AiTLAS: Artificial Intelligence Toolbox for Earth Observation

Ivica Dimitrovski[1,2,*] , **Ivan Kitanovski**[1,2] , **Panče Panov**[1,3] ,
Nikola Simidjievski[1,4] and **Dragi Kocev**[1,3*]

[1]Bias Variance Labs, Ljubljana, Slovenia
[2]University of Ss Cyril and Methodius, FCSE, Skopje, N. Macedonia
[3]Jožef Stefan Institute, Ljubljana, Slovenia
[4]University of Cambridge, United Kingdom
{ivica, ivan, pance, nikola, dragi}@bvlabs.ai

Abstract

The AiTLAS toolbox (Artificial Intelligence Toolbox for Earth Observation) includes state-of-the-art machine learning methods for exploratory and predictive analysis of satellite imagery as well as repository of AI-ready Earth Observation (EO) datasets. It can be easily applied for a variety of Earth Observation tasks, such as land use and cover classification, crop type prediction, localization of specific objects (semantic segmentation), etc. The main goal of AiTLAS is to facilitate better usability and adoption of novel AI methods (and models) by EO experts, while offering easy access and standardized format of EO datasets to AI experts which further allows benchmarking of various existing and novel AI methods tailored for EO data.

1 Introduction

According to the Sentinel Data Access Annual Reports [Castriotta and Volpi, 2020], the volume of data produced by the Sentinel satellite mission(s) is substantially growing – doubling each year since 2017 (to 19.5 PB in 2019), with data downloads growing at a similar rate (to 178.5 PB in 2019). On the other hand, this growth rate in data production is well matched by the rapidly growing development in AI, which probes various aspects of natural sciences, technology and society. Recent trends in machine learning, and particular deep learning, have ushered to a new era of image analysis and raised the predictive performance bar in many application domains, including remote sensing and Earth observation [Ball *et al.*, 2017]. Nevertheless, despite the success of some individual attempts, given the amount of available satellite images, the application of AI in the EO supply chain is still scarce and, more importantly, still unstructured and unverifiable. This is due to a number of challenges that require immediate attention: (1) lack of benchmark data sets produced in an AI-ready format, (2) lack of data and methods for benchmarking and model comparisons, and (3) lack of ready-to-use, comprehensive resources.

AiTLAS[1] addresses the aforementioned challenges and

---

aims at resolving them. It is a recent open-source library designed to facilitate the use of EO data in the AI community and, more importantly, to accelerate the uptake of (advanced) machine learning methods and approaches by EO experts. More specifically, AiTLAS provides resources such as: benchmarking tools, ready-to-use models, tools for learning models de novo, semantically annotated datasets created in a format that can be used directly by AI methods.

2 AiTLAS

We designed and developed a toolbox that contains state-of-the-art AI methods and techniques and follows the best practices for machine learning use and deployment. The toolbox is based on three main concepts: *models*, *datasets* and *tasks*. The *models* concept defines the architecture of the deep learning models as well as their behavior. The *datasets* concept defines the data by encapsulating certain operations over datasets, such as loading and preparing the data as well as acquiring and, optionally, transforming a given item. Finally, the *task* concept is used to make a complete workflow where a given model is used over a given dataset. In the toolbox, we have defined several tasks that are common for these types of problems, such as, tasks for *training* and *evaluating* a model over a given dataset, etc.

Additionally, AiTLAS also includes *transformations*, *visualizations* and *evaluation measures*, as supportive concepts to the previous ones. In particular, various *transformations* can be applied to the data if needed before training. The *evaluation measures* are used to gauge the performance of the model (e.g., accuracy, F1 score etc.) that, depending on the tasks, can also be illustrated with a variety of *visualizations* (e.g., showing mask overlays makes sense for segmentation). Note that, for increasing usability, the evaluation measures, along with the loss functions, are logged in a Tensorboard [Abadi *et al.*, 2015] friendly format.

2.1 Software implementation

AiTLAS is developed using `Python` and distributed under the MIT license. It is developed as a library but can be used as a standalone application as well. All the models, datasets and tasks can be configured via a proprietary JSON format or programatically by initializing the appropriate classes. The toolbox has a variety of dependencies that are used to power it,

but the main one is PyTorch [Paszke *et al.*, 2019]. This serves as the backbone over which deep learning models and behavior are built.

All the functionalities of AiTLAS are organized into five separate modules:

- `aitlas.base` - The core module that contains the abstract definitions of everything we have in AiTLAS. The definitions of the tasks, models, datasests, and transformations, evaluation measures and visualizations. To extend the functionality of AiTLAS, one would need to extend the definitions implemented here.

- `aitlas.models` - Contains specific implementations of the deep learning models.

