Universiteit Leiden

Master Computer Science

Automated Machine Learning for Satellite Data: Integrating remote sensing pre-trained models into AutoML systems

Name:	Nelly Rosaura Palacios Salinas
Student ID:	s2105713
Date:	25/02/2021
Specialisation:	Advanced Data Analytics
Supervisor:	Mitra Baratchi (LIACS)
Supervisor:	Jan N. van Rijn (LIACS)
Supervisor:	Andreas Vollrath (External)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Creating machine learning models from satellite data is useful for various applications ranging from environmental mapping and monitoring to urban planning and emergency response. However, many remote sensing practitioners deal with a knowledge gap in taking full advantage of the advancements in machine learning. Automated Machine Learning (AutoML) addressed this issue by allowing the creation of high-performing models through automatically making machine learning design choices in a data-driven manner. Current AutoML systems have been benchmarked with traditional natural image datasets, which differ from satellite images in various aspects. These differences open questions about the applicability of current AutoML systems for satellite data tasks. In this project, we studied how AutoML can be applied to satellite data by examining the deployed AutoML system of Auto-Keras and creating two new variants of its image classification task. These variants integrate discoveries made in the remote sensing research field by using models pre-trained with ImageNet and remote sensing datasets. We compared the performance against high performed, manually designed, architectures on a varied set of 7 satellite datasets. Our results show that in 66% of the cases the AutoML systems outperformed the remote sensing literature. In 5 out of 7 cases the pre-trained variants got better performance than the original Auto-Keras image classification task. Our new variant, using remote sensing pre-trained models, performed better than the ImageNet variant for small datasets and found the best-automated model for the datasets with 3 channels different from RBG. The subset of such different channels is a common practice in the remote sensing community and the number of samples available for training in remote sensing real-world problems is small. This highlights the usefulness of a customized satellite data search space in AutoML systems.

Contents

Chapter 1

Introduction

The rapid and large increase of geospatial and temporal data, in addition to new technologies, emphasizes the need for automated discovery of spatio-temporal knowledge [1]. Within the field of remote sensing, satellite data gained a massive boost throughout the last years due to the opening of imagery archives to the public.

Satellites continuously monitor the Earth's surface and provide information on the state and health of the planet that would not be possible by any other means. Its applications range from environmental mapping and monitoring to urban planning, natural risk assessment, emergency response, military reconnaissance and many more [2, 3, 4]. The most used data among researchers and practitioners are coming from Landsat and Sentinel missions, which have multi-spectral instruments.

Concepts of computer vision and machine learning are commonly used by remote sensing practitioners. However, they do not always leverage the latest developments within the field due to the difference in disciplines. Publications in remote sensing have mainly focused on classification tasks. Deep neural networks are state-of-the-art techniques for most image classification tasks [5]. Currently, there are two obstacles limiting the use of deep learning for satellite data. The first one is the lack of sufficient labeled data. Techniques such as transfer learning [6] and meta-learning [7] can help to overcome this limitation. The transfer learning approach re-uses the knowledge gained from previously seen tasks and applies it to a newly created model in another task [8]. Last year, the authors of [9] showed the potential of using pre-trained models for remote sensing datasets. The second obstacle lies in the difficulty of designing appropriate architectures that take the peculiarities of satellite images that differ considerably as compared to natural images into account. Creating new high-performing models for satellite data would require designing new architectures taking into account such differences and facing a large set of design choices, such as the hyperparameters. The solution to overcome this obstacle can be making such important decisions in a data-driven and automatic way using Automated Machine Learning (AutoML) [10].

Open AutoML systems [11, 12, 13] have been deployed to make machine learning available for non-machine learning experts. Most of those systems focus on shallow machine learning (modeling techniques as Random Forest, Support Vector Machines, K-neighbors). But, based on a literature review in remote sensing and previous experiments, the use of deep learning for remote sensing, especially CNNs, gives the most promising results. On the subject of deep learning, Neural architecture search (NAS) aims to find the best neural network architecture given a task and a dataset. NAS has also become an effective computational tool in AutoML.

To the best of our knowledge, in the field of computer science, NAS research has been benchmarked with natural image datasets but not for satellite images. In the field of remote sensing, the research community has some findings that if we combine them with the advances in AutoML this could leverage the implementation of deep learning for satellite data in a more efficient way. Bringing us the questions: *what is the performance of current AutoML systems for satellite data?* and *what happens when we integrate the knowledge gained from previous research in the field of remote sensing into*

AutoML systems?

In this thesis, we propose to answer these questions by (i) composing a benchmark of diverse satellite datasets, (ii) evaluating the performance of such datasets in a deployed AutoML system, and finally, (iii) designing a satellite data tasks-oriented NAS and evaluating the datasets on it.

To design a satellite data tasks-oriented NAS, we can tailor the neural architecture search space of current systems by integrating findings of the remote sensing field. Specifically, the use of models pre-trained with ImageNet and remote sensing datasets is considered.

We know that positive results in specific applications are based on human priors, the results presented in [14] regarding task-oriented NAS can be an example of it. The state-of-the-art in deep learning for remote sensing applications includes the use of pre-trained models. Our previous experiments shown how using AutoML methods for tuning deep learning hyperparameters can improve the performance of the model. Therefore, we expect that by integrating the remote sensing pre-trained models into AutoML systems we can obtain good performance and help to bridge the gap between the use of satellite data with state-of-the-art methods from the field of machine learning.

Instead of building a completely new framework to test our approach, we can build upon one of the existing ones. We use Auto-Keras [14], a framework enabling Bayesian optimization to guide the network morphism for efficient NAS. We focus on multi-spectral satellite datasets and classification tasks. In summary, our contributions are the following:

- Assessing the performance of the deployed Auto-Keras system within the context of remote sensing data and propose 2 new variants by exploiting transfer learning (from natural and satellite images).
- Extending the use of transfer learning from models trained with remote sensing data in an automated framework, passing on special domain data representations.
- Improving the efficiency of AutoML systems for satellite data by reducing the search space using a customized initial architecture inspired from the state-of-the-art remote sensing research.
- Evaluating the performance of the three variants of Auto-Keras by comparing them against high-performance manually designed architectures and achieving the benchmark performance for the 13-channels EuroSAT dataset.
- Making the resulting framework and source codes publicly available (with the possibility to be integrated into Auto-Keras), allowing the AutoML system to be adopted by the remote-sensing community for creating high-performing models on available satellite data for a variety of applications. [1]

The organization of this thesis is the following. Chapter 2 presents the background information and Chapter 3 the related work regarding deep learning in the remote sensing field. In Chapter 4, we present the methodology and the three proposed AutoML variants. In Chapter 5 the experiments are explained and in Chapter 6 the results are presented. Last, in Chapter 7 we discuss the concluding remarks as well as the future work of this study.

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

Chapter 2

Background Information

In this chapter, we will present the definitions regarding remote sensing, transfer learning, and automated machine learning. Part of this background information and literature review was made during a previous Introductory research project.

2.1 Remote Sensing and Satellite Data

Remote sensing is the acquisition of information about an object or phenomena without the instrument used to collect the data being in direct contact with the object [16]. In our context, it refers to the use of satellites to detect and classify objects on Earth through images.

Satellite instruments acquire data from various ranges of frequencies along the electromagnetic spectrum. The electromagnetic spectrum, illustrated in Figure 2.1, makes reference to the entire range of existing light. Light is a wave of alternating electric and magnetic fields that can be visible or invisible to the human eye. One important characteristic of the light is wavelength, which is the distance from the peak of one wave to the peak of the next wave. The visible light (to humans) wavelengths ranges from 380 to 700 nanometers. A satellite image combines measurements of different light wavelength ranges, both visible and invisible to humans [?]. When the number of spectral bands ranges from 3 to 20 or each band has a descriptive title this data is known as **multi-spectral**. When there are hundreds of bands without descriptive names, the data is called hyper-spectral [17]. Currently, researchers and practitioners used data mainly coming from Landsat and Sentinel missions, which provide multi-spectral information.

Earth's atmosphere absorbs and transmits various wavelengths of electromagnetic radiation. Carbon dioxide, water vapor, and other trace gases in the atmosphere absorb the longer wavelengths. Figure 2.2 shows the different band designations for Landsat 7, Landsat 8, and, Sentinel-2 satellites. The Sentinel-2 data has spectral bands similar to Landsat 8. The X-axis is the wavelength in nanometres and the Y-axis is how much the electromagnetic radiation will penetrate the atmosphere. Satellites bring instruments for land, ocean, and atmospheric monitoring, hence, a multi-spectral satellite image captures information of the electromagnetic spectrum related to it.

Satellite imagery is characterised by 4 types of resolutions [17]. Dependent on the satellite's instrument, all four parameters will vary as there are trade-offs to be taken into account when conceptualizing a satellite and its instrument. Those resolutions are:

- Spatial resolution. It refers to the size of the pixel on the Earth's surface and ranges from a few centimeters for very-high-resolution (VHR) satellites up to a kilometer for low-resolution.
- Temporal resolution. It is given by the revisit time, which is the time elapse before the satellite

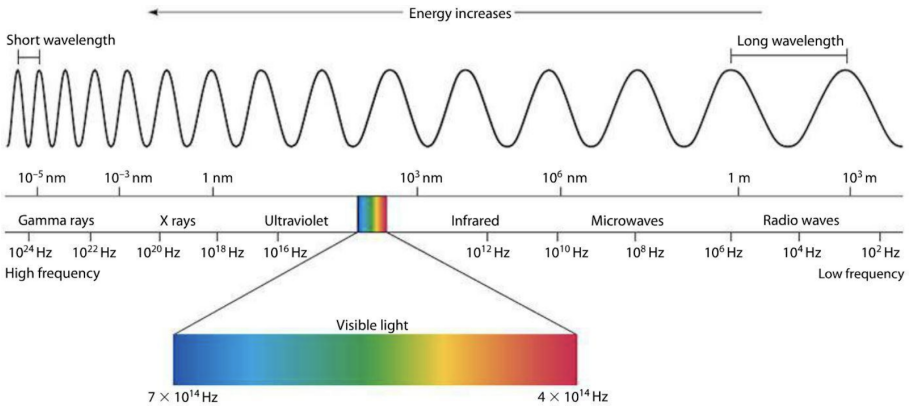


Figure 2.1: The Electromagnetic Spectrum: Remote sensing instruments are designed to capture specific ranges of the electromagnetic spectrum. [15]

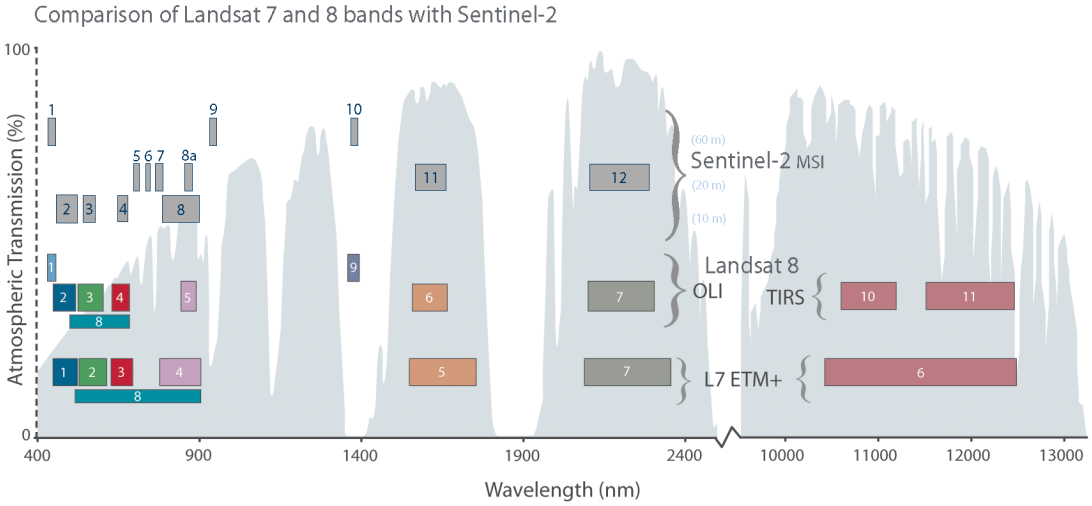


Figure 2.2: Comparison of Landsat 7 and 8 bands with Sentinel-2 [18].

captures the same point again. It also varies from satellite to satellite, from daily revisit time up to 2 weeks for Sentinel satellites.

