016/S2095-3119(17)61859-8.
3. Wang, X.; Xie, H. A review on applications of remote sensing and geographic information systems (GIS) in water resources and flood risk management. *Water (Switzerland)* **2018**, *10*, 608. doi:10.3390/w10050608.
4. Khatami, R.; Mountrakis, G.; Stehman, S.V. 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* **2016**, *177*, 89–100. doi:10.1016/J.RSE.2016.02.028.
5. Gavade, A.B.; Rajpurohit, V.S. Systematic analysis of satellite image-based land cover classification techniques: literature review and challenges. *International Journal of Computers and Applications* **2019**, pp. 1–10. doi:10.1080/1206212x.2019.1573946.
6. Wulder, M.A.; Coops, N.C.; Roy, D.P.; White, J.C.; Hermosilla, T. Land cover 2.0. *International Journal of Remote Sensing* **2018**, *39*, 4254–4284. doi:10.1080/01431161.2018.1452075.
7. Costa, H.; Benevides, P.; Marcelino, F.; Caetano, M. 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* **2020**, *42*, 29–34.
8. Vermote, E.F.; Skakun, S.; Becker-Reshef, I.; Saito, K. Remote Sensing of Coconut Trees in Tonga Using Very High Spatial Resolution WorldView-3 Data. *Remote Sensing* **2020**, *12*, 3113.
9. Costantino, D.; Pepe, M.; Dardanelli, G.; Baiocchi, V. USING OPTICAL SATELLITE AND AERIAL IMAGERY FOR AUTOMATIC COASTLINE MAPPING. *Geographia Technica* **2020**, pp. 171–190. doi:10.21163/gt_2020.152.17.
10. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing* **2018**, *39*, 2784–2817. doi:10.1080/01431161.2018.1433343.
11. Růžička, V.; D'Aronco, S.; Wegner, J.D.; Schindler, K. Deep Active Learning in Remote Sensing for data efficient Change Detection. *arXiv preprint arXiv:2008.11201* **2020**.
12. Liu, S.J.; Luo, H.; Shi, Q. Active Ensemble Deep Learning for Polarimetric Synthetic Aperture Radar Image Classification. *IEEE Geoscience and Remote Sensing Letters* **2020**, pp. 1–5, [2006.15771]. doi:10.1109/lgrs.2020.3005076.
13. Su, T.; Zhang, S.; Liu, T. Multi-spectral image classification based on an object-based active learning approach. *Remote Sensing* **2020**, *12*, 504. doi:10.3390/rs12030504.
14. Pasolli, E.; Yang, H.L.; Crawford, M.M. Active-metric learning for classification of remotely sensed hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing* **2016**, *54*, 1925–1939. doi:10.1109/TGRS.2015.2490482.
15. Pelletier, C.; Valero, S.; Ingla, J.; Champion, N.; Sicre, C.M.; Dedieu, G. Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sensing* **2017**, *9*, 173. doi:10.3390/rs9020173.
16. Stromann, O.; Nascenti, A.; Yousif, O.; Ban, Y. 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* **2020**, *12*, 76. doi:10.3390/RS12010076.
17. Alonso-Sarria, F.; Valdivieso-Ros, C.; Gomariz-Castillo, F. Isolation forests to evaluate class separability and the representativeness of training and validation areas in land cover classification. *Remote Sensing* **2019**, *11*, 3000. doi:10.3390/rs11243000.
18. Feng, W.; Huang, W.; Ye, H.; Zhao, L. Synthetic minority over-sampling technique based rotation forest for the classification of unbalanced hyperspectral data. International Geoscience and Remote Sensing Symposium (IGARSS). Institute of Electrical and Electronics Engineers Inc., 2018, Vol. 2018-July, pp. 2651–2654. doi:10.1109/IGARSS.2018.8518242.
19. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter* **2004**, *6*, 1–6. doi:10.1145/1007730.1007733.
20. Fernández, A.; López, V.; Galar, M.; del Jesus, M.J.; Herrera, F. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems* **2013**, *42*, 97–110. doi:10.1016/J.KNOSYS.2013.01.018.
21. Kaur, H.; Pannu, H.S.; Malhi, A.K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys* **2019**, *52*, 1–36. doi:10.1145/3343440.
