Automated machine learning for satellite data: integrating remote sensing pre-trained models into AutoML systems

Salinas, N.R.P.; Baratchi, M.; Rijn, J.N. van; Vollrath, A.; Dong, Y.; Kourtellis, N.; ... ; Lozano, J.A.

Automated Machine Learning for Satellite Data: Integrating Remote Sensing Pre-trained Models into AutoML Systems

Nelly Rosaura Palacios Salinas[1(✉)], Mitra Baratchi[1], Jan N. van Rijn[1], and Andreas Vollrath[2]

[1] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
n.r.palacios.salinas@umail.leidenuniv.nl
[2] Phi-Lab, ESA/ESRIN, Frascati, Italy

Abstract. Current AutoML systems have been benchmarked with traditional natural image datasets. Differences between satellite images and natural images (e.g., bit-wise resolution, the number, and type of spectral bands) and lack of labeled satellite images for training models, pose open questions about the applicability of current AutoML systems on satellite data. In this paper, we demonstrate how AutoML can be leveraged for classification tasks on satellite data. Specifically, we deploy the Auto-Keras system for image classification tasks and create two new variants, IMG-AK and RS-AK, for satellite image classification that respectively incorporate transfer learning using models pre-trained with (i) natural images (using ImageNet) and (ii) remote sensing datasets. For evaluation, we compared the performance of these variants against manually designed architectures on a benchmark set of 7 satellite datasets. Our results show that in 71% of the cases the AutoML systems outperformed the best previously proposed model, highlighting the usefulness of a customized satellite data search space in AutoML systems. Our RS-AK variant performed better than IMG-AK for small datasets with a limited amount of training data. Furthermore, it found the best automated model for the datasets composed of near-infrared, green, and red bands.

Keywords: Remote sensing · AutoML · Transfer learning · Classification

1 Introduction

Remote sensing satellites continuously monitor the Earth's surface and collect data representing the state and health of the planet. The range of applications that can benefit from such data varies from environmental mapping to urban planning, emergency response, and many more [3]. To make use of such data, remote sensing practitioners commonly adopt methods of computer vision and

A. Vollrath—Affiliated with FAO since 09/2020.

The original version of this chapter was revised: The given name and surname of the author Nelly Rosaura Palacios Salinas have been wrongly attributed in some parts of the original publication. This has now been corrected. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-86517-7_32

Y. Dong et al. (Eds.): ECML PKDD 2021, LNAI 12979, pp. 447–462, 2021.
https://doi.org/10.1007/978-3-030-86517-7_28

machine learning. Classical machine learning approaches benefit from domain-specific, hand-crafted features to account for dependencies in time or space, but rarely exploit spatio-temporal dependencies exhaustively. Modern deep learning methods can automatically extract such spatio-temporal features. However, currently, two obstacles are limiting the use of deep learning for satellite data. The first one is the lack of sufficient labeled data and the difficulty of getting labels considering that satellite images are not as interpretable as natural images for the human eye [3]. The second obstacle lies in the difficulty of designing appropriate architectures that take the characteristics of satellite images into account. Satellite images are different from natural images due to their additional spectral information content. Natural color images always include the same three channels (RGB) but for satellite images, the number and type of channels are variable, depending on the satellite instrument. A multi-spectral satellite image captures information of the electromagnetic spectrum related to different processes on Earth (e.g., land, ocean, atmosphere). The images from the most common satellites can have up to 13 spectral bands that each could be relevant for observing a different process. For instance, examples of channels related to vegetation features are near-infrared and short-wave infrared bands.

Furthermore, natural images have an 8 bits precision, while remote sensing input data usually comes at higher precision (16 or 32 bits). Creating new high-performing models for satellite data requires designing new architectures while taking into account these characteristics. Furthermore, the hyperparameters need to be set properly. These tasks can be complex for remote sensing experts.

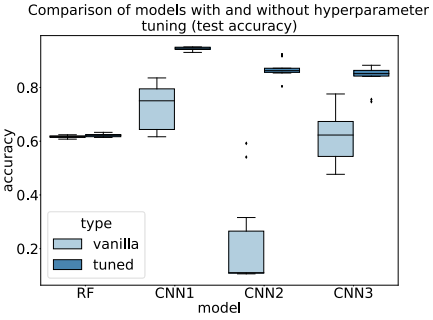


Fig. 1. Preliminary experiments using the EuroSAT dataset [9]. A random forest and three different CNNs built from scratch based on machine learning (a simple CNN with 3 convolutional layers (CNN1)) and remote sensing literature (CNN2 [1], CNN3 [15]) are compared. For each model, two versions are shown: a vanilla model performance using default configurations and a tuned model. The tuned models show the performance after applying hyperparameter tuning for the optimizer, learning rate, batch size, and the number of epochs in the case of the CNNs and the number of features for the random forest.

To overcome these obstacles, we propose to systematically leverage recent developments in two different machine learning fields: (i) transfer learning [22] and (ii) automated machine learning (AutoML) [11]. Transfer learning addresses the lack of labeled data by re-using the knowledge gained from previously seen tasks and transferring it to a newly created model in another task (e.g., through using pre-trained models). AutoML [11] aims to automatically design high-performing models for each dataset in a data-driven manner and thus making machine learning accessible to non-machine learning experts. Hyperparameter optimiza-

tion and Neural Architecture Search (NAS) are both exemplars of techniques that are scoped in this field. Specifically, with the increased interest in using deep learning algorithms, NAS has become an important area that aims at finding the best neural network architecture given a task and a dataset by automatically tuning various hyperparameters.

