EffBaGAN: An Efficient Balancing GAN for Earth Observation in Data Scarcity Scenarios

Nicolás Vilela-Pérez, Dora B. Heras, *Member, IEEE*, and Francisco Argüello

***Abstract*—Generative adversarial networks (GANs) can be used as a data augmentation technique in scenarios with limited labeled information and class imbalances, common issues in remote sensing datasets. The EfficientNet architecture has gained attention for achieving high accuracy with moderate computational cost. This work introduces efficient balancing generative adversarial network (EffBaGAN), a generative network specifically designed for the classification of multispectral remote sensing images based on EfficientNet, addressing data scarcity and class imbalances while minimizing network complexity. EffBaGAN is built upon a balancing generative adversarial network (BAGAN) architecture, incorporating a custom EfficientNet-based discriminator and generator. In particular, for the discriminator we propose reduced EfficientNet discriminator, a reduced version of EfficientNet-B0 adapted to multispectral imagery. The generator, residual EfficientNet generator, includes a residual EfficientNet-based path, which enhances the quality of the generated synthetic samples. In addition, a superpixel-based sample extraction procedure is used to further reduce the computational cost of the method. Experiments were conducted on large, very high-resolution multispectral images of vegetation, demonstrating that EffBaGAN achieves higher accuracy than other advanced classification methods, including vision transformers and residual BAGAN, while maintaining a significantly lower computational cost. In fact, EffBaGAN is more than twice as fast as the residual BAGAN, making it an efficient solution for remote sensing image classification in data-scarce environments.**

***Index Terms*—Balancing generative adversarial network (BAGAN), classification, data augmentation, EfficientNet, multispectral, residual generator, transformer, vegetation.**

Nomenclature

AA	Average accuracy.
ACGAN	Auxiliary classifier generative adversarial network.
Adam	Adaptive moment estimation.
BAGAN	Balancing generative adversarial network.
CGAN	Conditional generative adversarial network.
CNN	Convolutional neural network.
ConViT	Convolutional-like vision transformer.
DWConv	Depthwise convolution.
DWSConv	Depthwise separable convolution.
EffBaGAN	Efficient balancing generative adversarial network.
ELU	Exponential linear unit.
FLOPs	Floating-point operations.
GAN	Generative adversarial network.
κ	Cohen's kappa coefficient.
LeakyReLU	Leaky rectified linear unit.
MBConv	Mobile inverted bottleneck.
OA	Overall accuracy.
PReLU	Parametric rectified linear unit.
PWConv	Pointwise convolution.
RedEffDis	Reduced EfficientNet discriminator.
ResBaGAN	Residual balancing generative adversarial network.
ResEffGen	Residual EfficientNet generator.
ResNet	Residual network.
SE	Squeeze-and-excitation.
SEEDS	Superpixels extracted via energy-driven sampling.
SiLU	Sigmoid linear unit.
SLIC	Simple linear iterative clustering.
TrSp	Training speedup.
TTPE	Training time per epoch.
ViT	Vision transformer.
WP	Waterpixels.

Received 5 August 2024; revised 22 October 2024; accepted 1 December 2024. Date of publication 4 December 2024; date of current version 3 January 2025. This work was supported in part by the Agencia Estatal de Investigación, Government of Spain, Contract PID2022–141623NB–I00, and Contract TED2021–130367B–I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR funds, in part by the Consellería de Cultura, Educación, Formación Profesional e Universidades, Xunta de Galicia, through the aid of accreditation of Galician Research Center 2024-2027 ED431G-2023/04, and accreditation of competitive Research Group ED431C 2022/16; all of them are cofounded by the European Regional Development Fund (ERDF). *(Corresponding author: Nicolás Vilela-Pérez.)*

Nicolás Vilela-Pérez and Dora B. Heras are with the Singular Research Center on Intelligent Technologies (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain (e-mail: nicolas.vilela.perez@usc.es; dora.blanco@usc.es).

Francisco Argüello is with the Department of Electronics and Computing, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain (e-mail: francisco.arguello@usc.es).

Digital Object Identifier 10.1109/JSTARS.2024.3510859

I. Introduction

REMOTE sensing multi- and hyperspectral images are valuable tools for classifying elements in a scene [1]. Applications of such classification include forest mapping, monitoring the evolution of invasive species in watersheds [2], and estimating crop yield. For example, crop yield estimation can be achieved by analyzing the electrical conductivity of the soil from various spectral bands of the images [3].

Various machine learning techniques have been employed for image classification tasks in remote sensing [4]. In recent years, the focus has shifted primarily toward deep learning-based techniques, which have proven more effective for multi-

and hyperspectral image classification compared to traditional approaches [5], [6]. Among these techniques, CNN [7], such as ResNet [8], are particularly prominent and are commonly chosen for this purpose [9], [10], [11]. The integration of skip connections in ResNets facilitates the training of deep networks by mitigating the vanishing gradient problem, which can prevent the learning process in very deep networks [12].

However, these techniques are characterized by a high computational cost [13]. Thus, in recent years, research has increasingly focused on developing fast and effective models that require fewer computational cost, thus minimizing training and testing times [14]. This is particularly relevant for applications on mobile devices with limited computing capacity, such as those utilizing MobileNet networks [15], or in scenarios requiring real-time or near real-time processing. One of the techniques that are applied to reduce the computational cost in deep learning are network compression techniques, such as pruning and quantization [16]. Pruning involves removing parameters that do not significantly impact the network's performance, thereby enhancing computational efficiency. Quantization, on the other hand, reduces the number of operations by lowering the precision of the data type used for weights or activations in the networks. For instance, Hernández et al. [17] demonstrated the application of quantization in models for IoT devices within a facial recognition system, highlighting how this technique allows model training on IoT devices by reducing computational cost without significantly compromising accuracy.

Another alternative to reduce the computational cost is to develop network models focused on this purpose, being able to highlight the EfficientNet architecture [18], proposed by Google in 2019. The main novelty introduced in this architecture is that different networks can be designed by uniformly varying the three dimensions that define them: depth, width, and resolution. This is done through a compound coefficient, which is defined as an exponent affecting the three dimensions mentioned above. It allows an adaptation to the computational resources of the device used and/or to the computational time requirements without affecting the resulting accuracy. EfficientNet achieves comparable and often superior image classification accuracies compared to other CNN-based architectures, such as ResNet. There are previous works in the literature where EfficientNets have been used for the classification of images in remote sensing, but not for the classification of the different elements present in watersheds.

One problem with deep learning networks is that they require a large amount of data to train properly [19]. Added to this is the fact that data scarcity and class imbalances are prevalent in multi- and hyperspectral remote sensing datasets, leading to biased classifiers that do not generalize well. Data augmentation helps alleviate this issue. Although the most common way to perform data augmentation is by applying simple transformations to existing samples [20], it has also recently been performed by using a type of neural architecture known as GAN [21]. These architectures allow the creation of completely new synthetic samples from an estimate of the reference data distribution [22], [23]. GAN architectures that generate synthetic images from random noise are known as noise-to-image, but some architectures generate synthetic images from input images, known as image-to-image. This work is focused on noise-to-image architectures. Different GAN designs are found in the literature [24].

- CGAN [25]: incorporates the corresponding information to synthesize a sample of a specific class on demand.
- ACGAN [26]: a natural extension of CGAN that allows the discriminator to assign each sample the most probable class, as well as distinguish between whether it is a real or synthetic sample.
- BAGAN [27]: when dealing with imbalanced datasets among the different classes present in them, GANs may not have enough information from the minority classes to train on the most relevant features of these classes. Furthermore, in these cases, GANs synthesize identical samples for each class, with no variety to enrich the dataset, sometimes even failing to synthesize noise. To solve these two problems, this architecture was developed. It introduces multiple modifications to stabilize the training of small and imbalanced datasets. These improvements are the introduction of an autoencoder [28] and the combination of both discriminator outputs (the predicted class and whether it is a real or synthetic sample).

In summary, the main problem with deep learning methods for classification of remote sensing images is their high computational cost. Thus, in this work, we have designed a proposal to maintain very good accuracy metrics in this type of classification while reducing the computational cost compared to other state-of-the-art methods.

This work presents an alternative to perform data augmentation by taking advantage of the computational efficiency of EfficientNet networks, proposing EffBaGAN as a result. EffBaGAN is a novel EfficientNet-based augmentation and classification method for multispectral remote sensing imaging for environmental monitoring. The EffBaGAN architecture includes a custom discriminator and generator. In particular, the computational cost reduction is achieved through the RedEffDis, our simplified proposal of EfficientNet. The proposed residual EfficientNet generator (ResEffGen) includes an EfficientNet-based residual path to allow the synthesis of higher-quality samples. It preserves the features of the initial latent vector while creating new ones and synthesizing these samples in two steps: one for the spatial expansion and one for the spectral information. The challenges of data scarcity and class imbalances are addressed through a data augmentation technique that combines traditional transformations with sample synthesis by the proposed EffBaGAN. Thus, the main contributions of this work are the following.

1) The proposed method incorporates an EfficientNet-based architecture and a sample extraction process based on superpixel segmentation and traditional augmentation techniques combined with sample synthesis by BAGAN. This achieves high accuracies due to the enrichment obtained through the sample extraction procedure and both data augmentation techniques, as well as reduced computational cost due to the sophisticated building blocks used in EfficientNet-based architectures: the MBConv blocks.

The combination of these technical innovations is crucial in scarce and imbalanced datasets, where the available labeled samples should be used to their full potential.

2) The discriminator RedEffDis allows achieving very good accuracy metrics while significantly reducing computational cost compared to other classifiers based on residual architectures such as ResNet. This discriminator is characterized by having a reduced number of blocks, adapting to the samples extracted from the datasets used. Each of these blocks extracts spatial features from each channel independently, thus avoiding interaction between them and being faster and more computationally efficient. It does this through a DWSConv, which has two steps: first the independent feature extraction mentioned above, and then the combination of these through a one-point convolution.
3) The generator ResEffGen includes an EfficientNet-based residual path, which allows for the combination of the most relevant features of the initial noise and the new ones generated through the main path. As a result, richer and more complete samples are generated through a two-step process, being one for the spatial resolution and the other for the spectral information. This proposed residual path first expands the initial noise to have the spatial resolution of the desired samples, and then synthesizes the values of the different bands of the synthesized sample. This is done through a transposed DWSConv, which performs the two steps of this operation but with the transposed behavior.

The rest of this article is organized as follows. Section II shows the EfficientNet architecture and those of GAN used for the proposed method. Then, Section III describes EffBaGAN in detail. Thereafter, Section IV presents the different experiments for evaluating EffBaGAN in terms of classification performance and computational efficiency. Next, Section V carries out the discussion of our proposal. Finally, Section VI concludes this article.

II. Related Work

As discussed above, the basis of this work is the EfficientNet to reduce computational cost and GANs to address the problem of scarcity and imbalance in remote sensing images.

A. EfficientNet

Its low computational cost makes this type of network a very interesting alternative for the classification task. The EfficientNet [18] is characterized by a uniform scaling of the three dimensions of this type of network through a compound coefficient, thus adapting to the computational or temporal constraints imposed while trying to obtain a scheme as accurately as possible, as already explained in the introduction.

The main component of the EfficientNet architecture is MBConv, first introduced in MobileNetV2 networks [29]. This block consists of the following two phases.

- Expansion through a PWConv: the block starts with a convolutional operation with 1×1 size filter called PWConv. This convolution aims to increase the input dimensionality, projecting it into a high-dimensional space to capture more complex representations. The channel expansion is defined by the expansion factor e.
- DWSConv: the result of the previous phase is then passed through a DWSConv, significantly reducing the computational cost compared to a traditional convolution by avoiding intensive interactions between channels. This operation is further divided into the following two parts.
  1) DWConv: applies a convolution to each input channel separately, thus identifying spatial patterns independently to each channel, preserving the number of channels but reducing the spatial resolution.
  2) Reduction through a PWConv: is the last operation in the block, and is applied to learn more complex representations by combining the spectral information from the output of the previous operation. In turn, this block reduces the number of output channels to adapt it to the processing of the subsequent stages.

It should be noted that, in EfficientNets, MBConv blocks additionally include an SE block [30], which consists of two phases: a first phase that obtains a representative value of each feature map/channel as a global summary of everything present in it, and a second phase that consists of learning through fully connected layers the weights that will be applied to each feature. The percentage of channels processed through this block is defined by the reduction rate se.

The same authors as EfficientNet have later proposed a second version of it: EfficientNet V2 [31]. The main difference between this network and the first version is the modification of the MBConv block architecture, becoming a Fused MBConv block, introduced in [32]. The Fused MBConv block differs from the standard by replacing the first two operations (PWConv and DWConv) by a standard convolution.

The MBConv block used in the EfficientNet architecture, as well as the Fused MBConv block used in the EfficientNet V2 architecture [31], are depicted in Fig. 1, where the differences between them can be seen.

By using the MBConv block, CNNs will achieve higher computational efficiency, as this block reduces the number of computations compared to traditional convolutions. Furthermore, the addition of the SE block allows CNNs to focus on the most relevant features of the input sample and attenuate those that are less relevant with almost no added computational cost [33]. On the other hand, the standard convolution applied in the Fused MBConv block is faster than PWConv + DWConv at runtime due to the exploitation of the spatial locality of the processed data.

