ISPRS Journal of Photogrammetry and Remote Sensing 197 (2023) 18–35

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Current trends in deep learning for Earth Observation: An open-source benchmark arena for image classification

Ivica Dimitrovski [a,b], Ivan Kitanovski [a,b], Dragi Kocev [a,c,*], Nikola Simidjievski [a,c,d,*]

[a] Bias Variance Labs, d.o.o., Ljubljana, Slovenia
[b] Faculty of Computer Science and Engineering, University Ss Cyril and Methodius, Skopje, North Macedonia
[c] Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia
[d] Department of Computer Science and Technology, University of Cambridge, Cambridge, United Kingdom

ARTICLE INFO

Keywords:
Deep learning (DL)
Earth observation (EO)
Image classification
Benchmark study

ABSTRACT

We present *AiTLAS: Benchmark Arena* – an open-source benchmark suite for evaluating state-of-the-art deep learning approaches for image classification in Earth Observation (EO). To this end, we present a comprehensive comparative analysis of more than 500 models derived from ten different state-of-the-art architectures and compare them to a variety of multi-class and multi-label classification tasks from 22 datasets with different sizes and properties. In addition to models trained entirely on these datasets, we benchmark models trained in the context of transfer learning, leveraging pre-trained model variants, as it is typically performed in practice. All presented approaches are general and can be easily extended to many other remote sensing image classification tasks not considered in this study. To ensure reproducibility and facilitate better usability and further developments, *all of the experimental resources* including the trained models, model configurations, and processing details of the datasets (with their corresponding splits used for training and evaluating the models) are *publicly available on the repository*: https://github.com/biasvariancelabs/aitlas-arena.

1. Introduction

Recent trends in machine learning (ML) have ushered in a new era of image-data analyses, repeatedly achieving great performance across a variety of computer-vision tasks in different domains (Khan et al., 2020, 2021). Deep learning (DL) approaches have been at the forefront of these efforts — leveraging novel, modular and scalable deep neural network (DNN) architectures able to process large amounts of data. The inherent capabilities of these approaches also extend to various areas of remote sensing, in particular Earth Observation (EO), employed for analyzing different types of large-scale satellite data (Ball et al., 2017). Many of these contributions are instances of image-scene classification, such as land-use and/or land-cover (LULC) identification tasks, focusing on image-scene analyses, characterizations, and classifications of changes in the landscape caused either by human activities or by the elements.

Historically, from the perspective of ML, many of these tasks have been addressed mostly through the paradigms of either pixel-level (Tuia et al., 2009; Li et al., 2014) or object-level classification tasks (Blaschke, 2010). The former refers to classification tasks focusing on each pixel in the image, associating it with the appropriate semantic label. Such approaches typically do not scale well on high-resolution images, but more importantly, many times struggle to capture more high-level patterns in the image that can span over many pixels (Blaschke and Strobl, 2001). The latter, object-level classification methods, focus on analyzing distinguishable and meaningful objects in the image (as a collection of pixels) rather than independent pixels. This generally allows for better scalability and performance; however, such approaches may struggle with images containing more diverse and hardly-distinguishable objects, which prevail in most high-resolution remote-sensing data. Methods based on pixel-level and object-level paradigms have shown decent performance and are still actively researched, mostly as instances of image segmentation and object detection tasks. More recently, however, methods based on a new paradigm of scene-level classification (Cheng et al., 2017; Yang and Newsam, 2010) have shown significant performance improvements, focusing on learning semantically meaningful representations of more sophisticated patterns in an image by leveraging the capabilities of deep learning.

Deep learning approaches have been successfully applied in various remote-sensing scenarios, be it learning models from scratch or via transfer learning (Marmanis et al., 2016; Chen et al., 2019), in a fully supervised or self-supervised setting (Castillo-Navarro et al., 2022; Wang et al., 2022a), exploiting the heterogeneity (Neumann et al.,

* Corresponding authors at: Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia.
E-mail addresses: dragi.kocev@ijs.si (D. Kocev), nikola.simidjievski@ijs.si (N. Simidjievski).

https://doi.org/10.1016/j.isprsjprs.2023.01.014
Received 16 September 2022; Received in revised form 5 December 2022; Accepted 20 January 2023
Available online 1 February 2023

2020) and temporal properties (Ienco et al., 2017) of the available data. As a result, this synergy of accurate DL approaches, on the one hand, and accessible high-resolution aerial/satellite imagery, on the other, has led to important contributions in various domains ranging from agriculture (Chlingaryan et al., 2018; Johnson et al., 2016; Xu et al., 2021a), ecology (Ayhan et al., 2020; Jo et al., 2018), geology (Shirmard et al., 2022) and meteorology (Zhang et al., 2018; Sadeghi et al., 2019; Chen et al., 2019) to urban mapping/planning (Longbotham et al., 2012; Lv et al., 2022; Huang et al., 2018) and archaeology (Somrak et al., 2020).

Nevertheless, most of these efforts typically focus on very narrow tasks stemming from domain-specific and/or spatially constrained datasets. As a result, models have been evaluated in different settings and under different conditions (Cheng et al., 2020) — hardly reproducible and comparable. These persistent challenges, akin to a lack of standardized and consistent validation and evaluation of novel approaches, have also been identified by the community (Schneider et al., 2022). Citing the lack of available documentation on the design and evaluation of the employed machine learning approaches, the community highlights the urgent need for standardized benchmarks that will not only enable proper and fair model comparison across datasets and similar tasks but will also facilitate faster progress in designing better and more accurate modeling approaches.

Motivated by this, in this work, we introduce *AiTLAS: Benchmark Arena* — an *open-source EO benchmark suite* for evaluating state-of-the-art DL approaches for EO image classification. To this end, we present extensive comparative analyses of models derived from ten different state-of-the-art architectures, comparing them on a variety of multi-class and multi-label classification tasks from 22 datasets with different sizes and properties. We benchmark models trained from scratch and in the context of transfer learning, leveraging pre-trained model variants as it is typically performed in practice. While in this work, we mainly focus on EO-image classification tasks, such as LULC, all presented approaches are general and easily extendable to other remote-sensing image classification tasks. More importantly, to ensure reproducibility, facilitate better usability, and further exploitation of the results from our work, we provide *all of the experimental resources* — freely available on our repository.[1] The repository includes the complete study details, such as the trained models, model parameters, train/evaluation configurations, and measured performance scores, as well as the details on all of the datasets and their prepossessed versions (with the appropriate train/validation/test splits) used for training and evaluating the models.

To our knowledge, we present a unique systematic review and evaluation of different state-of-the-art DL methods in the context of EO image classification across many classification problems — benchmarked in the same conditions and using the same hardware. Related efforts, while relevant, have mainly focused on evaluating approaches on particular datasets (Cheng et al., 2017, 2020; Papoutsis et al., 2022; Xia et al., 2017); evaluating different aspects of method-design (Zhai et al., 2019; Neumann et al., 2020) relevant to remote-sensing classification tasks; or providing a more general overview of the common tasks at hand Zhang et al. (2016), Zhu et al. (2017). In particular, Cheng et al. (2017) introduce a dataset and surveys several ML representation-learning approaches commonly used for remote-sensing classification tasks, comparing their performance when combined with traditional convolutional neural network (CNN) architectures. Xia et al. (2017) also introduce a benchmark dataset for aerial-image classification, providing a comparison similar to Cheng et al. (2017) of representation-learning approaches combined with three deep networks. Another, more recent study (Cheng et al., 2020), discusses and compares more recent DL approaches and surveys several applications on three different datasets. In particular, the authors showcase the performance of the different methods for each dataset, as reported in the respective papers. The underlying, persistent conclusions from these studies show that model performances are associated with a particular dataset and study design, presenting difficulties for fair and general model comparisons. This is expected, but in our work, we seek to remedy this issue by training and evaluating all models under the same conditions.

In this context, our work is related to one of Zhai et al. (2019), which presents a large-scale study on more recent representation-learning approaches, benchmarking different aspects of method design and model parameters. However, Zhai et al. (2019) considers a relatively broad scope of different datasets with only a few relevant to remote-sensing and LULC classification. Neumann et al. (2020) present a large-scale study on five different benchmark datasets; however, they investigate the effect of transfer learning on these tasks. More specifically, they evaluate different variants of the same model architecture, trained under different circumstances, rather than comparing different model architectures. Another related study by Stewart et al. (2021) reports on the comparison of different variants of ResNets on EO-image classification tasks from four datasets. More recently, and arguably most related to our work in terms of the number of evaluated models, Papoutsis et al. (2022) present an extensive empirical evaluation of different state-of-the-art DL architectures suitable for EO-image classification tasks, specifically LULC tasks, focusing exclusively on the BigEarthNet (Sumbul et al., 2021) dataset. Namely, the authors benchmark different classes of model architectures across different criteria and introduce an efficient and well-performing model tailored specifically for BigEarthNet.

In this work, we go beyond all the aforementioned studies, significantly extending the scope of research in two directions: the number of model architectures (and model variants) being evaluated and the datasets being considered. This results in assessing more than 500 different models with different architectures, varying designs, and learning paradigms across 22 datasets. We provide essential study-design principles and model training details that will aid in more systematic and rigorous experiments in future work. The proposed *AiTLAS: Benchmark Arena* builds on the AiTLAS toolbox (Dimitrovski et al., 2022)[2] — a recent open-source library for exploratory and predictive analysis of satellite imaginary pertaining to different remote-sensing tasks. AiTLAS implements various methods and libraries for data handling, processing, and analysis, with PyTorch (Paszke et al., 2019) as a backbone for constructing and learning DL models. By having all of the methods and datasets under the same umbrella, we provide the means for a fair, unbiased, and reproducible comparison of approaches across different criteria that include: overall model performance, data- and task-dependent model performance, model size, and learning efficiency as well as the effect of transfer learning via model pre-training.

The results, summarized in Fig. 1, show that many of the current state-of-the-art architectures for vision tasks can lead to decent predictive performance when applied to EO image classification tasks. While, in some cases, training models from scratch can lead to satisfactory performance, using pre-trained models and fine-tuning them on each dataset leads to the best performance overall. We observed this in all cases, regardless of the dataset properties, the type of classification tasks, or the model architecture. We found more considerable performance gains on tasks from smaller datasets, which, as expected, benefited more from the pre-training process than models trained on larger datasets. In terms of model architectures, our experiments showed that pre-trained Transformer models, i.e. both Vision Transformer (Dosovitskiy et al., 2020) and Swin Transformer (Liu et al., 2022a) models, were, in general, able to achieve the best performance. Specifically, Vision Transformer models showed the best performance on various multi-classification tasks, while Swin Transformer models led to much better performance on multi-label tasks, albeit at the cost of

[1] https://github.com/biasvariancelabs/aitlas-arena

[2] https://aitlas.bvlabs.ai

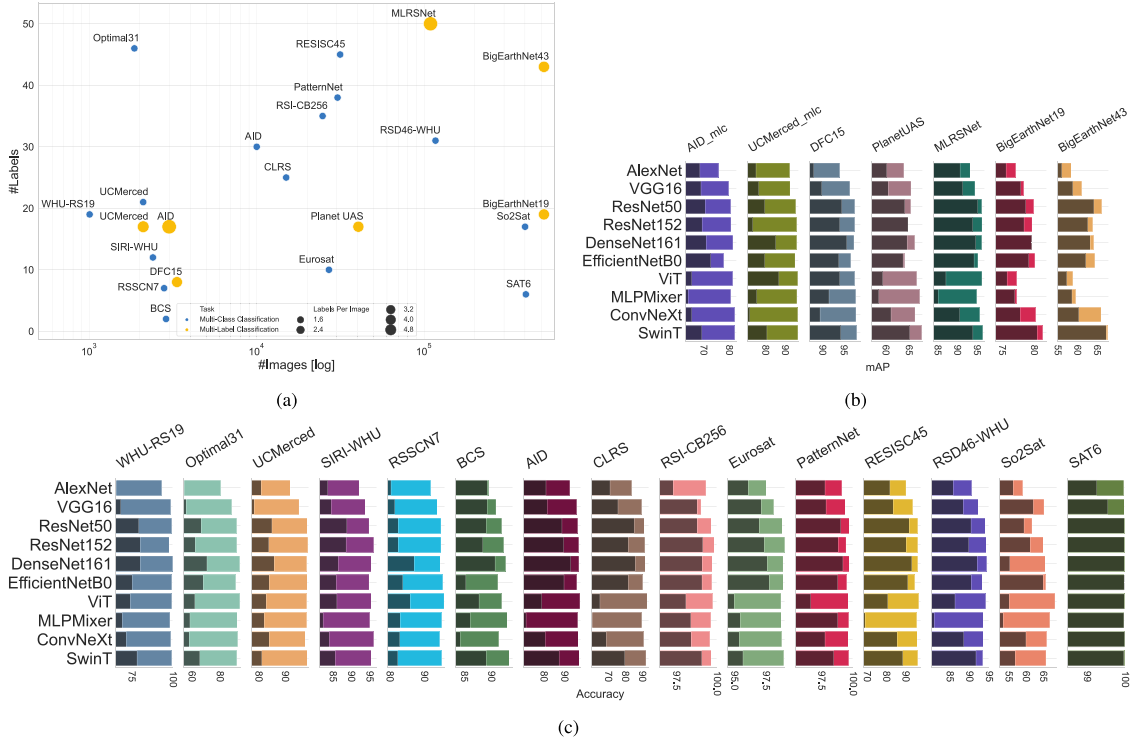


Fig. 1. Overview of the study: We benchmarked more than 500 models from 10 different model architectures on tasks from **(a)** 22 datasets with different sizes and properties; comparing them on **(b)** multi-label and **(c)** multi-class classification tasks. We evaluate two versions of each model architecture: (i) trained from scratch (denoted with *darker shading*) and (ii) pre-trained on ImageNet-1K (denoted with *lighter shading*). Note the varying scales in (b) and (c), made purposely for better visibility. Detailed results are presented in Section 4 and Appendices B, C and D in the Supplementary material.

much longer training time. Throughout the paper, we further evidence and discuss these findings.

