UNIVERSITY OF ICELAND

Scalable Deep Learning for Remote Sensing with High Performance Computing

Rocco Sedona

May 2023

Ph.D. thesis
in Computational Engineering

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Scalable Deep Learning for Remote Sensing with High Performance Computing

Rocco Sedona

Philosophiae Doctor **degree in Computational Engineering**

Supervisor
Prof. Morris Riedel

Ph.D. Committee
Prof. Morris Riedel
Prof. Gabriele Cavallaro
Prof. Matthias Book

Opponents
Prof. Steven C. Riesing
Prof. Sergio Bernabé

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, May 2023

Scalable Deep Learning for Remote Sensing with High Performance Computing

180 ECTS thesis submitted in partial fulfillment of a 210 ECTS Ph.D. degree in Computational Engineering

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Dunhagi 5
107, Reykjavik Iceland

Telephone: +354 525 4700

Bibliographic information:
Rocco Sedona (2023) *Scalable Deep Learning for Remote Sensing with High Performance Computing*, Ph.D. thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland.

ISBN 978-9935-9697-8-1

Printing: Háskólaprent, Fálkagata 2, 107 Reykjavík

Reykjavik, Iceland, May 2023

To the Glory of the Grand Architect of the Universe

Abstract

Advances in remote sensing (RS) missions in recent decades have greatly increased the volume of data that is continually acquired and made available to end users, who can utilize it in a variety of Earth observation (EO) applications. land cover (LC) maps play a key role in monitoring the Earth's surface, providing scientists and policymakers with an accurate view of the evolution of the landscape and helping them address pressing questions, from efficient resource planning to resilience to climate change. Due to the use of classical machine learning (ML) and more recently of deep learning (DL) methods, the information content of RS data can be exploited to an unprecedented degree, fostering research, development, and deployment of workloads to address open challenges for EO applications, including LC classification. However, the larger size of the datasets needed to train state-of-the-art (SotA) DL models and the need to utilize them at scale increases the time to deployment, which can hinder their effective utilization. Adopting strategies for distributed deep learning (DDL) on high performance computing (HPC) systems provides the opportunity to speed up the training of the models, allowing faster development times for researchers. Since space agencies operate a variety of missions, data acquired by different sensors can be used to increase the temporal resolution at which a certain area is observed, with potential improvements in the accuracy of the ML/DL models. The thesis objectives are formulated with these premises in mind and were investigated using a combination of methodologies to exploit the dedicated resources of HPC systems, contributing to addressing new questions on the adoption of DDL methods for EO applications and to familiarize the RS community with such approaches, which can be of great value.

Keywords— High Performance Computing, Remote Sensing, Distributed Deep Learning, Land Cover Classification

Útdráttur

Framför fjarkönnunar á síðustu áratugum hefur leitt til mikillar aukningar í stöðugri öflun gagna sem gerð eru aðgengileg til enda notenda sem geta hagnýtt þau í forritum fyrir skoðun jarðarinnar. Landþekjukort spila lykil hlutverk í eftirliti með yfirborði jarðar. Þau veita vísindafólki og stefnumótendum skýra sýn á þróun í landslaginu og hjálpar þeim að finna svör við aðkallandi vandamálum allt frá skilvirkri skipulagningu auðlinda að þoli jarðar gegn loflagsbreytingum. Vegna notkunnar klassískra aðferða í vélnámi og nýlegrar notkunnar djúptauganetsaðferða er hægt að nýta efni úr fjarkönnun að umfangi sme að ekki hefur sést áður. Það ýtir undir rannsóknir, þróun og dreifingu á nýjum kerfum sem að leita til þess að leysa núverandi áskoranir á sviði EO forrita eins og LC flokkunar. Aukin stærð gagnasafna sem þarf til þess að þjálfa nýjustu SotA DL módel og þörfin til þess að nota þau á stórum skala hefur hins vegar orsakast í auknum tíma þróunar sem að getur komið í veg fyrir skilvirka nýtingu þeirra. Innleiðing stefnu fyrir dreifðan djúplærdóm á ofurtölvukerfum gefur tækifæri til þess að hraða þjálfun módelanna og rannsakendum að auuka hraða í þróun og minnka tíma þangað til virði er fengið úr ferlinu. Þar sem að geimferðastofnanir framkvæma mismunandi verkefni þá er hægt að nota gögnin sem safnað er til þess að auka “temporal resolution” sem að svæði eru skoðuð með, einnig með mögulegum bætingu á nákvæmni í vélnáms og djúplærdóms módelum. Markmið ritgerðarinnar eru gerð út frá þessum forsendum og voru rannsökuð með blöndu af aðferðum til þess að nýta auðlindir í ofurtölvukerfum, leggja af mörkum að svara spurningum um nýtingu djúplærdómsaðferða fyrir EO hugbúnað og til þess að kynna fjarkönnunarsamfélaginu fyrir aðferðum líkt og þessum sem að geta skapað mikið verðmæti.