- `aitlas.datasets` - Contains The datasets and dataset types which are supported by AiTLAS, out of the box.

- `aitlas.tasks` - Includes readily executable workflows. There are several implemented tasks in AiT-LAS, such as: training task, evaluating tasks, task to run some data preparations, tasks for extracting features etc. While the ones currently implemented are can be applied in many different scenarios, they can also serve as a blueprint for creating/instantiating new, more specific, task.

- `aitlas.utils` - Contains handy utility functions, which can be used within the toolbox or outside of it.

2.2 Methods and models

The toolbox supports a number of models that have been shown to lead to improved predictive performance. Its flexible design of the software architecture allows users to easily include other models, including implementing their own custom made models. AiTLAS currently includes the following models: DeepLabv3 [Chen *et al.*, 2017], Fast R-CNN [Girshick, 2015], ResNet [He *et al.*, 2015], VGG16 [Simonyan and Zisserman, 2015], and Unsupervised DeepCluster [Caron *et al.*, 2019].

The implemented models cover a variety of EO use cases: land use and land cover classification, semantic segmentation and object detection in the context of EO, etc. The models support both multi class and multi label classification. The multi class models assign a single label per image, whereas the multi label models assign multiple labels (or, in practice, land covers) in each image, which is closer to what is used in real life.

The Unsupervised DeepCluster [Caron *et al.*, 2019] model can leverage large amounts of unlabeled images to learn robust features. It uses clustering in combination with deep neural networks to provide pseudo-labels for a convolutional neural network. Within the toolbox, we are using the unsupervised deep learning to learn general features from a large set of unlabeled satellite images and then apply fine-tuning using datasets with a limited number of labeled images for a given task.

2.3 Datasets

AiTLAS has a number of datasets available. We have separated them into two groups: (i) land-use/cover classification datasets and (ii) object segmentation datasets. Note that, land use datasets are where the annotation problem is a multi class or a multi label type, where as the latter tackles segmentation. To this end, the former includes: UC Merced [Yang and Newsam, 2010; Chaudhuri *et al.*, 2017], Eurosat [Helber *et al.*, 2019], BigEarthNet [Sumbul *et al.*, 2019], DFC15 [Hua *et al.*, 2019], PatternNet [Zhou *et al.*, 2018], AID [Xia *et al.*, 2017] and RESISC45 [Cheng *et al.*, 2017]. The latter includes: Chactun [Somrak *et al.*, 2020], SpaceNet 6 [Dukai, 2018] and Landcover AI [Boguszewski *et al.*, 2020].

3 AiTLAS use case

In order to demonstrate the functionalities of the toolbox, we present a Jupyter Notebook for multi label land cover classification of satellite images. We will show how to load the data and inspect the classes and class imbalances in the dataset. Next, we train our model using convolution neural networks, and finally, we test the model with external images for inference.

We use the multi label UC Merced dataset with 17 land cover classes. Once we get the data and unzip it[2], we are ready to explore it. In order to achieve this we need to create and instance of the UcMercedMultiLabelDataset, which is a class within the toolbox that can handle loading of the data. The instance is created by providing the root folder in which the images and annotations are unzipped, additionally we can supply the batch size to be used in the data loader during the process of training, the shuffle parameter to have the data reshuffled at every epoch and number of workers to specify the subprocesses to use for data loading. As an illustration the code snippet is given in Listing 1.

```
train_dataset_config = {
    "batch_size": 16,
    "shuffle": True,
    "num_workers": 4,
    "root": "/ucmerced/images"
}
train_dataset = UcMercedMultiLabelDataset(
    train_dataset_config)
fig = train_dataset.show_image(340)
```

Listing 1: Load train dataset

To display a given image from the dataset you can use the function `show_image` implemented in the class for the dataset. The function expects an index/order number and will display the image and the annotations/labels for the image. The example image with index/order number 340 and the annotations/labels are shown on Figure 1.

Before we move on to the machine learning task, we can inspect the distribution of the classes in the dataset. Checking the distribution of the dataset is an important step to check for data imbalances in your dataset. The distribution can be easily calculted using the function `data_distribution_table` implemented in the class for the dataset. The distribution for the dataset is given in Figure 1. The data reveals the class imbalances in the dataset, the pavement class has 987 images while Airplane class has 78 images.