In summary, in this paper, we make several contributions. Specifically, we:

- Introduce *AiTLAS: Benchmark Arena* — an open-source benchmark suite that enables standardized evaluation of machine learning models for Earth Observation (EO) applications;
- Provide study-design principles for training and evaluating state-of-the-art deep learning models on various supervised EO image classification tasks from 22 datasets with different sizes and properties;
- Implement and benchmark more than 500 models stemming from 10 state-of-the-art architectures, including models trained from scratch and their pre-trained variants;
- Investigate models' generalization abilities to unseen in-domain datasets;
- Evaluate different pre-training strategies that relate to pre-training models from in-domain EO datasets and investigate their effect on the downstream predictive performance;
- Discuss common issues that typically affect the models' performance, specifically in the context of EO tasks.
- Provide open-source access to all experimental details, including trained models, dataset details, train/evaluation configurations, and detailed performance scores.

2. Data & models

2.1. Data description

With the ever-growing availability of remote sensing data, there has been a significant effort by many research groups to prepare, label, and provide proper datasets that will support the development and evaluation of sophisticated machine learning methods. While there are many such datasets, both proprietary and publicly available, in this work, we focus on the latter — open-access publicly available dataset. Given this criterion, we select 22 open-access datasets usually considered in different EO studies for benchmarking DL approaches. The selected datasets have varying sizes (number of images), varying image types, image sizes, and formats, and, more importantly, related to different classification tasks.

Namely, we consider datasets related to multi-class and multi-label classification tasks, mainly addressing LULC applications. The objective of *multi-class classification* tasks is to predict one (and only one) class (label) from a set of predefined classes for each image in a dataset. *Multi-label classification*, on the other hand, refers to predicting multiple labels from a predefined set of labels for each image in the dataset (Tsoumakas and Katakis, 2009) (e.g., an image can belong to more than one class simultaneously). In our experimental study, we consider 15 multi-class and seven multi-label datasets.

Tables 1 and 2 summarizes the properties of the considered multi-class (MCC) and multi-label (MLC) classification datasets, respectively. The number of images across datasets is quite diverse, ranging from datasets with $\sim 2K$ images to datasets with $\sim 500K$ images. This also extends towards the number of labels per image, ranging from 2 to 60. Fig. 1(a) visualizes the datasets with respect to their sizes, with the x-axis denoting the number of images (on a log scale) and the y-axis indicating the number of labels (with marker size denoting the number of labels per image) for each of the different datasets. Most of the datasets consist of Aerial RGB images (with only a few comprised of satellite multi-spectral data) that are different in spatial resolution, size, and format. Finally, we note the datasets that include predefined splits (for training, validation, and testing) given by the original authors and provide the splits for the ones that are missing, as further discussed in Section 3.1. An extended description of each dataset is given in the Supplementary material (Appendix D in the Supplementary material).

Table 1
Summary of the multi-class classification (MCC) datasets.

Name	Image type	#Images	Image size	Spatial resolution	#Labels	Predefined splits	Image format
UC Merced (Yang and Newsam, 2010)	Aerial RGB	2100	256×256	0.3 m	21	No	tif
WHU-RS19 (Xia et al., 2010)	Aerial RGB	1005	600×600	0.5 m	19	No	jpg
AID (Xia et al., 2017)	Aerial RGB	10000	600×600	0.5 m - 8 m	30	No	jpg
Eurosat Helber et al. (2019)	Sat. Multispectral	27000	64×64	10 m	10	No	jpg/tif
PatternNet (Zhou et al., 2018)	Aerial RGB	30400	256×256	0.06 m - 4.69 m	38	No	jpg
Resisc45 (Cheng et al., 2017)	Aerial RGB	31500	256×256	0.2 m - 30 m	45	No	jpg
RSI-CB256 (Li et al., 2020a)	Aerial RGB	24747	256×256	0.3 - 3 m	35	No	tif
RSSCN7 (Zou et al., 2015)	Aerial RGB	2800	400×400	n/a	7	No	jpg
SAT6 (Basu et al., 2015)	RGB + NIR	405000	28×28	1 m	6	Yes	mat
Siri-Whu (Zhu et al., 2016)	Aerial RGB	2400	200×200	2 m	12	No	tif
CLRS (Li et al., 2020b)	Aerial RGB	15000	256×256	0.26 m - 8.85 m	25	No	tif
RSD46-WHU (Long et al., 2017)	Aerial RGB	116893	256×256	0.5 m - 2 m	46	Yes	jpg
Optimal 31 (Wang et al., 2019)	Aerial RGB	1860	256×256	n/a	31	No	jpg
Brazilian Coffee Scenes (BSC) (Penatti et al., 2015)	Aerial RGB	2876	64×64	10 m	2	No	jpg
So2Sat Zhu et al. (2020)	Sat. Multispectral	400673	32×32	10 m	17	Yes	h5

Table 2
Summary of the multi-label classification (MLC) datasets.

Name	Image type	#Images	Image size	Spatial resolution	#Labels	#Labels per image	Predefined splits	Image format
UC Merced (mlc) (Chaudhuri et al., 2018)	Aerial RGB	2100	256×256	0.3 m	17	3.3	No	tif
MLRSNet (Qi et al., 2020)	Aerial RGB	109161	256×256	0.1 m - 10 m	60	5.0	No	jpg
DFC15 (Hua et al., 2019)	Aerial RGB	3342	600×600	0.05 m	8	2.8	Yes	png
AID (mlc)(Hua et al., 2020)	Aerial RGB	3000	600×600	0.5 m - 8 m	17	5.2	Yes	jpg
PlanetUAS (Planet and SCOON, 2022)	Aerial RGB	40479	256×256	3 m	17	2.9	No	jpg/tiff
BigEarthNet 19 (Sumbul et al., 2021)	Sat. Multispectral	519284	20×20 60 × 60 120 × 120	60 m 20 m 10 m	19	2.9	Yes	tif, json
BigEarthNet 43 (Sumbul et al., 2019)	Sat. Multispectral	519284	20×20 60 × 60 120 × 120	60 m 20 m 10 m	43	3.0	Yes	tif, json

2.2. Model architectures

Current trends in EO image classification leverage the capabilities of DL architectures for computer vision, learning data representations that often lead to superior predictive performance. We recognize that there are many different approaches stemming from different model architectures and model variants. These can differ in various 'finer' details (e.g., number and width of layers, hyper-parameter values, and learning regimes), often developed for a particular task. Rather than seeking a state-of-the-art performance for each EO problem/dataset, in this study, we are interested in providing a more general evaluation framework and benchmarking models by analyzing their characteristics and unique properties through the lens of their predictive performance and learning efficiency across all datasets.

Therefore, our model-architecture (and parameter) choices are motivated by different architecture 'classes', such as the traditional convolutional architectures and the more recent attentional and multilayer-perceptron (MLP) architectures. This renders models with different sizes, training/inference time, different abilities in a transfer-learning setting, etc. More specifically, we investigate several architectures which have been traditionally used for EO image classification tasks, such as: AlexNet (Krizhevsky et al., 2012), VGG16 (Simonyan and Zisserman, 2014), ResNet (He et al., 2016) and DenseNet (Huang et al., 2017). Moreover, we investigate more recent architectures, which include EfficientNet (Tan and Le, 2019), ConvNeXt (Liu et al., 2022b), Vision Transformer (Dosovitskiy et al., 2020), Swin Transformer (Liu et al., 2022a) and MLPMixer (Tolstikhin et al., 2021), that have shown state-of-the-art performance in various vision tasks. In the following, we provide a brief overview of these architectures, highlighting their properties in Table 3.

The first class of models we consider relies on convolutional architectures, which, in recent years, have driven many of the advances in computer vision. The architecture of convolutional neural networks (CNN) consists of many (hidden) layers stacked together, designed to process (image) data in the form of multiple arrays. Most typically, CNNs consist of a series of convolutional layers, which apply convolution operation (passing the data through a kernel/filter), forwarding the output to the next layer. This serves as a mechanism for constructing feature maps, with former layers typically learning low-level features (such as edges and contours), subsequently increasing the complexity of the learned features with deeper layers in the network. Convolutional layers are typically followed by pooling operations, which serve as a downsampling mechanism by aggregating the feature maps through local non-linear operations. In turn, these feature maps are fed to fully-connected layers, which perform the ML task at hand — in this case, classification. All the layers in a network employ an activation function. In practice, the intermediate, hidden layers employ a non-linear function such as rectified linear unit (ReLU) or Gaussian Error Linear Unit (GELU) as common choices. The choice of activation function in the final layer relates to the tasks at hand, typically a sigmoid function in the case of classification. CNN architectures can also include different normalization and/or dropout operators embedded among the different layers, which can further improve the network's performance.

CNN architectures have been widely researched, with models applied in many contexts of remote sensing, and in particular EO image classification (Chen et al., 2019; Weng et al., 2017; Castelluccio et al., 2015; Papoutsis et al., 2022). This includes *AlexNet* (Krizhevsky et al., 2012), a pioneering architecture that introduced and successfully demonstrated the utility of the CNN blueprint, mentioned earlier, for computer vision tasks. Namely, even though the architecture of AlexNet has a modest depth (relative to more recent architectures) consisting of eight layers, it remains an efficient baseline approach for a variety of EO tasks (Cheng et al., 2017; Marmanis et al., 2016), leading to decent performance, especially when pre-trained with large image datasets (Han et al., 2017). We also consider the more sophisticated *VGG* (Simonyan and Zisserman, 2014), which employs a deeper architecture inspired by AlexNet. VGG has shown great performance

Table 3
Summary of the representative model architectures considered in this study.

Model	Year	#Layers	#Parameters	FLOPS	Based on
AlexNet (Krizhevsky et al., 2012)	2012	8	$\sim 57 \cdot 10^6$	0.72 G	Marcel and Rodriguez (2010)
VGG16 (Simonyan and Zisserman, 2014)	2014	16	$\sim 134.2 \cdot 10^6$	15.47 G	Marcel and Rodriguez (2010)
ResNet50 (He et al., 2016)	2015	50	$\sim 23.5 \cdot 10^6$	4.09 G	Marcel and Rodriguez (2010)
ResNet152 (He et al., 2016)	2015	152	$\sim 58.1 \cdot 10^6$	11.52 G	Marcel and Rodriguez (2010)
DenseNet161 (Huang et al., 2017)	2017	161	$\sim 26.4 \cdot 10^6$	7.73 G	Marcel and Rodriguez (2010)
EfficientNet B0 (Tan and Le, 2019)	2019	237	$\sim 5.2 \cdot 10^6$	0.39 G	Marcel and Rodriguez (2010) version: B0
Vision Transformer (ViT) (Dosovitskiy et al., 2020)	2020	12	$\sim 86.5 \cdot 10^6$	17.57 G	Wightman (2019) version: b_16_224
MLPMixer (Tolstikhin et al., 2021)	2021	12	$\sim 59.8 \cdot 10^6$	12.61 G	Wightman (2019) version: b_16_224
ConvNeXt (Liu et al., 2022b)	2022	174	$\sim 28 \cdot 10^6$	4.46 G	Marcel and Rodriguez (2010) version: tiny
Swin Transformer (Liu et al., 2022a)	2022	24	$\sim 49.7 \cdot 10^6$	11.55 G	Marcel and Rodriguez (2010) version: v2 small

in a variety of vision tasks, including EO-image classification problems (Kang et al., 2018; Hu et al., 2015; Zhou et al., 2018). There are two variants of VGG in practice, VGG16 and VGG19; both extend AlexNet mainly by increasing the depth of the network with 13 and 16 convolutional layers, respectively. In this study, we evaluate the performance of the former *VGG16*. VGGs employ kernels with smaller sizes than the ones typically used in AlexNet, demonstrating that stacking multiple smaller kernels are better able to extract more complex representations than one larger filter. While, in general, increasing the network depth by adding convolutional layers helps for learning more complex and more informative representations thereof, in practice, this can lead to several issues, such as the vanishing gradient problem (Goodfellow et al., 2016), which impairs the network training.

The Residual neural networks (*ResNets*) (He et al., 2016; Zagoruyko and Komodakis, 2016) tackle this issue explicitly by employing skip connections between blocks, therefore enabling better backprop gradient flow, better training, and, in general, better predictive performance. ResNet architecture follows a typical CNN blueprint: Stacking residual blocks (typically same-size CNN layers) and convolutional blocks (typically introducing a bottleneck via different-size CNN layers) together, followed by fully-connected layers. By employing skip connections, the ResNet architecture allows stacking multiple layers in a block, therefore training models with much deeper architectures. Here we investigate two variants with varying depths, *ResNet50* and *ResNet152*, with 50 and 152 layers, respectively. Since their inception, ResNets have been a prevalent choice in practice. This also extends towards their utility for EO tasks, applied in the context of image classification and semantic segmentation (Cheng et al., 2017; Audebert et al., 2018; Stewart et al., 2021; Papoutsis et al., 2022). Dense Convolutional Networks (*DenseNets*) (Huang et al., 2017) are another well-performing architecture variant of ResNets that has demonstrated state-of-the-art results on many classification tasks, including applications in the domain of remote sensing (Zhang et al., 2019; Tong et al., 2020; Chen and Tsou, 2021). As the name suggests, DenseNets consist of dense blocks, where each layer is connected to every preceding layer, taking an additional (channel-wise) concatenated input of the feature maps learned in the former layers. This differs from the ResNets, which propagate (element-wise) aggregated feature maps through the network layers. The architecture of DenseNets encourages feature reuse throughout the network, leading to well-performing and more compact models (with fewer trainable parameters than a ResNet of equivalent size), albeit at the cost of increased memory during training.

EfficientNets (Tan and Le, 2019) are a recent class of lightweight architecture that alleviate such common computational difficulties, typical when scaling deep architectures on larger and/or harder problems. Namely, rather than scaling the architecture in one aspect of increasing the depth (number of layers) (He et al., 2016), width (number of channels) (Zagoruyko and Komodakis, 2016) or (input image) resolution (Lin et al., 2017); EfficientNets implement compound scaling, that uniformly scales the architecture along the three dimensions simultaneously. Compound scaling seeks an optimal balance between these three dimensions, given the available resources and the task at hand. In turn, such an approach leads to substantially smaller models (than CNN variants of equivalent performance) while retaining state-of-the-art predictive performance. In the context of EO tasks, (variants of) EfficientNets have been successfully applied in different settings (Liu et al., 2020a; Tian et al., 2020; Alhichri et al., 2021; Chen and Tsou, 2021), and have also been thoroughly investigated in the context of multi-label image classification tasks from BigEarthNet (Papoutsis et al., 2022). While there are eight variants of EfficientNets, differing in the size and complexity of the architectures, here we investigate the performance of the baseline *EfficientB0* architecture with 5.2M parameters, substantially lower than any of the other competing model architectures. Most recently, (Liu et al., 2022b) introduce *ConvNeXt*, a novel class of convolutional architectures that leverage various successful design decisions of preceding convolutional and attentional architectures typically applied for vision tasks. Namely, ConvNeXt models implement various techniques at different levels: from reconfiguring details like activation functions and normalization layers, to redesigning more general architecture details related to residual/convolutional blocks, to modifications in the training strategies. This, in turn, leads to models that achieve good predictive performance, not only better than popular models from the class of convolutional architectures but also better than the more recent attentional architectures, such as transformers, discussed next. While there are several variants of the ConvNeXt architecture that mainly differ in their size, in this study, we evaluate the performance of the smallest variant, *ConvNeXt_tiny*. Note that, to our knowledge, this is the first application of ConvNeXt on EO-image classification tasks.