- Spectral resolution. It refers to the spectral width of each band in the dataset. Often, a general resolution of the sensor is provided but not all sensors collect information within bands of uniform widths.
- Radiometric resolution. It is how finely the radiance received in each band is distinguished. It is expressed as the number of bits for each band. 8-bits resolution was common in satellites but newer sensors can generate 16 and 32 bits data resolution.

Different characteristics, resolutions, and types of satellite instruments create a big difference between natural color images and satellite images. These differences were listed in [9] and are the following:

- Natural images have an 8 bits precision meanwhile remote sensing input data usually comes at higher precision (16 or 32 bits).
- Natural color images are always the same 3 channels (RGB) but for satellite images, the number and type of channels are variable, depending on the satellite instrument.
- Different from camera photos, in satellite images the range of values, varies largely from dataset to dataset and between channels. The values distribution can be highly skewed.
- Many quantitative remote sensing applications rely on the absolute values of the pixels.
- Lower resolution satellite data can aggregate a lot of information about the illuminated surface in a single pixel, whereas natural image coverage is small.
- Satellite image axes might be non-standard, for example representing range and azimuth dimensions.

Therefore, the pre-processing of satellite images and their model construction requires special attention. Many hyperparameters are involved in such steps, especially when deep learning models are applied. We propose to make use of the advances in automated machine learning to provide data-driven decisions during this process.

2.2 Transfer Learning

In many machine learning algorithms, when the training and the data to predict are part of the same feature space and have the same distribution the results are impressive. However, in many real-world applications, this scenario is not always possible. Either because the data required is outdated or because the training data is not sufficient knowledge transferring is a useful approach in such cases. Transfer learning consists of re-using the knowledge gained from previously seen tasks to be applied to a newly created model in another task [8]. Different methods have been developed to answer questions regarding what to transfer', 'when to transfer', and 'how to transfer'. In the context of deep learning, **pre-trained** networks are the basic form of transfer learning. The two most popular approaches are: (i) using pre-trained models as feature extractors; (ii) fine-tuning pre-trained models. The main idea of using pre-trained models as feature extractors is to only replace the top layer of the source model and use it with the new data without updating the model weights during training. The fine-tuning strategy does not only replace the final layer, it also does a retrain of previous layers.

2.3 AutoML and Neural Architecture Search

AutoML is the process of automating the different stages of a machine learning pipeline. Figure 2.3 presents the steps in a machine learning pipeline. These steps typically are: data collection, data preparation, feature engineering, preprocessing, algorithm selection, parameter optimization, model training, and deployment [10]. The most common and most developed task is to automatically set hyperparameters to optimize the performance of an algorithm for a given dataset.

Current AutoML systems commonly cover from data preparation to model training phase [10]. Auto-Sklearn [11], Auto-WEKA [12] and T-POT [13] are examples of AutoML systems focusing on traditional machine learning (such as SVM, Random Forest, K-neighbors). In deep learning, Neural Architecture Search (NAS) is one of the most challenging tasks. It is equivalent to the algorithm selection step in classic machine learning and aims at finding the best neural network architecture

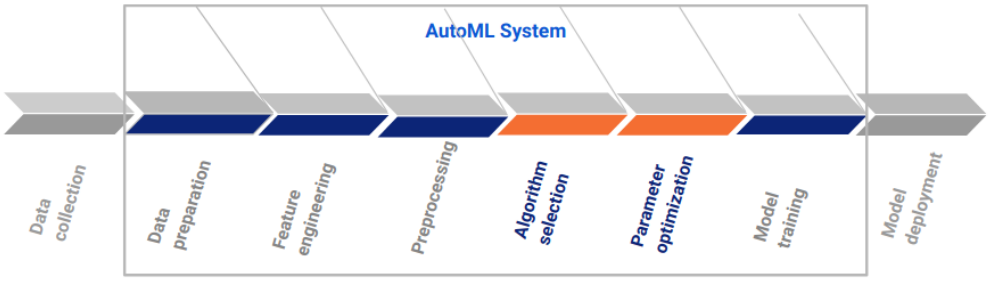


Figure 2.3: A Machine Learning pipeline and the common AutoML system. [10]

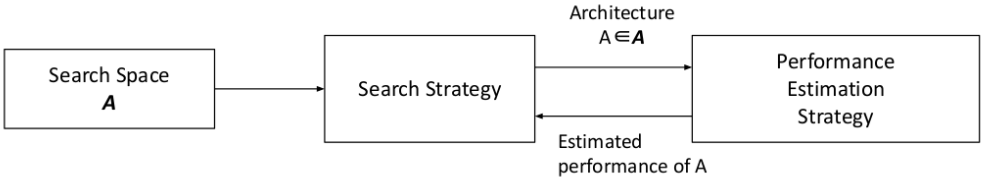


Figure 2.4: Abstract illustration of NAS. A search strategy selects an architecture A from a predefined search space A. The architecture A is passed to a performance estimation strategy, which returns the estimated performance of A to the search strategy [10]

for a given task and dataset [5]. NAS consists of the following steps: (i) using a search strategy to select an architecture from a predefined search space; (ii) evaluating the selected architecture using a performance estimation strategy; (iii) returning the estimated performance that can be utilized for the search strategy in order to select the next promising promising architecture. This loop ends until certain conditions are met. Figure 2.4 illustrates this process. Different methods have studied for each of the elements in the figure (search space, search strategy, performance estimation strategy). The search space determines all the possible architectures that can be discovered by NAS. Techniques to explore the space of neural architectures involving reinforcement learning [19, 20] and evolutionary algorithms [21, 22, 23] have achieved extraordinary performance at the cost of a large amount of computational resources. To address the need for computational efficiency, the authors of [24, 25] have considered reducing the complexity of the search space by proposing to search for motifs, called cells or blocks, rather than for whole architectures. By stacking these blocks in a predetermined mode the final architecture is created. Even with such a reduced search space, one of the biggest challenges of applying NAS in practice is still the computational time needed to find a good architecture for the following reason: 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. Training a neural network from scratch is costly. Therefore, an approach to speed up the training is to initialize the weights of new architectures based on the weights of another architecture. In order to apply this transferability of learned weights, the concept of network morphism in the context of neural networks was introduced by [26] using a parameter-transferring map from a parent network to a child network. The child network keeps its function and outputs. Based on this, the authors of [27] formalized the network morphism operations and presented a method to automatically search for CNN architectures. Continuing with the goal of creating efficient NAS methods, [14] proposed Auto-Keras, an efficient NAS with network morphism, where Bayesian optimization is used to guide through the search space.

It makes use of a neural network kernel and a tree-structured acquisition function to efficiently inspect the search space. Incorporating prior knowledge about properties well-suited for a task can reduce the size of the search space and simplify the search [10], but certainly, it will incorporate a human bias that would delimit the possible architectures. Our goal is to make high-performing deep learning models more accessible to remote sensing practitioners rather than discover novelty architectures, therefore this human bias is less relevant. We will use this approach of incorporating prior knowledge taken from remote sensing research literature in the creation of a customized search space for satellite data tasks.

Chapter 3

Related work

Scene classification is the basic problem in remote sensing. The satellite image scene classification task goal is to correctly label the given images with predetermined categories. Common problems while working on scene classification are related to the big intraclass diversity, low between-class separability, and large variance of scene scales represented. Over the years, three approaches have been studied: pixel-level, object-level, and scene-level classification. When the spatial resolution is very low, a pixel-level approach is suitable due to the semantic meaning carried by to the size of the pixel, which corresponds to the sizes of the objects. However, in present years the benchmark datasets have a higher resolution so pixel-level approaches have been replaced with scene-level approaches, where deep learning proclaim promising results.

3.1 Deep Learning and Remote Sensing

The authors of [28] categorize the different models applied for remote sensing into three types: statistical models, physical models, and data-driven models. To apply statistical models considerable level of prior knowledge is required, such models use digital signal processing methods. The physical models are based on physical approximations made by analytic computation methods and also require a good level of knowledge about the problem. Data-driven models, as inferred from name, are based on the knowledge extracted from the data; consequently only low prior knowledge is required. According to [4], deep learning techniques started to be used for remote sensing in 2014. In 2017, large-scale benchmarks for satellite image scene classification were released [29, 30], which open the door for the development of more deep learning-based models.

In [31], more than 200 publications in the field of remote sensing were reviewed and meta-analysed showing that the main focus of previous research is on classification. The authors of [4] reviewed the state of deep learning for satellite image scene classification. CNN-based methods have obtained impressive results when numerous annotated samples to fine-tune or train a network from scratch are available.

AlexNet [32], DenseNet [33], VGG [34] and ResNet [35] are some of the most powerful CNN architectures in computer vision tasks. The findings of [36], after examining seven deep convolutional networks suggest that an ensemble of Inception and ResNet modules is an effective architecture for land cover classification.

Transfer learning

Supervised deep learning normally have the need of variety and volume of data for training. However, for a majority of remote sensing applications the available training data is limited. Therefore, different approaches have been applied. Three commonly used techniques are: transfer learning, unsupervised learning, and generative adversarial networks.

The authors of [37] studied the performance of CNNs for satellite image scene classification using datasets containing a few annotated samples. They analyse different learning strategies: full training, fine-tuning, and using CNNs as feature extractors. According to their conclusions the fine-tuning approach tends to be a good option in different scenarios.
AutoML have made advances using meta-learning for transferable models. Meta-learning, is the science of learn from the experience gained by looking on how different machine learning approaches perform on a wide range of learning tasks (meta-data) to learn new tasks much faster than other possible models [38]. Based on [7], last year the model agnostic meta-learning (MAML) algorithm was put on the context of few-shot land cover classification [39]. The results indicate that this approach could be preferable only when the data has a high degree of diversity from region to region.
Emphasizing the importance of transfer learning for remote sensing, the authors of [9] investigated the in-domain representation learning to develop generic remote sensing representations. Their experimental results in 5 remote sensing datasets suggest that representations trained on the large weakly-supervised datasets were not as successful as that of a smaller and more diverse human-curated dataset. However, some results left matters open and need more examination. The best trained in-domain representations are published for easy reuse by the public.

