Cloud service of Geoportal ISDCT SB RAS for machine learning

Yuriy V. Avramenko, Anastasiya K. Popova and Roman K. Fedorov

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Lermontova str. 134, Irkutsk, 664033, Russia

Abstract

The paper describes the cloud service of ISDCT SB RAS for machine learning research. The introduction discusses the relevance of creating a service. Further, the theoretical part is considered, which describes the component of the services and the principle of their interaction. Then the results of practical application and discussion are presented. In the conclusion, the results of the work are summarized.

Keywords

WPS, remote sensing, machine learning

1. Introduction

Researchers often choose between existing methods and the development of new ones for solving practical problems [1-3]. In most cases, a known method is used with some modifications. This choice is associated with both the development of the technical part and the software. This problem is especially acute with free software, where patches and fixes are often released, and sometimes the required library is not supported by the developers at all. There are several ways to get around this limitation, for example, create a virtual machine or environment and then install the necessary software, use a Docker image with preinstalled software, use cloud services with necessary software. After choosing a suitable method, the researchers are testing the algorithms on the data given by the method author. If the test data matches the custom data in its characteristics, then we can assume that the method works well.

In our work, we use machine learning methods to classify the land cover with multispectral remote sensing images. On the remote sensing data the spectral characteristics can significantly differ from each other, since they depend on many factors so the same algorithms can give different results. Therefore, it is important to be able to apply the method and get the expected result on a custom dataset.

We tried to repeat the classification method used in work [4] to classify the south of the Irkutsk region and got negative results - the entire territory was classified as water. This happened due to the fact that the values of the spectral bands of the EuroSAT set, on which the training was carried out, significantly differ from the values of the corresponding bands of the studied territory. Therefore, we need an environment where we can flexibly customize methods for specific tasks.

A cloud service for machine learning was created at the ISDCT SB RAS as part of an applied digital platform. The goal is to provide technical and software base for the development of new and testing of known methods. The service takes into account factors such as speed of deployment, customization flexibility, user preferences and scalability.

2. Main idea

ITAMS 2021 – Information Technologies: Algorithms, Models, Systems, September 14, 2021, Irkutsk, Russia
EMAIL: avramenko@icc.ru (A. 1); popova@icc.ru (A. 2); fedorov@icc.ru (A. 3)
ORCID: 0000-0002-3082-1155 (A. 1); 0000-0001-6209-678X (A. 2); 0000-0002-2944-7522 (A. 3)

In the process of the service developing, the existing approaches were studied [5-7], the functionality for the effective and convenient work of users was determined for processing remote sensing data with machine learning. As a result, we have defined the requirements for the services:

- automation of repetitive user actions;
- support for multi-user work;
- sharing results;
- fine tuning.

Description and role of the main components of the Geoportal ISDCT cloud service [8-10]:

- JupyterHub provides JupyterLab capabilities to users groups. Contains a set of rules for running Docker containers.
- NextCloud is a set of client-server programs for creating and using cloud storage. Provides users with OAuth 2.0 sign-in and network storage access.
- Kubernetes is open source software for orchestrating Docker containers, to automate their deployment, to scale and coordinate in a cluster environment. Allows to add compute nodes and define rules for their use.
- Docker is software for automating the deployment and management of applications in virtualized environments. Create custom images or run existing ones.
- JupyterLab is an interactive web-based Python and R code and data development environment. Algorithm development and testing.
- PyWPS allows create and deploy custom geospatial operations (as processes) on the server. Provides algorithms for processing remote sensing data, automatically updates the remote sensing database.
- Compute nodes – physical or virtual machines for users.
- Network storage – contains the remote sensing database, user files.
- Interactive map displays remote sensing data, simplifies the users work withn creating a training sample, and allows call tools for processing remote sensing.

Figure 1 shows a general diagram of the cloud service components interaction.

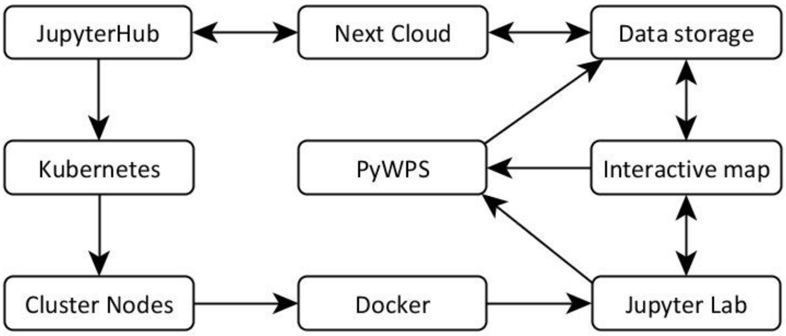


Figure 1: Cloud service components interaction diagram

The cloud service is focused on the development, implementation and testing of remote sensing methods. Distinctive features of the service from existing analogues are:

- unlimited working time;
- increased data storage;
- interaction of an interactive map with the development environment;
- the ability to run custom Docker images.