We next take the notion of the recent success of the class of attentional network architectures and study the performance of *Vision Transformers* (ViT) (Dosovitskiy et al., 2020) in the context of EO-image classification tasks. Namely, ViTs inspire by the popular NLP (natural language processing) Transformer architecture (Devlin et al., 2018), leveraging an attention mechanism for vision tasks. Much like the original Transformer that seeks to learn implicit relationships in sequences of word-tokens via multi-head self-attention, ViTs focus on learning such relationships between image patches. Typically they employ a standard transformer encoder that takes a lower-dimensional (linear) representation of these image patches together with additional positional embedding from each, in turn, feeding the encoder-output to a standard MLP head. ViTs have shown excellent performance on various vision tasks, particularly when combined with pre-training from large datasets. This also includes several applications in remote sensing (Bazi et al., 2021; Papoutsis et al., 2022; Gong et al., 2022).

More recent and sophisticated, attentional network architectures such as the *Swin Transformers* (SwinT) (Liu et al., 2021, 2022a) rely on additional visual inductive biases by introducing hierarchy, translation invariance, and locality in the attention mechanism. Like ViTs, SwinT architectures also attempt to learn relationships between image patches but operate on image windows (a group of neighboring image patches). SwinTs focus on computing attention between patches within a window (locality), in turn shifting these windows to allow learning of cross-window attention (translation invariance). Starting with windows with smaller patches and increasing their size at each subsequent stage, SwinTs also allow for learning representations at different granularity

(hierarchy). All this leads to SwinTs performing well in practice on a variety of vision tasks, including in the domain of remote sensing (Scheibenreif et al., 2022; Zhang et al., 2022a; Wang et al., 2022b), often outperforming ViTs and other convolutional architectures. In this study, we evaluate the 'small' architecture variant of the latest version of Swin Transformers V2 (Liu et al., 2022a).

In the context of vision tasks, an attention mechanism can be achieved differently (e.g., attending over channels and/or spatial information, etc.) and even employed with typically convolutional architectures (Liu et al., 2020a; Xu et al., 2021b; Alhichri et al., 2021). One alternative that builds only on the classical MLP architecture is the *MLPMixer* (Tolstikhin et al., 2021). Namely, similar to a transformer architecture, an MLPMixer operates on image patches; and contains two main components: A block of MLP layers for 'mixing' the spatial, patch-level information on every channel; and a block of MLP layers for 'mixing' the channel-information of an image. This renders lightweight models, with performance on par with many much more sophisticated architectures, on a variety of vision problems, both more general as well as specific EO tasks (Meng et al., 2021; Papoutsis et al., 2022; Gong et al., 2022). We employ an MLPMixer with an input size of 224 × 224 and a patch resolution of 16 × 16 pixels.

From each of the ten highlighted architectures, we evaluate two model versions: trained entirely on a given dataset and fine-tuned models that have been pre-trained on a different image dataset. This results in comparing 20 models on each predictive task, which are available on our repository.

3. Experimental design

3.1. Training and evaluation protocol

To establish a unified evaluation framework and support the results' reproducibility, we generated train, validation, and test splits using 60%, 20%, and 20% fractions, respectively. All of the data splits were obtained using stratified sampling. This technique ensures that the distribution of the target variable(s) among the different splits remains the same (Sechidis et al., 2011). We performed such stratification for all datasets except the ones which include predefined splits provided by the original authors. More specifically, for the *BigEarthNet* and *So2Sat* datasets, we use the train, validation, and test splits as provided in Sumbul et al. (2019, 2021), Zhu et al. (2020). Since *SAT6*, *RSD46-WHU*, *DFC15* and *AID* datasets consist only of predefined train and test splits, we further take 20% from the train part for validation. Finally, note that the PlanetUAS dataset was part of a competition, and as such, the test data is not publicly available. Therefore, we generated train, validation, and test splits from the original train data using the 60%, 20%, and 20% fractions, respectively.

All the models were trained using the same train splits, with parameters selection/search performed using the same validation splits. Additionally, to overcome over-fitting, we perform early stopping on the validation split for each dataset; the best checkpoint/model found (with the lowest validation loss) is saved and then applied to the original test split to obtain the final assessment of the predictive performance. All the train/validation/test splits for each dataset are available on our repository.

To better assess the generalization capabilities of the trained models, we evaluate their performance on different (in-domain) datasets not used for training. Specifically, we present two schemes of this evaluation: (1) performance measured on a holdout set compiled of test images with the same labels but from different datasets; (2) an exhaustive cross-dataset evaluation between pairs of datasets that contain the same labels. The former variant refers to a new test set consisting of 3216 images from the test splits of seven datasets (*RESISC45, UC Merced, CLRS, PatternNet, AID, RSI-CB256* and *WHU-RS19*) with labels present in all datasets (in our experiments, this results in five common labels: 'Forest', 'Parking', 'River', 'Harbor' and 'Beach'). We employ this evaluation setting only for multi-class classification tasks. In the latter variant, in a pairwise fashion, we evaluate every model on test splits from other datasets not used for training it. We measure the performance only on images with labels shared between the pairs of source (used for training the model) and target (used for evaluating the model) datasets. We employ this setting in both multi-class and multi-label classification scenarios. Note that in all cases, the models are only evaluated on the unseen datasets without additional fine-tuning. These configurations are also available on our repository.

During training, we perform *data augmentation* for each dataset by first resizing all the images to 256 × 256, followed by selecting a random crop of size 224 × 224. We then perform random horizontal and/or vertical flips. During evaluation/testing, we first resize the images to 256 × 256, followed by a central crop of size 224 × 224. We believe that this, in general, helps our models to generalize better on a given dataset. Also note that in the study, we are using only RGB images. In the case of the multispectral datasets (*Eurosat*, *So2Sat* and *BigEarthNet*), we computed the images in the RGB color space by combining the red (B04), green (B03), and blue (B02) bands. For the *Brazilian Coffee Scenes* dataset, we use images in green, red, and near-infrared spectral bands since these are most useful and representative for distinguishing vegetation areas, as suggested by the authors.

Since we train models on 22 datasets with a different number of classes, different training samples, and class distributions (as shown in Tables 1 and 2), we perform a hyperparameters search for each model and each dataset, to account for these variations. Namely, we search over different learning-rate values: 0.01, 0.001, and 0.0001. We use *ReduceLROnPlateau* as a learning scheduler which reduces the learning rate when the loss has stopped improving. Models often benefit from reducing the learning rate by a factor once learning stagnates. This scheduler tracks the values of the loss measure, reducing the learning rate by a given factor when there is no improvement for a certain number of epochs (denoted as 'patience'). In our experiments, we track the value of the validation loss with patience set to 5 and a reduction factor set to 0.1 (the new learning rate will be $lr * factor$). The maximum number of epochs is set to 100. Additionally, we also apply early stop criteria if no improvements in the validation loss are observed over 10 epochs. We use fixed values for some of the hyperparameters, such as batch size, which we set to 128. For optimization, we use *RAdam optimizer* (Liu et al., 2020b) without weight decay. RAdam is a variant of the standard Adam (Kingma and Ba, 2014), with a mechanism that rectifies the variance from the adaptive learning rate. This, in turn, allows for an automated warm-up tailored to the particular dataset at hand.

For each model architecture, we train two variants: (i) models trained entirely on a given dataset and (2) fine-tuned models previously trained on a different (and larger) image dataset. The former, which we refer to as models 'trained from scratch', refer to models trained only on the dataset at hand and initialized with random weights in the training procedure. The latter leverages transfer learning via model pre-training. The next section provides further details on how we use and fine-tune these pre-trained models. All models were trained on NVIDIA A100-PCIe GPUs with 40 GB of memory running CUDA version 11.5. We used the AiTLAS toolbox[3] to configure and run the experiments. All configuration files for each experiment are also available in our repository, along with the trained models. We believe this provides a standardized evaluation framework for EO image classification tasks.

3.2. Transfer learning strategy

In this study, we take the notion of *transfer learning* as a strategy that can lead to performance improvements of vision models on image classification tasks (Zhai et al., 2019), in particular in EO

[3] https://github.com/biasvariancelabs/aitlas

domains (Risojevic and Stojnic, 2021). In our problem setting, transfer learning allows downstream, task-specific models to leverage learned representations from model architectures pre-trained on much larger image datasets. This, in turn, often leads to (fine-tuned) models with much better generalization power using fewer training data (and training iterations), which is especially useful for tasks that stem from smaller datasets. In the case of DL models for image classification, two strategies are often used for performing transfer learning: (1) fine-tuning the model weights only for the last classifier layer or (2) fine-tuning the model weights of all layers in the network. The former approach retains the values of all but the last layer's weights of the model from the pre-training, keeping them 'frozen' during fine-tuning. The latter, on the other hand, allows the weights to change throughout the entire network during fine-tuning. In practice, this can lead to better generalization (Yosinski et al., 2014; Kornblith et al., 2019) and higher accuracy.

In our experiments, we implement the latter approach. Starting with a pre-train model, we fine-tune each network entirely (the entire parameter set) for each specific dataset. Note that the choice of the pre-training dataset, and its relation to the domain of the downstream task, may also influence the predictive performance of the fine-tuned model (Neumann et al., 2020). Since here we are interested in a more general evaluation that considers 22 different datasets, we evaluate a standard approach for transfer learning using pre-trained model architectures on the ImageNet-1K (Krizhevsky et al., 2012) dataset (version V1). More specifically, we use implementations from the PyTorch vision catalog (Marcel and Rodriguez, 2010) for most models, except ViT and MLPMixer, for which we base the implementations on (Wightman, 2019).

Furthermore, to evaluate the effect of the pre-training dataset on the performance of the downstream model, in a set of smaller-scale experiments, we benchmark architectures that have been pre-trained using different 'in-domain' EO datasets. In particular, we evaluate two strategies: (i) models pre-trained entirely on an EO dataset and (ii) models pre-trained on both ImageNet-1K and an EO dataset. The latter relates to a two-stage pre-training strategy, where models are first pre-trained on ImageNet-1K, followed by intermediate tuning on an in-domain EO dataset, and finally, fine-tuning them on the target EO dataset. We evaluate these pre-training strategies by comparing models from two architectures (ViT and DenseNet) using four in-domain EO datasets for pre-training.

3.3. Evaluation measures

Evaluating the performance of machine learning models is a non-trivial task that is specific to the learning task at hand and dependent on the general objectives of the model being learned. Different evaluation metrics capture different aspects of the models' behavior and their predictive capabilities measured on image samples not used for training. Since the goal of this study analyzing the predictive performance of different DL models across different datasets on multi-class and multi-label classification tasks — we examine the experimental work through the lens of evaluation measures most suitable for these two tasks.

More specifically, for multi-class classification tasks, we report the following measures: Accuracy, Macro Precision, Weighted Precision, Macro Recall, Weighted Recall, Macro F1 score, and Weighted F1 score. Note that, since for these tasks, the micro-averaged measures such as F1 score, Micro Precision, and Micro Recall have values equal to accuracy, we do not report them. Note that, for image classification tasks, it is customary to report *top-n accuracy* (typically n is set to 1 or 5) (Krizhevsky et al., 2012), where the score is computed based on the correct label being among the n most probable labels outputted by the model. In this paper, we report *top-1 accuracy*, denoted as 'Accuracy' unless stated otherwise. For multi-label classification tasks, we report Micro Precision, Macro Precision, Weighted Precision, Micro Recall, Macro Recall, Weighted Recall, Micro F1 score, Macro F1 score, Weighted F1 score, and mean average precision (mAP). Since all measures, but mAP, require setting a threshold on the predictions, we choose a threshold value of 0.5 for all models and settings. Further details and definitions of the evaluation measures used in the study are given in Appendix A in the Supplementary material. We also provide additional performance details in terms of confusion matrices of each experiment, allowing for a more detailed (per class/label) analysis of model performance (reported in Appendix D in the Supplementary material).

4. Results

We present the results of a large-scale study comparing different DL models for multi-class (MCC) and multi-label classification (MLC) tasks from 22 datasets. To this end, we evaluate models from 10 architectures: AlexNet, VGG16, ResNet50, ResNet152, DenseNet162, EfficientNetB0, ConvNeXt, Vision Transformer (ViT), Swin Transformer (SwinT) and MLPMixer. For each model architecture, we evaluate two variants: (i) models trained from scratch and (2) fine-tuned models previously trained on the ImageNet-1K dataset. We additionally assess the performance of models pre-trained using in-domain EO datasets. In the remainder, we outline and discuss the following:

- Performance of models trained from scratch with respect to the two types of tasks
- Benefits of pre-training models of different architectures and their effect in view of the dataset properties
- Models' ability to generalize on unseen in-domain datasets
- The choice of the pre-trained dataset and its effect on the performance of the downstream model
- The 'performance vs. cost of model training' trade-off between the considered modeling approaches
- Common issues that affect the models' predictive performance in the context of EO applications.

Detailed results of each experiment, with additional performance measures, are given in the Supplementary material (Appendices B, C and D in the Supplementary material).