Hyperparameter tuning
To the best of our knowledge, studies in remote sensing have only considered optimising a subset of hyperparameters using a parameter sweep approach [40, 9]. A most sophisticated hyperparameter tuning technique could improve the performance of their deep learning models.

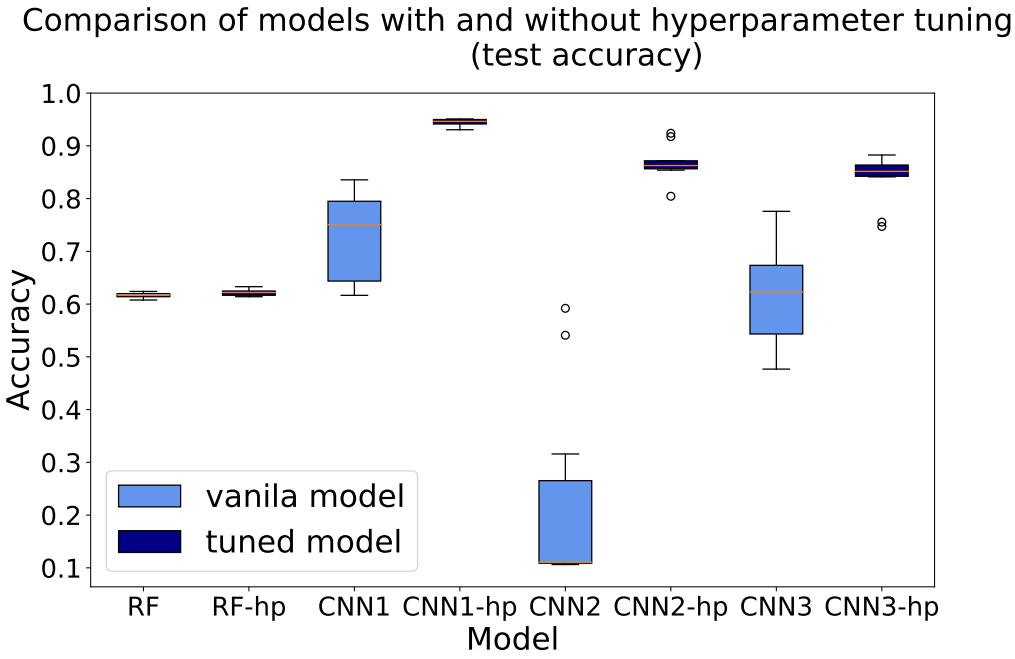


Figure 3.1: Preliminary experiments using the EuroSAT dataset [41]. A random forest and three different CNNs built from scratch based on machine learning and remote sensing literature [42, 36] are compared. For each model two versions are shown: a vanilla model performance using default configurations and a tuned model (suffix -hp). 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.

In previous experiments, we tested our assumption regarding the strong influence of hyperparameter settings in the performance of modern machine learning models for satellite image classification. Figure 3.1 shows the performance comparison of a random forest model (which is popular within the remote sensing community) and three different CNNs. Light blue color represents the model using default configurations and dark blue shows the performance after applying hyperparameter tuning. We can see that all the CNNs outperformed the results of the random forest approach, and contrary to this one, the results of the CNNs after hyperparameter tuning improved significantly illustrating how the tuning can make a difference between the resulting models. Supported by those results, in this thesis, we will address the remote sensing research gap of using more sophisticated hyperparameter tuning techniques through the use of automatic hyperparameter tuning incorporated in AutoML systems. As mentioned in Chapter 2.3, we will also incorporate the findings of remote sensing research into the creation of automatically generated deep learning models. Hopefully, this will lead us to improve or at least be comparable with the results found previously by the remote sensing research community.

3.2 AutoML system: Auto-Keras

Research in AutoML is very diverse and has developed software packages and methods targeted at both researchers and end-users [10]. Over the last few years, several open-source packages have been developed providing automated machine learning. Auto-Keras [14] is an efficient neural architecture search with network morphism, which utilizes Bayesian optimization to guide through the search space by selecting the most promising operations each time. Applying Bayesian optimization to NAS is not trivial. There are 3 main challenges for the design of this method. The first one is that neural architecture search space is not Euclidean, which affects the assumption of traditional Gaussian Process; the easy solution would be vectorizing, but due to the uncertain number of layers and parameters this solution is impractical. Therefore, instead of vectorizing Auto-Keras authors proposed an approximated solution of an Edit-Distance Neural Network Kernel for Gaussian Process. The second challenge is the acquisition function. The traditional acquisition functions are defined on Euclidean space, the optimization methods are not applicable to the tree-structured search via network morphism. Hence, the Auto-Keras authors propose a method to optimize the acquisition function for a tree-structured space. They selected the Upper confidence bound [43] as the acquisition function and set an explicit balance factor for exploration and exploitation. The third challenge is the tensor shape consistency while morphing the architecture. To solve this challenge Auto-Keras uses a graph level morphism to find and morph the layers influenced by a layer-level operation in the entire network. Previous work solved how to preserve the functionality at a layer lever morphism.

By tackling these challenges, the Auto-Keras framework is developed and the authors of [14] use a three-layer convolutional network as starting architecture for their experiments presented in the paper to test the efficiency of their approach compared with other methods.

Auto-Keras search space is built upon network morphism where the search space of NAS is created using morphism operations. These operations can insert a layer, make an existing layer wider, or add additive or concatenative connections and are denoted as follows: $deep(G, u)$, $wide(G, u)$, $add(G, u, v)$ and $concat(G, u, v)$, where G is a network architecture and u and v are nodes representing different positions on the network graph [14]. After G is initialized, any of the 4 morphism operations can be chosen to modify the network. However, when $wide$, add, or $concat$ operations are selected, extra actions are needed to ensure the shape consistency of the output.

The deployed Auto-Keras system has a task-oriented API, in which different initial architectures are applied for different tasks including classification and regression for structured data, text, and image data.

Chapter 4

Methodology

In this chapter, we will explain the methods applied to test the performance of AutoML for the satellite image classification task. We first describe how the search space is defined in the AutoML system, and after the different approaches regarding the search space for satellite data are described.

4.1 Neural Architecture Search Space

In order to discover automatically generated high-performance architectures for satellite data classification tasks, we set up the findings of deep learning in remote sensing literature in an AutoML framework. We proposed 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 first selected one of the deployed open AutoML systems to build upon. Two popular AutoML systems that focus on deep learning are Auto-Keras [14] and Auto-Pytorch [44], both supporting image classification tasks. Auto-Pytorch uses multi-fidelity optimization and Bayesian optimization (BOHB) 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 based on network morphism, it encloses all the 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. Due to the network morphism approach and the memory adaptation implemented in Auto-Keras, we decided to build up our proposed changes for the remote sensing applications on top of it.

In order 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 sections.

4.2 Approaches

4.2.1 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. As explained in Chapter 3.2, an initial network architecture G is given, and with the use of morphism, new networks are created derived from different operations. The authors of [14] use a three-layer convolutional network as starting architecture for their experiments presented in the 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 different initial architectures are applied for different tasks. The image classification task, which is the most relevant one for satellite data, uses 3 initial architectures: 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.

For the first part of this study, we want know how good is the current image classification task in Auto-Keras when it is used for satellite data. Therefore we do not do any changes in the search space and we keep the deployed initialization of G for the image classification task in Auto-Keras.

4.2.2 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 [45, 9, 46, 41]. The Auto-Keras search space already includes blocks with weights acquired by pre-training on ImageNet. Figure 4.1 is 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 we make use of the available resources in the current systems but we modify the configuration based on the remote sensing findings. We expect to improve the classification results by only changing the initial architecture choosing a block pre-trained with ImageNet. We influenced the search space of the image classification task by morphing a different initial architecture G. The new G can be selected based on the blocks with available ImageNet weights that are part of the best models found by the original Auto-Keras. In case the results do not contain relevant information we can take the findings about suitable architectures (mentioned in Chapter 3) and select the ResNet block. For this approach, we configure the initial G with the selected block and we set the parameter $pretrained$ to $true$ so the model pre-trained with ImageNet will be considered first.

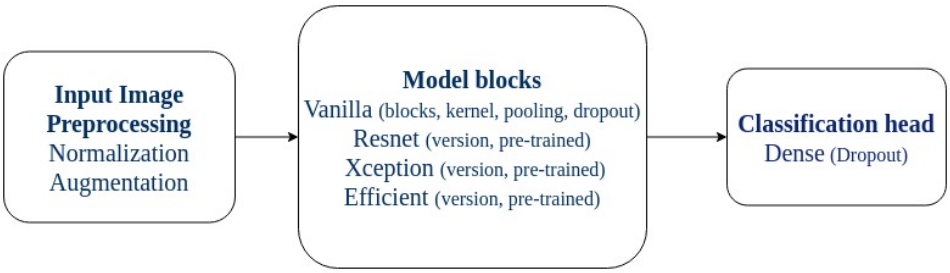


Figure 4.1: Abstract illustration of how the final architecture can be build based on pre-defined blocks existing in Auto-Keras.

4.2.3 Models pre-trained using remote sensing datasets (RS-AK)

We know that the use of transfer learning techniques can be successful when the source and target dataset and task are similar [47, 48, 9]. Within the remote sensing community, there are models pre-trained with remote sensing datasets [9, 49] but none of these are available yet in AutoML systems.

Therefore, we proposed to customize the Auto-Keras image classification task for its use with satellite data by incorporating the usage of models pre-trained with various remote sensing datasets.

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 for image classification was not fully explained in the original paper but it can be inferred from the open-source deployed system. After analysing the algorithm of the system, we deduce that the image classification task builds an architecture based on pre-defined cells or blocks. These blocks can be divided into three categories: pre-processing, model, and classification head. Inside the pre-processing category, two blocks are considered: (i) a normalization block, 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 conform to 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 consists of dense layers that will be added to the network to create the output layer of the network and it is connected to the model blocks, its parameters are related to 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 presented in Chapter 5.1. Furthermore, 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 will interact with the classification head block. Figure 4.2 is an abstract illustration of this. For future developments and other satellite data-specific tasks, however, we might need to modify more than the model blocks.

The $RS\ Block$ chooses between different pre-trained module versions (trained with satellite data). This choice is considered as another hyperparameter to tune. Therefore, it uses the same hyperparameter tuner that is used for all the other blocks. The current optimization method is explained next.