In the course of practical experiments, the algorithms for the land cover classification from [4] were repeated. The main difficulties in reproducing the results were conflicts between versions of the libraries and operating system, the way of reading the data. We tested Ubuntu versions 16 and 18. The versions of the data processing libraries were selected in two ways, based on the existing build in Google Colab and by comparing the release date.

On the technical side, the method is demanding on computational resources, so two Docker images were created: a regular one for writing algorithms and a high-performance one with support for CUDA technology, which is necessary for neural networks training. Switching between images occurs with KubeSpawner, a plug-in for Kubernetes integration in JupyterHub. KubeSpawner allows determine the number of processor cores, the amount of RAM, access to video cards and other parameters. Next, Kubernetes looks for a suitable node to run the image. The resulting images are used to solve machine learning tasks. Next, we consider the developed and implemented algorithms for solve the machine learning tasks.

3. Practical application

Training sample balancing algorithm. The algorithm input is the path to the data divided by classes in various directories. Each directory contains N files - containers for training data. First, a list of pairs is formed (file name, sample serial number). Next, a list of characteristics for clustering is built (the arithmetic mean and standard deviation of the sample pixels for each band). Clustering is performed based on the list of characteristics. The result is ordered in ascending order of the number of elements in the cluster. After that, samples are taken from each cluster according to the rule – if number of cluster elements are less than a specified threshold value, then we take everything, otherwise with a certain step. This approach guarantees that rare samples will definitely be included in the training set, and frequent ones will be thinned out. Figure 2-3 shows results of the algorithm work.

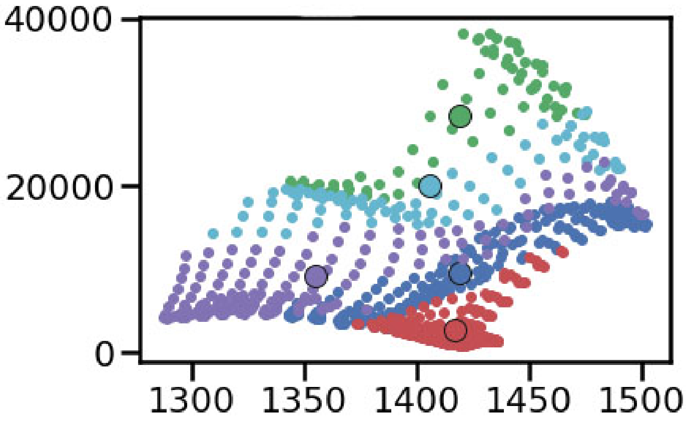
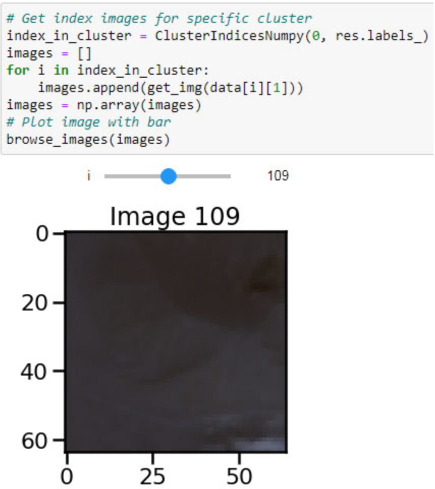