As mentioned before, the main building block of EfficientNets, MBConv, was previously introduced in MobileNet networks, also with the same objective of reducing the computational cost. MobileNet-based networks demonstrated, on RGB remote sensing images, to obtain very competitive accuracies while the computational operations used by such models are lower than for other standard networks [34], [35]. Similarly, EfficientNet-based networks also obtain very competitive results

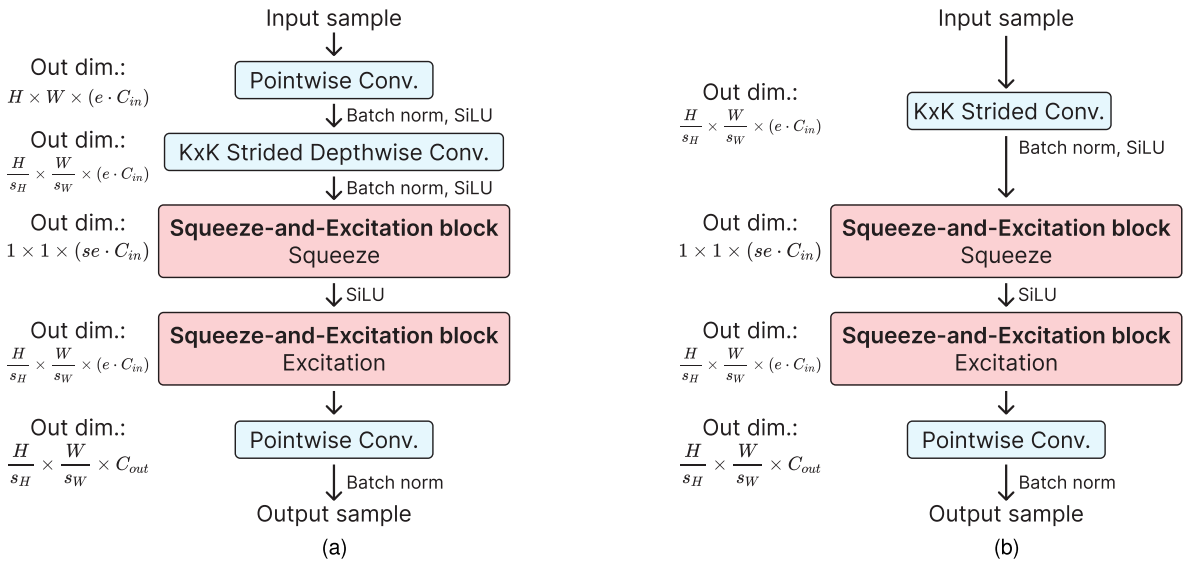

Fig. 1. Structure of (a) MBConv and (b) Fused MBConv blocks. The input sample has C_{in} channels with height H and width W. The accumulated output dimensions of each operation are shown on its left. The blocks MBConv and Fused MBConv apply expansion factor e on the first operation, stride $s_H \times s_W$ on the second operation in MBConv and on the first in Fused MBConv, and reduction rate $se \in [0, 1]$ on the first red operation. The number of output channels is C_{out}, selected in the last operation of both blocks.

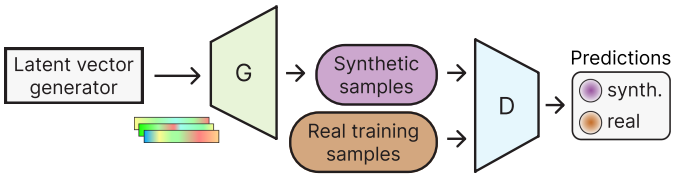

Fig. 2. Architecture of a GAN. It can be observed how the generated latent vectors pass through G to generate synthetic samples.

in terms of classification accuracy for remote sensing problems, also with a lower computational cost [36], [37], [38], [39].

B. Generative Adversarial Networks

We chose GANs as a data augmentation technique on scarce and imbalanced datasets. A GAN consists of two neural networks, the generator G, and the discriminator D, which compete with each other in an adversarial learning process. G creates synthetic data samples from a random latent input space, while D evaluates the authenticity of these samples, trying to distinguish them from the real ones. As the networks are trained together, G improves its ability to produce increasingly realistic samples, while D improves its ability to discern between real and synthetic samples. This process of competition and feedback allows GANs to generate synthetic samples that are indistinguishable from real ones, making them especially useful as a data augmentation technique, with great potential compared to traditional techniques [40]. The GAN architecture is depicted in Fig. 2.

Both G and D use convolutional neural architectures. While D remains a common architecture, G uses transposed convolutions as its main operation, working inversely. So then D needs to transform an input sample into a unidimensional feature vector of length Z, while G has as input a vector of length Z and will transform it into a sample whose sizes are those needed for D's input. This unidimensional vector of length Z is known as latent space, being Z the size of said space. This space is nothing more than the abstract representation of the characteristics and attributes of the samples synthesized by G.

BAGAN's features make it a very good choice for applying data augmentation technique on datasets with significant class imbalances, such as remote sensing datasets [27]. This architecture introduces various improvements to stabilize training, especially in situations of data scarcity and class imbalances. First, the parameters of the two networks are initialized by an autoencoder [28], thus initiating the training from a good starting point and subsequently learning how to represent the different classes in the latent space. In addition, thanks to the initialization of G with the autoencoder's decoder, the generator can learn an accurate class-conditioning in the latent space.

Various proposals for using GANs for data augmentation in the classification of images with EfficientNet have been presented in the bibliography. In [41], different methods to improve the classification of an EfficientNet-B0 network on a COVID-19 chest X-ray image set are proposed, one of them being a GAN-based augmentation one, which did not obtain the best results, but instead obtained the best results by balancing the dataset. Kwak and Kim [42] used a GAN-based augmentation method to further classify very high-resolution multispectral remote sensing images with an EfficientNet-based classifier, obtaining better results than the alternatives without augmentation. Regarding this method, it is worth mentioning that they used a CycleGAN [43], being this an image-to-image augmentation method, and not a noise-to-image one. Finally, Abady et al. [44] used GAN-based architectures to augment multispectral satellite images from different regions through season transfer with these

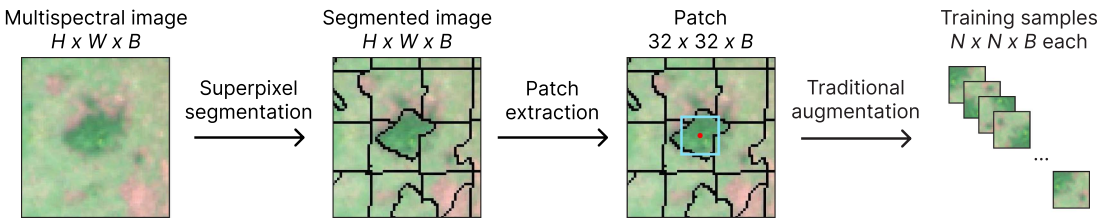

Fig. 3. Procedure for extracting training samples from the dataset in EffBaGAN. It starts with a segmentation in superpixels to subsequently obtain the patch of each superpixel and enrich the data through traditional augmentation, introducing the output of this procedure to the network architecture of Fig. 4. The augmentation is not performed on the validation and test sets. The multispectral image has height H, width W, and B bands.

architectures. Then, they performed detection and localization of these areas with EfficientNet-B4. The results show that in scenarios where the training and test sets were generated with the same GAN architecture, it is very accurate, but in those where this was not the case the results can be improved. All of the abovementioned proposals have been the starting point for our thinking that a method combining the low computational cost of EfficientNet and data augmentation through GAN would be promising.

Transversely to the previous proposals, it is also worth mentioning that Feng et al. [45] demonstrated, on hyperspectral remote sensing images, how a residual generator obtains better accuracies in GAN-based methods on noise-to-image than one that does not have these residual features. This last approach was the starting idea for our later proposal for a residual generator with EfficientNet features (which in turn has residual features), presented in more detail in Section III-C. Shortcuts in the residual generator allow the creation of synthetic samples preserving the most important features of the initial latent vector while generating new ones, and the EfficientNet features present in it allow the synthesis of the samples in two steps: one for the spatial expansion and one for the spectral information.

III. Proposed Method

EffBaGAN, the proposed method for remote sensing classification applied to data scarcity scenarios, is illustrated in Figs. 3 and 4. This section describes its different components as follows. First, the sample extraction procedure that combines superpixel segmentation and traditional augmentation techniques is introduced in Section III-A. Next, the designed network architecture is discussed globally and jointly in Section III-B. Finally, the design of the neural architectures in EffBaGAN is detailed in Section III-C.

A. Sample Extraction Via Superpixel Segmentation and Traditional Augmentation

Aiming to improve EffBaGAN for scarce and imbalanced datasets, a procedure for extracting samples from these datasets has been used [46]. This procedure combines two techniques widely used in remote sensing: superpixel segmentation and traditional augmentation techniques. Fig. 3 shows this procedure, step by step, the result of which are input samples to the network architecture described in Section III-B.

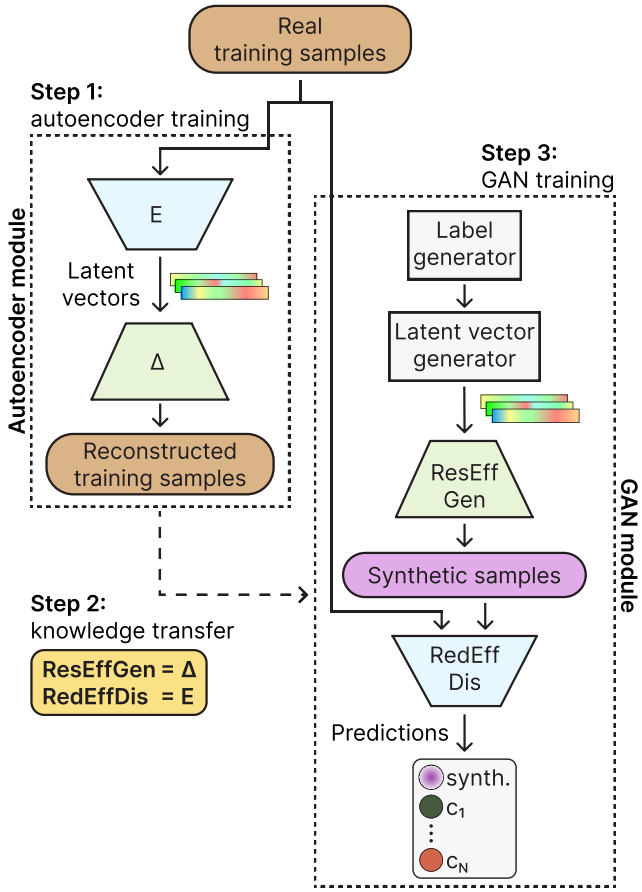

Fig. 4. Neural network architecture of EffBaGAN. Its main features are an EfficientNet-based BAGAN design that incorporates an EfficientNet-based residual generator (ResEffGen) and an EfficientNet-based classifier (RedEffDis). The input real training samples are obtained using the extraction procedure presented in Section III-A.

Superpixel segmentation groups contiguous pixels with similar characteristics, resulting in a homogeneous region. These regions do not have to be of a specific and/or regular shape, but have to be adapted to the image to group pixels with similar characteristics.

Since these are large and very high-resolution images, a representative patch of the superpixel will be selected for classification to classify the central pixel of it, and then the class resulting from this classification will be propagated to the whole superpixel. In this way, the number of samples to be processed by the classification scheme is significantly reduced, making

the computational cost of the scheme much lower than if it were done at the level of individual pixels. This representative patch is obtained from the already segmented image, determining the minimum quadrilateral that contains the superpixel within it, and setting the center point as the superpixel's central pixel. Once we have this central pixel, we establish a search range of size $N \times N$ pixels centered on this central pixel, forming the patch of this superpixel. It is worth mentioning that the class associated to the superpixel will be the class of the aforementioned central pixel. The patches have a spatial resolution of 32×32.

To perform the superpixel segmentation, different algorithms in the literature, such as SEEDS [47] or SLIC [48], could be used. In this work, WP [49] is used to facilitate comparison with other classification techniques already published as well as for its low computational time and good performance, providing homogeneous, compact regions with high adherence to the edges [2].

All extracted patches are further subjected to a traditional augmentation technique, which consists of, first, a random rotation of 0°, 90°, 180°, or 270°, and then a possible horizontal and/or vertical flip, with a 50% probability of occurrence each.

B. Network Architecture

The EffBaGAN is an EfficientNet-based BAGAN architecture. It has been designed to include a new residual generator ResEffGen with EfficientNet features, allowing the synthesis of higher quality samples. The discriminator RedEffDis, also based on EfficientNet, aims at reducing the computational cost. Since the high-resolution datasets for the classification applied to forest mapping have very limited labeled samples, an autoencoder module [28] is inserted. It stabilizes the subsequent training of the GAN, thus achieving a more realistic sample synthesis even with scarce and imbalanced datasets.

The EffBaGAN training process consists of three steps, as shown in Fig. 4.

1) First, the autoencoder is trained with all the available samples without considering their label, to produce an initial understanding of the data distribution. To guide the unsupervised training, mean squared error loss is used, with the autoencoder aiming to minimize it as much as possible. This autoencoder module consists of an encoder and a decoder, which must have the same topology as the discriminator (RedEffDis) and the generator (ResEffGen), respectively, to allow knowledge transfer through the sharing of weights of the autoencoder with the uninitialized GAN.
2) Once the autoencoder has been trained, all the learned parameters are transferred to the GAN, thus making it acquire the knowledge of the autoencoder. The objective of the autoencoder is to learn how to compress and reconstruct the samples as accurately as possible.
3) Finally, the last step is the training of the GAN module itself, where due to the transfer of the previous step, it starts from a more stable point than if the parameters were randomly initialized. To guide the training of this step, categorical cross-entropy loss is used in both the ResEffGen and the RedEffDis.

   It should be noted that the total loss of the discriminator results from the sum of the categorical cross-entropy loss of the real samples and that of the synthetic samples. The first part results from the calculation of the loss function between the predicted class probabilities of the real samples and their labels. On the other hand, the second part results from the calculation of the loss function between the predicted class probabilities of the synthetic samples and their labels.

Once the EffBaGAN networks have been trained, the synthetic output of the discriminator is disabled, so that the final classifier only makes predictions about the classes in the dataset.

C. Network Topologies

As mentioned in Section III-B, the autoencoder and the GAN modules share topology. First, the topology of the EfficientNet-based discriminator RedEffDis, also applicable to the encoder, will be discussed. Subsequently, the topology of the EfficientNet-based residual generator ResEffGen, also applicable to the decoder, will be discussed.

It should be noted that the topologies have been selected after a study that will be detailed in Section IV-B1 for the discriminator, and in Section IV-B2 for the generator.

For the discriminator topology, we have proposed the RedEffDis architecture, intending to obtain a better adaptation to the BAGAN and to the multispectral remote sensing datasets. We can highlight the following features.