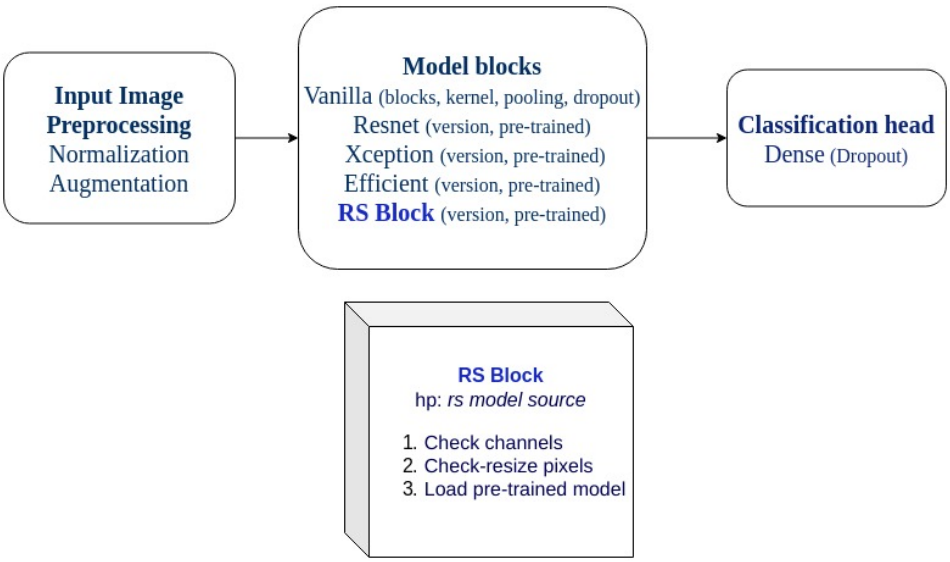


Figure 4.2: Abstract illustration of how the final architecture can be build based on pre-defined blocks. The RS Block takes as extra hyperparameter the remote sensing source model.

Hyperparameter tuning

Different tuners can be used to determine which combination of hyperparameters will be send for training in each trial during NAS. We used an oracle combining random search and greedy algorithm [50, 14] presented inside of Auto-Keras. It groups the hyperparameters 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 by modules of different satellite learning representations. Such representations are taken from different pre-trained models. Based on the number of spectral bands of the collected datasets we considered two types of pre-trained models.

3-channel models. As mentioned in Chaper 3, Google Research [9] made available 5 remote sensing pre-trained models. All the models use the same ResNet50 V2 architecture [35] and the tuning was made by sweeping only a small set of hyper-parameters. Each model was trained either from scratch or by fine-tuning ImageNet on each full dataset. Only one dataset per model was used for training. The best representation, based on the accuracy of correspondent dataset, was chosen.

13-channel models. Inspired by the findings in [36] and the selected architecture in [9], we decided to create our own in-domain representations for 13-channel datasets using ResNet architectures training in the EuroSAT dataset[41]. Keras Applications [51] are deep learning models that are made available alongside pre-trained weights. The use of such weights (from ImageNet) is optional. There are 6 versions of ResNet architectures [35, 52]: ResNet50, ResNet101, ResNet152, ResNet50V2, ResNet101V2 and ResNet152V2. In order to be able to use them for a 13-channel dataset, we need to modify the input shape parameter to the dimensions of the dataset (64,64,13) and do not include the top layer. We use the output of such a model to and we add a global spatial average pooling layer and a dense layer with the number of classes (10) as units to create the model architecture. We compiled it using the Adam optimizer and categorical cross-entropy loss. We trained the models with 50 epochs and 80% of the full dataset and evaluating the performance in the remaining 20%. Performances between 88% and 95% accuracy were found. Trying to improve the performance by using a more tuned architecture, we included the best model architecture found by the AutoML system, which has an accuracy of 98%, as another 13-channel pre-trained model.

After saving the trained models, we call them using the keras load model function and we remove the top layer because this last layer was adapted to the number of classes in the dataset. The model In order 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. As part of the study of the different remote sensing learning representations, we want to analyse which of the current representations has promising results and is more generalisable among the datasets. We can do this by studying the $version$ parameter of the $RS\ Block$

Chapter 5

Experiments

Within this chapter, we will present the experiments formulated to solve our research questions. Our original questions are:

what is the performance of current AutoML systems for satellite data? and
what happens when we integrate the knowledge gained from previous research in the field of remote sensing into AutoML systems?

In order to answer such questions, we created 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 is the performance of different Auto-Keras variants regarding the number of spectral bands, the size of the dataset, and the different class distribution datasets?
- Q3. When RS-AK is applied, which pre-trained modules (dataset source) are often chosen as part of the best model?

Based on Q1, we want to show the potential of the usage of AutoML systems for satellite data tasks. With the findings of Q2 we can compare the three approaches and their behavior in different types of datasets. As $RS - AK$ integrates remote sensing transfer learning representations and we are interested in studying which ones work the best, Q3 will help us to have an idea about it.

Our experiments were designed to show the performance of the different variants of the proposed AutoML framework described in Chapter 4. We compared the different variants against themselves and against the results presented in remote sensing research, which were obtained with high performed manually designed architectures. We ran the three methods for a different number of trials (10, 50, 100, 200) and for the different datasets using mainly accuracy as the performance metric. We extracted the configuration of the selected best model of each run to analyse the frequency of appearance of each pre-trained module.

The following sections present the datasets and their correspondent baseline (high performed manually designed architecture), the remote sensing pre-trained models used, and the experimental setup of our experiments.

5.1 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 multi-spectral satellite datasets.

Table 5.1 presents several characteristics of these datasets. These datasets are considered benchmarks in the remote sensing literature and well-known by the community. Furthermore, this selection shows a variety of classification tasks with presumably different degrees of difficulty and complexity.

BigEarthNet [49] and EuroSAT [41] are the most recent datasets extracted from Sentinel 2 images. For EuroSAT [41], Coffee scenes [46] and UC Merced [53] datasets the number of samples per class is quite similar. However, the BigEarthNet [49], So2Sat [54] and Cerrado-Savanna scenes [37] datasets have different class distribution. Among these datasets, the Brazilian Cerrado-Savanna scenes [37] is a challenging dataset, as explained by the authors, this is due to its high intraclass variance, 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 [37]. Moreover, from 1,311 samples included in this dataset, 73% correspond to the Arboreal vegetation. Figure 5.1 can give an idea of the difficulties when classifying these images.

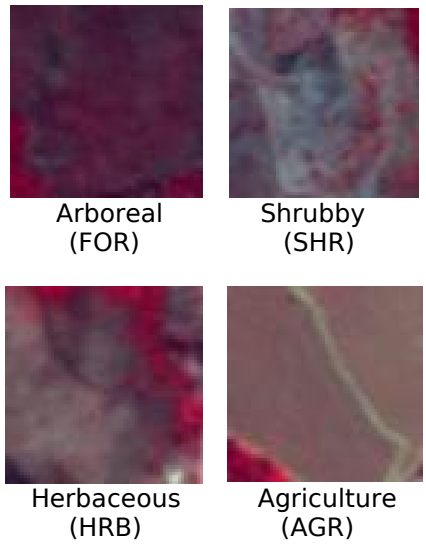


Figure 5.1: Examples of the Brazilian Cerrado-Savanna scenes dataset [37]. The identification of the 4 classes is challenging even for human eye. The labelling was performed by biologists and specialists

5.2 Baselines

We compare the performance of the three AutoML approaches against well-performing models from the literature. The following baselines are selected representing high-performance manually designed architectures for each dataset. The deep learning approach applied that gives the state-of-the-art (SOTA) performance for each case is explained in this section.

Table 5.1: Overview of available labeled datasets and the presented accuracy from the paper in which the dataset was introduced.

Dataset	Satellite (Bands)	Resolution	Images	Label	Number of classes	Performance	Reference
BigEarthNet	Sentinel-2 (3/12)	Medium-High	590,326(L)	Land cover	43	67.59%	[49]
BrazilDam	Sentinel-2 (13)	High	1,925(S)	dam/not dam	2	94.1%	[40]
Brazilian Coffee Scenes	SPOT (3)	High	2,876 (S)	coffee/non-coffee	2	83.04%	[46]
Cerrado-Savanna scenes	RapidEye (3)	High	1,311 (S)	Scene vegetation	4	90.5%	[37]
EuroSAT	Sentinel-2 (3/13)	High	27,000(L)	Land use	10	98.57%	[41]
So2Sat	Sentinel-2 (3)	High	376,000 (L)	Land cover	17	61%	[54]
UC Merced	USGS(3)	Very High	2,100 (S)	Land use	21	NA	[53]

5.2.1 BigEarthNet

Until now, BigEarthNet [49] is the largest benchmark dataset in the remote sensing field, consisting of 590,326 non-overlapping Sentinel-2 image patches. The spatial resolution on the ground is 1.2 x 1.2 km with 3 different image sizes depending on the spectral band resolution. Each sample was labeled with multiple land cover classes that were provided from the CORINE Land Cover database 2018 [55]. This is a non-balanced multi-label dataset with 43 classes.
The paper in which BigEarthNet was introduced [49] only presents the precision (65.06%), recall (75.57%), F1 and F2 metrics (67.59%, 71.39%) of a CNN architecture that consists of 3 convolutional layers and max-pooling.

5.2.2 BrazilDam

BrazilDam [40] is a public dataset based on Sentinel-2 and Landsat-8 satellite images covering all tailings dams classified using the Brazilian National Mining Agency data coordinates. The dataset was created from georeferenced images between 2016 and 2019 of 769 dams in Brazil. The dams have different shapes, areas, and volumes and they also contain some mining waste, this makes BrazilDam a challenging dataset to be used in machine learning benchmarks. Dam classification, Ore classification, and Risk Category are tasks that can be tested with this dataset.

The approach taken in [40] for the tailing Dam classification task consisted of evaluating six different neural network architectures: AlexNet [32], DenseNet [33], Inception [56], ResNet [35], SqueezeNet [57] and VGG [34], using Adam algorithm as optimizer and trying 3 different learning rate parameters (10^{-2}, 10^{-3}, 10^{-4}). The resultant best model was the DenseNet network with a learning rate of 10^{-4}.

5.2.3 Brazilian Coffee Scenes

This dataset consists of scenes related to coffee crops over four counties in the State of Minas Gerais, Brazil taken by the SPOT sensor in 2005.This dataset contains images with different plant ages with spectral distortions caused by shadows and the crop management techniques cause many intraclass variances [46]. Each county image was divided into multiple tiles of 64 x 64 pixels. The coffee crops distinguishing was made manually by agricultural researchers. The dataset folds are balanced with coffee and non-coffee samples (50% each).

In [46], the authors evaluated different feature descriptors including low-level, mid-level, and convolutional networks to use a SVM classifier. OverFeat [58] and Caffe [59] were considered for the CNN descriptors, both trained with the ImageNet dataset [60]. The best average accuracy was acquired by using a color descriptor but the use of two OverFeat networks gave an accuracy of 83.04%$\pm$2.