As far as we are aware, NAS research systems have been benchmarked with natural image datasets but not with satellite images. This brings us to the questions: *what is the performance of current AutoML systems for satellite data?* and *how can we further improve their performance for satellite data by transferring the knowledge gained from previous research in the field of remote sensing?* Fig. 1 shows the results of one of our preliminary experiments, demonstrating the potential of applying the most recent advances in AutoML regarding hyperparameter optimization to a remote sensing dataset. We know that positive results in specific applications are based on human priors. By incorporating domain expert prior knowledge into machine learning systems the performance of resultant models can significantly improve. Therefore, in this paper, we propose to tailor the neural architecture search space of Auto-Keras [12] (a popular AutoML system) by integrating findings of the remote sensing field in form of pre-trained models on ImageNet and remote sensing datasets.

To the best of our knowledge, this is the first work considering the design of AutoML systems for machine learning tasks based on remote sensing datasets. More specifically, to achieve this goal our contributions in this paper are as follows: (i) composing a diverse benchmark of already available satellite datasets using a standardized format, (ii) evaluating the performance of the deployed AutoML NAS system on these datasets, and finally, (iii) enriching this system by incorporating pre-trained models on remote sensing datasets in a new block called RS-AK.

2 Related Work

In this section, we review the most popular deep learning approaches applied to satellite data and the current status of AutoML in remote sensing.

Deep Learning in Remote Sensing: The remote sensing research community increasingly relies on the use of deep learning models. The authors of [3] indicate that CNN-based methods have obtained impressive results when numerous annotated samples to fine-tune or train a network from scratch are available. Due to the difficulty of acquiring labeled data, researchers typically rely on techniques from transfer learning, with models pre-trained on natural image datasets (e.g., ImageNet) but also remote sensing benchmark datasets (e.g., EuroSat [9], BigEarthNet [25]). Some works that rely on this technique are [9,16,18,23]. The authors of [19] analyze three different transfer learning strategies to improve the performance of CNNs for satellite image scene classification, i.e., full training, fine-tuning, and using CNNs as feature extractors. They conclude that the fine-tuning approach tends to be a good option in various scenarios. The authors of [9] evaluated various CNN architectures on the EuroSAT dataset, achieving the best accuracy using a fine-tuned ResNet-50 pre-trained on ImageNet for the RGB data. The

authors of [19,30] reported high-performance results using CNNs too. The authors of [15] suggest that an ensemble of Inception and ResNet modules is an effective architecture for land cover classification. Current remote sensing research does not fully exploit hyperparameter tuning to further improve these models; researchers have mainly considered optimizing a subset of hyperparameters using a parameter sweep approach [6,18]. The authors of [13] have considered AutoML for a specific application of high-throughput image-based plant phenotyping. They use Auto-Keras and compare its results with human-designed ImageNet pre-trained CNN architectures, finding the best performance while using the pre-trained network. However, they did not use all the potential of Auto-Keras. In this paper, we consider more general applicability by performing a systematic analysis on a diverse benchmark of problems and we propose the customization of Auto-Keras for satellite tasks. Moreover, we see that many architectures have been applied to remote sensing problems, but no clear consensus has been reached about which one works best. This makes a compelling argument for using AutoML, which can explore and select the best option in a data-driven way.

AutoML and Neural Architecture Search: AutoML aims to automate the different stages of a machine learning pipeline. These steps typically are data collection, data preparation, feature engineering, preprocessing, algorithm selection, hyperparameter optimization, model training, and deployment. Current AutoML systems commonly cover stages from data preparation to model training [11]. Auto-Sklearn [7], Auto-WEKA [26] and T-POT [20] are examples of AutoML systems focusing on traditional machine learning (such as SVM, random forest, K-nearest neighbors). So far, only a few open-source AutoML systems focus on deep learning. One of the biggest challenges of NAS compared to previously mentioned AutoML systems is maintaining computational efficiency. The time required to successfully solve the NAS problem is linked to the time needed to train a candidate network and the number of candidates existing in the search space. Two popular AutoML systems that focus on deep learning are Auto-Keras [12] and Auto-Pytorch [17], both supporting image classification tasks. Auto-Pytorch uses multi-fidelity optimization and Bayesian optimization (BOHB) [5] while Auto-Keras uses a Bayesian optimization with a neural network kernel and a tree-structured acquisition function to search for the best settings. The search space of Auto-Keras is defined based on network morphism, it encloses all architectures that can be created by morphing the initial architecture. Auto-Pytorch is delimited to multi-layer perceptron networks and funnel-shaped residual networks. To deal with the memory limitations, Auto-Pytorch asks the user to choose between small, medium, and full configuration spaces, whereas Auto-Keras adapts the configuration space automatically based on a memory estimation function. Both systems have focused on solving traditional machine learning tasks, in the case of image classification the attention is only on natural images. Our goal in this paper is to focus on Earth observation data. We propose to customize AutoML systems for satellite data tasks. The challenge of adapting NAS for specific problems falls into a right delimitation of the search space. By doing this, the remote sensing prac-

titioners can reduce the amount of time needed to find a suitable model for their data and instead focus on other major tasks.