---

[2]https://bigearth.eu/datasets.html

Next, we need to initialize and create the model with configuration. We will use the ResNet50 architecture which is implemented in the toolbox. The configuration parameters are the number of epoch, the path in which to save the final model parameters, the visualizations and the results, the number of classes, the learning rate, the threshold to obtain the predictions and calculate the various metrics. By setting the parameter pretrained to $true$ we state that we want to use the pretrained variant of ResNet50 over the ImageNet dataset and to apply fine tuning using the UC merced multi label dataset. Now, we can start training our model with the data. We use the `train_and_evaluate_model` function. The function accepts the train dataset created using the code from Listing 1, similar to this we can create the test dataset using different root folder (UC Merced multi label dataset has predefined train and test splits). We can also select to apply transformations over the images, this can be configured using the parameters `transforms` and `target_transforms` defined for the dataset class. As an ilustration the code snippet for initializing the model and initiating the training is given in Listing 2.

```
1  epochs = 50
2  model_directory = "/ucmerced/experiments/"
3  model_config = {"num_classes": 17, "
       learning_rate": 0.0001,"pretrained": True
       , "threshold": 0.5}
4  model = ResNet50MultiLabel(model_config)
5  model.prepare()
6  model.train_and_evaluate_model(
7      train_dataset=train_dataset,
8      epochs=epochs,
9      model_directory=model_directory,
10     val_dataset=test_dataset,
11     run_id='1',)
```
Listing 2: Creating a model and start of model training

Once the training starts, the AiTLAS toolbox displays the running time for each epoch, calculated loss and the evaluation metrics. The final model will be saved in the model

Figure 1: Example image from the UC Merced multi-label dataset (Image#340) for which the predictions are 'buildings', 'cars', 'pavement' and 'trees'.

directory given in the configuration parameter. The calculated metrics and losses are logged and can be visualized using *tensorboardX*. To test the model, we can predict several images from an external source and see how the model performs. This can be done using the code from Listing 3. First we load the image using the utility function `image_loader` from the toolbox, load the model parameters from the saved file and call the predict function. The predict function returns the predicted labels and the probability for each label.

```
1  model.load_model('ucmerced/experiments')
2  image = image_loader('images/predict')
3  plt.imshow(image)
4  y_true, y_pred, y_prob = model.predict_image(
       image)
```
Listing 3: Loading trained model and predicting image

The entire code and Jupyter Notebook for this tutorial is available in the Github repository of the AiTLAS toolbox[3].

4 Summary

We have described AiTLAS, an open-source, state-of-the-art toolbox for exploratory and predictive analysis of satellite imaginary pertaining to a variety of different tasks in Earth Observation. AiTLAS has several distinguishing properties. First, it is modular and flexible - allowing for easy configuration, implementation and extension of new data and models. Next, it is general and applicable to a variety of tasks and workflows. Finally, it is user-friendly. This, besides aiding the AI community by providing access to structured EO data, more importantly, facilitates and accelerates the uptake of (advanced) machine learning methods by the EO experts, thus bringing these two communities closer together.

Acknowledgments

---

[3]https://github.com/biasvariancelabs/aitlas/tree/master/examples

References

[Abadi *et al.*, 2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[Ball *et al.*, 2017] John E. Ball, Derek T. Anderson, and Chee Seng Chan Sr. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 11(4):1 – 54, 2017.

[Boguszewski *et al.*, 2020] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Anna Zambrzycka, and Tomasz Dziedzic. Landcover.ai: Dataset for automatic mapping of buildings, woodlands and water from aerial imagery, 2020.

[Caron *et al.*, 2019] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features, 2019.

[Castriotta and Volpi, 2020] Adriana Grazia Castriotta and Federica Volpi. "copernicus sentinel data access anual report, 2020.

[Chaudhuri *et al.*, 2017] Bindita Chaudhuri, Begüm Demir, Subhasis Chaudhuri, and Lorenzo Bruzzone. Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1144–1158, 2017.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[Cheng *et al.*, 2017] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017.

[Dukai, 2018] Balázs Dukai. 3d registration of buildings and addresses (bag) / 3d basisregistratie adressen en gebouwen (bag), Nov 2018.

[Girshick, 2015] Ross Girshick. Fast r-cnn, 2015.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[Helber *et al.*, 2019] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[Hua *et al.*, 2019] Yuansheng Hua, Lichao Mou, and Xiao Xiang Zhu. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional lstm network for multi-label aerial image classification. *ISPRS journal of photogrammetry and remote sensing*, 149:188–199, 2019.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[Somrak *et al.*, 2020] Maja Somrak, Sašo Džeroski, and Žiga Kokalj. Learning to classify structures in als-derived visualizations of ancient maya settlements with cnn. *Remote Sensing*, 12(14), 2020.

[Sumbul *et al.*, 2019] Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5901–5904. IEEE, 2019.

[Xia *et al.*, 2017] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

[Yang and Newsam, 2010] Yi Yang and Shawn Newsam. 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*, pages 270–279, 2010.

[Zhou *et al.*, 2018] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018.