1) RedEffDis is based on EfficientNet-B0: EfficientNet has multiple versions, ranging from EfficientNet-B0 to EfficientNet-B7 uniformly increasing the three dimensions of the network. These three dimensions are depth, width and resolution, so that the EfficientNet-B7 will have more layers, and a larger number of channels and sample sizes than the EfficientNet-B0. All of these versions are designed for much larger datasets than those used in this work (such as ImageNet [50]), so we have focused on the simplest architecture of this type of network to avoid possible overfitting to the training data, which is EfficientNet-B0.
2) RedEffDis avoids the excessive reduction of spatial resolution provided by EfficientNet-B0 by eliminating the first 3 stages with MBConv of this network. In other case, the last stages of the network will not be able to extract the spatial features correctly even having the necessary operations to do so, since these stages have an input spatial resolution of 1×1. The proposed elimination decreases the reduction of spatial resolution across the network, so that the final stages are able to extract spatial features by having an input spatial resolution greater than 1×1.
3) Finally, RedEffDis includes several modifications focused on MBConv block stages. It should be noted that the same proportion among the different stages of the network architecture than for EfficientNet-B0 is respected. The modifications focused on the following aspects.

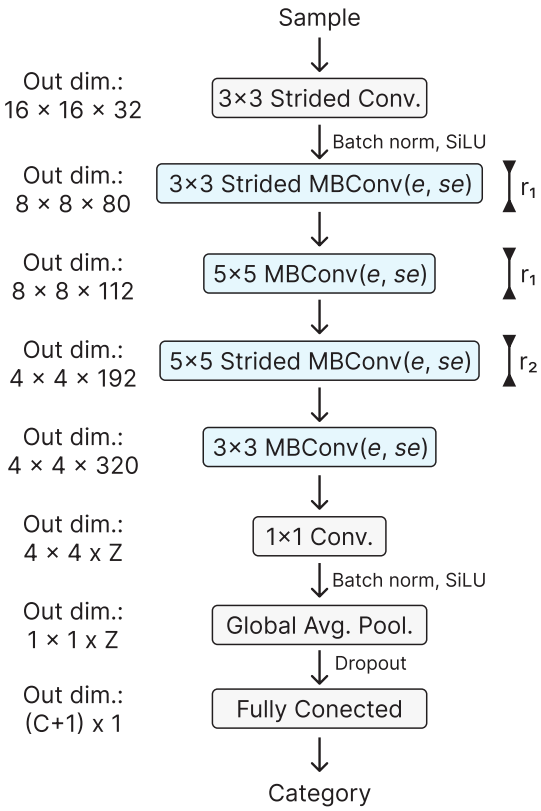

Fig. 5. Diagram of the RedEffDis for EffBaGAN. The output dimensions correspond to an input sample with dimensions $H \times W \times B = 32 \times 32 \times B$, being B the number of spectral bands. The values of the modifiable parameters (r_1, r_2, e, se) will be those of the corresponding configuration.

- Number of repetitions for each MBConv stage, r_1 and r_2: these values are selected complying with the constraint $r_1 \leq r_2$. This is shown in Fig. 5, where each stage with MBConv blocks is repeated the number of times indicated on its right.
- Expansion factor e: the MBConv block of each stage performs an expansion of the number of input channels in the first operation, as shown in Fig. 1. The number of channels resulting from the expansion is the number of input channels C_{in} multiplied by e. The values of these factors are chosen to be the same for all stages with MBConv, thus respecting the proportions of EfficientNet-B0.
- Reduction rate se: the MBConv block of each stage includes an SE module that performs the SE operations. The reduction rate se indicates the percentage of channels that are selected for these operations. It is a decimal number in the range [0,1], and, similar to the expansion factor e, se is the same for all stages with MBConv.

It is worth mentioning that the modifications proposed for the MBConv block can also be applied to the Fused MBConv one. Following the approach of EfficientNet V2 [31], two additional versions were tested: one with the first half being Fused MBConv blocks, and another with all of them. We call these versions RedEffDis-FusedHalf and RedEffDis-FusedAll, respectively.

TABLE I
DETAILS OF THE LAYERS IN THE REDEFFDIS FOR EFFBAGAN

#	Component	# of layers	Output dimensions	Filter size	Stride
1	2D Convolution	1	$16 \times 16 \times 32$	3×3	2×2
2	MBConv$(e,\ se)$	r_1	$8 \times 8 \times 80$	3×3	2×2
3	MBConv$(e,\ se)$	r_1	$8 \times 8 \times 112$	5×5	1×1
4	MBConv$(e,\ se)$	r_2	$4 \times 4 \times 192$	5×5	2×2
5	MBConv$(e,\ se)$	1	$4 \times 4 \times 320$	3×3	1×1
	2D Convolution		$4 \times 4 \times Z$	1×1	1×1
6	Global Avg. Pooling	1	$1 \times 1 \times Z$	-	-
	Fully Connected		$(C+1) \times 1$	-	-

The output dimensions correspond to an input sample with dimensions $H \times W \times B = 32 \times 32 \times B$, being B the number of spectral bands. The values of the modifiable parameters (r_1, r_2, e, se) will be those of the corresponding configuration.

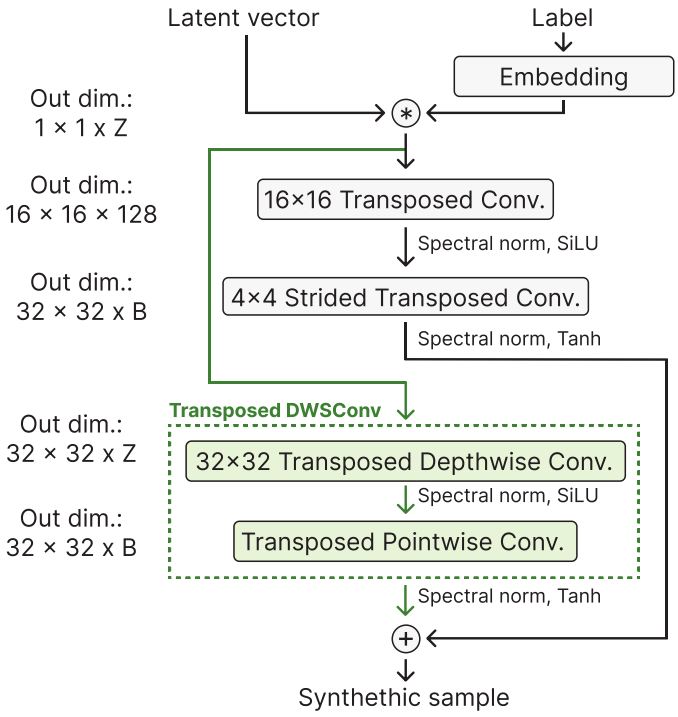

Fig. 6. Diagram of the ResEffGen for EffBaGAN. The size of the input latent vector is Z. The output dimensions of the generator are $32 \times 32 \times B$, being B the number of spectral bands. These output dimensions must be the same as the real samples and as the input dimensions of the RedEffDis.

The proposed RedEffDis topology is depicted in Fig. 5 and detailed in Table I. As a result of the discriminator study in terms of accuracy and training time, we have proposed two different configurations of EffBaGAN depending on the values of (r_1, r_2, e, se) defined for RedEffDis. EffBaGAN-Base is defined by values $(3, 4, 6, 0.5)$, which means that it has $(2 \cdot 3 + 4 + 1) = 11$ MBConv blocks with expansion factor $e = 6$ and reduction rate se $= 0.5$. EffBaGAN-Small is defined by $(1, 1, 6, 0)$, thus having only $(2 \cdot 1 + 1 + 1) = 4$ MBConv blocks with expansion factor $e = 6$ and reduction rate $se = 0$. These two configurations EffBaGAN-Base and EffBaGAN-Small will be evaluated in Section IV.

Regarding the generator ResEffGen, it is depicted in Fig. 6 and detailed in Table II. We have proposed to add an EfficientNet-based residual path next to a main path that includes two transposed convolutions. The EfficientNet-based residual path performs the transposition of the DWSConv operation of the EfficientNet (see the two transposed convolutions in the green path,

TABLE II
DETAILS OF THE LAYERS IN THE RESEFFGEN FOR EFFBAGAN

#		Component	Output dimensions	Filter size	Stride
1		Embedding	$Z \times 1$	C	-
M.1	Main	2D Transposed Conv.	$16 \times 16 \times 128$	16×16	1×1
M.2	path	2D Transposed Conv.	$32 \times 32 \times B$	4×4	2×2
R.1	Residual	2D Transposed DWConv	$32 \times 32 \times Z$	32×32	1×1
R.2	path	2D Transposed PWConv	$32 \times 32 \times B$	1×1	1×1

The size of the input latent vector is Z. The output dimensions of the generator are $32 \times 32 \times B$, being B the number of spectral bands. These output dimensions must be the same as the real samples and as the input dimensions of the RedEffDis.

the residual one, in Fig. 6). The transposed convolutions of the main path increase the spatial resolution of the synthetic sample by two steps to finally obtain a sample of the desired spatial and spectral resolution. For its part, the residual path consists of a transposed DWSConv which, as previously explained, is formed by a DWConv followed by a PWConv, and in this case we have adapted them to have this transposition behavior.

This combination of the RedEffDis and the ResEffGen is effective for the following two main reasons.

- The EfficientNet-based discriminator trains faster thanks to the optimized MBConv block. Its design includes the DWSConv operation, which has a lower computational cost than a standard convolution as it avoids interaction between channels.
- The EfficientNet-based residual generator produces an enhanced-accuracy network thanks to the increased quality of the synthetic samples. This is achieved by complementing the information contained in the main path with the synthetic samples produced by the residual path with transposed DWConv and transposed PWConv.

IV. EXPERIMENTS

In this section, EffBaGAN is evaluated in terms of classification performance to assess its overall effectiveness. EffBaGAN's capabilities are compared with other classification methods, such as a CNN, a ResNet, a BAGAN-based method, a ViT, and other methods designed for low computational cost such as MobileNet or EfficientNet-B0.

The section is organized as follows. First, the datasets, metrics, and execution environment used for the experiments are outlined in Section IV-A. The optimizer, weight initialization, and the hyperparameter optimization procedure, are also detailed here. Then, the EffBaGAN's discriminator and generator topology selection, as well as the experimental results, are presented and discussed in Section IV-B.

A. Experimental Setup

1) Datasets: Eight large, very high-spatial resolution multispectral images of natural regions with dense vegetation, which were used in [2], were considered. These images have been captured in 2018, 2019, and 2020 flying an autonomous aerial vehicle at 120 m altitude over several river basins in Galicia (Spain), resulting in a spatial resolution of 10 cm/px. The vehicle carried a MicaSense RedEdge-MX multispectral camera, capturing five bands corresponding to wavelengths of 475 nm (blue), 560 nm (green), 668 nm (red), 717 nm (red-edge), and 840 nm (near-infrared). Table III details the specific locations and dimensions of the scenes.

Fig. 7 shows the composite color images corresponding to the eight river images, together with their reference information. Table IV lists the ten classes identifiable in the reference information, detailing the number of samples in each dataset. These classes range from native vegetation to human-made structures such as roads or buildings. It is important to note the data scarcity and large imbalances between classes in all datasets. These imbalances introduce a bias toward the majority classes, which could prevent a balanced classification accuracy across classes.

All datasets were segmented into superpixels using the WP algorithm choosing an average size of 400 px/superpixel, with a minimum of 100 px/superpixel, and utilizing a compactness factor of 0.5 points, following the approach of [2]. The extracted patches have a spatial dimension of $N \times N = 32 \times 32$ px. In addition, all data were normalized to the range $[-1, 1]$. For all datasets, a training set of 15% of the samples and a validation set of 5% of the samples will be used to monitor training progress to identify potential problems such as overfitting. This leaves the remaining 80% of the samples for the test set.

2) Metrics: The classification performance of EffBaGAN is determined by class prediction of each labeled sample and comparison of results with reference information. For this, three standard pixel-level metrics in remote sensing classification [51] will be used, excluding for their calculation only the central pixels of the superpixels of the training set: OA, AA, and Cohen's kappa coefficient (κ).

The computational cost of the EffBaGAN will be evaluated through different metrics. Execution time is evaluated in terms of TTPE, in seconds. Based on this metric, the TrSp evaluates how much faster a classification method trains per epoch compared to a pre-established one. Three metrics related to the size and complexity of the network will also be obtained: network size in memory (in MiB), number of trainable parameters (floating-point variables) of the network, and number of operations, in particular FLOPs, of the network for one forward pass with batch size one.

3) Execution Environment: All experiments have been performed on a computing cluster. The used node has two AMD EPYC 7543 CPUs with 32 cores each, 256 GB of RAM, and two NVIDIA Ampere A100 GPUs with 40 GB of VRAM each, but only one core and one GPU have been used in the experiments carried out. Regarding the software, the code has been executed within a Conda environment, with Python 3.8.13, and CUDA 11.3 with cuDNN 8.3.2. The programming language to be used will be Python, on which the following main packages should be highlighted: PyTorch 1.12.0, for the creation, training, and testing of the networks; NumPy 1.24.3, for the manipulation of the datasets; scikit-learn 1.3.2, for the preprocessing of the datasets and obtaining κ; fvcore 0.1.5.post20221221, for obtaining the FLOPs; torchinfo 1.7.2, for obtaining the detailed information on the architecture of the networks; and Guild AI 0.8.1, for the registration of the experiments.