5.2.4 Cerrado-Savanna Scenes

This dataset contains 1,311 multi-spectral images acquired by the RapidEye satellite sensors. It covers the southern-central Brazil region, which contains a extremely bio-diverse and heterogeneous mountains landscape [37]. The RapidEye has 5 spectral bands: blue, green, red, red edge, and near-infrared; this dataset contains only the information from the near-infrared, green, and red bands. Those can help to discriminate vegetation areas. The Cerrado-Savanna dataset is a quite a challenge due to its high intraclass variance, caused by different spatial configurations and densities of the same vegetation type, as well as its high inter-class similarity, given a similar appearance of different types of vegetation species. The images are 64 x 64 pixels, and are classified into Agriculture, Arboreal, Herbaceous, and Shrubby Vegetation. Biologists and specialists manually labeled the data (ground-truth annotation).

The approach in [37] to solve the classification task of this dataset is similar to the Coffee Scenes approach. It also uses CNNs as feature extractors but it also explores the use of CNN with a fine-tuning strategy. AlexNet [32] was the network used, the features were extracted from the last fully-connected layer, and for the fine-tuning, two approaches were considered: freezing the first 3 layers, and without freezing any layers. [37] found the best results with the fine-tuning strategy achieving a 90.54%$\pm$1.83 overall accuracy.

5.2.5 EuroSAT

EuroSAT dataset [41] is one of the newest benchmark datasets, it was made available in 2019. It is composed of 27,000 labeled and georeferenced Sentinel-2 satellite (13 spectral bands) images of 64x64 pixels classifying 10 different land use and land cover classes. Each class has between 2,000 and 3,000 samples. The satellite images are associated with the cities covered in the European Urban Atlas.

For this dataset, the authors evaluated the performance of the Bag of-Visual-Words approach using SIFT features and a trained SVM and three different CNNs: a CNN with two layers, ResNet-50 [35] and GoogleNet [56]. The best accuracy (98.57%) was obtained using a fine-tuned ResNet-50 (pre-trained on ImageNet) for RGB bands. The paper does not show the performance of using all the 13 bands together as input. [9] established a new benchmark of 99.2% using the same architecture but pre-trained on remote sensing datasets.

5.2.6 So2Sat

So2Sat LCZ42 [54] is a benchmark dataset consisting of co-registered synthetic aperture radar and multi-spectral optical image patches acquired by the Sentinel-1 and Sentinel-2 remote sensing satellites, and the corresponding local climate zones label. The dataset is distributed over 42 cities across different continents and cultural regions of the world. The labelling of this dataset was performed by 15 domain experts.

For this dataset, only three baselines were tested: a Random forest, a SVM and ResNeXt-CBAM [61]. The best achieved overall accuracy was 61% using the ResNet approach. [9] also found out that using fine-tuning with in-domain representations an accuracy of 63.25% can be achieved.

5.2.7 UC Merced

UC Merced dataset [53] is composed of 2,100 aerial scene images extracted from the United States Geological Survey (USGS) National Map Urban Area Imagery collection for various urban areas around the country. Each image measures 256x256 pixels, with a pixel resolution of 1 foot.

The authors of this dataset investigate Bag of-Visual-Words approaches but do not present the best accuracy for the classification task. Other papers had achieved high performance (around 99%) for the dataset. [9]'s approach shows 99.61% accuracy.

5.3 Remote sensing representations

The remote sensing learning transfer representations used in this project were originated from three different sources.

The first ones came from the TensorFlow Hub [62] library. These are 5 models deployed by [9], consist of ResNet50 architectures trained for BigEarthNet, EuroSAT, So2Sat, UC Merced, and Resisc45 dataset. The training approach was explained in Chapter 4.2.3

The second source is the models trained using the Keras Applications [51] model definitions. These are 6 different ResNet versions that we trained using the EuroSAT dataset. Its specifications were also described in Chapter 4.2.3. The weights created by these models were saved in h5 format (easily reusable by the community).

The third source is a model from the AutoML system itself. The best model found for the 13-channel EuroSAT dataset using the original Auto-Keras. This architecture is composed of a convolutional layer and a Resnet50 module.

Although, there are no exact rules yet defining what makes a good and generalisable pre-trained model. Their performance in the unseen data can give a clue of how good the model will be. It is expected that models created with a big dataset can be applied for a large range of tasks. But there is also a threshold related to the similarity of the tasks [63, 64].

In order to study the best model representations, we used the first type of models with different data sources for the RS-AK approach for all 3-channel datasets and the second and third source only for the BrazilDam dataset.

The remote sensing learning transfer representations used in the $RS\ Block$ were taken from removing the top layer of the trained networks, so the output of this block can be connected to the classification head block from Auto-Keras.

5.4 Performance metrics and validation

Accuracy:
Accuracy is defined as the fraction of predictions that the model identified correctly.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

F1-score, Precision and Recall:
Precision and recall are useful performance measures when the classes are not balanced. Precision is the resulting score from the division of the true positives over the union of the number of true positives and the number of false positives.

$$Precision = \frac{TP}{TP+FP}$$

Recall is the resulting score from the division of the number of true positives over the true positives plus the number of false negatives.

$$Recall = \frac{TP}{TP+FN}$$

F1-score is the harmonic mean of precision and recall.

$$F1 - score = 2\frac{Precision \times Recall}{Precision + Recall}$$

Overall F1-score
The simplest way is to take the macro averaged F1-score. It is computed as an arithmetic mean of the F1-score per class.

5.5 Experimental setup

For all our experiments, the datasets were first randomly divided into a training set and a testing set. 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 5 folds. The first 4 were used for training and the last one is considering the testing set. Next, another split of 80-20 was applied to the training set, assigning 20% of it for validation, which was 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, during the third experiment when testing on a specific dataset, the pre-trained weights of the same dataset were discarded. For example, while testing on the EuroSAT dataset, the models pre-trained with EuroSAT were not considered as part of the search space. To exclude the corresponding pre-trained models, before running the task, we removed this option from the set of pre-trained remote sensing models to consider inside the $RS\ Block$.

To be able to show the significance of the results, the outcomes presented in this thesis are based on the 10 trials experiments. 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 with 10 trials. Each trial, varying per dataset, ranges from few minutes to around 6 hours.

The Auto-Keras version was 1.0.9, the modified versions used for the experiments will be available in https://github.com/palaciosnrps/automl-rs-project and https://github.com/palaciosnrps/autokeras.
The existing datasets were build using Tensorflow datasets version 1.2 and to add the new ones tfds-nightly 4.2.0dev was used. For the models we took Tensorflow hub version 0.9 and the repository of [9] models available in https://tfhub.dev/google/collections/remote_sensing/1.

Chapter 6

Results

Within this chapter, we will present the results of our experiments. We divided this chapter into three sections. The first section shows the comparison of the AutoML approaches against the non-automated models and answers Q1. The second section presents the findings of comparing the three different approaches and answer Q2. The third section focuses on answering Q3 while analysing the results of the approach using our new remote sensing block.

6.1 AutoML vs non-automated models

As mentioned in Chapter 4.2.2, we first ran the original Auto-Keras (V-AK) and we analysed the best models obtained from it. Figure 6.1, shows that in 78% of these models a ResNet module was chosen as part of the architecture. These results agree with the findings in the literature regarding the popular use of ResNet architectures for satellite datasets. Therefore, we are using the ResNet block pre-trained with ImageNet as the initial architecture for the IMG-AK approach.

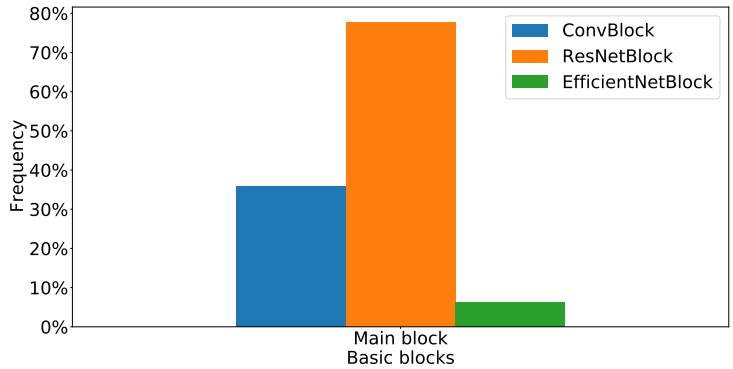


Figure 6.1: Image blocks chosen as part of the best model acquired by V-AK. Please note that, the sum of the frequency exceeds 100% because the final architecture can include more than one model block.

Table 6.1 summarizes the performance of the three different AutoML approaches on the test set for the different datasets. The performance metric shown here, compatible with the baseline papers, is the overall classification accuracy. For the BigEarthNet-rgb dataset, we decided to change the performance metric (See Table 6.3) to be able to compare with the baseline. We achieved an F1-score of 67.84% using an ImageNet pre-trained module, meanwhile the result presented in [49] is 67.59%.

Table 6.1: Average overall accuracy on test dataset considering 10 runs of each experiment and the state-of-the-art (SOTA) for each dataset. EuroSAT-all was not included in the RS-AK approach because the 13-channel the model was pre-trained on the same dataset, no benchmark was found for this version of the dataset. Boldfaced entries are the best approach among the 3 AK variants and the entry marked with * shows that the results are statistically significant.

Dataset	Type	SOTA	Vanilla Auto-Keras (V-AK)	ImageNet Auto-Keras (IMG-AK)	Remote Sensing Auto-Keras (RS-AK)
BrazilDam	Small-13	94.1[40]	**89.09**±.05	76.54±.13	85.57 ±.01
Coffee scenes	Small-3	83.4[46]	86.18±.02	82.96±.04	**88.84±.00***
Cerrado-Savanna scenes	Small-3	90.5 [37]	85.79±.01	84.33±.03	**89.92±.01**
UCMerced	Small-rgb	99.61 [53]	**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[9]	99.18±.00	**99.54±.00***	95.90±.01
So2Sat-rgb	Large-rgb	63.25[9]	95.47±.00	**97.80±.00***	76.92±.00

There is not benchmark performance for the full spectral version of EuroSAT, resultant from our experiments we established one with 97.8% overall accuracy.

We can see that the overall accuracy of V-AK is already above 80% for all the datasets. Comparing these results with the state-of-the-art performance (also in Table 6.1), V-AK got a better performance for the Coffee scenes, So2Sat-rgb, and UC Merced dataset. IMG-AK outperformed the results on EuroSAT-rbg and also So2Sat. RS-AK improved the performance on the Coffee scenes dataset.

To answer $Q1$: *Can we achieve performance similar to the non-automated deep learning research in remote sensing by using AutoML systems?*, we grouped the results of the three variants (V-AK, IMG-AK, and RS-AK) and we took the maximum accuracy. In this way we can analyse the AutoML competency against the non-automated architectures. Figure 6.2 helps us to illustrate it. It compares the best obtained result from our experiments versus the state-of-the-art accuracy obtained by non-automated methods taken from the literature. We outperformed the literature in 4 out of 6 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.