4.1. Training models from scratch

We begin by analyzing the performance of models trained from scratch, i.e., models initialized with random weights during training. Tables 4 and 5 present these results for the MCC and MLC tasks, respectively. Table 4 reports the accuracy (%) of the models learned from scratch for the 15 MCC datasets. It also reports the rank of the models, estimated based on their performance and averaged over the 15 datasets. The results show that, in general, convolutional architectures, especially the DenseNet, the EfficientNet, and the two ResNets, consistently perform well. This is even more evident for datasets such as PatternNet, RSI-CB256, and SAT6, where the DenseNet (and the other top-ranked models) lead to near-perfect results (accuracy greater than 99%). More specifically, DenseNet is the best-performing model in more than half of the tasks (9 out of 15) and achieves accuracy greater than 90% in 8 tasks. These performances are generally much lower for smaller datasets, such as WHU-RS19, Optimal31, UC Merced, SIRI-WHU, RSSCN7, and CLRS. However, the most challenging task is *So2SAT*, where EfficientNetB0 achieves the highest accuracy of 65.17%, while many of the models trail behind with a performance of 55%–60%. These results are consistent with previous findings (Stewart et al., 2021), suggesting clear signs of over-fitting, influenced by the quality and size of the images in the dataset. The two transformer architectures (SwinT and ViT), the MLPMixer, and the latest ConvNeXt models are ranked at the bottom (only better than AlexNet), with lower, but, in many cases, still practically comparable performance to the leading DenseNets.

Next, we shift our focus to MLC tasks. Table 5 reports the mean average precision (%) of the models learned from scratch across the

Table 4
Accuracy (%) of models trained from scratch on multi-class classification datasets. Bold indicates best performing model for a given dataset. We report the *average rank* of a model (lower is better), ranked based on the performance and averaged across the 15 datasets.

Dataset \Model	AlexNet	VGG16	ResNet50	ResNet152	DenseNet161	EfficientNetB0	ViT	MLPMixer	ConvNeXt	SwinT
WHU-RS19	66.169	68.657	79.602	80.597	**80.597**	75.622	74.627	69.652	72.139	78.607
Optimal31	55.108	56.720	67.204	62.903	**71.237**	68.548	62.634	59.140	58.871	66.129
UC Merced	81.190	78.571	85.238	84.048	**86.190**	84.286	83.095	82.381	84.286	81.429
SIRI-WHU	83.750	84.792	**88.958**	88.750	86.667	86.042	86.250	82.500	84.167	85.833
RSSCN7	80.536	81.607	82.679	82.679	**87.321**	83.929	86.071	83.214	83.036	82.500
BCS	89.410	89.410	89.236	88.542	**90.799**	85.417	87.847	86.285	84.375	89.236
AID	81.350	81.950	89.050	89.900	**93.300**	90.050	79.350	71.750	81.100	87.700
CLRS	71.400	76.067	85.567	82.300	**86.167**	82.267	65.467	61.133	69.167	80.000
RSI-CB256	97.354	98.828	98.828	**99.152**	99.131	99.111	98.121	98.424	98.444	99.091
Eurosat	96.167	97.185	97.000	97.407	97.630	**97.796**	95.037	95.500	95.426	95.722
PatternNet	97.829	97.911	99.063	98.882	**99.243**	98.832	96.694	98.832	97.829	98.520
RESISC45	82.159	83.889	92.333	90.683	**93.460**	91.365	81.016	69.413	85.937	88.730
RSD46-WHU	86.032	88.625	90.549	89.944	92.211	90.612	86.466	81.253	88.693	91.806
So2Sat	56.511	62.271	59.587	61.477	55.428	**65.173**	55.333	53.580	60.154	57.128
SAT6	99.272	99.564	**100.00**	99.998	99.995	99.998	99.985	99.984	99.998	99.980
Avg. Rank	8.13	6.60	3.27	3.47	**2.00**	3.33	7.33	8.07	6.60	5.47

Table 5
Mean average precision (mAP %) of models trained from scratch on multi-label classification datasets. Bold indicates best performing model for a given dataset. We report the *average rank* of a model (lower is better), ranked based on the performance and averaged across the 7 datasets.

Dataset \Model	AlexNet	VGG16	ResNet50	ResNet152	DenseNet161	EfficientNetB0	ViT	MLPMixer	ConvNeXt	SwinT
AID (mlc)	68.780	69.206	70.867	69.646	71.218	**72.889**	65.581	64.235	65.595	69.548
UC Merced (mlc)	75.516	76.797	79.867	73.657	85.414	79.874	**87.142**	75.677	72.271	81.071
DFC15	88.099	89.871	94.675	94.188	**95.848**	93.973	94.164	91.663	89.564	94.349
Planet UAS	60.282	60.682	64.192	64.956	64.738	63.868	59.414	58.550	61.277	**65.229**
MLRSNet	90.850	91.524	**95.259**	93.982	94.745	94.395	87.250	85.281	90.710	94.099
BigEarthNet 19	75.711	77.989	78.726	78.519	79.725	79.211	75.871	77.005	77.909	**80.586**
BigEarthNet 43	56.082	58.969	64.343	62.736	63.390	62.173	57.410	58.772	60.472	**67.487**
Avg. Rank	8.57	6.57	3.00	4.71	**2.14**	3.86	7.29	8.57	7.71	2.57

7 MLC datasets. While DenseNets rank the best, they achieve the best result in only 1 out of 7 tasks. The second-ranked SwinT models achieve the best performance in 3 tasks with comparable performance in the remaining 4. Unlike the MCC tasks, the performance difference to other convolutional models (i.e., the two ResNets and the EfficientNetB0) here is much smaller. Moreover, most models were only able to achieve high performance (above 90%) on two tasks, *DFC15* and *MLRSNet*, with DenseNet and ResNet50 achieving the best results. However, this is an expected result, as MLC tasks are generally more challenging than MCC tasks. This can be attributed to two things in particular: First, in many cases, the semantic labels can be very similar, which makes many of the models struggle. Second, MLC datasets tend to have a more significant class/label imbalance, in contrast to MCC datasets' more uniform class distribution. In this context, the most challenging MLC tasks overall are *PlanetUAS* and *BigEarthNet43*, where the best performing SwinT models achieve mAP od 65.229% and 67.487%, respectively. Finally, similar to the previous MCC analysis, ViT, MLPMixer, and ConvNeXt remain only better ranked than AlexNet. Nevertheless, their performance on these MLC tasks is much more competitive, for instance, in the case of ViT, which is the best model on the *UC Merced* task.

4.2. The benefits of model pre-training

While training models from scratch leads to decent performance, in practice, leveraging pre-trained models can lead to significant performance improvements on image classification tasks (Zhai et al., 2019), and in particular on tasks in EO domains (Risojevic and Stojnic, 2021).

This is also the general conclusion from our analysis. When using models that were first pre-trained on ImageNet-1K and then fine-tuned on the specific datasets, we found that: *Pre-trained models lead to substantial performance improvements compared to models trained from scratch.* Fig. 2 illustrates this performance-improvement trend for different models across the 22 MCC and MLC tasks. We find that pre-training significantly improves the performance of all the evaluated models. Notably, we observe that the transformer models, based on either ViT or SwinT architectures, benefit the most from pre-training, followed by MLPMixer and ConvNeXt models. This is a significant improvement over the models trained from scratch. These results, especially for the case of ViT, are consistent with previously reported findings (Dosovitskiy et al., 2020; Papoutsis et al., 2022).

Tables 6 and 7 present the detailed results of these analyses for MCC and MLC tasks, respectively. Similar to the analyses in the previous section, we report model accuracy (%) in the case of MCC tasks and mean average precision (%) in the case of MLC tasks. We also report the rank of the models, averaged over the respective datasets. Considering MCC tasks (Table 6), most models achieve very good performance (accuracy over 90%) on 14 (out of 15) tasks, with (almost) perfect results in five of those. Notably, we observed significant performance improvements, compared to model counterparts trained from scratch, on smaller datasets (such as *WHU-RS19, Optimal31, UC Merced, SIRI-WHU, RSSCN7, and CLRS*), reaffirming the utility of transfer learning from large datasets in the context of EO image classification tasks. In terms of model architectures, the ViT ranks at the top among the model architectures, achieving the best performance in 6 out of 15 cases, followed by DenseNet161, SwinT, and ResNet152 with lower but comparable performance. Transformer architectures, and ViTs in particular, typically require large amounts of training data (Dosovitskiy et al., 2020; Paul and Chen, 2022) for learning robust, good performing models. As a result, using pre-trained models and fine-tuning them leads to substantial performance improvements, compared to training them from scratch. The performance of ViTs is further highlighted for the case of the challenging *So2SAT* task, where the ViT model leads to an accuracy of 68.55%, in contrast to the next ranked DenseNet and SwinT with an accuracy of 65.75% and 65.95%, respectively. In this specific case of *So2SAT*, we observed that over-fitting remains an issue, even for pre-trained models. Our further investigation of the train/validation loss trends showed that, regardless of the model at hand, with the training loss decreasing, the validation errors increase almost instantly (after 1–2 epochs) — a typical trend observed in over-fitting models (see Figure D.46 in Appendix D.15 in the Supplementary material, that illustrates such behavior in a ViT model). This, fortunately, is not the case for the remaining tasks, where we observed

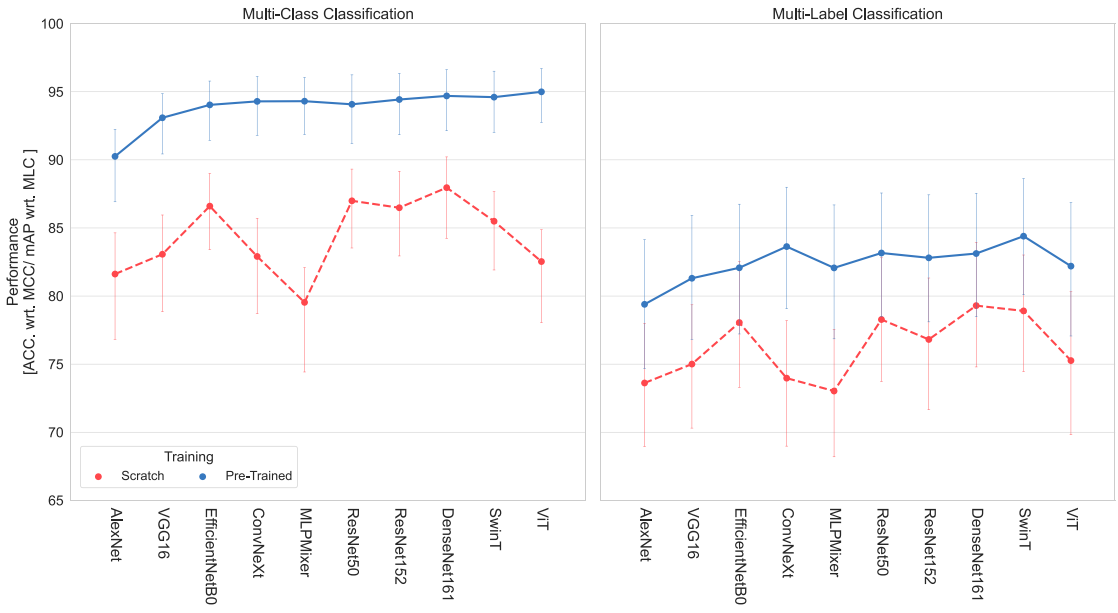


Fig. 2. Comparison of average performance improvement of models from the 10 different architectures when trained from scratch **(red)** and employing pre-trained models **(blue)** across **(left)** MCC and **(right)** MLC datasets. Error bars indicate confidence interval of 68%. Models are ordered (worst to best) based on the average performance-rank of the pre-trained variants across all of the 22 datasets. Model pre-training leads to substantial performance improvements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Accuracy (%) of models pre-trained on ImageNet-1K on multi-class classification datasets. Bold indicates best performing model for a given dataset. We report the *average rank* of a model (lower is better), ranked based on the performance and averaged across the 15 datasets.

Dataset\Model	AlexNet	VGG16	ResNet50	ResNet152	DenseNet161	EfficientNetB0	ViT	MLPMixer	ConvNeXt	SwinT
WHU-RS19	93.532	99.005	99.502	98.010	**100.00**	99.502	99.502	98.507	99.005	99.502
Optimal31	80.914	88.710	92.204	92.473	94.355	91.667	**94.624**	92.742	93.011	92.473
UC Merced	92.143	95.476	98.571	**98.810**	98.333	98.571	98.333	98.333	97.857	98.571
SIRI-WHU	92.292	93.958	95.000	**96.250**	95.625	95.000	95.625	95.208	**96.250**	95.625
RSSCN7	91.964	93.929	95.000	95.000	94.821	95.536	**95.893**	95.179	94.643	95.179
BCS	89.583	90.972	92.014	92.361	92.708	91.319	92.014	93.056	91.493	**93.403**
AID	92.900	96.100	96.550	97.200	97.250	96.250	**97.750**	96.700	96.950	97.400
CLRS	84.100	89.900	91.567	91.900	92.200	90.500	**93.200**	90.100	91.100	92.533
RSI-CB256	99.354	99.051	99.677	**99.859**	99.737	99.717	99.758	99.657	99.596	99.677
Eurosat	97.574	98.148	98.833	**99.000**	98.889	98.907	98.722	98.741	98.778	98.944
PatternNet	99.161	99.424	**99.737**	99.490	**99.737**	99.539	99.655	99.704	99.671	99.688
RESISC45	90.492	93.905	96.460	96.540	96.508	94.873	**97.079**	95.952	96.270	96.587
RSD46-WHU	90.646	92.422	94.158	94.404	**94.507**	93.387	94.238	93.673	93.627	93.536
So2Sat	59.203	65.375	61.903	65.169	65.756	65.801	**68.551**	67.066	66.169	65.950
SAT6	99.980	99.993	**100.00**	**100.00**	**100.00**	99.988	99.998	99.995	99.999	99.999
Avg. Rank	9.93	8.67	4.67	3.80	3.13	5.87	**3.07**	5.33	5.47	3.20

Table 7
Mean average precision (mAP %) of models pre-trained on ImageNet-1K on multi-label classification datasets. Bold indicates best performing model for a given dataset. We report the *average rank* of a model (lower is better), ranked based on the performance and averaged across the 7 datasets.

Dataset \Model	AlexNet	VGG16	ResNet50	ResNet152	DenseNet161	EfficientNetB0	ViT	MLPMixer	ConvNeXt	SwinT
AID (mlc)	75.906	79.893	80.758	80.942	81.708	78.002	81.539	80.879	**82.298**	82.254
UC Merced (mlc)	92.638	92.848	95.665	96.010	96.056	95.384	96.699	96.340	96.431	**96.831**
DFC15	94.057	96.566	97.662	97.600	97.529	96.787	97.617	97.941	97.994	**98.111**
Planet UAS	64.048	65.584	65.528	64.825	66.339	64.157	66.804	67.330	66.447	**67.837**
MLRSNet	93.399	94.633	96.272	96.432	96.306	95.391	96.410	95.049	95.807	**96.620**
BigEarthNet 19	77.147	78.418	79.983	79.776	79.686	80.221	77.310	77.288	80.283	**81.384**
BigEarthNet 43	58.554	61.205	66.256	64.066	64.229	64.589	58.997	59.648	66.166	**67.733**
Avg. Rank	10.00	7.86	5.14	5.43	5.00	6.86	4.86	5.71	3.00	**1.14**

a decent performance overall. Most models, especially the top half ranked, achieved stable and mostly comparable performance.