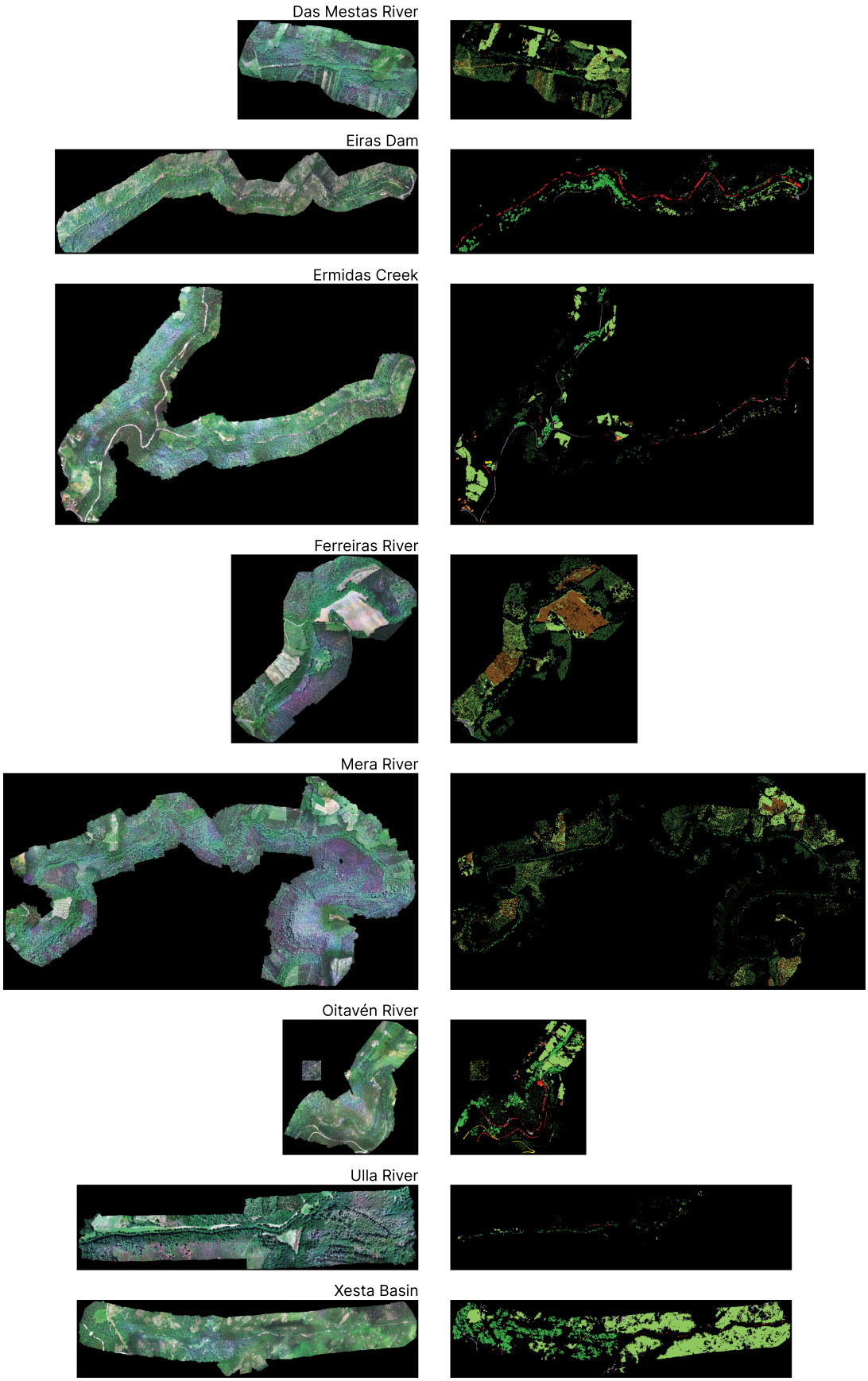

Fig. 7. Composite color images (left) and reference information (right) of the datasets used in this work. All representations follow the same size scale. The class corresponding to each color in the reference information is described in Table IV, while black means that there is no reference information about those pixels.

TABLE III
DESCRIPTIONS OF THE DATASETS USED IN THIS WORK

Dataset	Location	Size	Data pixels	Labeled pixels	Labeled superpixels
Das Mestas River	43°38′30.24″ N 7°58′44.08″ W	9 040 × 4 915 px (920 × 510 m^2)	27 169 125	10 341 813	39 229
Eiras Dam	42°20′45.26″ N 8°30′10.81″ W	18 221 × 5 176 px (2 260 × 660 m^2)	38 352 791	4 023 134	15 435
Ermidas Creek	42°22′48.43″ N 8°24′53.36″ W	18 972 × 11 924 px (2 190 × 1 390 m^2)	65 557 313	6 914 486	23 188
Ferreiras River	43°32′46.96″ N 7°57′16.66″ W	9 219 × 9 335 px (740 × 750 m^2)	40 193 098	15 019 880	54 565
Mera River	43°34′31.15″ N 7°52′34.81″ W	22 116 × 10 718 px (2 770 × 1 370 m^2)	99 244 153	15 588 216	59 987
Oitavén River	42°22′15.48″ N 8°25′47.07″ W	6 722 × 6 689 px (760 × 760 m^2)	22 041 591	6 067 179	21 648
Ulla River	42°49′14.32″ N 7°54′5.29″ W	16 555 × 4 220 px (1 420 × 380 m^2)	46 011 758	355 516	1 461
Xesta Basin	42°23′34.95″ N 8°21′21.23″ W	17 202 × 3 848 px (1 945 × 435 m^2)	39 998 952	17 037 589	61 393

The number of labeled superpixels is obtained through WP segmentation.

TABLE IV
ABUNDANCE OF CLASSES IN THE DATASETS

#	Color	Class	Das Mestas River	Eiras Dam	Ermidas Creek	Ferreiras River	Mera River	Oitavén River	Ulla River	Xesta Basin
1		Water	0	1 929	447	0	0	912	0	210
2		Bare soil	4 583	424	472	18 404	9 856	484	210	182
3		Rocks	0	726	747	0	0	349	0	2 170
4		Asphalt	0	233	2 418	510	241	128	14	467
5		Concrete	0	111	95	0	5	458	0	0
6		Tiles	0	29	453	0	8	294	0	0
7		Meadows	17 099	2 989	10 413	14 087	22 777	6 896	243	42 821
8		Native trees	1 414	8 604	3 154	2 059	5 129	7 573	913	15 543
9		Pines	0	351	800	11	0	955	81	0
10		Eucalyptus	16 133	39	4 189	19 494	21 971	3 599	0	0
		Total samples:	39 229	15 435	23 188	54 565	59 987	21 648	1 461	61 393

The number of samples is in superpixels, obtained through WP segmentation.

4) On Training Neural Networks: Adam [52] was used as the network optimization algorithm, which has proven to be suitable for problems with large amounts of data and large number of parameters. The parameters used for this optimizer were set to $\beta_1 = 0.5$ and $\beta_2 = 0.999$, as usual in adversarial architectures. All the network parameters that could be were initialized using Xavier's (or Glorot's) initialization [53]. The number of training epochs, the losses, and the validation accuracy of EffBaGAN have been monitored throughout the training epochs. The conclusion is that 600 epochs are sufficient to reach the maximum performance of the full classification method.

5) Hyperparameter Optimization: In Section III, certain configuration features of EffBaGAN were left unaddressed, such as the size of the latent vectors or the activation function, among others. To determine them, a hyperparameter optimization procedure has been carried out evaluating the performance of EffBaGAN in several scenarios. Specifically, the following configurations were tested.

- *Batch size ($batch_{size}$):* larger batches accelerate training but may worsen generalization abilities. The batch size has been limited to small values to prioritize classification accuracy over speedup, testing $batch_{size} = \{32, 64, 128\}$.
- *Learning rate (α):* all the values in the range [0.0005, 0.0030] with a step of 0.0005 were tried.
- *Size of latent vectors (Z):* guided by the dimensionalities of the convolutional layers in EffBaGAN, the values $Z = \{32, 64, 128\}$ were tried.
- *Activation function:* ELU [54], LeakyReLU [55], PReLU [56], and SiLU [57] were tested.
- *Dropout probability ($p_{dropout}$):* to cover sufficient possibilities, the values $p_{dropout} = \{0.05, 0.20, 0.35, 0.50\}$ were tested.

All the possible combinations of these hyperparameters result in 864 configurations to evaluate. Since the computational cost of this optimization is very high and the initial behavior of the network is a good indicator of future performance, the number of training epochs of each configuration was limited to 20

TABLE V
RECORDED METRICS FOR EIRAS DAM DATASET USING DIFFERENT CONFIGURATIONS OF THE EFFBAGAN DISCRIMINATOR

Discriminator version	(r_1, r_2, e, se)	**OA** (%)	**AA** (%)	κ (%)	**TTPE** (s)
EfficientNet-B0	Not applicable	95.30 ± 1.32	64.97 ± 9.76	91.89 ± 2.30	21.88 ± 3.43
RedEffDis	$(2, 4, 6, 0.25)$	97.73 ± 0.66	89.26 ± 4.30	96.13 ± 1.12	16.18 ± 2.40
RedEffDis	$(3, 4, 6, 0)$	98.15 ± 0.29	90.35 ± 3.23	96.87 ± 0.48	13.94 ± 1.34
RedEffDis	$(3, 4, 6, 0.5)$	98.23 ± 0.28	88.15 ± 3.15	97.00 ± 0.47	19.21 ± 1.20
RedEffDis	$(3, 4, 6, 0.75)$	97.87 ± 0.78	90.68 ± 3.70	96.40 ± 1.29	19.52 ± 0.98
RedEffDis	$(1, 1, 6, 0)$	97.78 ± 0.64	86.67 ± 6.52	96.22 ± 1.10	8.92 ± 0.75
RedEffDis-FusedHalf	$(1, 1, 5, 0)$	96.87 ± 1.36	86.72 ± 4.08	94.71 ± 2.25	8.55 ± 0.65
RedEffDis-FusedAll	$(1, 1, 5, 0)$	96.76 ± 0.49	83.42 ± 3.64	94.47 ± 0.84	8.29 ± 0.77

The colored cells are the first, second and third best values for each metric.

during this optimization. These configurations have been tested on the Oitavén River dataset with EffBaGAN-Small repeating each one 5 times for consistency. Once all these tests were run, the configuration with the highest classification accuracies (determined by OA, AA, and κ) was chosen.

Following this optimization process, the hyperparameter choices that maximized the performance of the EffBaGAN were a batch size of 64 samples, a learning rate of about 0.0005, latent vectors of size $Z = 32$, the SiLU activation function, and dropout probability of 5%.

B. Experimental Results

1) Selection of the Discriminator Topology: Before going on to explain the selection of the discriminator (this section) and the generator (next section), it should be noted that all modifiable parameters other than those under study have been kept constant. In addition, each discriminator or generator configuration has been run a total of 5 times for consistency. Thus, the values shown for the different metrics for each configuration are the average and the deviation after the 5 repetitions of each.

For selecting the adequate configurations of the discriminator of the EffBaGAN called RedEffDis, a study of all possible configurations was carried out. In this study, the EfficientNet-B0, as well as several parameter configurations of the RedEffDis, RedEffDis-FusedHalf, and RedEffDis-FusedAll proposed in Section III-C were considered. A total of 19 different discriminator configurations were tested over four of the eight datasets: Eiras Dam, Ermidas Creek, Oitavén River, and Ulla River.

Table V shows the average values of the experimental metrics for the nine most representative EffBaGAN discriminator configurations on the Eiras Dam dataset in terms of high accuracy and low execution time. Several RedEffDis configurations varying the number of repetitions of the different stages, the expansion factor e, and the reduction rate se both separately and combined, one configuration of RedEffDis-FusedHalf, one of RedEffDis-FusedAll, and EfficientNet-B0 are shown in the Table. It is worth commenting that these results follow the same trend in all the tested datasets.

The final configurations of the discriminator selected are those considered to be the best combinations of accuracy and training time metrics. The discriminator version with the best tradeoff between accuracy and training time is the RedEffDis with parameters $(r_1, r_2, e, se) = (1, 1, 6, 0)$, since it has very competitive accuracy metrics and one of the shortest TTPE (note the wide range of this metric among the different versions). However, the RedEffDis with $(r_1, r_2, e, se) = (3, 4, 6, 0.5)$ obtains the highest accuracies. These two configurations are shown to be the most promising for the discriminator topology since one of them obtains the highest accuracy metrics while the other has a reduced computational cost. So, two configurations of EffBaGAN have been proposed.

- EffBaGAN-Base: it has the discriminator topology with the higher accuracy metrics. This topology has a total of 11 MBConv blocks with expansion factor $e = 6$ and reduction rate $se = 0.5$. This configuration corresponds to the fourth row of Table V: RedEffDis being its tuple of modifiable parameters $(r_1, r_2, e, se) = (3, 4, 6, 0.5)$.
- EffBaGAN-Small: it has one of the discriminator topologies with the highest accuracy metrics but a reduced TTPE. This topology has a total of 4 MBConv blocks with expansion factor $e = 6$ and reduction rate $se = 0$. This configuration corresponds to the sixth row of Table V: RedEffDis, being its tuple of modifiable parameters $(r_1, r_2, e, se) = (1, 1, 6, 0)$.

2) Selection of the Generator Topology: On the other hand, the EffBaGAN generator is characterized by incorporating a residual path. We tested various generator topologies with a varying number of transposed convolutions from 2 to 5, obtaining an optimal number of 2. Then, different residual path topologies were tested.

- Transposed convolution: a simple transposed convolution was tested as the residual operation.
- Upsampling with convolution: to obtain a synthetic sample with a specific spatial resolution and number of bands, we first proposed to perform a bilinear upsampling to obtain the desired spatial resolution, and then with a 2D convolution to adapt the number of channels to the number of bands desired for the synthetic sample.
- Transposed DWSConv: following the philosophy of the main EfficientNet block, we have proposed the creation of a transposed DWSConv, consisting first of a transposed DWConv, which obtains the desired spatial resolution, and then a transposed PWConv, which obtains the desired number of bands.