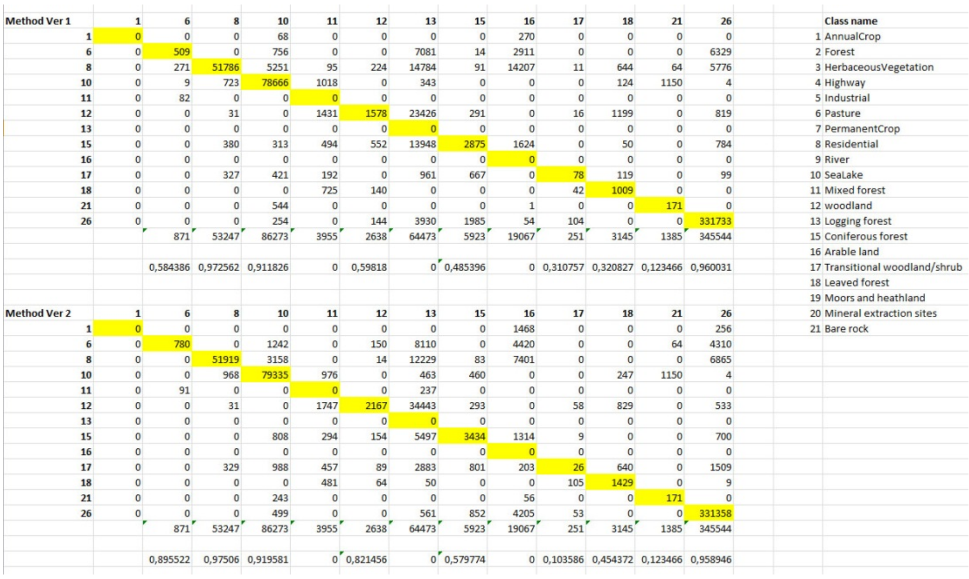
Figure 2: Data clustering

Classification quality control algorithm. The algorithm input is two files, the result of the classification and the labeled dataset. Since it is not possible for a specialist to completely label the whole image, the check is carried out only in the areas corresponding to the markup. The PyCM library is used to calculate statistics. PyCM is a multi-class confusion matrix library written in Python that supports both input data vectors and direct matrix, and a proper tool for post-classification model evaluation that supports most classes and overall statistics parameters. ACC (Accuracy), PPV (Precision or positive predictive value) indicators were used as the main criteria. As a result of the check, it became possible to compare two versions of the method and choose the best one. The error matrix for each version of the method is shown in Figure 4.



```
# Get index images for specific cluster
index_in_cluster = ClusterIndicesNumpy(0, res.labels_)
images = []
for i in index_in_cluster:
    images.append(get_img(data[i][1]))
images = np.array(images)
# Plot image with bar
browse_images(images)
```

Figure 3: Checking the sequence of images



Method Ver 1	1	6	8	10	11	12	13	15	16	17	18	21	26
1	0	0	0	68	0	0	0	0	270	0	0	0	0
6	0	509	0	756	0	0	7081	14	2911	0	0	0	6329
8	0	271	51786	5251	95	224	14784	91	14207	11	644	64	5776
10	0	9	723	78666	1018	0	343	0	0	0	124	1150	4
11	0	82	0	0	0	0	0	0	0	0	0	0	0
12	0	0	31	0	1431	1578	23426	291	0	16	1199	0	819
13	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	380	313	494	552	13948	2875	1624	0	50	0	784
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	327	421	192	0	961	667	0	78	119	0	99
18	0	0	0	0	725	140	0	0	0	42	1009	0	0
21	0	0	0	544	0	0	0	0	1	0	0	171	0
26	0	0	0	254	0	144	3930	1985	54	104	0	0	331733
		871	53247	86273	3955	2638	64473	5923	19067	251	3145	1385	345544
		0,584386	0,972562	0,911826	0	0,59818	0	0,485396	0	0,310757	0,320827	0,123466	0,960031

Method Ver 2	1	6	8	10	11	12	13	15	16	17	18	21	26
1	0	0	0	0	0	0	0	0	1468	0	0	0	256
6	0	780	0	1242	0	150	8110	0	4420	0	0	64	4310
8	0	0	51919	3158	0	14	12229	83	7401	0	0	0	6865
10	0	0	968	79335	976	0	463	460	0	0	247	1150	4
11	0	91	0	0	0	0	237	0	0	0	0	0	0
12	0	0	31	0	1747	2167	34443	293	0	58	829	0	533
13	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	808	294	154	5497	3434	1314	9	0	0	700
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	329	988	457	89	2883	801	203	26	640	0	1509
18	0	0	0	0	481	64	50	0	0	105	1429	0	9
21	0	0	0	243	0	0	0	0	56	0	0	171	0
26	0	0	0	499	0	0	561	852	4205	53	0	0	331358
		871	53247	86273	3955	2638	64473	5923	19067	251	3145	1385	345544
		0,895522	0,97506	0,919581	0	0,821456	0	0,579774	0	0,103586	0,454372	0,123466	0,958946

	Class name
1	AnnualCrop
2	Forest
3	HerbaceousVegetation
4	Highway
5	Industrial
6	Pasture
7	PermanentCrop
8	Residential
9	River
10	SeaLake
11	Mixed forest
12	woodland
13	Logging forest
15	Coniferous forest
16	Arable land
17	Transitional woodland/shrub
18	Leaved forest
19	Moors and heathland
20	Mineral extraction sites
21	Bare rock

Figure 4: Checking the sequence of images

The result of applying adapted method from work [1] is shown in Figure 5.