3 Methodology

To discover automatically generated high-performance architectures for satellite data classification tasks, we integrate the deep learning solutions for remote sensing in an AutoML framework. We propose to increase the efficiency of AutoML systems by reducing the complexity of the search space focusing on the most likely well-performance architectures for satellite data tasks.

We selected one of the deployed AutoML systems to build upon. We select Auto-Keras [12], an efficient NAS with network morphism, where Bayesian optimization is used to guide the exploration of the search space. The search space of Auto-Keras is based on network morphism, enclosing all architectures that can be created by morphing the initial architecture. The generation of the candidate architectures depends on the acquisition function of the Bayesian optimization. As the NAS space is not Euclidean, Auto-Keras uses an edit-distance neural network kernel for the Gaussian Process. This kernel measures the number of operations needed to morph one network into another one. It considers morphing the layers and the skip-connections. Different from fixed layer width methods [14], the morphism operations can change the number of filters in a convolutional layer and then make the search space larger. Therefore, finding a good architecture could take more time. By focusing on the most likely well-performance architectures for specific tasks, the searching time would be reduced.

To measure the benefits of the development of specific tasks for satellite data, we decided to gradually enhance the search space of the system and proposed three different settings for our experiments. Those settings and the motivation behind them are explained in the following subsections.

3.1 Original Auto-Keras System (V-AK)

Auto-Keras search space is built upon network morphism where the search space of NAS is created using morphism operations. An initial network architecture G is given and, with the use of morphism operations, new networks are created. Auto-Keras' authors use a three-layer convolutional network as starting architecture for the experiments presented in their paper to test the efficiency of their approach compared with other methods. However, the deployed Auto-Keras system has a task-oriented API, in which 3 different initial architectures are applied for the image classification task: first, it tries a vanilla network with 2 layers, second a ResNet50 model without pre-training, and thirdly an Efficientb7 network pre-trained with ImageNet. This change influences the possible architectures to select and outperforms the system initialized with a three-layer convolutional network. To the best of our knowledge, the selection of the initial architectures was based on human expert knowledge and state-of-the-art architectures for specific tasks based on natural image data.

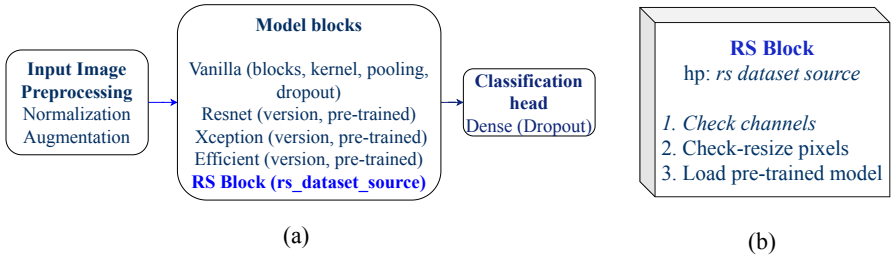


Fig. 2. (a) An abstract illustration of how the final architecture can be build based on pre-defined blocks. A network consists of one preprocessing block, several model blocks, and one classification head. V-AK and IMG-AK compose the model by using Vanilla, Resnet, Xception and/or Efficient blocks. In addition to these, RS-AK can make use of the RS Block as well. (b) The RS Block, only available to RS-AK. It can be extracted from various different remote sensing datasets, which is controlled by the hyperparameter $rs\_dataset\_source$

3.2 Models Pre-trained Using ImageNet Dataset (IMG-AK)

Based on remote sensing research, we know that models pre-trained with ImageNet can lead to promising results for satellite data classification tasks [3]. The Auto-Keras search space already includes blocks with weights acquired by pre-training on ImageNet. However, the decision to use such blocks depends on the process of selecting new candidate architectures. It could be the case that, due to trials budget and the vast search space, these pre-trained architectures are not considered. Figure 2 provides an abstract illustration of how the final architecture for the image classification task can be build based on pre-defined blocks existing in Auto-Keras. The model blocks in which the ImageNet weights are available have a hyperparameter called *pretrained*, which defines whether or not a pre-trained version of the model will be used.

Therefore, in this approach, although we make use of the available pre-trained models in the current systems, we still modify the configuration of G by defining an initial architecture for the new specific task: satellite image classification. We expect to improve the classification results by starting the neural architecture search with a block pre-trained with ImageNet. The model block can be selected based on the remote sensing literature findings. As reviewed in Sect. 2, ResNet architectures have shown promising results in classification of satellite images in the literature (see, e.g., [9,15]). Thus, we configure the initial G with a ResNet block and we set the parameter *pretrained* to *true*.

3.3 Models Pre-trained Using Remote Sensing Datasets (RS-AK)

Transfer learning can be most successful when the source and target domain are similar [4,18,27]. Within the remote sensing community, there are models pre-trained with remote sensing datasets [18,25] but none of these are available yet in AutoML systems. Therefore, we proposed to incorporate this type of pre-trained models and customize the Auto-Keras image classification task for satellite data.