The results of the different generator configurations are shown in Table VI. The values correspond to the average after executing them on the eight datasets. The selected configuration, which

TABLE VI
AVERAGE RECORDED METRICS FOR ALL THE DATASETS USING DIFFERENT CONFIGURATIONS OF THE EFFBAGAN GENERATOR

Main path operations	Residual path single operation	OA (%)	AA (%)	κ (%)	TTPE (s)
2 transposed convolution	Transposed convolution	95.53 ± 2.73	85.50 ± 8.96	92.51 ± 4.16	10.06 ± 5.65
2 transposed convolution	Upsampling with convolution	95.80 ± 2.74	86.33 ± 9.33	93.00 ± 4.17	12.01 ± 6.66
2 transposed convolution	Transposed DWSConv	95.90 ± 2.81	86.51 ± 9.54	93.17 ± 4.34	13.28 ± 7.82
3 transposed convolution	Transposed convolution	95.89 ± 2.69	86.36 ± 9.75	93.07 ± 4.13	9.99 ± 5.27
3 transposed convolution	Upsampling with convolution	95.61 ± 2.57	85.91 ± 9.01	92.62 ± 3.88	12.65 ± 7.37
3 transposed convolution	Transposed DWSConv	95.67 ± 2.79	85.96 ± 9.52	92.81 ± 4.27	13.33 ± 8.02
4 transposed convolution	Transposed DWSConv	95.53 ± 3.01	85.65 ± 9.43	92.58 ± 4.54	14.69 ± 9.43
5 transposed convolution	Transposed DWSConv	95.70 ± 2.79	85.91 ± 9.41	92.84 ± 4.26	14.08 ± 8.64

The colored cells are the first, second and third best values for each metric.

will be called ResEffGen, is that offering the best accuracy metrics. ResEffGen consists of 2 transposed convolutions in the main path and a transposed DWSConv, i.e., based on EfficientNet, in the residual path. It should be noted that the generator is the most critical part of the method design, so a thorough analysis has been carried out on all available datasets. In the case of the discriminator, the design of which is less sensitive to parameter selection, only a representative subset of the images has been used, as mentioned in Section IV-B1.

3) Overfitting Analysis: To better understand the selection of the proposed reduced version of EfficientNet-B0 (RedEffDis) as the discriminator of our method, we will discuss the overfitting issues encountered with larger models from the EfficientNet family. As previously mentioned when introducing RedEffDis in Section III-C, the EfficientNet family was originally designed for much larger datasets than those used in this work. Therefore, the simplest model, EfficientNet-B0, with further reductions has been chosen to suit our needs. Fig. 8 illustrates the training and validation losses of an experiment on the Eiras Dam dataset for three networks: EfficientNet-B0, EfficientNet-B3, and EfficientNet-B7. The figure highlights that as the network size increases, overfitting becomes a more significant issue, as evidenced by the widening oscillations and instability of the validation losses for the larger models, such as EfficientNet-B3 and even more so, EfficientNet-B7. These big and sudden oscillations are an indicator that these networks are not generalizing well, but are learning specific features from the training data. The reduced version of EfficientNet-B0 RedEffDis was, therefore, proposed to mitigate this overfitting while still maintaining reasonable performance on smaller remote sensing datasets.

4) Evolution of Metrics During Hyperparameter Optimization and Training: To understand how the learning rate value influences model training, the relationship between the accuracy metrics and the discriminator and generator losses during hyperparameter optimization are analyzed with respect to the different learning rates tested, as depicted in Fig. 9(a) and (b). Fig. 9(a) justifies the selection of 0.0005 as the learning rate for the hyperparameter optimization. As the learning rate increases, we observe a decrease in accuracy metrics. In Fig. 9(b), we can see that increasing the learning rate results in a higher discriminator loss and a lower generator loss, which is detrimental in the former case but beneficial in the latter. However, the chosen learning rate of 0.0005 provides a good balance between these competing factors, optimizing the overall learning performance.

On the other hand, to ensure correct and sustained training, the evolution of various training metrics for both discriminator and generator is also analyzed in an experiment of EffBaGAN-Base on the Ermidas Creek dataset, as depicted in Fig. 10(a)–(c). Fig. 10(a) shows a consistent generalization of the classifier, as the variation of the discriminator validation accuracy between epochs decreases markedly between the first and last training epochs, reaching the minimum variation from about 450 epochs onwards. Fig. 10(b) and (c) shows how the discriminator and generator losses decrease together as the training epochs progress, reaching their minimum values in the final epochs. From these two observations, it can be seen that running 600 training epochs allows the EffBaGAN to optimize its performance to the experimental datasets used in this work.

5) Overall Classification Performance: Table VII presents the resulting OA, AA, κ, TTPE, and TrSp metrics for EffBaGAN and other classification methods on all the experimental datasets. The classification methods compared include a CNN, implemented in [58], a ResNet, proposed in [8], and a BAGAN-based approach called ResBaGAN, proposed in [46]. They also include MobileNet and EfficientNet approaches, proposed in [15] and [18], respectively, as baselines for efficient deep learning methods, and a ViT, ConViT, and Mobile-Former approaches, proposed in [59], [60], and [61], respectively, as state-of-the-art references for image classification and previously used in remote sensing [62]. Each classification method has been run 5 times for consistency.

First, we can observe in Table VII, how the two classification methods considered in the literature as especially efficient (MobileNet and EfficientNet-B0) do not achieve as high accuracy metrics values as the simpler methods CNN and ResNet. Furthermore, the two efficient methods have similar or even higher TTPE than the simpler ones. The reason is that these two efficient methods are designed for large datasets, being as a consequence deeper, while the simpler methods consist of very shallow networks. This supports our motivation of proposing a method with a discriminator based on a reduced version of EfficientNet-B0 to achieve high classification accuracy at low computational cost in classification problems with data scarcity.

Analyzing the results shown in Table VII by other state-of-the-art techniques based on transformers such as ViT, ConViT, and Mobile-Former even been faster in average (TTPE values) they obtain OA average values around 1% lower than EffBaGAN. ResBaGAN also presents similar OA values to both ViT and

TABLE VII
RECORDED METRICS FOR ALL THE DATASETS USING ALL THE CLASSIFICATION METHODS

Dataset		CNN	ResNet	MobileNet	EfficientNet-B0	ResBaGAN	ViT	ConViT	Mobile-Former	EffBaGAN (ours) Base	EffBaGAN (ours) Small
Das Mestas River	OA	91.69 ± 0.27	92.10 ± 0.28	91.07 ± 0.30	91.60 ± 0.29	92.02 ± 0.24	91.20 ± 0.07	91.05 ± 0.41	91.56 ± 0.24	**92.34 ± 0.22**	91.85 ± 0.56
	AA	87.48 ± 0.93	88.80 ± 0.92	85.73 ± 1.72	87.67 ± 1.13	87.78 ± 1.24	87.81 ± 0.19	85.52 ± 1.37	87.04 ± 0.78	**89.21 ± 0.74**	88.86 ± 0.72
	κ	85.63 ± 0.44	86.32 ± 0.46	84.49 ± 0.50	85.48 ± 0.45	86.18 ± 0.40	84.75 ± 0.13	84.61 ± 0.61	85.42 ± 0.41	**86.79 ± 0.39**	85.94 ± 0.99
	TTPE	**2.86 ± 0.04**	4.65 ± 0.08	4.26 ± 0.06	7.09 ± 0.05	18.53 ± 0.15	3.37 ± 0.02	5.75 ± 0.04	9.51 ± 0.07	16.53 ± 0.74	8.06 ± 0.13
	TrSp	**6.48 ± 0.05**	3.99 ± 0.03	4.35 ± 0.03	2.61 ± 0.02	-	5.49 ± 0.04	3.22 ± 0.03	1.95 ± 0.02	1.12 ± 0.01	2.30 ± 0.02
Eiras Dam	OA	94.39 ± 1.53	89.79 ± 6.96	85.02 ± 13.25	89.53 ± 8.40	97.15 ± 1.51	97.72 ± 0.10	93.34 ± 3.74	91.53 ± 1.43	97.85 ± 0.83	**98.07 ± 0.55**
	AA	71.79 ± 3.33	79.87 ± 7.32	57.50 ± 21.00	56.63 ± 8.04	84.88 ± 2.79	89.39 ± 0.79	76.41 ± 5.17	75.02 ± 3.59	85.92 ± 2.18	**90.69 ± 1.50**
	κ	90.35 ± 2.41	82.49 ± 12.34	74.43 ± 22.27	81.92 ± 14.49	95.12 ± 2.53	96.08 ± 0.17	88.09 ± 7.01	85.22 ± 2.69	96.33 ± 1.39	**96.68 ± 0.95**
	TTPE	2.18 ± 0.01	3.07 ± 0.02	2.95 ± 0.04	4.53 ± 0.02	11.29 ± 0.54	**1.76 ± 0.01**	2.61 ± 0.01	4.22 ± 0.02	8.27 ± 0.12	4.79 ± 0.05
	TrSp	5.18 ± 0.25	3.67 ± 0.18	3.83 ± 0.18	2.49 ± 0.12	-	**6.41 ± 0.31**	4.32 ± 0.21	2.67 ± 0.13	1.37 ± 0.07	2.36 ± 0.11
Ermidas Creek	OA	95.99 ± 1.36	98.15 ± 0.70	97.42 ± 0.38	96.05 ± 2.21	98.61 ± 0.49	**98.79 ± 0.06**	97.09 ± 0.68	97.22 ± 1.84	98.64 ± 0.32	98.78 ± 0.22
	AA	89.29 ± 2.69	94.83 ± 1.56	91.89 ± 1.02	90.54 ± 5.69	95.82 ± 1.01	95.56 ± 0.05	92.23 ± 0.65	92.63 ± 3.43	94.56 ± 2.33	**96.21 ± 0.38**
	κ	92.50 ± 2.74	96.56 ± 1.30	95.21 ± 0.70	92.63 ± 4.11	97.42 ± 0.94	**97.76 ± 0.10**	94.64 ± 1.24	94.89 ± 3.34	97.49 ± 0.60	97.75 ± 0.41
	TTPE	2.99 ± 0.05	4.36 ± 0.05	4.16 ± 0.07	6.30 ± 0.16	15.67 ± 0.31	**2.59 ± 0.04**	3.76 ± 0.01	6.14 ± 0.06	13.40 ± 0.81	7.88 ± 0.12
	TrSp	5.24 ± 0.10	3.60 ± 0.07	3.77 ± 0.07	2.49 ± 0.05	-	**6.05 ± 0.12**	4.17 ± 0.08	2.55 ± 0.05	1.17 ± 0.02	1.99 ± 0.04
Ferreiras River	OA	91.91 ± 0.24	92.83 ± 0.30	91.47 ± 0.50	91.63 ± 0.35	92.60 ± 0.31	92.38 ± 0.05	91.76 ± 0.19	92.19 ± 0.28	**93.14 ± 0.20**	92.99 ± 0.15
	AA	74.71 ± 1.03	76.79 ± 0.94	73.96 ± 1.60	73.45 ± 3.06	75.79 ± 1.07	**78.81 ± 0.12**	74.83 ± 0.73	76.87 ± 0.92	77.38 ± 0.24	77.06 ± 0.38
	κ	87.52 ± 0.35	88.91 ± 0.46	86.87 ± 0.74	87.12 ± 0.51	88.59 ± 0.48	88.26 ± 0.08	87.30 ± 0.27	87.95 ± 0.42	**89.42 ± 0.28**	89.19 ± 0.21
	TTPE	**4.15 ± 0.10**	7.63 ± 0.10	7.05 ± 0.11	12.19 ± 0.10	35.15 ± 1.62	4.51 ± 0.02	8.36 ± 0.08	14.38 ± 0.23	19.65 ± 0.40	12.68 ± 2.44
	TrSp	**8.48 ± 0.39**	4.60 ± 0.21	4.98 ± 0.23	2.88 ± 0.13	-	7.79 ± 0.36	4.21 ± 0.19	2.44 ± 0.11	1.79 ± 0.08	2.77 ± 0.13
Mera River	OA	92.54 ± 0.16	92.85 ± 0.20	92.08 ± 0.23	92.60 ± 0.15	92.62 ± 0.70	92.81 ± 0.11	92.08 ± 0.23	92.65 ± 0.24	**93.34 ± 0.15**	92.94 ± 0.39
	AA	68.29 ± 1.44	68.57 ± 1.87	66.74 ± 1.02	66.50 ± 0.42	70.48 ± 4.90	**71.63 ± 3.86**	66.16 ± 2.07	67.47 ± 0.23	68.42 ± 1.20	67.17 ± 0.85
	κ	88.51 ± 0.21	88.99 ± 0.29	87.75 ± 0.35	88.58 ± 0.22	88.64 ± 1.01	88.90 ± 0.17	87.80 ± 0.35	88.65 ± 0.36	**89.72 ± 0.24**	89.10 ± 0.59
	TTPE	**4.36 ± 0.03**	7.21 ± 0.02	6.70 ± 0.03	11.43 ± 0.08	34.69 ± 3.89	5.25 ± 0.01	10.14 ± 0.17	21.62 ± 2.25	30.42 ± 1.28	15.22 ± 0.30
	TrSp	**7.95 ± 0.89**	4.81 ± 0.54	5.18 ± 0.58	3.04 ± 0.34	-	6.61 ± 0.74	3.42 ± 0.38	1.60 ± 0.18	1.14 ± 0.13	2.28 ± 0.26
Oitavén River	OA	95.15 ± 0.31	96.23 ± 0.47	94.05 ± 0.64	94.66 ± 0.76	97.15 ± 0.29	96.98 ± 0.18	95.10 ± 0.56	96.02 ± 0.21	**97.52 ± 0.08**	97.41 ± 0.16
	AA	88.44 ± 1.10	90.63 ± 1.25	83.79 ± 2.98	86.74 ± 1.16	92.33 ± 1.60	92.30 ± 0.60	91.42 ± 0.53	90.02 ± 1.66	**92.77 ± 1.04**	92.19 ± 0.90
	κ	92.25 ± 0.47	93.97 ± 0.75	90.47 ± 1.03	91.48 ± 1.19	95.45 ± 0.45	95.16 ± 0.29	92.19 ± 0.87	93.63 ± 0.31	**96.02 ± 0.12**	95.84 ± 0.24
	TTPE	2.17 ± 0.07	3.49 ± 0.11	3.28 ± 0.09	6.19 ± 0.11	14.18 ± 0.93	**2.04 ± 0.01**	4.78 ± 0.06	7.53 ± 0.46	11.37 ± 0.16	6.00 ± 0.05
	TrSp	6.52 ± 0.43	4.06 ± 0.27	4.32 ± 0.28	2.29 ± 0.15	-	**6.95 ± 0.45**	2.97 ± 0.19	1.88 ± 0.12	1.25 ± 0.08	2.36 ± 0.15
Ulla River	OA	96.96 ± 1.53	98.26 ± 1.68	87.01 ± 14.88	42.09 ± 0.00	96.18 ± 0.52	97.95 ± 0.82	95.74 ± 0.31	95.98 ± 1.55	89.27 ± 16.00	**98.36 ± 0.87**
	AA	90.95 ± 5.63	**95.30 ± 3.97**	70.41 ± 11.37	20.00 ± 0.00	84.46 ± 9.11	92.76 ± 4.20	76.11 ± 4.08	87.76 ± 4.91	86.62 ± 11.27	93.07 ± 3.17
	κ	95.32 ± 2.38	97.32 ± 2.59	79.40 ± 23.92	0.00 ± 0.00	94.09 ± 0.81	96.86 ± 1.25	93.41 ± 0.48	93.78 ± 2.40	83.21 ± 25.18	**97.47 ± 1.34**
	TTPE	1.48 ± 0.09	1.56 ± 0.14	1.61 ± 0.10	1.78 ± 0.12	2.29 ± 0.02	**0.60 ± 0.00**	0.97 ± 0.02	1.24 ± 0.03	2.16 ± 0.03	1.74 ± 0.03
	TrSp	1.54 ± 0.02	1.47 ± 0.02	1.42 ± 0.01	1.29 ± 0.01	-	**3.84 ± 0.04**	2.36 ± 0.02	1.84 ± 0.02	1.06 ± 0.01	1.32 ± 0.01
Xesta Basin	OA	97.24 ± 1.20	98.40 ± 0.63	97.93 ± 0.27	98.03 ± 0.47	98.54 ± 0.22	98.58 ± 0.03	97.79 ± 0.09	98.36 ± 0.54	**99.17 ± 0.08**	98.83 ± 0.20
	AA	88.82 ± 2.81	92.20 ± 1.64	84.41 ± 2.35	85.34 ± 1.94	86.17 ± 5.34	93.10 ± 0.52	87.78 ± 2.34	91.02 ± 2.31	**93.94 ± 0.41**	91.93 ± 2.45
	κ	91.15 ± 3.37	94.87 ± 1.93	93.20 ± 0.86	93.64 ± 1.38	95.23 ± 0.71	95.37 ± 0.09	92.80 ± 0.31	94.58 ± 1.82	**97.29 ± 0.26**	96.21 ± 0.62
	TTPE	**4.60 ± 0.12**	8.29 ± 0.25	7.83 ± 0.19	14.89 ± 0.24	38.95 ± 1.09	4.92 ± 0.02	11.03 ± 0.13	19.14 ± 0.27	29.53 ± 0.39	14.80 ± 0.12
	TrSp	**8.47 ± 0.24**	4.70 ± 0.13	4.97 ± 0.14	2.62 ± 0.07	-	7.92 ± 0.22	3.53 ± 0.10	2.04 ± 0.06	1.32 ± 0.04	2.63 ± 0.07
Average	OA	94.48 ± 0.83	94.82 ± 1.40	92.01 ± 3.81	87.02 ± 1.58	95.61 ± 0.54	95.80 ± 0.18	94.24 ± 0.78	94.44 ± 2.77	95.16 ± 2.23	**96.16 ± 0.38**
	AA	82.47 ± 2.37	85.87 ± 2.43	76.80 ± 5.38	70.86 ± 2.68	84.72 ± 3.38	**87.67 ± 1.29**	81.31 ± 2.12	83.48 ± 9.00	86.10 ± 2.43	87.15 ± 1.29
	κ	90.40 ± 1.55	91.18 ± 2.52	86.48 ± 6.30	77.61 ± 2.79	92.59 ± 0.92	92.89 ± 0.28	90.10 ± 1.39	90.52 ± 4.27	92.03 ± 3.56	**93.52 ± 0.67**
	TTPE	**3.10 ± 0.06**	5.03 ± 0.09	4.73 ± 0.09	8.05 ± 0.11	21.34 ± 1.07	3.13 ± 0.02	5.92 ± 0.07	10.47 ± 6.90	16.42 ± 0.49	8.90 ± 0.41
	TrSp	6.23 ± 0.30	3.86 ± 0.18	4.10 ± 0.19	2.46 ± 0.11	-	**6.38 ± 0.29**	3.53 ± 0.15	2.12 ± 0.09	1.28 ± 0.05	2.25 ± 0.10