6.2 AutoML variants and the different type of datasets

To address **Q2**, we firstly group our selected datasets per number and type of spectral bands (channels). We have 5 datasets with 3 channels. Coffee scenes and Cerrado-Savanna scenes datasets are composed of images with near-infrared, green, and red bands. The EuroSAT and the So2Sat datasets contain the red, green, and blue bands (RGB). We have two datasets (BrazilDam and EuroSAT-all) with 13 channels from the Sentinel-2 satellite. As we used EuroSAT to build a pre-trained model for 13 channel datasets, we should not test the same source dataset as the target dataset hence the corresponding entries in Tables 6.1, 6.3 and 6.2 are left blank. If we do include the same dataset, such a pre-trained model is chosen as the best and gives an accuracy near 100%. In Table 6.1 the boldfaced

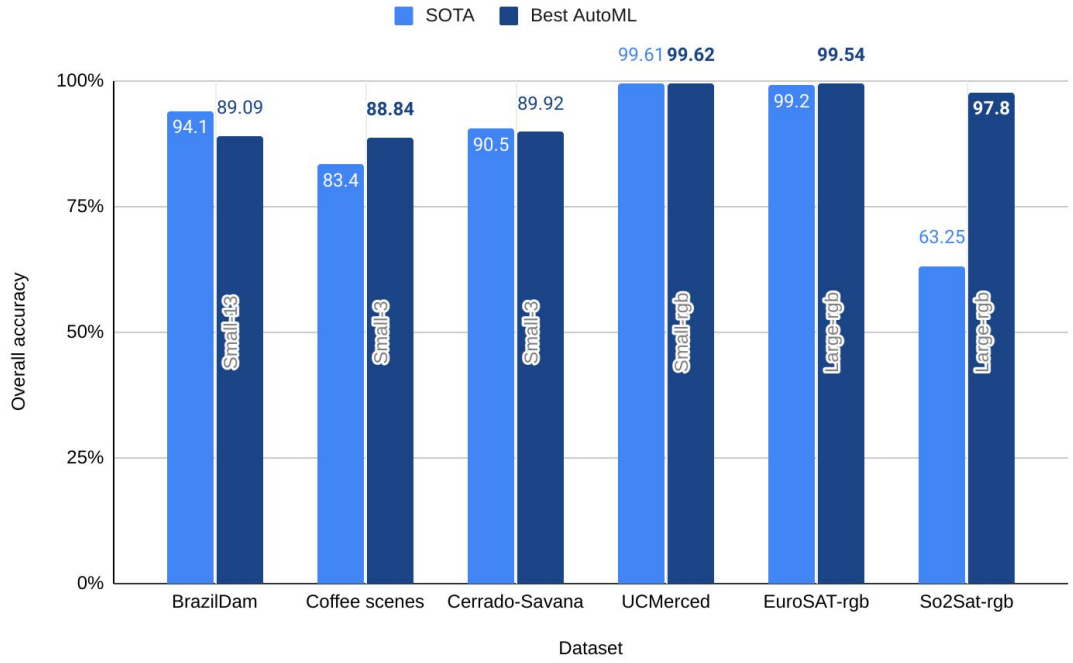


Figure 6.2: Comparison between the state-of-the-art (SOTA) performance obtained with manually designed models versus the best performance found by using any of the three AutoML approaches. In 4 out of 6 datasets the AutoML models outperformed the previous SOTA results.

entries indicate the best approach among the three AK variants. In order to determine the statistical 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. If the results are statistically significant for that best approach the entry is also marked with * (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 similarly on the EuroSAT-all dataset. For the case of the BrazilDam dataset, the initialization with a pre-trained ImageNet model did not benefit the performance (see IMG-AK Table 6.1), and on the contrary, it decreased the average accuracy of the runs. This can be explained considering the difference in the number of input channels. BrazilDam dataset is a 13-channel image dataset; therefore, the direct use of pre-trained models from ImageNet (3-channel) does not apply. Only blocks without weights can be selected. Different from the EuroSAT dataset, the number of labeled samples of BrazilDam is small. When the RS-AK version was used for BrazilDam, we can notice an improvement considering this initialization, however, the performance did not beat the results of the V-AK best models. This suggests that for this particular case, training the network from scratch could lead to better performance.

We can see that for the RGB channel datasets either V-AK or IMG-AK approaches lead to the best performance, meanwhile for Coffee scenes and Cerrado-Savanna scenes datasets the RS-AK variant obtained the best results. These two datasets are composed of near-infrared, green, and red bands and the classification task differs from land cover-land use identification. Considering such observations, 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 the 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

Table 6.2: Maximum average misclassification rate per class on test dataset. EuroSAT-all was not included in the RS-AK approach because the 13-channel model was pre-trained on the same dataset.

Dataset	Type	Vanilla Auto-Keras (V-AK)	ImageNet Auto-Keras (IMG-AK)	Remote Sensing Auto-Keras (RS-AK)
BrazilDam	Small-13	**11.1%**	32.3 %	22.0%
Coffee scenes	Small-3	15.0%	25.5%	**13.0%**
Cerrado-Savanna scenes	Small-3	83.9%	71.4%	**54.1%**
UCMerced	Small-rgb	**3.8%**	28.9%	19.1%
EuroSAT-all	Large-13	11.6%	**4.3%**	-
EuroSAT-rgb	Large-rgb	2.2 %	**1.3%**	7.3%
So2Sat-rgb	Large-rgb	12.5%	**6.3%**	57.0%

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

Secondly, we can group our datasets per size. Considering BrazilDam, Coffee scenes, Cerrado-Savanna and UC Merced as small datasets and EuroSAT and So2Sat as large ones.

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. Probably this is linked to the transfer learning technique and the dataset we are using.

The overall accuracy only gives a general idea of the performance, we need to look with more detail into each class to know if there is still any room for improvement. We can generate the confusion matrices per each run. Figure 6.3a 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 (Figure 6.3b), 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.

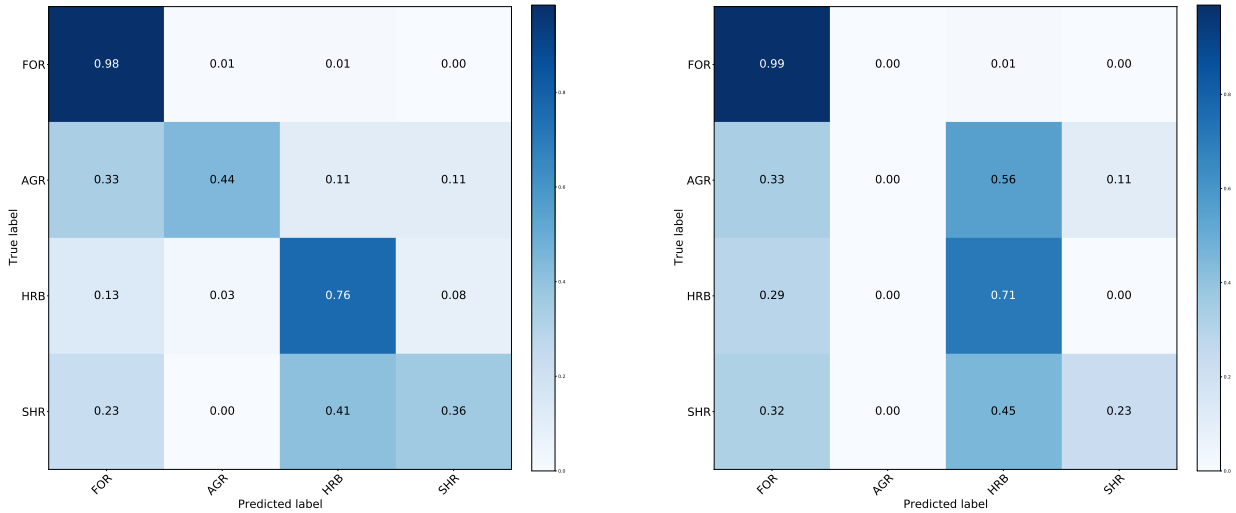
Some of our datasets have only 2 classes, while others have more than 2. Also, for some of them the distribution on classes is highly variant whereas for others the number of samples per class are similar.

Table 6.2 summarises the insights acquired from confusion matrices by showing the maximum misclassification rate found (it can belong to any class) after averaging the misclassification rate obtained on the test set results for each class. Lower values show higher performance by achieving less misclassification error. This metric will give us a better understanding of the improvements made by the different AK variants on the classification task. The EuroSAT and So2Sat (our large datasets) achieved lowest misclassification rate while using the IMG-AK variant. The Coffee scenes and Cerrado-Savanna datasets obtained lower values using the RS-AK version. The UC Merced dataset got the best performance with the V-AK approach. UC Merced is the only dataset consisting of very high resolution images, each pixel represents 0.3m whereas the other datasets are 10m.

Considering a more standard metric, Table 6.3 shows the average f1-score achieved on the test

Table 6.3: F1-score on test dataset considering 10 runs of each experiment per dataset, except for BigEarthNet which had 3 runs. EuroSAT-all was not included in the RS-AK approach because the 13-channel the model was pre-trained on the same dataset.

Dataset	Type	Vanilla Auto-Keras (V-AK)	ImageNet Auto-Keras (IMG-AK)	Remote Sensing Auto-Keras (RS-AK)
BrazilDam	Small-13	**85.62**±.02	83.13±.04	84.59±.01
Coffee scenes	Small-3	82.91±.10	86.07±.02	**89.24**±.00*
Cerrado-Savanna scenes	Small-3	85.63±.01	85.63±.01	**88.94**±.01*
UCMerced	Small-rgb	**97.75**±.01*	73.94±.15	82.70±.11
EuroSAT-all	Large-13	94.75±.01	**94.78**±.01	-
EuroSAT-rgb	Large-rgb	97.96±.00	**98.14**±.00	95.41±0.01
So2Sat-rgb	Large-rgb	94.58±0.0	**95.71**±.00*	75.52±.00
BigEarthNet-rgb	Large-rgb	50.62±.00	**67.84**±.00	65.29±.00



(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).

Figure 6.3: Comparison of confusion matrices for Cerrado-Savanna dataset. Classes are Agriculture (AGR), Arboreal Vegetation (FOR), Herbaceous Vegetation (HRB) and Shrubby Vegetation (SHR).

dataset considering 3 runs for the BigEarthNet dataset and 10 runs for all the other datasets. It is worth mentioning that the AutoML system chooses the best model based on a performance metric, accuracy by default. Only in the case of BigEarthNet, the objective metric was changed to f1-score for the IMG-AK and RS-AK approaches. In the rest of the cases maximizing the accuracy remained

the objective. The F1-score and Overall accuracy of the datasets showed in the tables came from the same best model. The same conclusions regarding the three variants can be drawn using F1-score. But these scores can be used as point of reference for other researches and for our future experiments changing the objective metric. The precision and recall tables can be found in Appendix B.

The most evident conclusions can be taken from the dataset size. The results presented so far in Tables 6.1, 6.2 and 6.3 suggest that for small datasets the RS-AK is outperforming IMG-AK. However, for large datasets, IMG-AK achieves the best performance. This can 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 [63, 47, 64]. Bigger datasets should produce better representations. However, 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 accuracy 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.