The benefits of pre-training models also extend to MLC tasks (Table 7), in several cases with significant performance gains, compared to model counterparts trained from scratch. In particular, we found that pre-training can lead to minor improvements (1%–2%) on challenging tasks such as *PlanetUAS* and *BigEarthNet43* (mAP of 67.837% and 67.733% achieved by SwinTs); to more considerable improvements (up to 15%) in some cases such as *AID* and *UCMerced* (mAP of 82.298% and 96.83% obtained by ConvNeXt and SwinT, respectively). Also, in this case, we found that the transformer models benefited the most from pre-training. This is in line with studies (Liu et al., 2021, 2022a) that highlight the significance of pre-training to the generalization performance of these types of models. Notably, SwinT models ranked the best overall and achieved the best performance on 6 (out of the 7) tasks. They are followed by ViT and ConvNeXt, with comparable performance on most tasks.

Table 8
Accuracy (%) of models pre-trained on ImageNet-1K and fine-tuned on a specific source dataset and evaluated on the common test dataset with shared labels. Bold indicates best performing model for a given source dataset.

Dataset \Model	AlexNet	VGG16	ResNet50	ResNet152	DenseNet161	EfficientNetB0	ViT	MLPMixer	ConvNeXt	SwinT
RESISC45	66.853	78.514	81.063	84.080	84.111	77.985	**86.007**	82.121	84.422	83.706
UC Merced	63.371	67.040	76.057	73.010	74.254	74.440	75.995	**79.478**	75.902	72.326
CLRS	80.037	83.427	89.801	88.557	89.024	86.070	**92.600**	89.646	89.303	90.299
PatternNet	43.501	52.332	56.965	54.540	56.716	60.044	64.739	62.687	59.391	**65.205**
AID	71.393	69.714	79.384	80.100	66.169	77.892	**83.862**	77.954	79.851	79.789
RSI-CB256	56.872	61.412	58.893	63.650	64.832	61.723	66.014	**66.791**	64.677	66.294
WHU-RS19	61.101	62.624	71.953	73.321	72.388	68.284	72.917	74.036	**74.876**	71.144
Avg. Rank	9.71	8.71	5.43	5.29	5.86	6.86	**2.14**	3.29	3.57	4.14

4.3. Generalization capabilities to unseen data

We further investigate the generalization ability of the trained models by evaluating their performance across datasets not used during training. In particular, we present results from two evaluation settings: (1) performance measured on a holdout set compiled of test images with shared labels and (2) an exhaustive cross-dataset evaluation between pairs of datasets with overlapping labels. First, we analyze the predictive performance of all models when applied to the same holdout set with 3216 images sampled from the test splits from seven MCC datasets (*RESISC45, UC Merced, CLRS, PatternNet, AID, RSI-CB256* and *WHU-RS19*) using only images with labels shared among the seven datasets: 'Forest', 'Parking', 'River', 'Harbor', and 'Beach'. Figure C.1 (in Appendix C in the Supplementary material) presents further details of the distribution of images in the holdout set w.r.t. source datasets and labels. We evaluate and report the predictive performance of pre-trained models from all ten architectures. Note that here we only evaluate the models on the holdout set without additional fine-tuning. Table 8 reports the predictive performance assessed using accuracy (%) as an evaluation measure.

The results show that ViT models are able to generalize well to unseen images from other in-domain datasets. Namely, in many cases, ViT models perform better than the competitors, further supporting previous results regarding their performance on MCC tasks. The performance of ViTs is followed by models based on more recent architectures, such as SwinT, MLPMixer, and ConvNeXt, which show worse but, in many cases, practically comparable performance. With respect to specific datasets, our experiments show that models fine-tuned on the *CLRS* and *RESISC45* datasets were able to achieve much better performance than the others (with ViT models achieving 92.6% in the case of *CLRS*). We hypothesize that such performance may be related to the particular properties of these datasets: Both *CLRS* and *RESISC45* are multi-resolution datasets (containing images at different spatial resolutions) with a large number of diverse labels. However, this is not the case for models fine-tuned on *PatternNet* and *RSI-CB256*. While models trained and evaluated on these datasets separately show great performance (99% accuracy), this performance decreases significantly when evaluated on a holdout set (down to 66.79% and 65.2% for *RSI-CB256* and *PatternNet*, respectively). These results, along with results from models learned from scratch (Table 4), are indicative of both datasets being easily learned, producing models that are not able to generalize well to other unseen images and classification tasks.

In the second experimental setup, we employ the following pairwise evaluation scheme. We consider pairs of *source* and *target* datasets: We take pre-trained models that have been fine-tuned on a *source* dataset and evaluate them on test images from a *target* dataset. Note that we only *evaluate* the models on the target dataset without additional fine-tuning. We measure the performance only on a subset of images with shared labels between the source and target datasets. Therefore, for this experiment, we selected datasets with at least 0.15 IoU[4] overlap of labels with at least one other dataset. This resulted in pairs from 12 (out of 15) MCC datasets and 4 (out of 7) MLC datasets, yielding 256 comparisons of pre-trained models from each of the ten considered architectures. Figs. 3 and 4 present the performance of the best model for each MCC and MLC comparison in terms of accuracy (%) and mAP (%), respectively. They also provide a summary of the overlap between each pair of datasets in terms of IoU. Detailed results of all comparisons, per architecture, are given in Appendix C in the Supplementary material.

The results support our earlier findings that the *transformer*-based models, in particular the ViT models (on MCC tasks) and the SwinT models (on MLC tasks), perform best when applied to other in-domain datasets. More specifically, when considering MCC tasks, the transformer-based models perform best in almost 2/3 of the comparisons, with the ViT models alone performing best in ~40% of them. ViTs are followed by SwinT, ConvNeXt, and MLPMixer models that, in many cases, showed practically comparable performance. We observed that convolutional models such as DenseNets, which exhibited good performance in our previous analyses (when evaluated on test images from the same dataset), generally lead to worse performance than models from more recent architectures. The dominance of the transformer-based models also extends to MLC tasks, with SwinT models producing the best overall performance, followed closely by ViT models. Note that these empirical results are also consistent with other studies (Bhojanapalli et al., 2021; Paul and Chen, 2022; Zhang et al., 2022b), that highlight the robustness and good generalization capabilities of transformer-based models for general-domain images.

4.4. Domain-adaptive transfer learning

Having demonstrated the practical benefits and generalization capabilities of using pre-trained models, we further investigate the impact of the pre-trained dataset on the performance of the downstream model. As we focus on particular domains of interest that leverage satellite imagery, we evaluate whether and how choosing more appropriate in-domain EO pre-training datasets (and strategies) affects downstream predictive performance. Our experimental setup aims to investigate two different strategies for such in-domain pre-training: (i) in-domain only, where models are pre-trained entirely on an EO dataset (ii) two-stage pre-training, where models are pre-trained on a combination of ImageNet-1K and an EO dataset. The former strategy is analogous to the ImageNet-1K pre-training strategy but uses a different EO dataset. In the second strategy, on the other hand, the models are first pre-trained on ImageNet-1K, followed by intermediate tuning on an in-domain EO dataset, before fine-tuning the models on the target EO dataset.

Rather than evaluating all architectures, in this set of experiments, we evaluate two types of architectures: a ViT and a DenseNet161, as representatives of transformer and convolutional architectures that have shown overall good performance in our previous experiments. Specifically, we analyze their performance on six tasks (3 MCC and 3 MLC) that proved somewhat challenging for these models: *CLRS, Optimal31, So2SAT, AID (mlc), PlanetUAS, and BigEarthNet 19*. We select four different in-domain datasets for our pre-training: *SAT6, RSD46-WHU, MLRSNet* and *RESISC45*; based on the overall performance achieved in the previous analyses, their size (number of images),

[4] Intersection over Union (IoU), measures the overlap between two sets. Values range from 0 to 1, where 0 indicates no overlap and 1 indicates complete overlap between the sets

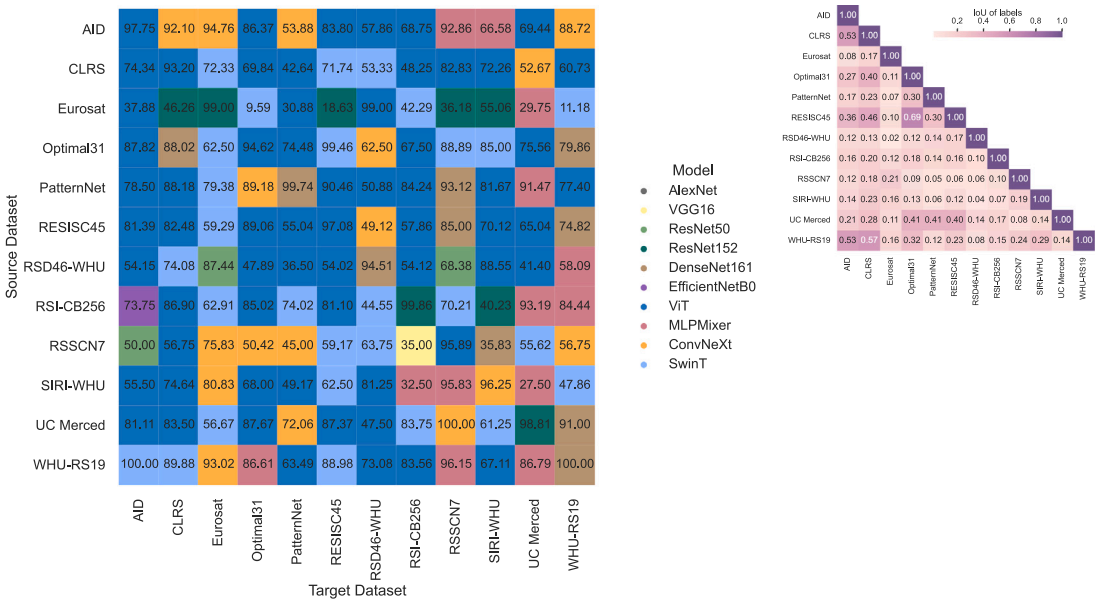


Fig. 3. Model generalization on multi-class classification tasks: Comparison of the best performing pre-trained models (**left**) from the 10 different architectures (**color-coded**) in terms of **accuracy** (% acc. is indicated in each field); the models are fine-tuned on *source* dataset and evaluated on images with common/overlapping labels in *target* dataset. The heatmap (**right**) reports the label overlap between each pair of datasets, in terms of IoU. Transformer-based models, in particular the ViT models, perform the best when evaluated on other in-domain MCC datasets.

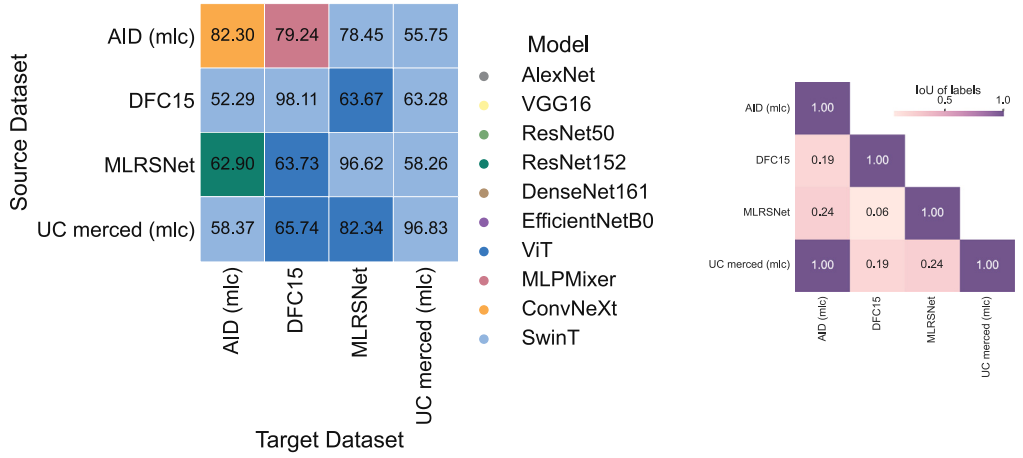


Fig. 4. Model generalization on multi-label classification tasks: Comparison of the best performing pre-trained models (**left**) from the 10 different architectures (**color-coded**) in terms of **mean average precision** (% mAP is indicated in each field); the models are fine-tuned on *source* dataset and evaluated on images with common/overlapping labels in *target* dataset. The heatmap (*right*) reports the label overlap between each pair of datasets, in terms of IoU. In general, transformer-based models, in particular the SwinT models, lead to the best performance on MLC tasks.

and their heterogeneity (in terms of semantic labels). Table 9 reports the results of these experiments.

Our general conclusion regarding pre-training remains: Pre-trained models based entirely on EO datasets can still outperform their counterparts trained from scratch. However, we find that the choice of the pre-training dataset has a significant impact on the downstream performance and is not necessarily related to the quality of the pre-training dataset (measured as stand-alone performance) or solely to its size. For instance, we found that models pre-trained entirely using *SAT6* (a dataset on which most models performed very well) performed much worse than the other pre-trained counterparts and, in some cases, even worse than models trained from scratch. This is not the case when pre-training models on *RSD46-WHU*, *MLRSNet*, and *RESISC45*, which led to better performance, compared to their counterparts trained from scratch (in both cases of ViT and DenseNets), albeit worse than models pre-trained on ImageNet-1K.

Importantly, we found that using a combined pre-training procedure, with ImageNet-1K followed by an in-domain dataset, can lead to improvements (up to 5%), especially when combined with *MLRSNet* or *RESISC45* datasets. This is specifically the case for *Optimal31* and *AID (mlc)*, where models from both ViT and DenseNet161 architectures were able to outperform their counterparts pre-trained only on ImageNet-1K. These results suggest that using datasets for intermediate fine-tuning that contain images at different resolutions with heterogeneous (but potentially semantically similar) labels, in addition to ImageNet-1K, can lead to performance improvements. However, in most cases, we did not observe neither practical nor significant benefits for using a combined pre-training procedure with an additional in-domain dataset that would justify the additional computational overhead for training such models.