CNN is from [58], ResNet is from [8], ResBaGAN is from [46], MobileNet is from [15], EfficientNet-B0 is from [18], ViT is from [59], ConViT is from [60], and Mobile-Former is from [61].
The OA, AA and κ are in percentages in the range $[0, 100]$, the TTPE, in positive decimals representing seconds, and the TrSp, in positive decimals representing the number of times that the training for the network is faster than for ResBaGAN. Results in bold are the best for each metric for each dataset.

ConViT, despite being also adapted to data scarcity scenarios by utilizing data augmentation via a GAN, just like EffBaGAN. A more detailed analysis of the differences in the classification maps between ViT and EffBaGAN will be conducted at the end of this section.

EffBaGAN also obtains reduced computational cost (average TTPE values) compared to ResBaGAN, the other high accuracy method that is specially adapted to data scarcity scenarios. The reduction in computational cost with respect to ResBaGAN is, as we explained along this work, due to the introduction in the discriminator RedEffDis of the main component of the EfficientNet networks, the MBConv block. It consists of operations with reduced computational cost, such as DWSConv, thereby making the overall method more computationally efficient. Regarding the generator ResEffGen, it helps to synthesize richer samples for training due to the introduction of the EfficientNet-based residual path.

If we analyze the results for the different datasets, we can observe how, depending on the specific dataset, the highest accuracy metrics are obtained by one or the other configuration of the proposed EffBaGAN, or by ViT. However, the mean values of two of the three accuracy metrics are higher in EffBaGAN-Small, being the other in ViT. Comparing the two proposed EffBaGAN configurations, EffBaGAN-Small is more computationally efficient by having a reduced discriminator with fewer operations, making it a better choice than EffBaGAN-Base. Regarding the training time, it can be seen how EffBaGAN-Small is faster than ResBaGAN, the other classification method adapted to data scarcity scenarios, being up to $2.25\times$ faster than it. The EffBaGAN-Small configuration obtains very high accuracy metrics, achieving average values of 96.16% of OA and 87.15% of AA due to the fact that the reduced discriminator is better adapted to this type of data scarcity scenarios, as already seen in the study on discriminator selection in Section IV-B1. A specific case could be that of the Ulla River dataset, which, as noted in Table IV, presents a specially low number of samples per class. Despite this, the proposed method achieves high accuracies with 98.36% of OA. This also

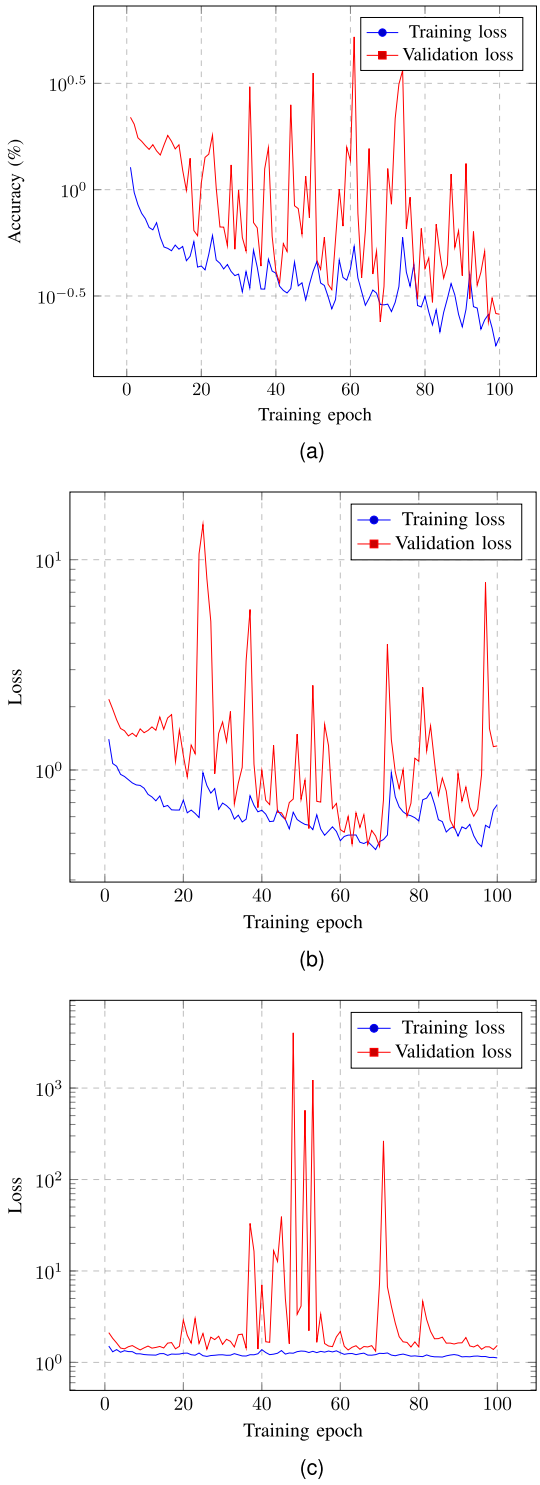

Fig. 8. Evolution of the training and validation losses from different EfficientNets in an experiment on the Eiras Creek dataset. Lower values are better. (a) EfficientNet-B0. (b) EfficientNet-B3. (c) EfficientNet-B7.

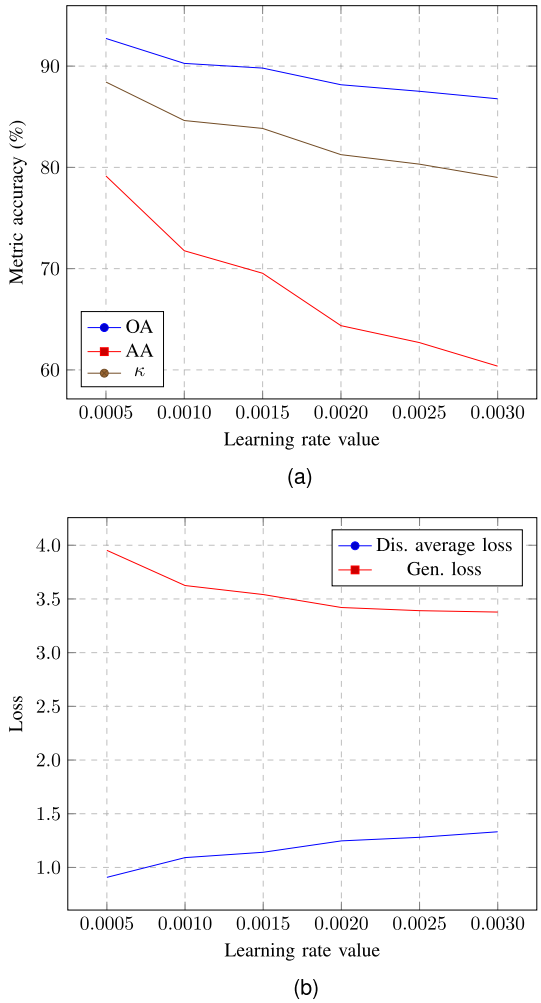

Fig. 9. Relationship between accuracy metrics (a) and discriminator and generator losses (b) with respect to the learning rate value used in hyperparameter optimization. Higher values are better in (a), while lower values are better in (b). (a) Accuracy metrics. (b) Discriminator and generator losses.

explains why EfficientNet-B0 is not able to converge with this dataset.

It is interesting to observe graphically the differences between EffBaGAN-Small and ViT, the two best performing methods in our comparative study, for the detection of the different types of vegetation. Figs. 11 and 12 illustrate different regions of the Eiras Dam dataset in composite color, alongside their reference information and the classification maps obtained by EffBaGAN-Small and ViT. The class and color identification is consistent with the details provided in Table IV. It can be observed that ViT does not discriminate correctly between different vegetation types, assigning different colors, which correspond to different vegetation types, to regions that are uniform in the reference information. These problems are partially solved by the proposed EffBaGAN-Small although some errors remain detectable.

6) Ablation Study: In order to evaluate the impact of each data augmentation technique (traditional and through BAGAN) applied by EffBaGAN in the performance, additional experiments have been carried out selectively removing each one of the augmentation techniques. These experiments have been performed on the Eiras Dam dataset, showing the mean values and confidence intervals of OA, AA, and κ for 5 repetitions of each experiment. Specifically, two variants of EffBaGAN-Small were tested: applying only traditional augmentation and applying only BAGAN-based augmentation.

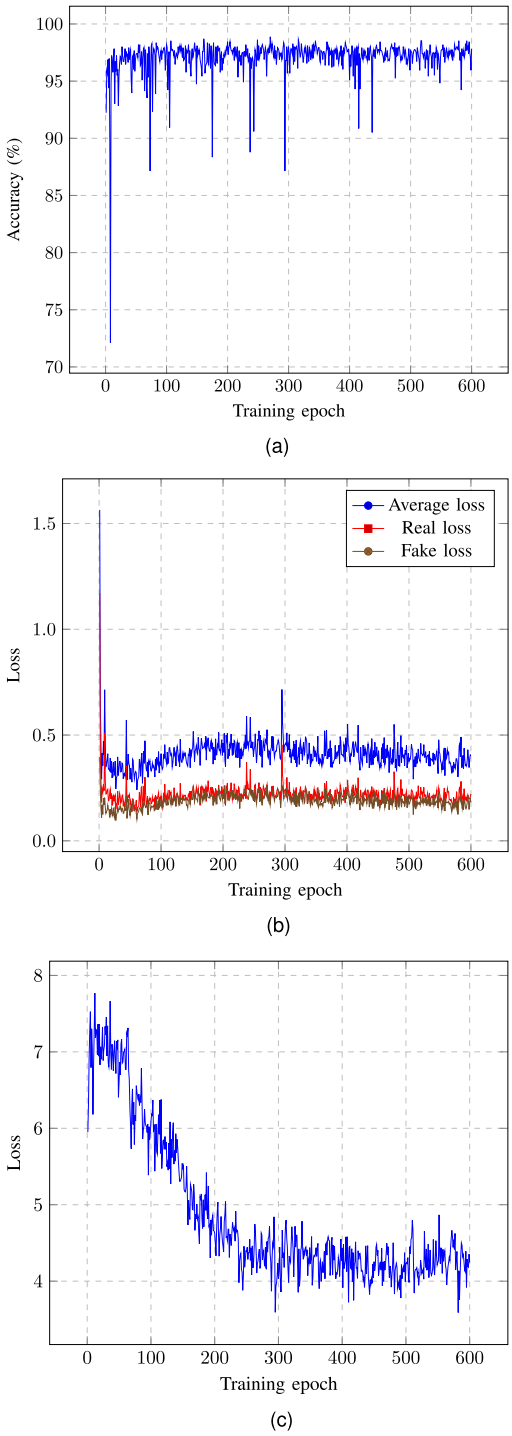

Fig. 10. Evolution of discriminator validation accuracy (a), discriminator losses (b) and generator loss (c) of EffBaGAN-Base in an experiment on the Ermidas Creek dataset. Higher values are better in (a), while lower values are better in (b) and (c). (a) Discriminator validation accuracy. (b) Discriminator losses. (c) Generator loss.