We need to initially decide what needs to be changed in Auto-Keras to be able to add this feature. The Auto-Keras task-oriented NAS approach can be inferred from the open-source deployed system. The image classification task builds an architecture based on pre-defined cells or blocks. These blocks can be divided into three categories: preprocessing, model, and classification head. For the preprocessing category, two blocks are considered: (i) normalization, which performs a feature-wise normalization on the data; and (ii) an image augmentation block, which can apply various methods including flipping, rotation, and translation. The addition of such blocks to the final architecture in Auto-Keras is treated as a hyperparameter. The model blocks represent all the possible cells that will shape the hidden layers of the network. Each block consists of parameterized modules of well-known CNNs with various hyperparameters to be tuned. The third category is the classification head block, which creates the output layer of the network based on the number of classes and the classification type. The only hyperparameter to tune in this block is a dropout value. The preprocessing steps correspond to the ones applied by the authors of our satellite datasets, and the classification head block does not need to be changed because the nature of the classification is the same as any image classification task. We only need to change the model blocks and how our new block (which we refer to as *RS Block*) will interact with the classification head block. Figure 2 is an abstract illustration of this. The *RS Block* first checks the shape of the input and resizes the pixels if necessary. It chooses between different pre-trained module versions (trained with satellite data). This choice is considered as another hyperparameter to tune. Hence, it uses the same hyperparameter tuner that is used for all the other blocks. The optimization method is explained next.

Hyperparameter Tuning. Different tuners can be used to determine which combination of hyperparameters will be sent for training in each trial during NAS. We used an oracle combining random search and greedy algorithm [12,21] presented inside of Auto-Keras. The hyperparameters are arranged by grouping them into several categories according to the level or functionality. The oracle tunes each category separately using random search. In each trial, it uses a greedy strategy to generate new values for one of the categories of hyperparameters and use the best trial so far for the rest.

Remote Sensing Pre-trained Models. Our *RS Block* is composed of modules of different satellite learning representations acquired from different pre-trained models. These pre-trained models were trained with 5 different satellite datasets (BigEarthNet [25], EuroSAT [9], RESISC-45 [2], So2Sat [30], and UC Merced [28]). Based on the number of spectral bands of the collected datasets we considered two types of pre-trained models: (i) 3-channels and (ii) 13-channels. Figure 3 shows the architectures and datasets used for pre-training. The 3-channel pre-trained models were taken from the publicly available models posted by [18]. Inspired by the findings in [15] and the selected architecture in [18], we decided to create in-domain representations for 13-channels datasets using ResNet architectures and training with the EuroSAT dataset [9].

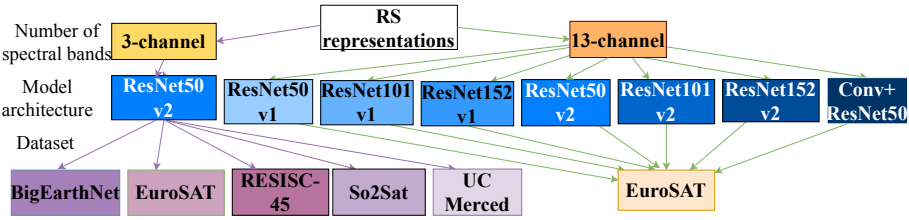


Fig. 3. Remote sensing pre-trained models considered for the RS Block. The first layer indicates the number of channels, the second layer the architecture used, and, the third layer the remote sensing dataset used. 3-channels models were created by Google Research [18], 13-channels were created by us.

To rapidly test the performance of our new block, we made two changes in the Auto-Keras search space. We first added the proposed *RS Block* to the model blocks structure. Secondly, we adapted the initial architecture G to start with our new remote sensing block. We would like to be able to study which of the remote sensing representations (pre-trained blocks) are used more often and, thus are more promising. We can inspect this, by studying the *rs_dataset_source* parameter of the *RS Block*, which indicates the source dataset used for pre-training in the case of the 3-channel datasets.

4 Experiments

In our evaluation we aim to address the following research questions:

- **Q1.** Can we achieve a performance similar to the non-automated deep learning research in remote sensing by using AutoML systems?
- **Q2.** How do different Auto-Keras variants perform for datasets with different characteristics (different number of spectral bands, sizes, and class distributions)?
- **Q3.** Which of the remote sensing pre-trained modules used in the RS-AK shows more promising results for developing NAS systems for remote sensing?

4.1 Datasets

To have a broader idea of the applicability of this framework in the remote sensing field, we have composed a benchmark of 7 diverse and well-known multi-spectral satellite datasets. Furthermore, this selection shows a variety of classification tasks with presumably different degrees of difficulty and complexity. Table 1 presents the characteristics of these datasets and summarizes the approach taken by their corresponding authors, as well as its performance. Except for the EuroSAT, So2Sat, and UC Merced datasets, the performance and approach showed in this table is the state-of-the-art (SOTA) considered for our experiments. For the case of these 3 datasets, better results are reported by the Google research

Table 1. Overview of available labelled datasets and the presented approach and performance from the paper in which the dataset was introduced.