Lykilorð— High Performance Computing, Remote Sensing, Distributed Deep Learning, Land Cover Classification

Contents

List of Figures

List of Tables

Abbreviations

ANN	artificial neural network
API	application programming interface
ARD	analysis ready data
AoI	area of interest
BPMN	Business Process Modeling Notation
CNN	convolutional neural network
CPU	central processing unit
CSDB	Cloud Scenario Database
CUDA	Compute Unified Device Architecture
DL	deep learning
DDL	distributed deep learning
DTW	dynamic time warping
EM	electromagnetic
EO	Earth observation
ESA	European Space Agency
FLOPS	floating point operations per second
GAN	generative adversarial network
GPT	Generative Pre-trained Transformer
GPU	graphics processing unit

HDF Hierarchical Data Format

HLS Harmonized Landsat Sentinel-2

HMM hidden Markov models

HPC high performance computing

HR high resolution

I/O input/output

JSC Jülich Supercomputing Centre

JURECA Jülich Research on Exascale Cluster Architectures

JUWELS Jülich Wizard for European Leadership Science

L8 Landsat-8

LAMB Layer-wise Adaptive Moments Optimizer for Batch Training

LARS Layer-wise Adaptive Rate Scaling

LC land cover

LU land usage

LSTM long short-term memory

MIMD multiple instruction multiple data

MIPAS Michelson Interferometer for Passive Atmospheric Sounding

ML machine learning

MODIS Moderate Resolution Imaging Spectroradiometer

MPI Message-Passing Interface

MSI Multispectral Instrument

NASA National Aeronautics and Space Administration

NCCL NVIDIA Collective Communication Library

OLI Operational Land Imager

PCA principal component analysis

PSC polar stratospheric cloud

PM proposed method

PS parameter server

RBF radial basis function

RDMA remote direct memory access

RS remote sensing

RF random forest

RNN recurrent neural network

S2 Sentinel-2

SAR synthetic aperture radar

SGD stochastic gradient descent

SotA state-of-the-art

SRF standard random forest

STA standard transformer approach

SVM support vector machine

TS time series

TO thesis objective

TPU Tensor Processing Unit

UAV unmanned aerial vehicle

UoI University of Iceland

List of Publications

The following publications are included as Appendix, and their summary and explanation can be found in Chapter 4.

PAPER I

R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. A. Benediktsson, "Remote Sensing Big Data Classification with High Performance Distributed Deep Learning", Remote Sensing (MDPI), vol. 11, no. 24: 3056, 2019, https://doi.org/10.3390/rs11243056.

PAPER II

R. Sedona, L. Hoffmann, R. Spang, G. Cavallaro, S. Griessbach, M. Höpfner, M. Book, and M. Riedel, "Exploration of Machine Learning Methods for the Classification of Infrared Limb Spectra of Polar Stratospheric Clouds", Atmospheric Measurement Techniques (Copernicus), vol. 13, no. 7, pp. 3661–3682, 2020. https://doi.org/10.5194/amt-13-3661-2020.

PAPER III

R. Sedona, G. Cavallaro, M. Riedel and M. Book, "Enhancing Large Batch Size Training of Deep Models for Remote Sensing Applications", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1583-1586, 2021, https://doi.org/10.1109/IGARSS47720.2021.9555136.

PAPER IV

R. Sedona, C. Paris, G. Cavallaro, L. Bruzzone and M. Riedel, "A High-Performance Multispectral Adaptation GAN for Harmonizing Dense Time Series of Landsat-8 and Sentinel-2 Images", in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10134-10146, 2021, https://doi.org/10.1109/JSTARS.2021.3115604.

PAPER V

R. Sedona, C. Paris, L. Tian, M. Riedel and G. Cavallaro, "An Automatic Approach for the Production of a Time Series of Consistent Land-Cover Maps Based on Long-Short Term Memory", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 203-206, 2022, https://doi.org/10.1109/IGARSS46834.2022.9883655.

PAPER VI

R. Sedona, C. Paris, J. Ebert, M. Riedel, G. Cavallaro, "Toward the Production of Spatiotemporally Consistent Annual Land Cover Maps using Sentinel-2 Time Series", IEEE Geoscience and Remote Sensing Letters vol. 20, pp. 1-5, 2023, https://doi.org/10.1109/LGRS.2023.3329428.

Additional Papers

These additional publications, which do not directly address the TOs or to which the author has contributed to a lesser extent, are not included in the thesis.

R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and M. Book, "Scaling Up a Multispectral Resnet-50 to 128 GPUs", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1058-1061, 2020, https://doi.org/10.1109/IGARSS39084.2020.9324237.

D. Coquelin, R. Sedona, M. Riedel and M. Götz, "Evolutionary Optimization of Neural Architectures in Remote Sensing Classification Problems", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1587-1590, 2021, https://doi.org/10.1109/IGARSS47720.2021.9554309.