Fig. 13 depicts the results in terms of accuracy. It can be seen how both data augmentation techniques contribute to an increase in accuracy metrics, especially in terms of AA. The application of both augmentation techniques reaches higher accuracy, as it helps the classifier to learn the minority classes with limited data.

TABLE VIII
COMPUTATIONAL RESOURCES USED BY THE DIFFERENT NETWORKS

Network	Size (MiB)	# of trainable parameters	# of FLOPs	Average TTPE (s)
CNN	0.23	59 082	3 408 192	3.10 ± 0.06
ResNet	0.65	152 730	9 786 688	5.03 ± 0.09
MobileNet	12.41	3 217 802	11 950 592	4.73 ± 0.09
EfficientNet-B0	15.63	4 020 934	8 945 664	8.05 ± 0.11
ResBaGAN	10.79	2 756 960	48 858 496	21.34 ± 1.07
ViT	12.13	3 172 650	240 513 920	3.13 ± 0.02
ConViT	21.31	5 572 434	23 295 360	5.92 ± 0.07
Mobile-Former	8.66	2 218 182	4 720 512	10.47 ± 6.90
EffBaGAN-Base	40.80	10 481 354	175 547 744	16.42 ± 0.49
EffBaGAN-Small	16.25	4 129 002	61 942 112	8.90 ± 0.41

7) Computational Cost: In order to further analyze the trade-off between accuracy and computational cost of the different networks, their computational requirements have been analyzed. Table VIII shows the size, number of trainable parameters, number of FLOPs, and average TTPE, directly extracted from Table VII, for the different networks tested. It can be observed that the network with the biggest size in memory (and, therefore, that with the higher number of trainable parameters) is EffBaGAN-Base, while the one with the highest number of FLOPs is ViT. It is worth noting that the size of EffBaGAN-Small is less than 1 MiB larger in memory than EfficientNet-B0 achieving higher accuracy. It can also be seen that EffBaGAN-Small, despite having more trainable parameters than ResBaGAN, requires lower average TTPE than it. This is due to the MBConv blocks of EffBaGAN-Small discriminator RedEffDis, which speed up the convolution operations by performing them in two separate steps through DWSConvs, thus avoiding the interaction between channels, as mentioned earlier in this work.

In order to analyze how the computational cost of EffBaGAN-Small changes depending on the parameter selection, various configurations of its discriminator, RedEffDis, were tested on the Eiras Dam dataset. Specifically, the expansion factor e of the MBConv block stages was varied, while keeping all other configurable parameters set to their default values. Each configuration was run five times to ensure consistency in the accuracy metrics. As detailed in Table IX, the network size, number of trainable parameters, number of FLOPs, and accuracy metrics for the different RedEffDis configurations are presented. The results indicate that accuracy improves as the expansion factor e increases. Consequently, the value $e = 6$ was selected, despite the fact that the network size and number of FLOPs are higher compared to other configurations in the table. It is worth noting that if the priority were to minimize computational cost rather than maximize accuracy, lower values e would be more appropriate.

8) Other Datasets: The proposed method has also been tested on three widely used remote sensing datasets with higher spectral resolution and lower spatial resolution than the experimental datasets described in Table III: Salinas, Pavia University, and Indian Pines [63]. For the experiments, all parameters have been kept the same as those used in the other experiments in this work. However, segmentation into superpixels was not performed, as these new images are smaller (the largest of has

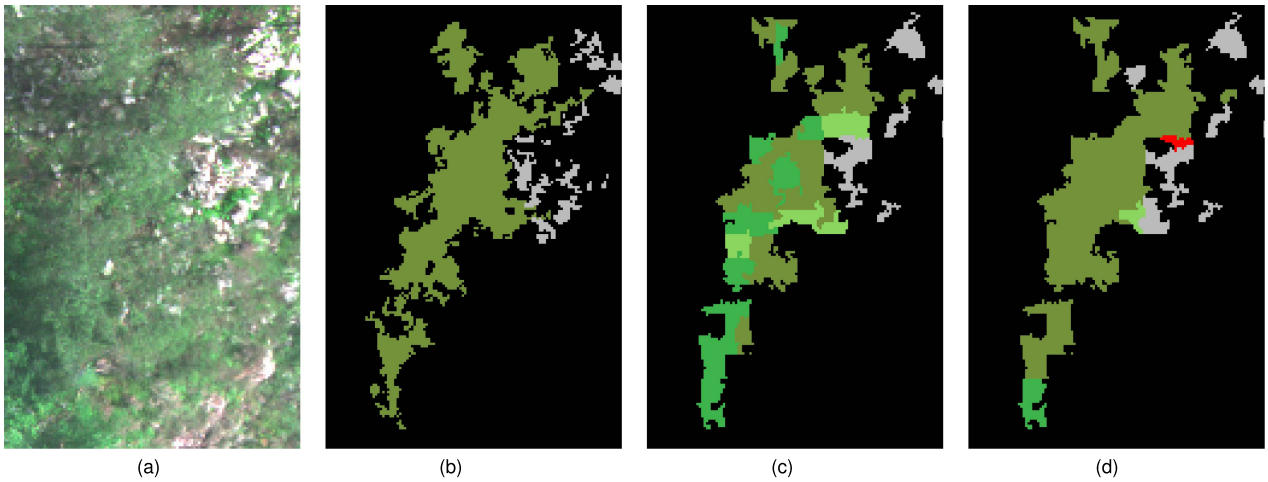

Fig. 11. Composite color image (a), reference information (b), ViT classification map (c), and EffBaGAN-Small classification map (d) of a region of the Eiras Dam dataset where pines and rocks are present. The class and color identification is consistent with the details provided in Table IV. (a) Composite color. (b) Reference information. (c) ViT. (d) EffBaGAN-Small.

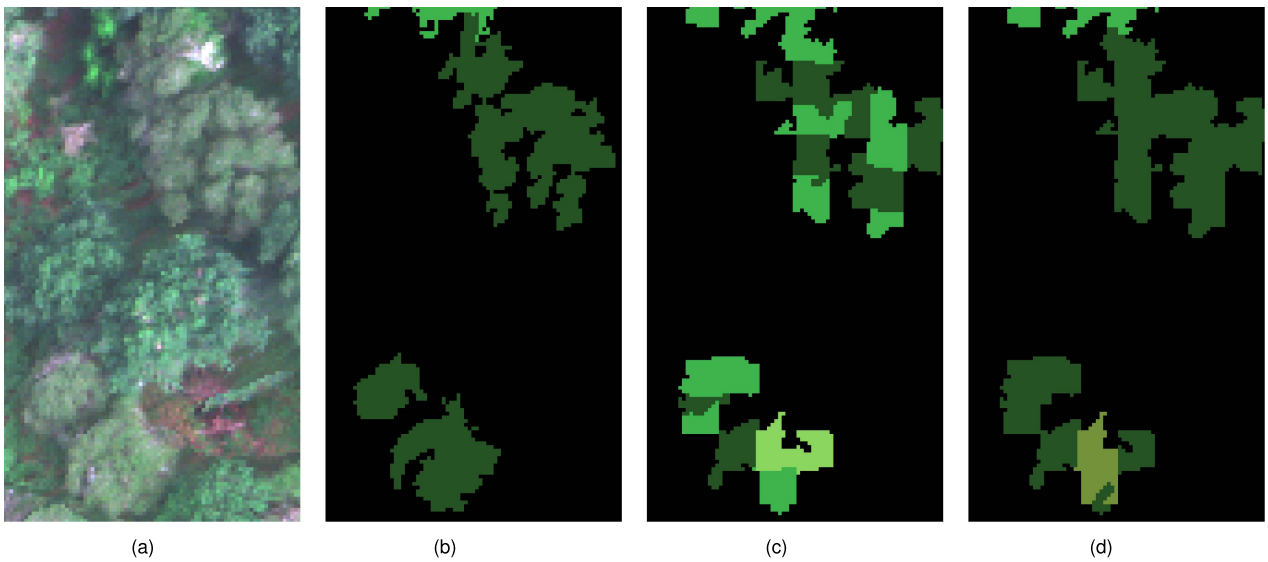

Fig. 12. Composite color image (a), reference information (b), ViT classification map (c), and EffBaGAN-Small classification map (d) of a region of the Eiras Dam dataset where eucalyptus are present. The class and color identification is consistent with the details provided in Table IV. (a) Composite color. (b) Reference information. (c) ViT. (d) EffBaGAN-Small.

TABLE IX
COMPUTATIONAL RESOURCES AND AVERAGE RECORDED ACCURACY METRICS FOR EIRAS DAM DATASET USING DIFFERENT EXPANSION FACTORS FOR THE EFFBAGAN-SMALL DISCRIMINATOR

Expansion factor	**Size** (MiB)	**# of trainable parameters**	**# of FLOPs**	**OA** (%)	**AA** (%)	κ (%)
$e = 1$	9.62	2 415 914	13 530 976	97.10 $\pm$ 0.40	85.59 $\pm$ 3.68	95.03 $\pm$ 0.68
$e = 2$	11.32	2 850 794	26 591 584	97.60 $\pm$ 0.26	89.40 $\pm$ 1.83	95.88 $\pm$ 0.43
$e = 3$	12.55	3 170 346	35 429 216	97.37 $\pm$ 0.45	91.01 $\pm$ 1.70	95.46 $\pm$ 0.81
$e = 4$	13.78	3 489 898	44 266 848	97.02 $\pm$ 1.12	87.98 $\pm$ 3.77	94.95 $\pm$ 1.81
$e = 5$	15.02	3 809 450	53 104 480	98.11 $\pm$ 0.18	88.37 $\pm$ 1.63	96.74 $\pm$ 0.31
$e = 6$	16.25	4 129 002	61 942 112	98.07 $\pm$ 0.55	90.69 $\pm$ 1.50	96.68 $\pm$ 0.95

Note that $e = 6$ is the default setting of the EffBaGAN-Small discriminator, therefore the row corresponding to $e = 6$ are the values for the proposed method EffBaGAN-Small.

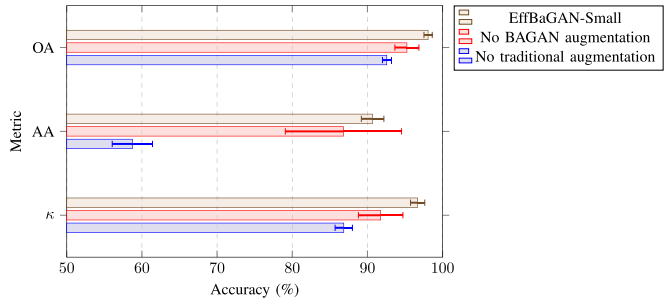

Fig. 13. Accuracy metrics in the ablation study of the two data augmentation techniques used in EffBaGAN on the Eiras Dam dataset. The bars represent the mean values along their confidence intervals. Higher values are better.

TABLE X
RECORDED ACCURACY METRICS FOR THREE WIDELY USED REMOTE SENSING DATASETS USING EFFBAGAN-SMALL

Dataset	OA (%)	AA (%)	κ (%)
Salinas	99.52 $\pm$ 0.63	99.32 $\pm$ 0.98	99.47 $\pm$ 0.70
Pavia University	99.80 $\pm$ 0.10	99.61 $\pm$ 0.16	99.74 $\pm$ 0.13
Indian Pines	98.31 $\pm$ 0.41	93.92 $\pm$ 2.90	98.07 $\pm$ 0.47

610 $\times$ 340 px), making pixel-by-pixel processing computationally feasible. Each run for these datasets was repeated five times. Table X shows that EffBaGAN-Small achieved OA and AA values above 99% for two out of the three datasets. For Indian Pines, with has a much lower number of labeled samples, specific parameter and architecture configurations would be required for maximum performance.

V. DISCUSSION

Deep learning-based techniques for the classification of remote sensing images present high classification performance, but also high computational cost and, typically, require large amounts of data to be adequately trained. Scenarios with data scarcity, such as the forest mapping one studied in this work, pose a challenge to these techniques. In general, in remote sensing for Earth observation, the scarcity of labeled data, and the class imbalances [4] are common issues. In this context, data augmentation techniques, especially those based on GANs [21] such as BAGAN [27], play a significant role.

Regarding the high computational cost of the deep learning classification techniques, several new efficient methods, such as EfficientNet [18], have been developed to specifically reduce it [14]. In this work, we focus on EfficientNets due to their structure based on the compound coefficient that modifies the depth, width and resolution of the network to adapt to the specific computational device used without significantly affecting classification accuracy. EffBaGAN, a combination of EfficientNet, an efficient classification method, and BAGAN, a data augmentation technique, has been shown to be effective in this work. The analyzed results show lower computational cost than other approaches specifically designed with similar objectives, such as ResBaGAN.