Dataset	Satellite (Bands)	Resolution	Images	Labels	No.	Perf (%)	Approach
BigEarthNet	Sentinel-2 (3/12)	Med-high	590k (L)	Land	43	67.59	CNN 3-Conv[25]
BrazilDam	Sentinel-2 (13)	High	1.92k (S)	Dam?	2	94.1	DenseNet [6]
Brazilian Coffee	SPOT (3)	High	2.87k (S)	Coffee?	2	83.04	2 OverFeat networks [23]
Cerrado-Savanna	RapidEye (3)	High	1.31k (S)	Veg.	4	90.5	Fine-tuning AlexNet [19]
EuroSAT	Sentinel-2 (3/13)	High	27k (L)	Land	10	98.57	Pre-trained ResNet [9]
So2Sat	Sentinel-2 (3)	High	376k (L)	Land	17	61	ResNet [30]
UC Merced	USGS(3)	Very high	2.1k (S)	Land	21	NA	BoVW [28]

team in [18] using ResNet models pre-trained with remote sensing datasets; thus their results are the SOTA in Table 2.

The use of bands different from the RGB spectrum is a common practice in remote sensing applications due to the additional information that can be extracted from other spectral bands. A clear example is the creation of vegetation indexes for different applications; such indexes involve non-RGB channels like near-infrared. The number of samples available for training in remote sensing real-world problems is usually small. The Coffee scenes, BrazilDam, and Cerrado-Savanna datasets meet these characteristics. The Cerrado-Savanna scenes [19] is one of the most challenging datasets for classification. As explained by the authors, this is due to the high intraclass variance of the dataset, caused by different spatial configurations and densities of the same vegetation type, as well as its high inter-class similarity, caused by the similar appearance of different vegetation species [19]. Moreover, from 1,311 samples included in this dataset, 73% correspond to the Arboreal vegetation.

4.2 Experimental Setup

For all our experiments, the datasets were first randomly divided into train and test sets. The test set was created by reserving 20% of all the available data from Eurosat, BigEartNet, So2Sat, and UC Merced datasets. In the case of the BrazilDam dataset, only the Sentinel fold from 2019 was extracted to study. The Coffee scenes and Savanna datasets are originally divided into five folds. The first four were used for training and the last fold is reserved for testing. Next, another split of 80-20 was applied to the training set, assigning 20% of it for validation, which is used for the AutoML system to tune hyperparameters and select the best model. As most of the datasets are also used as a source for creating pre-trained models, when evaluating RS-AK, we should be careful not to include pre-trained blocks from the dataset that we want to use to test on, to avoid being exposed to labels from the test set. As such, when evaluating on a given dataset, we remove the pre-trained blocks coming from this dataset from the search space. To exclude the corresponding dataset, before running the task, we keep out this option from the set of pre-trained models available for the *rs_dataset_source* hyperparameter in the *RS Block*.

To be able to show the significance of the results we performed a Wilcoxon signed-rank test, first ensuring that the data was not normally distributed and considering a p value of 0.05. The outcomes presented in this paper are based on the 10 trials experiments. Each trial, varying per dataset, ranges from few minutes to around 6 h. All the experiments were run on a compute cluster using nodes with 4 GPUs (PNY GeForce RTX 2080TI). We delimited the memory to 32 and 64 GB for the experiments. For better reproducibility, we have made the source code of our experiments available in a public repository.[1]

5 Results

In this section we will answer the research questions that were stated in Sect. 4.

5.1 AutoML vs Non-automated Models

Table 2 summarizes the performance of the three different AutoML approaches on the test set for the different datasets. The performance metric shown here, same as the baseline papers, is the overall classification accuracy. For the BigEarthNet-rgb dataset, we decided to change the performance metric to be able to compare with the baseline. We achieved an F1-score of 67.84% using an ImageNet pre-trained module, while the result presented in [25] is 67.59%. There is no benchmark performance available for the full spectral version of EuroSAT. Resultant of our experiments, we established one with 97.8% overall accuracy.

To answer **Q1** we grouped the results of the three variants (V-AK, IMG-AK, and RS-AK) and we took the maximum performance. In this way, we can analyze the AutoML competency against the non-automated architectures. We outperformed the literature in 5 out of 7 datasets, improving the state-of-the-art result for So2Sat by a rate of 34.5%. Therefore, we can conclude that the performance found by using AutoML systems can be competitive and even better for some of these datasets.

5.2 AutoML Variants and the Different Type of Datasets

To address **Q2**, we group our datasets based on size, number, and type of spectral bands (channels). We consider four small datasets. We have 2 datasets with 13 channels (BrazilDam and EuroSAT-all) and 6 with 3 channels. The 3-channels are either RGB bands or near-infrared, green, and red bands. Note that the EuroSAT-all dataset has an empty entry for the SOTA and RS-AK approach. Since the pre-trained blocks from the 13-band dataset come all from the EuroSAT-all dataset, we could not fairly deploy this model (see experimental setup). To facilitate comparisons with the SOTA found in literature and among our experiments, in Table 2 the boldfaced entries indicate the best approach among the 3 Auto-Keras variants.

[1] https://github.com/palaciosnrps/automl-rs-project.