22. Kottke, D.; Calma, A.; Huseljic, D.; Kreml, G.; Sick, B. Challenges of reliable, realistic and comparable active learning evaluation. CEUR Workshop Proceedings, 2017, Vol. 1924, pp. 2–14.

23. Sverchkov, Y.; Craven, M. A review of active learning approaches to experimental design for uncovering biological networks. *PLOS Computational Biology* **2017**, *13*, e1005466. doi:10.1371/journal.pcbi.1005466.
24. Li, J.; Huang, X.; Chang, X. 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* **2020**, *163*, 1–17. doi:10.1016/j.isprsjprs.2020.02.022.
25. Yoo, D.; Kweon, I.S. Learning Loss for Active Learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
26. Aghdam, H.H.; Gonzalez-Garcia, A.; Lopez, A.; Weijer, J. Active learning for deep detection neural networks. Proceedings of the IEEE International Conference on Computer Vision, 2019, Vol. 2019-Octob, pp. 3671–3679, [1911.09168]. doi:10.1109/ICCV.2019.00377.
27. Cawley, G. Baseline Methods for Active Learning. *Proceedings of Active Learning and Experimental Design workshop In conjunction with AISTATS* **2011**, *16*, 47–57.
28. Li, X.; Guo, Y. Active learning with multi-label SVM classification. In IJCAI, 2013, pp. 1479–1485.
29. Tuia, D.; Pasolli, E.; Emery, W.J. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment* **2011**, *115*, 2232–2242.
30. Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral Image Classification with Convolutional Neural Network and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing* **2020**, *58*, 4604–4616. doi:10.1109/TGRS.2020.2964627.
31. Wu, X.; Li, W.; Hong, D.; Tian, J.; Tao, R.; Du, Q. Vehicle detection of multi-source remote sensing data using active fine-tuning network. *ISPRS Journal of Photogrammetry and Remote Sensing* **2020**, *167*, 39–53, [2007.08494]. doi:10.1016/j.isprsjprs.2020.06.016.
32. Bi, H.; Xu, F.; Wei, Z.; Xue, Y.; Xu, Z. An Active Deep Learning Approach for Minimally Supervised PolSAR Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2019**, *57*, 9378–9395. doi:10.1109/TGRS.2019.2926434.
33. Shrivastava, V.K.; Pradhan, M.K. Hyperspectral Remote Sensing Image Classification Using Active Learning. In *Studies in Computational Intelligence*; Springer, 2021; Vol. 907, pp. 133–152. doi:10.1007/978-3-030-50641-4_8.
34. Muslea, I.; Minton, S.; Knoblock, C.A. Active learning with multiple views. *Journal of Artificial Intelligence Research* **2006**, *27*, 203–233, [1110.1073]. doi:10.1613/jair.2005.
35. Di, W.; Crawford, M.M. View generation for multiview maximum disagreement based active learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* **2012**, *50*, 1942–1954. doi:10.1109/TGRS.2011.2168566.
36. Zhou, X.; Prasad, S.; Crawford, M. Wavelet domain multi-view active learning for hyperspectral image analysis. Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing. IEEE Computer Society, 2014, Vol. 2014-June. doi:10.1109/WHISPERS.2014.8077528.
37. Zhang, Z.; Pasolli, E.; Yang, H.L.; Crawford, M.M. Multimetric Active Learning for Classification of Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters* **2016**, *13*, 1007–1011. doi:10.1109/LGRS.2016.2560623.
38. Tuia, D.; Ratle, F.; Pacifici, F.; Kanevski, M.F.; Emery, W.J. Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing* **2009**, *47*, 2218–2232. doi:10.1109/TGRS.2008.2010404.
39. Copo, L.; Tuia, D.; Volpi, M.; Kanevski, M. Unbiased query-by-bagging active learning for VHR image classification. Image and Signal Processing for Remote Sensing XVI; Bruzzone, L., Ed. SPIE, 2010, Vol. 7830, p. 78300K. doi:10.1117/12.864861.
40. Demir, B.; Persello, C.; Bruzzone, L. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* **2011**, *49*, 1014–1031. doi:10.1109/TGRS.2010.2072929.
41. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Hyperspectral image segmentation using a new bayesian approach with active learning. *IEEE Transactions on Geoscience and Remote Sensing* **2011**, *49*, 3947–3960. doi:10.1109/TGRS.2011.2128330.