It is important to mention that the proposed method in this work is adapted to the spatial and spectral resolution of the considered datasets. If it is desired to use this method with input samples with different spatial resolution or from remote sensing images with different number of bands than those shown here, the patch size should be changed. In our case the size is 32 $\times$ 32 $\times$ 5. The spatial resolution in the network is reduced by the consecutive convolutional layers, so the number of layers and the stride should be adapted to avoid an excessive reduction of the spatial resolution. If the number of bands is very different from that of the images used here, it will also be appropriate to change the number of feature maps that are obtained throughout the network, in both discriminator and generator.

If maximum performance is required, some additional changes can be applied. First, in the discriminator, the sizes of the kernels used in the convolution operations for the extraction of features should be selected according to the resolution of the input patches; and, second, in the generator, the size of the latent space should be adapted to the desired resolution for the synthetic samples. Overfitting should also be avoided as it was explained in Section IV-B3. In addition, as future work to improve the accuracy of the proposed method, more sophisticated architectures for the discriminator such as transformers could be proposed.

Although GANs are a very good alternative for sample synthesis and, consequently, dataset enrichment, they have some limitations. The main limitation of GANs lies in their instability during the training process. Due to the nature of their adversarial architecture, training can be affected by a difficult to achieve equilibrium. On many occasions, the discriminator tends to improve much faster than the generator, which can result in the latter eventually generating poor quality samples. Another limitation of GANs is the high computational cost involved in training, making their applicability difficult in scenarios with limited computational resources or strong temporal constraints. In addition, this type of network is difficult to adjust in terms of the selection of hyperparameters and the architecture itself, and this adjustment is done empirically, being laborious and unsystematic. For all these reasons, the design of the architecture of the proposed method in this work has been done following a careful process.

Another method that shows good results in our experiments is ViT, which also presents very high accuracy metrics but lower in a 1% on average than the proposed method EffBaGAN over the studied datasets. It is important to note that EffBaGAN and ViT are two deep learning architectures with different approaches and purposes. EffBaGAN represents an advancement of GAN-based networks, therefore, employing adversarial training, while ViT is based on the transformer architecture that captures complex relations among the input samples. Also, as discussed in the experimentation, ViT distinguishes less well between the different vegetation types in the datasets used, which is the critical point of the classification in this particular domain.

The computational cost of EffBaGAN opens an interesting line for future research in the use of GAN-based architectures in real-time applications. While EffBaGAN represents a significant step toward near real-time processing and suitability for

hardware architectures with limited capabilities in data-scarcity scenarios, it still presents some limitations. Although it achieves a good balance between performance and computational cost, the training time remains higher compared to other methods.

Future work on this network should focus on optimizing its architecture to further reduce computational costs without compromising accuracy. These enhancements could enable more efficient execution on resource-constrained platforms, making it even more practical for real-world applications. Also concerning potential future improvements for increasing performance, techniques such as quantization and pruning could reduce computational costs even further [16]. These techniques would allow for more efficient inference by minimizing the number of operations and memory requirements.

Finally, the adaptation to other computing platforms could also be an interesting line for future research. Although the current experiments were conducted using a single core and one high-performance GPU, aiming to minimize the amount of computing resources used, evaluating the performance on alternative computing infrastructures such as distributed computing infrastructures could prove beneficial. This flexibility would allow the model to scale efficiently for problems involving bigger datasets and, therefore, requiring more substantial computational resources.

VI. Conclusion

This work introduces EffBaGAN, a deep learning method tailored for multispectral remote sensing image classification, specifically addressing the challenges posed by data scarcity while maintaining computational efficiency. The method integrates an EfficientNet-based residual generator and an EfficientNet-based discriminator within a BAGAN-based data augmentation framework. In addition, the use of superpixel-based sample extraction helps to reduce computational costs. This work represents the first application of a fully EfficientNet-based architecture with BAGAN for remote sensing classification.

To optimize performance, various configurations of the EffBaGAN discriminator were tested, modifying factors such as the repetition of each stage, the expansion factor, and the reduction rate in the SE module. Generator topologies were also explored, with variations in the number of transposed convolutions in the main path and the type of residual paths used, including a simple transposed convolution, upsampling with convolution, or transposed DWSConv. As a result, a more effective and computationally efficient overall network topology was achieved.

The method was evaluated under a data scarcity scenario, focusing on classifying eight high-resolution multispectral images of forests in Galicia (Spain) with limited training data and pronounced class imbalances. Compared to other classification methods, including some based on transformers, EffBaGAN demonstrated high overall and average classification accuracy, with an average TTPE of 8.90 s, being more than twice as fast as ResBaGAN, another method designed for data scarcity scenarios. The best-performing configuration, EffBaGAN-Small, achieved an OA of 96.16% and an AA of 87.15%.

Finally, this work opens several research directions. One potential direction is replacing the EffBaGAN discriminator with more advanced architectures like transformers, which could enhance accuracy. The proper configuration to manage computational costs for such models would need thorough investigation. In addition, reducing the computational burden of EffBaGAN through techniques like quantization and pruning, as well as experimentation on different computing platforms, could improve its efficiency.

References

[1] D. Chutia, D. K. Bhattacharyya, K. K. Sarma, R. Kalita, and S. Sudhakar, "Hyperspectral remote sensing classifications: A perspective survey," *Trans. GIS*, vol. 20, no. 4, pp. 463–490, 2016.

[2] F. Argüello, D. B. Heras, A. S. Garea, and P. Quesada-Barriuso, "Watershed monitoring in Galicia from UAV multispectral imagery using advanced texture methods," *Remote Sens.*, vol. 13, no. 14, 2021, Art. no. 2687.

[3] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gürbüz, and U. Sakarya, "A short survey of hyperspectral remote sensing applications in agriculture," in *Proc. 6th Int. Conf. Recent Adv. Space Technol.*, 2013, pp. 171–176.

[4] T. A. W. Aaron, E. Maxwell, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018, doi: 10.1080/01431161.2018.1433343.

[5] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 158, pp. 279–317, 2019.

[6] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[9] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, 2017.

[10] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216310104

[11] Z. Zhong, J. Li, L. Ma, H. Jiang, and H. Zhao, "Deep residual networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 1824–1827.

[12] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[13] N. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," in *Proc. 9th Comput. Within Limits*, Jun. 2023. [Online]. Available: https://limits.pubpub.org/pub/wm1lwjce

[14] B. R. Bartoldson, B. Kailkhura, and D. Blalock, "Compute-efficient deep learning: Algorithmic trends and opportunities," *J. Mach. Learn. Res.*, vol. 24, no. 122, pp. 1–77, 2023. [Online]. Available: http://jmlr.org/papers/v24/22-1208.html

[15] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[16] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221010894

[17] N. Hernández, F. Almeida, and V. Blanco, "Performance and energy efficiency: Quantization of models for IoT devices," *Res. Square*, 2023, doi: 10.21203/rs.3.rs-3405705/v1.

[18] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Jun. 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[19] B. Zohuri and M. Moghaddam, "Deep learning limitations and flaws," *Modern Approaches Mater. Sci.*, vol. 2, pp. 241–250, 2020.

[20] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2204.08610

[21] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[22] Q. Su, H. N. A. Hamed, M. A. Isa, X. Hao, and X. Dai, "A GAN-based data augmentation method for imbalanced multi-class skin lesion classification," *IEEE Access*, vol. 12, pp. 16498–16513, 2024.

[23] E. Strelcenia and S. Prakoonwit, "A survey on GAN techniques for data augmentation to address the imbalanced data issues in credit card fraud detection," *Mach. Learn. Knowl. Extraction*, vol. 5, no. 1, pp. 304–329, 2023. [Online]. Available: https://www.mdpi.com/2504-4990/5/1/19

[24] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[25] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: https://arxiv.org/abs/1411.1784

[26] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 2642–2651. [Online]. Available: https://proceedings.mlr.press/v70/odena17a.html

[27] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," in *Proc. Int. Conf. Mach. Learn.*, 2018. [Online]. Available: https://research.ibm.com/publications/bagan-data-augmentation-with-balancing-gan

[28] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690370209

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[31] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 10096–10106. [Online]. Available: https://proceedings.mlr.press/v139/tan21a.html

[32] S. Gupta and M. Tan, "EfficientNet-EdgeTPU: Creating accelerator-optimized neural networks with AutoML," 2019. [Online]. Available: https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html

[33] V.-T. Hoang and K.-H. Jo, "Practical analysis on architecture of EfficientNet," in *Proc. 14th Int. Conf. Hum. Syst. Interact.*, 2021, pp. 1–4.

[34] L. Cao, "A MobileNetV2 model of transfer learning is employed for remote sensing image classification," *Adv. Eng. Technol. Res.*, vol. 10, no. 1, pp. 596–596, 2024.

[35] S. Du, J. Li, and M. Noto, "Comparison and analysis of three MobileNet-based models for wildfire detection," *J. Adv. Inf. Technol.*, vol. 15, no. 4, pp. 511–518, 2024.

[36] P. Charoenchittang, P. Boonserm, K. Kobayashi, and N. Cooharojananone, "Airport buildings classification through remote sensing images using EfficientNet," in *Proc. 18th Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inf. Technol.*, 2021, pp. 127–130.

[37] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of remote sensing images using EfficientNet-B3 CNN model with attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021.

[38] R. D. I. Puspitasari, F. Q. Annisa, and D. Ariyanto, "Flooded area segmentation on remote sensing image from unmanned aerial vehicles (UAV) using DeepLabV3 and EfficientNet-B4 model," in *Proc. Int. Conf. Comput., Control, Informat. Appl.*, 2023, pp. 216–220.

[39] R. Wang, Z. Yang, H. Qiu, X. Liu, and D. Wu, "Spatial and channel exchange based on EfficientNet for detecting changes of remote sensing images," in *Proc. 26th Int. Conf. Comput. Supported Cooperative Work Des.*, 2023, pp. 1595–1600.

[40] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218310749

[41] O. Fedoruk, K. Klimaszewski, A. Ogonowski, and R. Możdżonek, "Performance of GAN-based augmentation for deep learning COVID-19 image classification," *AIP Conf. Proc.*, vol. 3061, no. 1, 2024, Art. no. 030001, doi: 10.1063/5.0203379.

[42] T. Kwak and Y. Kim, "Semi-supervised land cover classification of remote sensing imagery using CycleGAN and EfficientNet," *KSCE J. Civil Eng.*, vol. 27, no. 4, pp. 1760–1773, Apr. 2023, doi: 10.1007/s12205-023-2285-0.

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[44] L. Abady et al., "Detection and localization of GAN manipulated multi-spectral satellite images," in *Proc. Euro. Symp. Artif. Neural Netw.*, 2022, pp. 339–344.

[45] B. Feng, Y. Liu, H. Chi, and X. Chen, "Hyperspectral remote sensing image classification based on residual generative adversarial neural networks," *Signal Process.*, vol. 213, 2023, Art. no. 109202. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168423002761

[46] Á. G. Dieste, F. Argüello, and D. B. Heras, "ResBaGAN: A residual balancing GAN with data augmentation for forest mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 6428–6447, 2023.

[47] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 298–314, Feb. 2015, doi: 10.1007/s11263-014-0744-2.

[48] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[49] V. Machairas et al., "Waterpixels," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3707–3716, Nov. 2015.

[50] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[51] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sens. Environ.*, vol. 37, no. 1, pp. 35–46, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/003442579190048B

[52] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *Proc. ICLR: Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[53] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist., Ser. Proc. Mach. Learn. Res.*, May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[54] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *Under Rev. ICLR*, 2015. [Online]. Available: https://www.researchgate.net/publication/284579051_Fast_and_Accurate_Deep_Network_Learning_by_Exponential_Linear_Units_ELUs

[55] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3. [Online]. Available: https://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[57] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2023. [Online]. Available: https://arxiv.org/abs/1606.08415

[58] Y. Choi, "PyTorch tutorial," 2017. Accessed: Oct. 21, 2024. [Online]. Available: https://github.com/yunjey/pytorch-tutorial/tree/master

[59] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[60] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "ConVit: Improving vision transformers with soft convolutional inductive biases," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2286–2296.

[61] Y. Chen et al., "Mobile-Former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5270–5279.

[62] W. Han et al., "A survey of machine learning and deep learning in remote sensing of geological environment: Challenges, advances, and opportunities," *ISPRS J. Photogrammetry Remote Sens.*, vol. 202, pp. 87–113, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271623001582

[63] ROSIS, "Hyperspectral remote sensing scenes," 2013, Accessed: Oct. 21, 2024. [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

Nicolás Vilela-Pérez received the B.S. degree in computer engineering and the M.Sc. degree in big data in 2023 and 2024, respectively, from the Universidade de Santiago de Compostela, Santiago de Compostela, Spain, where he is currently working toward the Ph.D. degree in computer science.

He is currently a Collaborative Researcher with the Singular Research Center on Intelligent Technologies (CiTIUS), Universidade de Santiago de Compostela. His research interests focus on computer vision tasks, while bringing together knowledge from diverse computing areas, such as artificial intelligence, high-performance computing, and big data.

Dora B. Heras (Member, IEEE) received the M.Sc. degree in physics and the Ph.D. degree in physics from the Universidade de Santiago de Compostela, Santiago de Compostela, Spain, in 1995 and 2000, respectively.

Since 2023, she is a Vice-Chair of the International Parallel Computing conference (Euro-Par). Since 2020, she has been the Chair for the High-Performance and Disruptive Computing in Remote Sensing (HDCRS) Working Group under the IEEE GRSS Earth Science Informatics Technical Committee (ESI TC). She is currently a Full Professor with the Department of Electronics and Computing, Universidade de Santiago de Compostela. Her research interests cover a range of topics in the combined fields of image processing, remote sensing, machine learning, and high-performance computing applied to Earth observation. In particular, she has published papers on registration, classification, domain adaptation, and change detection applied to multispectral and hyperspectral remotely sensed images.

Francisco Argüello received the B.S. and Ph.D. degrees in physics from the Universidade de Santiago de Compostela, Santiago de Compostela, Spain, in 1988 and 1992, respectively.

He is currently a Full Professor with the Department of Electronics and Computing, Universidade de Santiago de Compostela. His research interests include signal and image processing, computer graphics, parallel and distributed computing, and quantum computing.