Table 2. Performance on test dataset considering 10 runs (Except for BigEarthNet which had 3 runs) of each of our experiments and the state-of-the-art (SOTA) found in literature for each dataset. BigEarthNet performance metric is F1-score, all the other datasets use overall accuracy. An asterisk (*) represents statistically significant results.

Dataset	Type	SOTA	V-AK	IMG-AK	RS-AK
BrazilDam	Small-13	94.1 [6]	**89.09 ± .05**	76.54 ± .13	85.57 ± .01
Coffee scenes	Small-3	83.4 [23]	86.18 ± .02	82.96 ± .04	**88.84 ± .00***
Cerrado-Savanna	Small-3	90.5 [19]	85.79 ± .01	84.33 ± .03	**89.92 ± .01**
UCMerced	Small-rgb	99.61 [18]	**99.62 ± .00**	76.43 ± .13	91.19 ± .06
EuroSAT-all	Large-13	–	95.38 ± .02	**97.82 ± .00***	–
EuroSAT-rgb	Large-rgb	99.2 [18]	99.18 ± .00	**99.54 ± .00***	95.90 ± .01
So2Sat-rgb	Large-rgb	63.25 [18]	95.47 ± .00	**97.80 ± .00***	76.92± .00
BigEarthNet-rgb	Large-rgb	67.59 [25]	50.62 ± .00	**67.84 ± .00**	65.29 ± .00

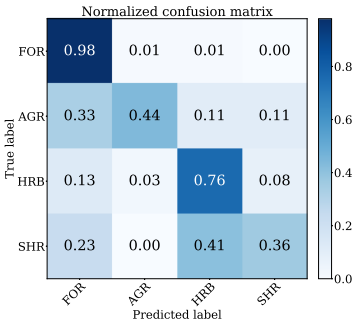
If the results are statistically significant to both other approaches according to the Wilcoxon Signed rank test the entry is marked with an asterisk (*). Please note that the paired comparison of second-best approaches is not shown in the table.

The original Auto-Keras V-AK and the IMG-AK version performed well on the EuroSAT-all dataset. In this case, IMG-AK performs better than V-AK. For the case of the BrazilDam dataset, the initialization with a pre-trained ImageNet model did not benefit the performance (see IMG-AK Table 2) and it even decreased the average accuracy. This can be explained considering the difference in the number of input channels (increasing the complexity) and the size of the dataset. BrazilDam dataset has 13 channels; therefore, the direct use of pre-trained models from ImageNet (3-channel) does not apply. Different from EuroSAT, the number of labeled samples of BrazilDam is small. We can notice an improvement using RS-AK but this is not enough to beat the baselines.

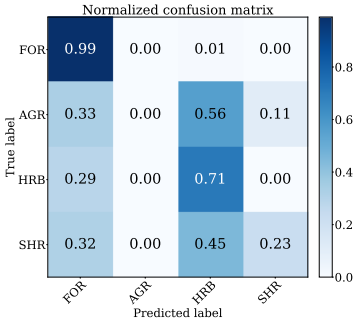
We can see that for the RGB channel datasets either V-AK or IMG-AK approaches lead to the best performance. We achieved a large improvement for the So2Sat-rgb dataset, compared to the work presented in [18]. Even though the authors of [18] also used pre-trained models, the variety of model versions and the more sophisticated hyperparameter tuning method provided by the AutoML systems played an important role in achieving better performance for this dataset. Conversely, the RS-AK variant obtained the best results for the Coffee scenes and Cerrado-Savanna datasets. These two datasets are composed of near-infrared, green, and red bands and the classification task differs from land cover identification. Based on that, we can infer that the 3-channel remote sensing representations are an option for transfer learning when the target dataset is different from the well-known RGB channel datasets. In the case of the 13-channel representations used for the BrazilDam dataset, the results were not as successful as what was obtained by manually designed architectures. The best-automated model generated using the original Auto-Keras consists of convolutional blocks without pre-trained modules, suggesting that for this dataset training from scratch rather than using the available pre-trained models is a better approach. Based on the results

of the non-RGB datasets, we can expect that improving the 13-channel representations could lead us to better performance.

Considering the dataset size, we notice that comparing the initialization of G with ImageNet pre-trained models (IMG-AK) versus the implementation of remote sensing pre-trained models (RS-AK), RS-AK gives better performance for the small datasets. Meanwhile, IMG-AK consistently results in better performance for large datasets. This could be explained by (i) the amount of data available for pre-training and (ii) the degree of similarity between the target and source domains that both determine the quality of the transfer-learning technique [24,27,29]. Bigger datasets should produce better representations. But data similarity also needs to be taken into account. It is possible that for the classes represented in the small datasets the current remote sensing representations are enough and the best performance is acquired, as the domain source is similar. However, in the case of the large datasets the quality of the representations generated with the ImageNet dataset (being over 2 times bigger than the BigEarthNet dataset) gain over the domain similarity. To improve the performance of classification for the bigger datasets using RS-AK, more studies are needed and some of those should investigate different fine-tuning strategies and improving the performance of the BigEarthNet representation, which so far is the most promising one.



(a) Confusion matrix for the Cerrado-Savanna dataset using a pre-trained remote sensing block.



(b) Confusion matrix, Cerrado-Savanna dataset using only convolutional blocks (no pre-trained versions).