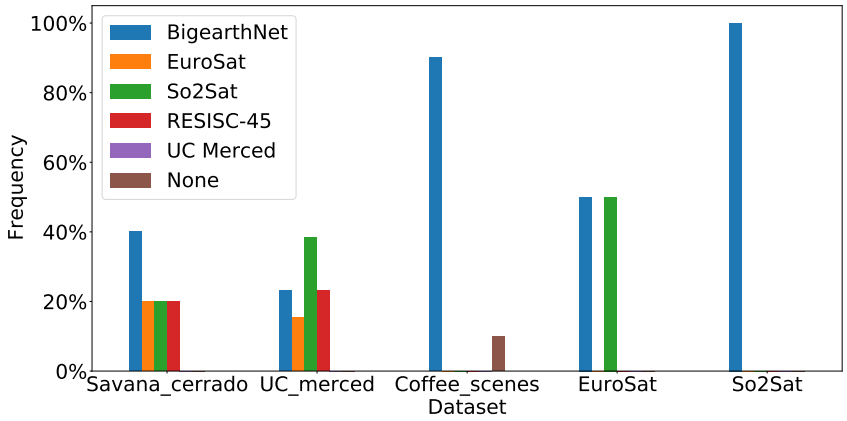


Figure 6.4: Remote sensing pre-trained models selected for the 3-channels datasets during the third experiment. The datasets in X-axis are sorted ascending by the size of the dataset. Cerrado-Savanna, UC Merced and Coffee scenes are small datasets.

6.3 The remote sensing block RS-AK

In this section, we aim to address **Q3**. Figure 6.4 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 source 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 in [9] that concludes RESISC-45 representation achieves the highest performance. Our experiments differ in the way we are using a more efficient framework for tuning the hyper-parameters and selecting the design choices using an oracle combining random search and a greedy algorithm (explained in Chapter 3) while the authors of [9] propose to optimize by sweeping only a fixed set of hyper-parameters. In our approach the pre-trained model is only a block that is part of the full architecture (see Figure 4.2)

and the selection of the best performed model was based on the validation set inside the Auto-Keras framework. The authors of [9] utilized the same ResNet50V2 architecture [52] to train the remote sensing datasets using SGD with momentum set to 0.9 and the comparison of the different pre-trained models was made after finishing the fine-tuning either using 100, 1000 or the full number of training samples. Considering that, we believe that our experiments have exploited the potential of each dataset representation by using a more sophisticated framework and our results are consistent with the expectations of the remote sensing community about the promising applications of BigEarthNet on remote sensing tasks [49].

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we addressed the questions *what is the performance of current AutoML systems for satellite data?* and *what happens when we integrate the knowledge gained from previous research in the field of remote sensing into AutoML systems?*

We demonstrated how automated machine learning can be used to leverage the implementation of deep learning models for satellite data tasks, achieving results comparable to 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 [14](V-AK) and two different variants of its image classification task, one initializing the architecture to morph with a model pre-trained on ImageNet (IMG-AK) and a second one 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, show that for three channel datasets, current AutoML systems can beat state-of-the-art results for land cover and land use classification tasks. However, the tasks with different spectral channels and regarding more specialized knowledge still can be improved.

Trying to improve our results and to answer our second question, we analysed the performance of the two Auto-Keras variants, IMG-AK and RS-AK, initialized with pre-trained blocks. When pre-trained blocks are used in the model a better representation of the classes is created. The use of remote sensing pre-trained models helps to improve these results when the channels are different from the standard true-color RGB. And for all the small datasets, RS-AK got better performance than the IMG-AK variant. The use of bands different from RGB is a common practice in remote sensing applications due to the extra spectral information that can be extracted from such bands. A clear example is the creation of vegetation indexes for different applications; such indexes involve non-RGB channels like near-infrared. Moreover, the number of samples available for training in remote sensing real-world problems is usually 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. However, based on our results for the studied 13-channel dataset there is still room for improvement in such remote sensing representations.

The practical milestone we achieved was to create a new module built on top of the Auto-Keras framework that can easily be used by the remote sensing community for NAS. This module re-uses previously trained remote sensing models and currently works without any additional requirement for 3-channel datasets. The 13-channel version is also available but it requires our pre-trained model to be downloaded and placed into a local directory. For easier deployment, we will work on integrating this feature in a more seamless manner for the final user.

7.2 Future Work

In future work, we will first aim at improving the transferability of the pre-trained models. The use of transfer learning techniques is most successful when certain conditions are met, those are regarding the size of the datasets or the similarities between the source and target tasks. Depends on the degree of similarity and the size of the data target available for training, different transfer learning strategies can be applied. The source datasets used for this study are all related to land use or land cover classification. Half of our target datasets are based on the same task as well, but the other half deals with more specific problems: dam detection, agricultural or vegetation species classification. Different dataset size are considered. In this case, a more sophisticated transfer learning method or customized techniques per dataset/task (based on the findings of [63, 64]) integrated into the AutoML system could lead to a better use the remote sensing data representations. Techniques for deep meta-learning surveyed in [65] to create a meta-learning model that leverages prior learning experience (taken from diverse source tasks) to learn new tasks with fewer data can also be a suitable option in the remote sensing domain.

Furthermore, we will focus on automating other meaningful remote sensing tasks like time series image classification, regression and anomaly detection helping more remote sensing practitioners to integrate state-of-the-art machine learning to real-world problems. We can start by adding a the temporal dimension to the classification problem studied in this thesis and adding Convolutional LSTM and RNN blocks to the search space.

Appendix A

New benchmark EuroSAT

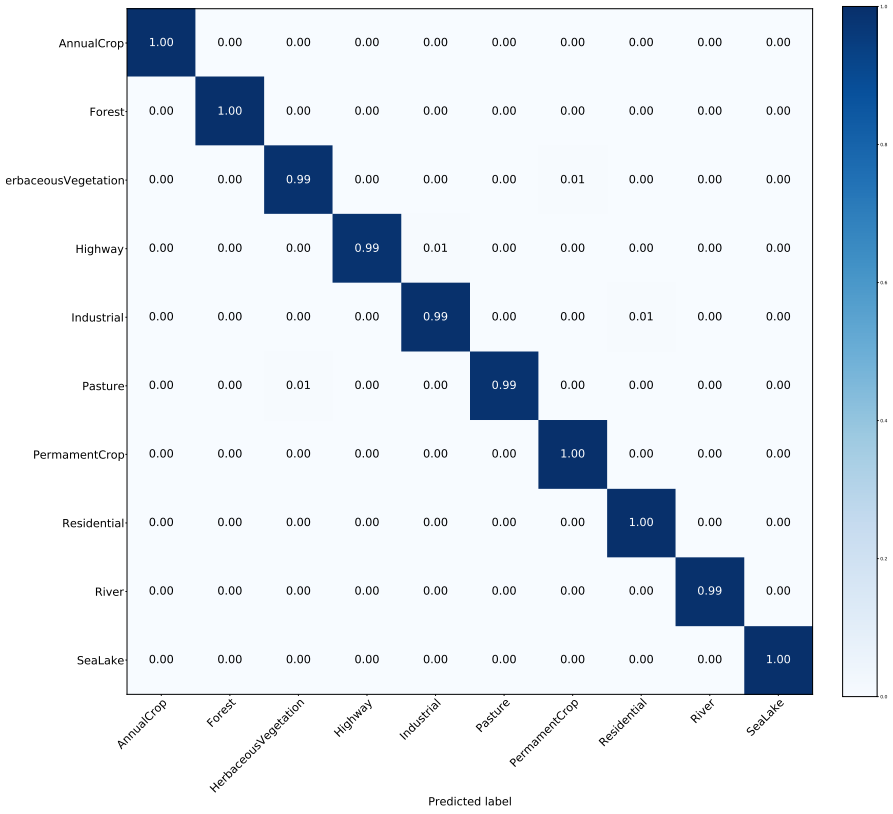


Figure A.1: Confusion matrix for EuroSAT dataset, results from best performed model. The performance sets a new benchmark for this dataset.

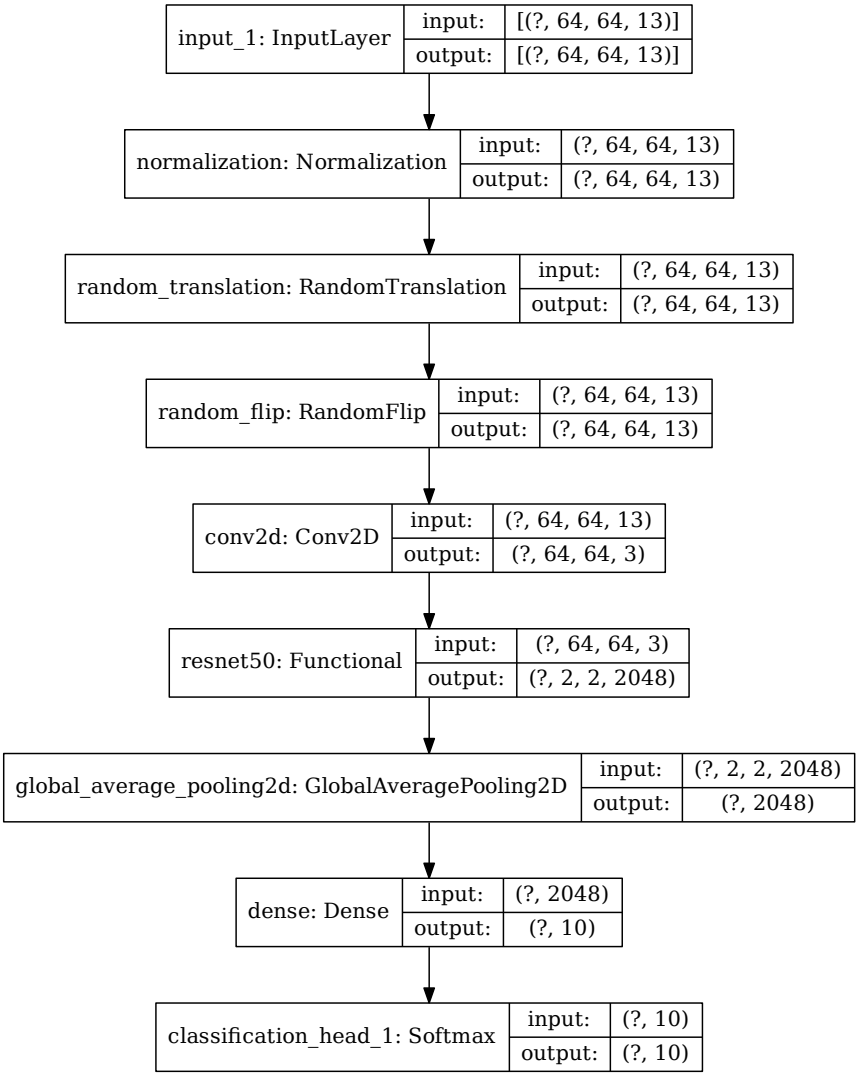


Figure A.2: Model structure of best model found for EuroSAT dataset.

Appendix B

Precision and Recall tables

Table B.1: Overall precision on test dataset considering 10 runs of each experiment per dataset, except for BigEarthNet which had 3 runs. EuroSAT-all was not included in the RS-AK approach because the 13-channel the model was pre-trained on the same dataset.