4.5. The 'performance vs. training cost' trade-off

Having established the performance of our evaluated models and demonstrated the clear benefits of using pre-trained models, we focus

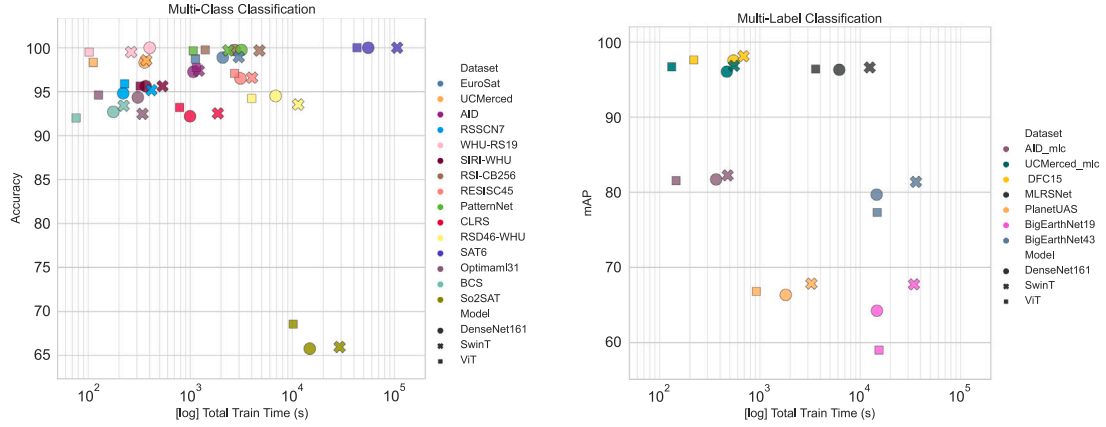
Table 9
Comparison of pre-training strategies for (a) Vision Transformers (ViT) and (b) DenseNets161 using 4 in-domain EO datasets (SAT6, RSD46-WHU, MLRSNet, RESISC45) and ImageNet-1K. We report their performance on 3 multi-class and 3 multi-labels classification tasks, in terms of accuracy (% Acc.) and mean average precision (% mAP), respectively.

(a)

In-domain dataset	Pre-training strategy	Target dataset					
		CLRS [%Acc.]	Optimal31 [%Acc.]	So2Sat [%Acc.]	AID (mlc) [$\%mAP$]	Planet UAS [$\%mAP$]	BigEarthNet 19 [$\%mAP$]
SAT6	In-domain only	60.767	58.065	54.672	62.200	59.538	74.618
	ImageNet-1K + In-domain	71.200	68.011	64.284	67.595	62.356	76.009
RSD46-WHU	In-domain only	72.267	75.269	57.859	71.209	61.015	75.529
	ImageNet-1K + In-domain	91.067	92.204	65.322	80.102	66.460	76.809
MLRSNet	In-domain only	71.700	77.688	54.746	72.915	60.985	74.827
	ImageNet-1K + In-domain	91.033	95.430	64.321	83.069	64.574	77.308
RESISC45	In-domain only	68.700	86.022	57.446	69.552	61.457	75.345
	ImageNet-1K + In-domain	92.533	98.925	66.876	82.888	66.654	76.682
/	ImageNet-1K only	93.200	94.624	68.551	81.539	66.804	77.310

(b)

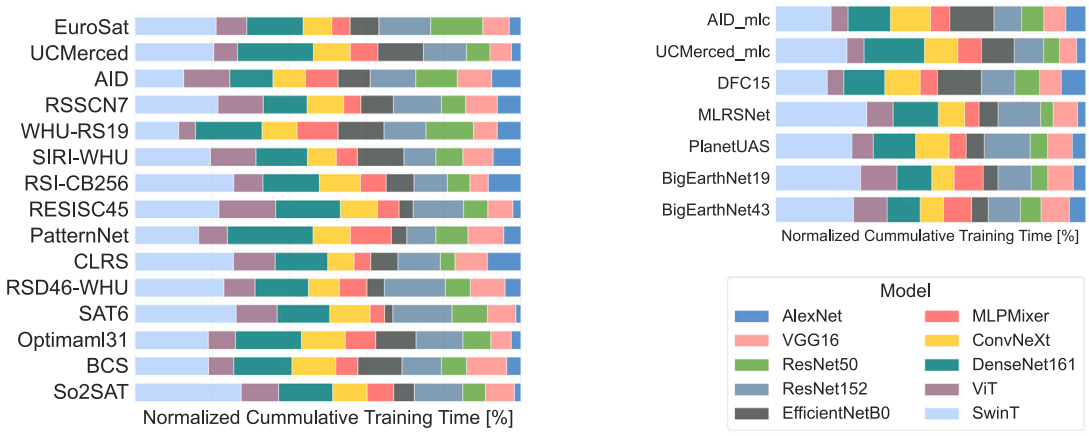
In-domain dataset	Pre-training strategy	Target dataset					
		CLRS [%Acc.]	Optimal31 [%Acc.]	So2Sat [%Acc.]	AID (mlc) [$\%mAP$]	Planet UAS [$\%mAP$]	BigEarthNet 19 [$\%mAP$]
SAT6	In-domain	65.467	55.914	58.455	59.363	59.419	76.745
	ImageNet-1K + In-domain	89.467	85.215	65.334	74.653	64.918	79.773
RSD46-WHU	In-domain	89.267	86.559	60.890	77.056	65.101	79.374
	ImageNet-1K + In-domain	91.800	93.280	65.152	82.339	66.161	79.867
MLRSNet	In-domain	89.700	92.742	61.030	80.144	64.530	79.646
	ImageNet-1K + In-domain	91.367	96.505	62.808	84.070	64.859	79.945
RESISC45	In-domain	86.433	93.011	60.009	73.199	63.532	78.309
	ImageNet-1K + In-domain	91.267	98.387	64.011	82.936	66.276	79.695
/	ImageNet-1K only	92.200	94.355	65.756	81.708	66.339	79.686



(a) MCC tasks (b) MLC tasks

Fig. 5. Performance vs. total training time comparison of the overall top-3 performing *pre-trained* model architectures, ViT, DenseNet161 and SwinT (denoted with different markers); evaluated on **(left)** MCC and **(right)** MLC datasets (color-coded). Performance is reported as accuracy (%) and mean average precision (mAP %) for MCC and MLC tasks, respectively. Note the log scale of the total training time (seconds).

here on another line of comparison — the cost of model training. Recall from Section 2.2, and in particular Table 3, that we study model architectures that differ significantly in the number of learnable parameters. Typically, larger models require more computing resources and much more training time than smaller models. In our experimental setup, we train all models on the same computing infrastructure, under the same conditions, and with the same training/evaluation setup (in terms of hyperparameters and data partitioning). Therefore we can directly analyze the 'performance vs. training cost' (in terms of total training time) trade-off for each model variant from the ten different



(a) MCC tasks (total time)

(b) MLC tasks (total time)

Fig. 6. Total training time of pre-trained models for each of the **(a)** MCC and **(b)** MLC datasets. The training time of each model architecture (denoted with different colors) is depicted as a fraction (%) of the cumulative training time for each dataset.

architectures (either pre-trained or trained from scratch) across the 22 datasets. This way, we can explicitly measure the benefits of each model and make further modeling decisions based on the performance of the models and the 'cost' of training them.

Fig. 5 illustrates the trade-off for the top-3 best performing model architectures overall (as shown in Tables 6 and 7), DenseNet, ViT, and SwinT; applied to the 22 MCC and MLC tasks. While the performance analyses showed many similarities between these models, the difference between them in terms of training times is much more pronounced. In general, ViT requires less training time than both DenseNets and SwinTs. DenseNets have nearly a quarter of the number of parameters of ViT but achieve almost half fewer FLOPS (floating-point operations per second) than them. For MCC tasks, ViT models generally result in comparable/better predictive performance than DenseNet models and, in many cases, require half the training time. SwinT models, on the other hand, are much more demanding. In almost all cases, training SwinT models takes up to 2–3 times longer than training ViTs and DenseNets. This is also true for MLC tasks, where SwinT models perform the best performance but at the cost of significant training time. These findings further support previous results (Liu et al., 2021), which point out that Swin transformers (the 'small' variant) have slower training and inference performance than Vision Transformers, which have significantly more parameters but achieve considerably more FLOPS. For an extended illustration of these trade-offs, covering all 10 model architectures, see Figure B.1 in Appendix B in the Supplementary material.

We can further analyze these trends in training time trends for each model and dataset, as presented in Fig. 6. In particular, Fig. 6 illustrates the training (fine-tuning) times of each pre-trained model as a fraction of the cumulative training time of all models summed across all (a) multi-class and (b) multi-label datasets. This shows that, in many cases, ViT models can be trained almost twice as fast as the models of the other best-performing architectures, such as DenseNet and SwinT. The training cost of ViT models is similar to that of EfficientNetB0, ConvNeXt, and MLPMixer, which are efficient but generally perform worse on these tasks. We can also observe that these variants of SwinT models are the slowest to train on all 22 tasks compared to the other architectures. This is also evident when comparing the time for each epoch (see Appendix B in the Supplementary material), with SwinT models taking twice longer to train compared to DenseNet161 models, the next slowest architecture. We also observed that fine-tuning pre-trained models almost halves the training time compared to training models from scratch, even though they take about the same time per epoch. Note, however, that we have not accounted for the time required to pre-train each model, which certainly increases the overall training times significantly. This is generally expected behavior but may help in the design and planning of DL pipelines for similar EO. Additional results presenting models' training costs are presented in the supplementary material (Appendix B in the Supplementary material).

4.6. A closer look on several tasks

To better understand the performance of the learned models on the various MCC and MLC tasks, we examine the model decisions in detail, focusing on datasets (and classes) where the models tend to perform poorly. We hypothesize that these cases are related to several overarching issues that often affect the performance of the models:

- High inter-class similarity between images from different classes;
- Many EO image-classification tasks, which are formulated as MCC, are, in fact, MLC problems. In many cases, an image has a single label, but there are more than one classes/concepts present;
- Presence of abstract/complex/compound classes within the datasets, can cause many difficulties in detecting useful and consistent patterns;
- Absence of additional spatio-temporal data which captures the dynamics of land-cover changes

To investigate these issues, we simultaneously analyze the models' confusion matrices and visualizations of localized activation maps that highlight the distinguishing parts of the image responsible for the model decision. To generate such visualizations, we use Gradient-weighted Class Activation Mapping (GradCAM) (Selvaraju et al., 2017), which is typically used to diagnose model predictions for various deep learning architectures (Gildenblat and contributors, 2021), including Earth Observation applications (Papoutsis et al., 2022; Li et al., 2020c). GradCAM uses the gradients of the target classes from the last convolutional layer and produces a coarse localization map highlighting important regions in the image for class prediction. In this set of analyses, we select several cases from the datasets considered datasets, especially those containing classes/land types for which the models perform poorly (based on the various evaluation scores, as reported in Appendix D in the Supplementary material), and calculate/visualize the corresponding GradCAM maps.

We start by investigating the inter-class similarities between images assigned to different classes. This is a common problem in practice in many similar EO applications, caused by the presence of visually similar (often indistinguishable) objects in an image. Fig. 7 illustrates this problem using GradCAM activation maps of some sample images with their respective classes/labels from the different datasets.

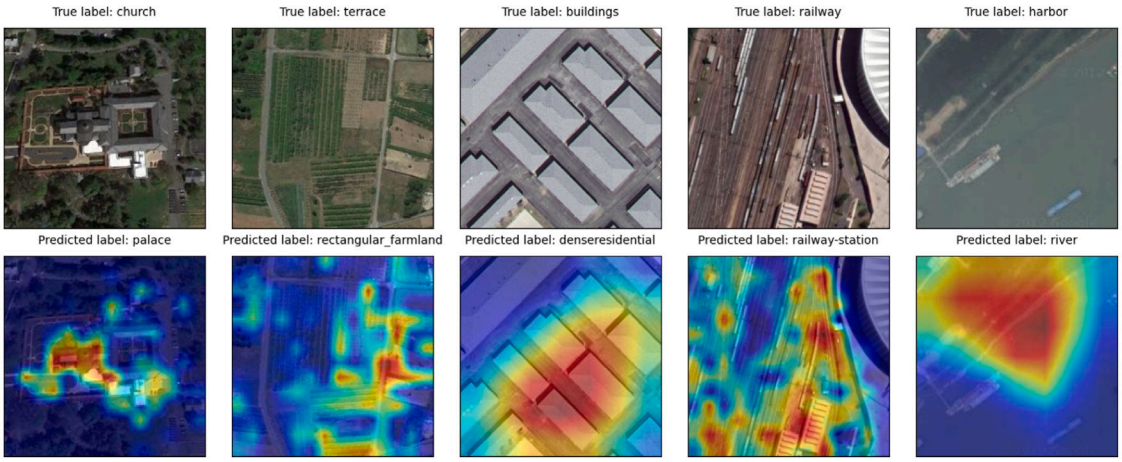


Fig. 7. GradCAM visualizations calculated for example images with high inter-class similarity. The input images with their ground-truth label are shown in the first row, while the corresponding activation maps with predicted labels are shown below in the second row. The datasets for the images and the models used to predict the labels are as follows, from left to right: (1) Resisc45, ViT model (2) Resisc45, ViT model (3) UC Merced, ResNet152 model (4) CLRS, ViT model and (5) SIRI-WHU, ResNet152 model.

Our qualitative analyses show that the predictive models are generally able to focus on the correct parts of the images (with distinguishable patterns) but cannot identify the correct object. This is the case, for example, when distinguishing between a 'church' and a 'palace' or a 'terrace' and a 'rectangular farmland', which are visually very similar but semantically different. As expected, the models also struggle with cases where the image labels are also semantically similar, such as in the distinction between 'railway' and 'railway station' or 'river' and 'harbor', which even a human expert would have difficulty classifying. Similar cases can be further analyzed by examining the confusion matrices. For example, the most challenging dataset, *So2Sat*, contains many such examples (see Figure D.45 in Appendix D.14 in the Supplementary material), which are the reason for the poor overall performance of the models.