	FOR	AGR	HRB	SHR
Precision	0.94	0.57	0.71	0.67
Recall	0.98	0.44	0.76	0.36

	FOR	AGR	HRB	SHR
Precision	0.90	0.00	0.61	0.83
Recall	0.99	0.00	0.71	0.23

Fig. 4. Comparison of confusion matrices for Cerrado-Savanna dataset. Classes are Agriculture (AGR), Arboreal Vegetation (FOR), Herbaceous Vegetation (HRB) and Shrubby Vegetation (SHR).

The overall accuracy only gives a general idea of the performance, for datasets in which the samples per class are not balanced we need to look with more detail into the performance achieved for each class to know if there is still any room for improvement. We generate confusion matrices to inspect the performance in more detail. Figure 4a is the confusion matrix of the best model found for the Cerrado-Savanna dataset by using RS-AK. The classes with originally more samples (FOR, HRB) are the classes with better performance. For the SHR and AGR classes, the misclassification is still high. However, while comparing with the results given by using a non-pre-trained model obtained with V-AK (Fig. 4b), we can appreciate a big improvement of 13% and 44% in the less representative classes (SHR, AGR) acquired by the use of pre-trained blocks.

Table 3 summarizes the findings of the confusion matrices for datasets with a major difference in the distribution of class samples. To measure the impact of pre-trained blocks, in this table, we compare the performance achieved for the minority and majority classes, with and without pre-training. We notice that while using pre-trained blocks, the recall of the least representative classes in all datasets increases between 7% and 44 % while the values for majority classes slightly decrease between 1% and 9%. However, the overall accuracy is impacted more by the majority class, ignoring the large improvements on the minority classes. For remote sensing applications in which the class distribution is non-balanced, this improvement for the minority class is important.

5.3 The Remote Sensing Block RS-AK

In this section, we aim to address **Q3**. Figure 5 shows the frequency at which each source model was selected as part of the customized block for each dataset. For the Savanna Cerrado, Coffee scenes, and So2Sat datasets the most chosen pre-trained model was BigEarthNet. So2Sat was the most selected model in the case of UC Merced dataset and it tied with BigEarthNet for the EuroSAT dataset. These results are expected due to the big size of the datasets but differ from the findings of [18] who conclude that the RESISC-45 representation achieves the highest performance. We found the RESISC-45 representation to achieve the best results only when used for classification on the BigEarthNet dataset. Our experiments differ in the way we are using a more efficient framework for tuning a large set of possible hyperparameters (including learning rate, optimizer, regularization, preprocessing) and selecting the design choices using an oracle combining random search and a greedy algorithm (explained in Sect. 3.3) while the authors of [18] optimize by sweeping only a fixed set of hyperparameters (learning rate, weight decay, training schedules, preprocessing). The authors of [18] utilized the same ResNet50V2 architecture [8] to fine-tune the remote sensing datasets using SGD with momentum set to 0.9, in our approach the pre-trained model is only a block that is part of the full architecture (see Fig. 2). In [18], the comparison of the different pre-trained models was made after finishing the fine-tuning using partial (100, 1000) and full training samples; in our study, the selection of the best-performed model was based on the validation set inside the Auto-Keras framework. Considering that, we believe that our experiments have exploited the potential of each

Table 3. Recall value of the classes with most and least samples for the non-balanced datasets.

Dataset	Class	Non-pre	Pre-trained
Cerrado	Majority (73.6%)	0.99	0.98
savanna	Minority (3.4%)	0.00	0.44
So2Sat	Majority (12.3%)	0.95	0.99
	Minority (0.6%)	0.76	0.94
BrazilDam	Majority (57.9%)	0.95	0.86
	Minority (42.1%)	0.78	0.85

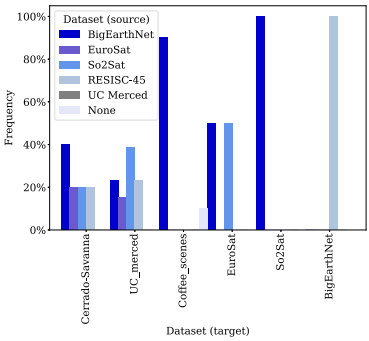


Fig. 5. Remote sensing pre-trained models selected for the 3-channels datasets during the 3rd experiment.

dataset representation by using a more sophisticated framework for the design of the architecture and the hyperparameter tuning; moreover, our results are consistent with the expectations of the remote sensing community about the promising applications of BigEarthNet on remote sensing tasks [25].

6 Conclusions and Future Work

We demonstrated how AutoML can be used to leverage the implementation of deep learning models for satellite data tasks, outperforming some state-of-the-art research results. We focused on classification tasks for multi-spectral satellite datasets. We assessed the performance of the original Auto-Keras [12] (V-AK) and modified its search space to create two different variants of its image classification task: (i) initializing the architecture to morph with a model pre-trained on ImageNet (IMG-AK) and (ii) adding models pre-trained on well-known remote sensing datasets (RS-AK) such as BigEarthNet and UC Merced. Our experimental results on a varied selection of satellite datasets showed that for 3-channel datasets, current AutoML systems can beat state-of-the-art results for land cover classification tasks. Analyzing the performance of the two Auto-Keras variants initialized with pre-trained blocks (IMG-AK and RS-AK), we noticed that RS-AK performed better for small datasets meanwhile IMG-AK was best for relatively large datasets. Moreover, we showed that these pre-trained versions exhibit superior performance on minority classes. The use of bands different from RGB is a common practice in remote sensing due to the extra spectral information that can be extracted from such bands. Besides, the amount of samples available for training in remote sensing real-world problems is often small. Our remote sensing block achieved the best results in such situations. This highlights the usefulness of a customized satellite data search space in AutoML systems for real-world datasets. The 13-channel pre-trained models can be downloaded and used for other remote