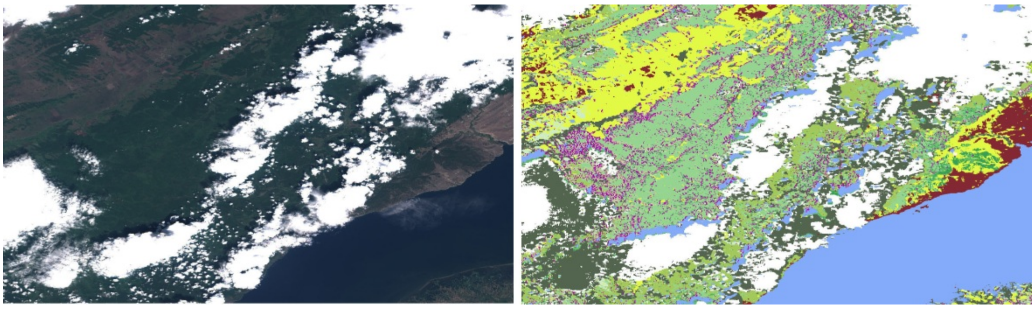


Figure 5: The result of the classification of the image of the southern part of Lake Baikal. Left - original image, right - land cover.

4. Conclusion

The purpose of this study is to create a cloud service for the ISDCT SB RAS Geoportal for machine learning tasks. We took into account the needs of users and the features of the tasks to be solved during the design stage. This service allows users to apply data processing methods to solve practical problems, develop and implement new methods. Unlike existing services, the proposed one has the following advantages: unlimited working time; increased data storage; connection of the interactive map with the development environment; the ability to run custom Docker images. During testing, we gained experience in adapting methods for specific tasks, taking into account the specifics of the processed data, repeating the method close to the original in the software part and partly in the technical one in the shortest possible time.

5. Acknowledgements

The results were obtained within the framework of the State Assignment of the Ministry of Education and Science of the Russian Federation for the project "Methods and technologies of cloud-based service-oriented platform for collecting, storing and processing large volumes of multi-format interdisciplinary data and knowledge based upon the use of artificial intelligence, model-guided approach and machine learning" (state registration number 121030500071-2). Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex».

6. References

[1] M. Martini, V. Mazzia, A. Khaliq, M. Chiaberge, Domain-adversarial training of self-attention based networks for land cover classification using multi-temporal sentinel-2 satellite imagery. Computer Vision and Pattern Recognition, p20, (2021) arXiv: 2104.00564.

[2] L. Alonso, J. Picos, and J. Armesto, Forest Land Cover Mapping at a Regional Scale Using Multi-Temporal Sentinel-2 Imagery and RF Models, Remote Sens, Volume 13, Issue 12 (2021). doi:10.3390/rs13122237.

[3] Klaudia Weronika Pałas, Jarosław Zawadzki. Sentinel-2 Imagery Processing for Tree Logging Observations on the Białowieża Forest World Heritage Site. Forests. Volume 11, Issue 8 (2020). doi:10.3390/f11080857.

[4] T. Chambon, Fighting Hunger through Open Satellite Data: A New State of the Art for Land Use Classification, 2019. URL: https://medium.com/omdena/fighting-hunger-through-open-satellite-data-a-new-state-of-the-art-for-land-use-classification-f57f20b7294b.

[5] Google Earth Engine Homepage. URL: https://earthengine.google.com/.

[6] Earth Observing System Homepage. URL: https://eos.com/.

[7] Sentinel Hub Homepage. URL: www.sentinel-hub.com/.

[8] J. Shah, D. Dubaria, Building modern clouds: Using docker, kubernetes google cloud platform. 2019 IEEE 9th Annu. Comput. Commun. Work. Conf. CCWC (2019) doi: 10.1109/CCWC.2019.8666479.

[9] D. Bernstein, Containers and cloud: From LXC to docker to kubernetes. IEEE Cloud Comput (2014) doi:10.1109/MCC.2014.51.

[10] A. Poniszewska-Marańda, E. Czechowska, Kubernetes cluster for automating software production environment. Sensors, 21(5):1910 (2021) doi: 10.3390/s21051910.