The second issue that we highlight is related to the fact that, in many cases, multiple land-cover classes/concepts are present in a single image, but the image itself is assigned to only one class — making it a multi-class instead of a multi-label problem. Fig. 8 shows several activation maps illustrating this issue. For example, consider the image-pair on the far left: The image is labeled only as 'river', but we can also see an 'overpass' (a label also present in the dataset) that causes the model to make an 'incorrect' prediction, albeit with a probability of 0.54. Similar situations can be observed for the remaining images: Objects from other classes that are substantially present in an image are detected, thus confusing the models. This, however, shows that the models have been trained well and are performing as expected, but instead of outputting multiple labels (as in a typical MLC setting), they have to choose a single one — which can lead to errors and lower performance.

To evaluate the third issue, which relates to complex/compound classes, we examine samples with lower F1 scores. Complex/compound classes refer to classes that consist of objects with different physical properties and spatial distribution, making it very difficult to detect useful and consistent patterns. This is also true for abstract classes, where the semantic gap (in terms of labels) is challenging to overcome, which is typically the case when the features learned from the models differ from human interpretation.

Fig. 9 illustrates these problems using the respective activation maps. In particular, in the case of *AID* (the two pairs of images on the far left), the model confuses 'school' with 'commercial', the latter being quite vague, for which the semantic gap is not easily dealt with. In the second case, the model has difficulty distinguishing between 'park' and 'resort' (which is also evident in the confusion matrix in Figure D.9 in Appendix D.9 in the Supplementary material). This could be because these classes consist of common objects but have different spatial distributions. Similar problems can be seen in the cases of *CLRS* and *SIRI-WHU* (the last three image pairs), where labels such as 'industrial' or 'meadow' are confused with labels such as 'commercial/residential/park', which are visually and semantically almost indistinguishable from the ground truth. Similar problems exist in MLC datasets, such as *BigEarthNet*, that contain multiple complex/compound classes. From the evaluation details (see Appendix D.17 in the Supplementary material), we can see that complex/compound classes such as 'Complex cultivation patterns', 'Land principally occupied by agriculture, with significant areas of natural vegetation', and 'Industrial and commercial units' have lower F1 scores.

Finally, our analysis shows that for some tasks (such as *So2Sat*), one needs additional and more sophisticated (spatio-temporal) data to improve the performance of the predictive models. For example, the *So2Sat* dataset is very challenging, not only because of the high inter-class similarity but also because of the relatively low spatial resolution of the images. Images labeled 'Open high rise' or 'Compact low rise' are often confused with 'Open middle rise' or 'Lightweight low rise', respectively, which is hardly surprising without additional data that can capture such subtle and often subjective differences. Moreover, in the case of *BigEarthNet*, classes such as 'Permanent crops', 'Coastal wetlands', and 'Natural grassland and sparsely vegetated areas' require additional spatio-temporal data that capture the dynamics caused by frequent land cover changes, making the process of classification more reliable and thus more accurate.

5. Conclusions

We present a systematic review and evaluation of several modern DL architectures applied in Earth Observation. Specifically, we introduce *AiTLAS: Benchmark Arena* — an *open-source EO benchmark suite* and demonstrate its utility with a comprehensive comparative analysis of models from ten different state-of-the-art DL architectures, comparing them to a variety of multi-class and multi-label image classification tasks from 22 datasets. We compare models trained from scratch and pre-trained models under the same conditions and with the same hardware. We evaluate more than 500 models with different architectures and learning paradigms across tasks from 22 datasets with different sizes and properties. To our knowledge, the evaluation of these different setups (in terms of machine learning tasks, model setups, model architectures, and datasets) makes this the largest and most comprehensive empirical study of deep learning methods applied to EO datasets to date. All of the important details about the study design, the results, and the trained models are freely available. This will contribute to more systematic and rigorous experiments in future

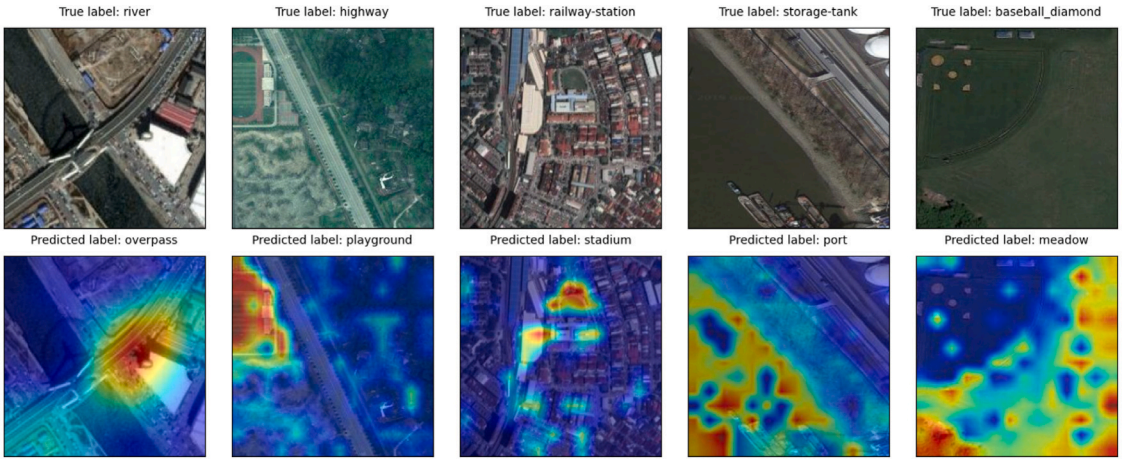


Fig. 8. GradCAM visualizations that illustrate the MCC/MLC issues. The input images with their ground-truth label are shown in the first row, while the corresponding activation maps with predicted labels are shown below in the second row. The datasets for the images and the models used to predict the labels are as follows, from left to right: (1) Resisc45, ViT model (2) Resisc45, ViT model (3) UC Merced, ResNet152 model (4) CLRS, ViT model and (5) SIRI-WHU, ResNet152 model.

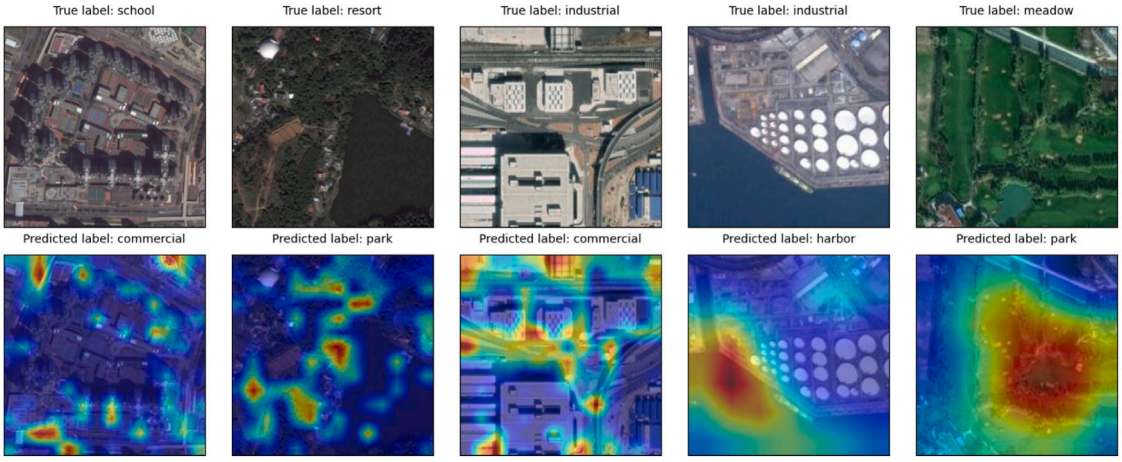


Fig. 9. GradCAM visualizations for images with complex/compound classes. The input images with their ground-truth label are shown in the first row, while the corresponding activation maps with predicted labels are shown below in the second row. The datasets for the images and the models used to predict the labels are as follows, from left to right: (1) AID, ViT model (2) AID, ViT model (3) CLRS, ViT model (4) SIRI-WHU, ResNet152 model and (5) SIRI-WHU, ResNet152 model.

work and, more importantly, will enable better usability and faster development of novel approaches. We believe that both this study and the associated repository can serve as a starting point and a guiding design principle for evaluating and documenting machine learning approaches in the different domains of EO. More importantly, we hope that with further involvement from the community, AiTLAS: Benchmark Arena can become a reference point for further studies in this highly active research area.

More broadly, we believe that this work, along with the developed resources, will strongly impact the AI and EO research communities. First, such ready-to-use resources containing trained models, clear experimental designs, and detailed results will facilitate better adoption of sophisticated modeling approaches in the EO community — bringing the EO and AI communities closer together. Second, it demonstrates the FAIRification process of AI4EO resources, i.e., making resources adhere to the FAIR principles (Findable, Accessible, Interoperable, and Reusable (Wilkinson et al., 2016)). Finally, it contributes to the 'Green AI' initiative by saving additional computational overhead. Since all experimental details, especially the trained models, are publicly available – other experts and researchers can compare, reproduce, and reuse these resources – reducing the need to (repeatedly) run unnecessary experiments.

Reproducibility

All the necessary details, in terms of the trained models, model parameters and implementations as well as details on all of the used datasets and their prepossessed versions are available at https://github.com/biasvariancelabs/aitlas-arena. All the models were trained/fine-tuned on NVIDIA A100-PCIE-40 GB GPUs, running CUDA Version 11.5 (www.nvidia.com/en-gb/data-center/a100/). Note that, we do not host the datasets. To obtain them, please refer to each of the respective studies (referenced in Tables 1 and 2) or follow the links provided in our repository. The study was performed using the AiTLAS Toolbox (Dimitrovski et al., 2022), a library for exploratory and predictive analysis of satellite imaginary pertaining to different remote-sensing tasks, available at https://aitlas.bvlabs.ai.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support of the European Space Agency ESA, France through the activity AiTLAS - AI4EO rapid prototyping environment. We thank Sofija Dimitrovska for her thoughtful feedback.

Appendix. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.isprsjprs.2023.01.014.

References

Alhichri, H., Alswayed, A.S., Bazi, Y., Ammour, N., Alajlan, N.A., 2021. Classification of remote sensing images using EfficientNet-B3 CNN model with attention. IEEE Access 9, 14078–14094. http://dx.doi.org/10.1109/ACCESS.2021.3051085.

Audebert, N., Le Saux, B., Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogramm. Remote Sens. 140, 20–32.

Ayhan, B., Kwan, C., Budavari, B., Kwan, L., Lu, Y., Perez, D., Li, J., Skarlatos, D., Vlachos, M., 2020. Vegetation detection using deep learning and conventional methods. Remote Sens. 12 (15), http://dx.doi.org/10.3390/rs12152502.

Ball, J.E., Anderson, D.T., Chan, Sr., C.S., 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. J. Appl. Remote Sens. 11 (4), 1–54.

Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., Nemani, R., 2015. DeepSat: A learning framework for satellite imagery. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. SIGSPATIAL '15, Association for Computing Machinery.

Bazi, Y., Bashmal, L., Rahhal, M.M.A., Dayil, R.A., Ajlan, N.A., 2021. Vision transformers for remote sensing image classification. Remote Sens. 13 (3), http://dx.doi.org/10.3390/rs13030516.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A., 2021. Understanding robustness of transformers for image classification. In: 2021 IEEE/CVF International Conference on Computer Vision. ICCV, IEEE Computer Society, Los Alamitos, CA, USA, pp. 10211–10221.

Blaschke, T., 2010. Object based image analysis for remote sensing. ISPRS J. Photogramm. Remote Sens. 65, 2–16.

Blaschke, T., Strobl, J., 2001. What's wrong with pixels? Some recent developments interfacing remote sensing and GIS. In: GIS – Zeitschrift für Geoinformationssysteme.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks, CoRR. http://dx.doi.org/10.48550/ARXIV.1508.00092.

Castillo-Navarro, J., Le Saux, B., Boulch, A., Lefèvre, S., 2022. Energy-based models in earth observation: From generation to semisupervised learning. IEEE Trans. Geosci. Remote Sens. 60, 1–11. http://dx.doi.org/10.1109/TGRS.2021.3126428.

Chaudhuri, B., Demir, B., Chaudhuri, S., Bruzzone, L., 2018. Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method. IEEE Trans. Geosci. Remote Sens. 56 (2), 1144–1158.

Chen, H., Chandrasekar, V., Tan, H., Cifelli, R., 2019. Rainfall estimation from ground radar and TRMM precipitation radar using hybrid deep neural networks. Geophys. Res. Lett. 46 (17–18), 10669–10678. http://dx.doi.org/10.1029/2019GL084771.

Chen, F., Tsou, J.Y., 2021. DRSNet: Novel architecture for small patch and low-resolution remote sensing image scene classification. Int. J. Appl. Earth Obs. Geoinf. 104, 102577. http://dx.doi.org/10.1016/j.jag.2021.102577.

Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: Benchmark and state of the art. Proc. IEEE 105 (10), 1865–1883. http://dx.doi.org/10.1109/JPROC.2017.2675998.

Cheng, G., Xie, X., Han, J., Guo, L., Xia, G.-S., 2020. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 3735–3756. http://dx.doi.org/10.1109/JSTARS.2020.3005403.

Chlingaryan, A., Sukkarieh, S., Whelan, B., 2018. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. Comput. Electron. Agric. 151, 61–69. http://dx.doi.org/10.1016/j.compag.2018.05.012.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dimitrovski, I., Kitanovski, I., Panov, P., Simidjievski, N., Kocev, D., 2022. AiTLAS: Artificial intelligence toolbox for earth observation. CoRR, abs/2201.08789. arXiv:2201.08789.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16×16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Gildenblat, J., contributors, 2021. PyTorch Library for CAM Methods. GitHub, https://github.com/jacobgil/pytorch-grad-cam.

Gong, N., Zhang, C., Zhou, H., Zhang, K., Wu, Z., Zhang, X., 2022. Classification of hyperspectral images via improved cycle-MLP. IET Comput. Vis. 16 (5), 468–478. http://dx.doi.org/10.1049/cvi2.12104.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Han, X., Zhong, Y., Cao, L., Zhang, L., 2017. Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. Remote Sens. 9 (8), http://dx.doi.org/10.3390/rs9080848.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Helber, P., Bischke, B., Dengel, A., Borth, D., 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.

Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. 7 (11), 14680–14707. http://dx.doi.org/10.3390/rs71114680.

Hua, Y., Mou, L., Zhu, X.X., 2019. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. ISPRS J. Photogramm. Remote Sens. 149, 188–199.