42. Liu, W.; Yang, J.; Li, P.; Han, Y.; Zhao, J.; Shi, H. A novel object-based supervised classification method with active learning and random forest for PolSAR imagery. *Remote Sensing* **2018**, *10*. doi:10.3390/rs10071092.
43. Luo, T.; Kramer, K.; Goldgof, D.; Hall, L.O.; Samson, S.; Remsen, A.; Hopkins, T. Learning to recognize plankton. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2003, Vol. 1, pp. 888–893. doi:10.1109/icsmc.2003.1243927.
44. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Transactions on Geoscience and Remote Sensing* **2013**, *51*, 844–856. doi:10.1109/TGRS.2012.2205263.
45. Ertekin, S.; Huang, J.; Giles, C.L. Active learning for class imbalance problem. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07; ACM Press: New York, New York, USA, 2007; pp. 823–824. doi:10.1145/1277741.1277927.
46. DeVries, T.; Taylor, G.W. Dataset augmentation in feature space. 5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings. International Conference on Learning Representations, ICLR, 2017, [1702.05538].
47. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357, [1106.1813]. doi:10.1613/jair.953.
48. Jozdani, S.E.; Johnson, B.A.; Chen, D. Comparing Deep Neural Networks, Ensemble Classifiers, and Support Vector Machine Algorithms for Object-Based Urban Land Use/Land Cover Classification. *Remote Sensing* **2019**, *11*, 1713. doi:10.3390/rs11141713.
49. Bogner, C.; Seo, B.; Rohner, D.; Reineking, B. Classification of rare land cover types: Distinguishing annual and perennial crops in an agricultural catchment in South Korea. *PLoS ONE* **2018**, *13*. doi:10.1371/journal.pone.0190476.
50. Douzas, G.; Bacao, F. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences* **2019**, *501*, 118–135. doi:10.1016/j.ins.2019.06.007.

51. Douzas, G.; Bacao, F.; Fonseca, J.; Khudinyan, M. Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric SMOTE algorithm. *Remote Sensing* **2019**, *11*, 3040. doi:10.3390/rs11243040.
52. Baumgardner, M.F.; Biehl, L.L.; Landgrebe, D.A. 220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3, 2015. doi:doi:/10.4231/R7RX991C.
53. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **1967**, *13*, 21–27. doi:10.1109/TIT.1967.1053964.
54. Nelder, J.A.; Wedderburn, R.W. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* **1972**, *135*, 370–384.
55. Ho, T.K. Random Decision Forests. Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1; IEEE Computer Society: USA, 1995; ICDAR '95, p. 278.
56. Olofsson, P.; Foody, G.M.; Stehman, S.V.; Woodcock, C.E. Making better use of accuracy data in land change studies: Estimating accuracy and area and quantifying uncertainty using stratified estimation. *Remote Sensing of Environment* **2013**, *129*, 122–131. doi:10.1016/j.rse.2012.10.031.
57. Pontius, R.G.; Millones, M. Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing* **2011**, *32*, 4407–4429. doi:10.1080/01431161.2011.552923.
58. Jeni, L.A.; Cohn, J.F.; De La Torre, F. Facing imbalanced data - Recommendations for the use of performance metrics. Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013, 2013, pp. 245–251. doi:10.1109/ACII.2013.47.
59. Kubat, M.; Matwin, S.; others. Addressing the curse of imbalanced training sets: one-sided selection. Icm. Citeseer, 1997, Vol. 97, pp. 179–186.
60. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* **2009**, *21*, 1263–1284, [arXiv:1011.1669v3]. doi:10.1109/TKDE.2008.239.
61. Reitmaier, T.; Sick, B. Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS. *Information Sciences* **2013**, *230*, 106–131.
62. Kagy, J.F.; Kayadelen, T.; Ma, J.; Rostamizadeh, A.; Strnadova, J. The Practical Challenges of Active Learning: Lessons Learned from Live Experimentation, 2019, [arXiv:cs.LG/1907.00038].
63. 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.; Duchesnay, É. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
64. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* **2017**, *18*, 1–5.
65. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* **2006**, *7*, 1–30.
66. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* **1945**, *1*, 80. doi:10.2307/3001968.