sensing tasks; due to the number of channels these models are useful when working with Sentinel-2 satellite images. There is still room for improvement in such remote sensing representations. In future work, we will first aim at improving the transferability of the remote sensing pre-trained models and work on covering the widely used image segmentation task. A more sophisticated transfer learning method, deep meta-learning [10], or customized techniques per dataset & task (based on [24,29]) integrated into AutoML systems could improve the usage of remote sensing data representations. Based on our experiments, we recommend the remote sensing practitioners to make use of the existing open-source AutoML tools. By making this framework publicly available, we enable the community to further experiment with relevant remote sensing datasets and expect to expand the use of AutoML for different applications.

Acknowledgement. This work is partially supported by PAME, a Dutch Research Council (NWO) project under grant number OCENW.KLEIN.425 and it has been performed using the ALICE & GRACE compute resources provided by Leiden University.

References

1. Basu, S., et al.: DeepSat: a learning framework for satellite imagery. In: 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (2015)
2. Cheng, G., et al.: Remote sensing image scene classification: benchmark and state of the art. Proc. IEEE **105**(10), 1865–1883 (2017)
3. Cheng, G., et al.: Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities. IEEE J. Sel. Topics Appl. Earth Observations Remote Sens. **13**, 3735–3756 (2020)
4. Cui, Y., et al.: Large scale fine-grained categorization and domain-specific transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4109–4118 (2018)
5. Falkner, S., et al.: BOHB: robust and efficient hyperparameter optimization at scale. In: 35th International Conference on Machine Learning, pp. 2323–2341 (2018)
6. Ferreira, E., et al.: BrazilDam: a benchmark dataset for tailings dam detection. In: Latin American GRSS & ISPRS Remote Sensing Conference, pp. 339–344 (2020)
7. Feurer, M., et al.: Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems, vol. 28, pp. 2962–2970 (2015)
8. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
9. Helber, P., et al.: EuroSat: a novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Topics Appl. Earth Observations Remote Sens. **12**(7), 2217–2226 (2019)
10. Huisman, M., van Rijn, J.N., Plaat, A.: A survey of deep meta-learning. Artif. Intell. Rev. 1–59 (2021). https://doi.org/10.1007/s10462-021-10004-4
11. Hutter, F., et al.: Automated machine learning: Methods, systems, challenges. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05318-5
12. Jin, H., et al.: Auto-keras: an efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946–1956 (2019)

13. Koh, J.C., et al.: Automated machine learning for high-throughput image-based plant phenotyping. Remote Sens. **13**, 858 (2021)
14. Liu, H., et al.: DARTS: differentiable architecture search. In: Proceedings of ICLR (2019)
15. Mahdianpari, M., et al.: Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sens. **10**(7), 1119 (2018)
16. Marmanis, D., et al.: Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. **13**(1), 105–109 (2016)
17. Mendoza, H., et al.: Towards automatically-tuned deep neural networks. In: AutoML: Methods, Systems, Challenges, vol. 7, pp. 141–156 (2018)
18. Neumann, M., et al.: Training general representations for remote sensing using in-domain knowledge. In: 2020 IEEE International Geoscience and Remote Sensing Symposium (2020)
19. Nogueira, K., et al.: Towards vegetation species discrimination by using data-driven descriptors. In: 9th IAPR Workshop on Pattern Recogniton in Remote Sensing, pp. 1–6(2016)
20. Olson, R., et al.: Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of GECCO 2016, pp. 485–492 (2016)
21. O'Malley, T., et al.: Keras Tuner (2019). https://github.com/keras-team/keras-tuner
22. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
23. Penatti, O.A., et al.: Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 44–51 (2015)
24. Soekhoe, D., van der Putten, P., Plaat, A.: On the impact of data set size in transfer learning using deep neural networks. In: Boström, H., Knobbe, A., Soares, C., Papapetrou, P. (eds.) IDA 2016. LNCS, vol. 9897, pp. 50–60. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46349-0_5
25. Sumbul, G., et al.: BigEarthNet: a large-scale benchmark archive for remote sensing image understanding. In: IEEE International Geoscience and Remote Sensing Symposium, pp. 5901–5904 (2019)
26. Thornton, C., et al.: Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 847–855 (2013)
27. Yang, F., et al.: Transfer learning strategies for deep learning-based PHM algorithms. Appl. Sci. **10**(7), 2361 (2020)
28. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279 (2010)
29. Yosinski, J., et al.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, vol. 27, pp. 3320–3328 (2014)
30. Zhu, X., et al.: So2Sat LCZ42: a benchmark data set for the classification of global local climate zones. IEEE Geosci. Remote Sens. Mag. **8**(3), 76–89 (2018)