Dataset	Type	Vanilla Auto-Keras (V-AK)	ImageNet Auto-Keras (IMG-AK)	Remote Sensing Auto-Keras (RS-AK)
BrazilDam	Small-13	84.40±.06	**88.10**±.08	87.19±02
Coffee scenes	Small-3	83.67±.06	82.55±.05	**87.62**±.00
Cerrado-Savanna scenes	Small-3	87.23±.01	85.22±.03	**90.28**±.02
UCMerced	Small-rgb	**97.88**±.01	94.41±.03	95.74±.02
EuroSAT-all	Large-13	94.98±.01	**95.00**±.01	-
EuroSAT-rgb	Large-rgb	98.02±.00	**98.20**±.00	96.14±.01
So2Sat-rgb	Large-rgb	95.11±.00	**95.93**±.00	85.38±.00
BigEarthNet-rgb	Large-rgb	**99.07**±.00	51.36±.00	48.48±.00

Table B.2: Overall recall on test dataset considering 10 runs of each experiment per dataset, except for BigEarthNet which had 3 runs. EuroSAT-all was not included in the RS-AK approach because the 13-channel the model was pre-trained on the same dataset.

Dataset	Type	Vanilla Auto-Keras (V-AK)	ImageNet Auto-Keras (IMG-AK)	Remote Sensing Auto-Keras (RS-AK)
BrazilDam	Small-13	**87.63**±.06	80.71±.12	86.34±.02
Coffee scenes	Small-3	84.24±.14	90.34±.04	**90.92**±.01
Cerrado-Savanna scenes	Small-3	84.10±.01	82.03±.05	**87.71**±.02
UCMerced	Small-rgb	**97.62**±.01	63.31±.21	74.31±.16
EuroSAT-all	Large-13	94.53±.02	**94.57**±.01	-
EuroSAT-rgb	Large-rgb	97.90±.00	**98.08**±.00	94.70±.01
So2Sat-rgb	Large-rgb	94.06±.01	**95.48**±.00	67.7±.00
BigEarthNet-rgb	Large-rgb	33.99±.00	99.89±.00	**99.98**±.00

Appendix C

Reproducibility guide

The code and the instructions to reproduce these experiments are being updated on https://github.com/palaciosnrps/automl-rs-project. Most of the datasets used for this paper are already available in the Tensorflow-datasets collection, however three more datasets were used (BrazilDam, Coffee scenes, Cerrado-Savanna scenes). The code for building them as part of the tensorflow-dataset package can be found in the folder 'datasets/tf-new-datasets', only the script 'load tfds.sh' needs to be executed.

The modifications in the initial architecture made for the IMG-AK variant are listed below. The code for the RS-AK variant is presented in detail in the github repository.

Pretrained initialization using ImageNet
image normalize: True
image augmentation: True
image block type: Resnet
Resnet block pretrained: True
Resnet block version: Resnet50
Classification head block: Global avg
Classification head block Droput:0
optimizer: adam
learning rate: 2e-5

Other settings
The preliminary experiments were executed on a compute cluster using nodes with 2 GPUs GeForce GTX 1080TI and 2 CPUs of 16 cores.

Name	Model	Values
Max. features	Random Forest	[0-1]
Epochs	All-CNN	[5-100]
Batch size	All-CNN	[128,256,384,512]
Optimizer	All-CNN	SGD, RMSprop, Adam
Learning rate opt	All-CNN	[1e-6 - 1e-2]
Dropout rate	CNN2	[0-0.5]

Table C.1: Configuration space for hyperparameters used in preliminary experiments.

Bibliography

[1] S. Shekhar, Z. Jiang, R. Y. Ali, E. Eftelioglu, X. Tang, V. M. Gunturi, and X. Zhou, "Spatiotemporal data mining: A computational perspective," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2306–2338, 12 2015.

[2] M. Lu, E. Hamunyela, J. Verbesselt, and E. Pebesma, "Dimension Reduction of Multi-Spectral Satellite Image Time Series to Improve Deforestation Monitoring," *Remote Sensing*, vol. 9, no. 10, p. 1025, oct 2017. [Online]. Available: http://www.mdpi.com/2072-4292/9/10/1025

[3] T. Zhang, J. Su, C. Liu, W. H. Chen, H. Liu, and G. Liu, "Band selection in sentinel-2 satellite for agriculture applications," in *ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing*. Institute of Electrical and Electronics Engineers Inc., oct 2017.

[4] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," may 2020. [Online]. Available: http://arxiv.org/abs/2005.01094

[5] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search," in *Springer*, 2019, pp. 69–86.

[6] DARPA, "TL Transfer Learning Proposer Information Pamphlet (PIP) for Broad Agency Announcement," Tech. Rep., 2005.

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *34th International Conference on Machine Learning, ICML 2017*, vol. 3, no. 2015, pp. 1856–1868, 3 2017. [Online]. Available: http://arxiv.org/abs/1703.03400

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, "Training general representations for remote sensing using in-domain knowledge," 2020.

[10] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning*, 2019.

[11] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2962–2970. [Online]. Available: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

[12] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. of KDD-2013*, 2013, pp. 847–855.

[13] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16. New York, NY, USA: ACM, 2016, pp. 485–492. [Online]. Available: http://doi.acm.org/10.1145/2908812.2908918

[14] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 1946–1956.

[15] M. Physics, "The Electromagnetic Spectrum | Mini Physics - Learn Physics." [Online]. Available: https://www.miniphysics.com/electromagnetic-spectrum{_}25.html

[16] ESA, "ESA - Eduspace EN - Home - What is remote sensing?" 2010.

[17] "MOOC Copernicus (Massive Open Online Course)." [Online]. Available: https://mooc.copernicus.eu/

[18] USGS, "Comparison of Landsat 7 and 8 bands with Sentinel-2." [Online]. Available: https://www.usgs.gov/media/images/comparison-landsat-7-and-8-bands-sentinel-2

[19] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *CoRR*, vol. abs/1611.02167, 2016. [Online]. Available: http://arxiv.org/abs/1611.02167

[20] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1eYHoC5FX

[21] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via lamarckian evolution," 2019.

[22] R. Miikkulainen, J. Z. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," *CoRR*, vol. abs/1703.00548, 2017. [Online]. Available: http://arxiv.org/abs/1703.00548

[23] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," 2018.

[24] Z. Zhong, J. Yan, W. Wu, J. Shao, and C. Liu, "Practical block-wise neural network architecture generation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2423–2432.

[25] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *CoRR*, vol. abs/1707.07012, 2017. [Online]. Available: http://arxiv.org/abs/1707.07012

[26] T. Wei, C. Wang, Y. Rui, and C. W. Chen, "Network morphism," *CoRR*, vol. abs/1603.01670, 2016. [Online]. Available: http://arxiv.org/abs/1603.01670

[27] T. Elsken, J.-H. Metzen, and F. Hutter, "Simple and efficient architecture search for convolutional neural networks," 2017.

[28] B. Zhang, Z. Chen, D. Peng, J. A. Benediktsson, B. Liu, L. Zou, J. Li, and A. Plaza, "Remotely sensed big data: Evolution in model development for information extraction [point of view]," pp. 2294–2301, 12 2019.

[29] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang, "AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 8 2016. [Online]. Available: http://arxiv.org/abs/1608.05167http://dx.doi.org/10.1109/TGRS.2017.2685945

[30] G. Cheng, J. Han, and X. Lu, "Remote Sensing Image Scene Classification: Benchmark and State of the Art," pp. 1865–1883, 10 2017.

[31] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271619301108

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[33] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Tech. Rep., 2015. [Online]. Available: http://www.robots.ox.ac.uk/

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[36] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery," *Remote Sensing*, vol. 10, no. 7, p. 1119, 7 2018. [Online]. Available: http://www.mdpi.com/2072-4292/10/7/1119

[37] K. Nogueira, J. A. Dos Santos, T. Fornazari, T. S. F. Silva, L. P. Morellato, and R. d. S. Torres, "Towards vegetation species discrimination by using data-driven descriptors," in *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*. Ieee, 2016, pp. 1–6.

[38] J. Vanschoren, "Meta-learning," in *AutoML: Methods, Sytems, Challenges*, 2018, pp. 39–68.

[39] M. Rußwurm, S. Wang, M. Körner, and D. Lobell, "Meta-Learning for Few-Shot Land Cover Classification," Tech. Rep.

[40] E. Ferreira, M. Brito, R. Balaniuk, M. S. Alvim, and J. A. dos Santos, "BrazilDAM: A Benchmark dataset for Tailings Dam Detection," 3 2020. [Online]. Available: http://arxiv.org/abs/2003.07948

[41] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," 2017.

[42] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat - a learning framework for satellite imagery," 2015.

[43] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, may 2002. [Online]. Available: https://link.springer.com/article/10.1023/A:1013689704352

[44] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, M. Urban, M. Burkart, M. Dippel, M. Lindauer, and F. Hutter, "Towards automatically-tuned deep neural networks," in *AutoML: Methods, Sytems, Challenges*, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Springer, Dec. 2018, ch. 7, pp. 141–156, to appear.

[45] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2016.

[46] O. A. Penatti, K. Nogueira, and J. A. Dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 44–51.

[47] F. Yang, W. Zhang, L. Tao, and J. Ma, "Transfer Learning Strategies for Deep Learning-based PHM Algorithms," *Applied Sciences*, vol. 10, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2361

[48] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," 2018.

[49] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," *CoRR*, vol. abs/1902.06148, 2019.

[50] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Keras Tuner," https://github.com/keras-team/keras-tuner, 2019.

[51] "Keras Applications." [Online]. Available: https://keras.io/api/applications/{#}build-inceptionv3-over-a-custom-input-tensor

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," 2016.

[53] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)*, 2010.

[54] X. Zhu, J. Hu, C. Qiu, Y. Shi, H. Bagheri, J. Kang, H. Li, L. Mou, G. Zhang, M. Häberle, S. Han, Y. Hua, R. Huang, L. Hughes, Y. Sun, M. Schmitt, and Y. Wang, "So2sat lcz42," 2018. [Online]. Available: https://mediatum.ub.tum.de/1454690

[55] "CORINE Land Cover — Copernicus Land Monitoring Service." [Online]. Available: https://land.copernicus.eu/pan-european/corine-land-cover

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June. IEEE Computer Society, oct 2015, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1409.4842v1

[57] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[58] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations (ICLR) (Banff)*, 12 2013.

[59] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093, 2014. [Online]. Available: http://arxiv.org/abs/1408.5093

[60] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[61] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam convolutional block attention module," 2018.

[62] TensorFlow, "TensorFlow Hub." [Online]. Available: https://www.tensorflow.org/hub

[63] D. Soekhoe, P. Van Der Putten, and A. Plaat, "On the Impact of data set Size in Transfer Learning using Deep Neural Networks," Tech. Rep.

[64] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" Tech. Rep.

[65] M. Huisman, J. N. van Rijn, and A. Plaat, "A survey of deep meta-learning," 2020.