Hua, Y., Mou, L., Zhu, X.X., 2020. Relation network for multilabel aerial image classification. IEEE Trans. Geosci. Remote Sens. 58 (7), 4558–4572.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708.

Huang, B., Zhao, B., Song, Y., 2018. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. Remote Sens. Environ. 214, 73–86. http://dx.doi.org/10.1016/j.rse.2018.04.050.

Ienco, D., Gaetano, R., Dupaquier, C., Maurel, P., 2017. Land cover classification via multitemporal spatial data by deep recurrent neural networks. IEEE Geosci. Remote Sens. Lett. 14 (10), 1685–1689. http://dx.doi.org/10.1109/LGRS.2017.2728698.

Jo, Y.-H., Kim, D.-W., Kim, H., 2018. Chlorophyll concentration derived from microwave remote sensing measurements using artificial neural network algorithm. J. Mar. Sci. Technol. 26 (10), http://dx.doi.org/10.6119/JMST.2018.02_(1).0004.

Johnson, M.D., Hsieh, W.W., Cannon, A.J., Davidson, A., Bédard, F., 2016. Crop yield forecasting on the Canadian Prairies by remotely sensed vegetation indices and machine learning methods. Agricult. Forest Meteorol. 218–219, 74–84. http://dx.doi.org/10.1016/j.agrformet.2015.11.003.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X.X., 2018. Building instance classification using street view images. ISPRS J. Photogramm. Remote Sens. 145, 44–59.

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M., 2021. Transformers in vision: A survey. ACM Comput. Surv. http://dx.doi.org/10.1145/3505244.

Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S., 2020. A survey of the recent architectures of deep convolutional neural networks. Artif. Intell. Rev. 53 (8), 5455–5516.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kornblith, S., Shlens, J., Le, Q.V., 2019. Do better ImageNet models transfer better? In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2656–2666.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Li, H., Dou, X., Tao, C., Wu, Z., Chen, J., Peng, J., Deng, M., Zhao, L., 2020a. RSI-CB: A large-scale remote sensing image classification benchmark using crowdsourced data. Sensors 20 (6), 1594. http://dx.doi.org/10.3390/s20061594.

Li, H., Jiang, H., Gu, X., Peng, J., Li, W., Hong, L., Tao, C., 2020b. CLRS: Continual learning benchmark for remote sensing image scene classification. Sensors 20 (4).

Li, J., Lin, D., Wang, Y., Xu, G., Zhang, Y., Ding, C., Zhou, Y., 2020c. Deep discriminative representation learning with attention map for scene classification. Remote Sens. 12 (9).

Li, M., Zang, S., Zhang, B., Li, S., Wu, C., 2014. A review of remote sensing image classification techniques: the role of spatio-contextual information. Eur. J. Remote Sens. 47 (1), 389–411. http://dx.doi.org/10.5721/EuJRS20144723.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 936–944. http://dx.doi.org/10.1109/CVPR.2017.106.

Liu, S., He, C., Bai, H., Zhang, Y., Cheng, J., 2020a. Light-weight attention semantic segmentation network for high-resolution remote sensing images. In: IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 2595–2598.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., Guo, B., 2022a. Swin transformer V2: Scaling up capacity and resolution. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 11999–12009. http://dx.doi.org/10.1109/CVPR52688.2022.01170.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J., 2020b. On the variance of the adaptive learning rate and beyond. In: Proceedings of the Eighth International Conference on Learning Representations. ICLR 2020.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision. ICCV, pp. 9992–10002. http://dx.doi.org/10.1109/ICCV48922.2021.00986.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022b. A ConvNet for the 2020s. arXiv preprint arXiv:2201.03545.

Long, Y., Gong, Y., Xiao, Z., Liu, Q., 2017. Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55 (5), 2486–2498.

Longbotham, N., Chaapel, C., Bleiler, L., Padwick, C., Emery, W.J., Pacifici, F., 2012. Very high resolution multiangle urban classification analysis. IEEE Trans. Geosci. Remote Sens. 50 (4), 1155–1170. http://dx.doi.org/10.1109/TGRS.2011.2165548.

Lv, Z., Liu, T., Benediktsson, J.A., Falco, N., 2022. Land cover change detection techniques: Very-high-resolution optical images: A review. IEEE Geosci. Remote Sens. Mag. 10 (1), 44–63. http://dx.doi.org/10.1109/MGRS.2021.3088865.

Marcel, S., Rodriguez, Y., 2010. Torchvision the machine-vision package of torch. In: Proceedings of the 18th ACM International Conference on Multimedia. pp. 1485–1488.

Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2016. Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. 13 (1), 105–109. http://dx.doi.org/10.1109/LGRS.2015.2499239.

Meng, Z., Zhao, F., Liang, M., 2021. SS-MLP: A novel spectral-spatial MLP architecture for hyperspectral image classification. Remote Sens. 13 (20), http://dx.doi.org/10.3390/rs13204060.

Neumann, M., Pinto, A.S., Zhai, X., Houlsby, N., 2020. Training general representations for remote sensing using in-domain knowledge. In: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium. pp. 6730–6733. http://dx.doi.org/10.1109/IGARSS39084.2020.9324501.

Papoutsis, I., Bountos, N.-I., Zavras, A., Michail, D., Tryfonopoulos, C., 2022. Efficient deep learning models for land cover image classification. arXiv:2111.09451.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.

Paul, S., Chen, P.-Y., 2022. Vision transformers are robust learners. Proc. AAAI Conf. Artif. Intell. 36 (2), 2071–2081.

Penatti, O.A., Nogueira, K., Dos Santos, J.A., 2015. 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. pp. 44–51.

Planet, SCOON, 2022. Planet: Understanding the amazon from space. URL https://www.kaggle.com/competitions/planet-understanding-the-amazon-from-space. (Last accessed 21 May 2022).

Qi, X., Zhu, P., Wang, Y., Zhang, L., Peng, J., Wu, M., Chen, J., Zhao, X., Zang, N., Mathiopoulos, P.T., 2020. MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. ISPRS J. Photogramm. Remote Sens. 169, 337–350.

Risojevic, V., Stojnic, V., 2021. Do we still need ImageNet pre-training in remote sensing scene classification? arXiv, abs/2111.03690. arXiv:2111.03690.

Sadeghi, M., Asanjan, A.A., Faridzad, M., Nguyen, P., Hsu, K., Sorooshian, S., Braithwaite, D., 2019. PERSIANN-CNN: Precipitation estimation from remotely sensed information using artificial neural networks–convolutional neural networks. J. Hydrometeorol. 20 (12), 2273–2289. http://dx.doi.org/10.1175/JHM-D-19-0110.1.

Scheibenreif, L., Hanna, J., Mommert, M., Borth, D., 2022. Self-supervised vision transformers for land-cover segmentation and classification. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 1421–1430. http://dx.doi.org/10.1109/CVPRW56347.2022.00148.

Schneider, R., Bonavita, M., Geer, A., Arcucci, R., Dueben, P., Vitolo, C., Le Saux, B., Demir, B., Mathieu, P.-P., 2022. ESA-ECMWF report on recent progress and research directions in machine learning for earth system observation and prediction. npj Clim. Atmospheric Sci. 5 (1), 51. http://dx.doi.org/10.1038/s41612-022-00269-z.

Sechidis, K., Tsoumakas, G., Vlahavas, I., 2011. On the stratification of multi-label data. In: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III. Springer-Verlag, pp. 145–158.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision. ICCV, pp. 618–626.

Shirmard, H., Farahbakhsh, E., Müller, R.D., Chandra, R., 2022. A review of machine learning in processing remote sensing data for mineral exploration. Remote Sens. Environ. 268, 112750. http://dx.doi.org/10.1016/j.rse.2021.112750.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Somrak, M., Dzeroski, S., Kokalj, Z., 2020. Learning to classify structures in ALS-derived visualizations of ancient Maya settlements with CNN. Remote Sens. 12 (14), 2215. http://dx.doi.org/10.3390/rs12142215.

Stewart, A.J., Robinson, C., Corley, I.A., Ortiz, A., Ferres, J.M.L., Banerjee, A., 2021. TorchGeo: deep learning with geospatial data. CoRR, abs/2111.08872. arXiv:2111.08872.

Sumbul, G., Charfuelan, M., Demir, B., Markl, V., 2019. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium. pp. 5901–5904.

Sumbul, G., de Wall, A., Kreuziger, T., Marcelino, F., Costa, H., Benevides, P., Caetano, M., Demir, B., Markl, V., 2021. BigEarthNet-MM: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. IEEE Geosci. Remote Sens. Mag. 9 (3), 174–180.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114.

Tian, Z., Wang, W., Tian, B., Zhan, R., Zhang, J., 2020. Resolution-aware network with attention mechanisms for remote sensing object detection. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 5 (2).

Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al., 2021. Mlp-mixer: An all-mlp architecture for vision. Adv. Neural Inf. Process. Syst. 34.

Tong, W., Chen, W., Han, W., Li, X., Wang, L., 2020. Channel-attention-based DenseNet network for remote sensing image scene classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 4121–4132. http://dx.doi.org/10.1109/JSTARS.2020.3009352.

Tsoumakas, G., Katakis, I., 2009. Multi-label classification: An overview. Int. J. Data Warehous. Min. 3, 1–13.

Tuia, D., Ratle, F., Pacifici, F., Kanevski, M.F., Emery, W.J., 2009. Active learning methods for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 47 (7), 2218–2232. http://dx.doi.org/10.1109/TGRS.2008.2010404.

Wang, Y., Albrecht, C.M., Braham, N.A.A., Mou, L., Zhu, X.X., 2022a. Self-supervised learning in remote sensing: A review. CoRR. http://dx.doi.org/10.48550/ARXIV.2206.13188.

Wang, Q., Liu, S., Chanussot, J., Li, X., 2019. Scene classification with recurrent attention of VHR remote sensing images. IEEE Trans. Geosci. Remote Sens. 57 (2), 1155–1167.

Wang, D., Zhang, J., Du, B., Xia, G.-S., Tao, D., 2022b. An empirical study of remote sensing pretraining. IEEE Trans. Geosci. Remote Sens. 1. http://dx.doi.org/10.1109/TGRS.2022.3176603.

Weng, Q., Mao, Z., Lin, J., Guo, W., 2017. Land-use classification via extreme learning classifier based on deep convolutional features. IEEE Geosci. Remote Sens. Lett. 14 (5), 704–708. http://dx.doi.org/10.1109/LGRS.2017.2672643.

Wightman, R., 2019. PyTorch image models. http://dx.doi.org/10.5281/zenodo.4414861, GitHub Repository. https://github.com/rwightman/pytorch-image-models.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al., 2016. The FAIR guiding principles for scientific data management and stewardship. Sci. Data 3 (1), 1–9.

Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., Lu, X., 2017. AID: A benchmark data set for performance evaluation of aerial scene classification. IEEE Trans. Geosci. Remote Sens. 55 (7), 3965–3981.

Xia, G.-S., Yang, W., Delon, J., Gousseau, Y., Sun, H., Maître, H., 2010. Structural high-resolution satellite image indexing. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. - ISPRS Arch. 38.

Xu, J., Yang, J., Xiong, X., Li, H., Huang, J., Ting, K., Ying, Y., Lin, T., 2021a. Towards interpreting multi-temporal deep learning models in crop mapping. Remote Sens. Environ. 264, 112599. http://dx.doi.org/10.1016/j.rse.2021.112599.

Xu, Z., Zhang, W., Zhang, T., Yang, Z., Li, J., 2021b. Efficient transformer for remote sensing image segmentation. Remote Sens. 13 (18), http://dx.doi.org/10.3390/rs13183585.

Yang, Y., Newsam, S., 2010. Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. Association for Computing Machinery, pp. 270–279.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 3320–3328.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. CoRR. http://dx.doi.org/10.48550/ARXIV.1605.07146.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., Houlsby, N., 2019. A large-scale study of representation learning with the visual task adaptation benchmark. arXiv:1910.04867.

Zhang, J., Lu, C., Li, X., Kim, H.-J., Wang, J., 2019. A full convolutional network based on DenseNet for remote sensing scene classification. Math. Biosci. Eng. 16 (5), 3345–3367. http://dx.doi.org/10.3934/mbe.2019167.

Zhang, C., Wang, L., Cheng, S., Li, Y., 2022a. SwinSUNet: Pure transformer network for remote sensing image change detection. IEEE Trans. Geosci. Remote Sens. 60, 1–13. http://dx.doi.org/10.1109/TGRS.2022.3160007.

Zhang, L., Zhang, L., Du, B., 2016. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. 4 (2), 22–40. http://dx.doi.org/10.1109/MGRS.2016.2540798.

Zhang, C., Zhang, M., Zhang, S., Jin, D., feng Zhou, Q., Cai, Z., Zhao, H., Yi, S., Liu, X., Liu, Z., 2022b. Delving deep into the generalization of vision transformers under distribution shifts. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 7267–7276.

Zhang, X., Zhang, Q., Zhang, G., Nie, Z., Gui, Z., Que, H., 2018. A novel hybrid data-driven model for daily land surface temperature forecasting using long short-term memory neural network based on ensemble empirical mode decomposition. Int. J. Environ. Res. Public Health 15 (5).

Zhou, W., Newsam, S., Li, C., Shao, Z., 2018. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J. Photogramm. Remote Sens. 145, 197–209.

Zhu, X.X., Hu, J., Qiu, C., Shi, Y., Kang, J., Mou, L., Bagheri, H., Haberle, M., Hua, Y., Huang, R., Hughes, L., Li, H., Sun, Y., Zhang, G., Han, S., Schmitt, M., Wang, Y., 2020. So2Sat LCZ42: A benchmark data set for the classification of global local climate zones [software and data sets]. IEEE Geosci. Remote Sens. Mag. 8 (3), 76–89.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. 5 (4), 8–36. http://dx.doi.org/10.1109/MGRS.2017.2762307.

Zhu, Q., Zhong, Y., Zhao, B., Xia, G.-S., Zhang, L., 2016. Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. IEEE Geosci. Remote Sens. Lett. 13 (6), 747–751.

Zou, Q., Ni, L., Zhang, T., Wang, Q., 2015. Deep learning based feature selection for remote sensing scene classification. IEEE Geosci. Remote Sens. Lett. 12 (11), 2321–2325. http://dx.doi.org/10.1109/LGRS.2015.2475299.