S. Kesselheim, A. Herten, K. Krajsek, J. Ebert, J. Jitsev, M. Cherti, M. Langguth, B. Gong, S. Stadtler, A. Mozaffari, G. Cavallaro, R. Sedona, A. Schug, A. Strube, R. Kamath, M. G. Schultz, M. Riedel and T. Lippert, "JUWELS Booster - A Supercomputer for Large-Scale AI Research", 2021, Part of the Lecture Notes in Computer Science book series, vol. 12761, https://doi.org/10.1007/978-3-030-90539-2_31.

M. Aach, R. Sedona, A. Lintermann, G. Cavallaro, H. Neukirchen, and M. Riedel, "Accelerating Hyperparameter Tuning of a Deep Learning Model for Remote Sensing Image Classification", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 263-266, 2022, https://doi.org/10.1109/IGARSS46834.2022.9883257.

Acknowledgments

I would like to begin by expressing my sincere gratitude to Prof. Morris Riedel, who provided mentorship during my Ph.D. journey at the University of Iceland. Not only did he offer guidance, but he also introduced me to the island's natural and human beauty.

I am particularly indebted to Prof. Gabriele Cavallaro for his exceptional guidance, encouragement, and invaluable insights throughout my program. Additionally, Prof. Matthias Book's feedback, especially early in my program, was very helpful.

Prof. Claudia Paris provided me with irreplaceable scientific advice and support, and inspiring humanity, for which I am grateful.

If my passion for Machine Learning and Deep Learning has been ignited, I owe a great deal to Prof. Farid Melgani. His inspirational lectures have influenced countless students, including myself.

Thanks to Prof. Helmut Neukirchen, I have gained a wealth of knowledge, not only in the specific topics covered in his lectures but also in the methods he employs. His teachings have been highly informative and impactful, and I aspire to model my own teaching activities after his exemplary approach.

The Jülich Supercomputing Centre's colleagues provided me with constant stimulation, motivation, and support throughout my research. Special thanks go to Shadi, Jan and Surbhi. I would like to express my sincere gratitude to Dr. Lars Hoffmann for his invaluable guidance and support at the beginning of my PhD. His expertise and advice have been of immense help in shaping my research direction.

Francesca's constant support and encouragement have been instrumental in my success, providing a constant source of inspiration.

My friends, both near and far, have been a constant source of encouragement, support, and understanding. Specifically, Gianmarco, Teo, Andrea, Lollo, Tommy, Edo, Elenita, Mattia, Ila, Gianluca, Giorgio, Dilly, Alby, Ben, comrades Natasha, Lyudmila, Irina and Robi, have supported me through challenging times. My friend Thorstein was always there for me during my stay in Iceland, proving that friendship knows no geographical distance.

My mother, Annachiara, and sister, Carlotta, have been unwavering in their love and

encouragement throughout my Ph.D. journey.

The enduring inspiration and guidance of my late father, Patrizio, and uncle, Bruno, have been a constant source of strength.

I am grateful to my friend, Bro ∴ Enrico, whose constant support during some of the most challenging years of my life has been invaluable.

I am deeply grateful to Dr. Enrico Conte for his compassionate and insightful guidance in helping me embark on a transformative journey of self-discovery. With his expertise and empathetic approach, he has played an active role in helping me explore the depths of my inner self and uncover my true potential.

Finally, I express my gratitude to the brethren of the Star of Saxony Lodge Nr 853 within the Grand Lodge of British Freemasons in Germany for their acceptance and warmth. Their support has played an important part in my moral journey towards becoming a better man and contributing effectively to my human community.

1. Introduction

1.1. Motivation

In recent decades remote sensing (RS) missions for Earth observation (EO) have been providing increasingly large amounts of multi-source data (e.g., optical, radar, lidar), fostering the development of applications in a variety of fields, including monitoring the evolution of land cover (LC), damage assessment during and after natural disasters and providing insights valuable to determine the impact of climate change [62]. Scientists and policymakers rely heavily on the analysis carried out on such data to study and address pressing issues faced by humankind [61]. Space agencies operate missions that utilize a variety of satellites, such as National Aeronautics and Space Administration (NASA)'s Earth Observing System project [1] and Copernicus, a program of the European Union [2], to acquire and provide information on Earth's land surfaces, oceans, and atmosphere to the users. With over 155 PB of data downloaded since the start of operations and an average volume of 9.6 TB published daily, the Sentinel-2 (S2) mission continually provides newly acquired imagery of the Earth's surface. ML has long been used for RS applications, and since an ever-increasing number of missions for EO are becoming operational, the field of RS is not excluded from the DL "gold rush", benefiting greatly from such methods [115]. In recent years, DL has provided a means to advance many applications, relying on greater availability of data and graphics processing units (GPUs), which have been shown to speed up the training of such models by a factor of 50 compared to central processing units (CPUs) [17, 86]. DL thrives on large datasets, learning complex representations by optimizing the parameters of each layer. Researchers utilize DL models for image fusion and registration, object and change detection, LC classification, and segmentation [61]. However, large DL models also require large datasets. Consequently, the training and deployment of these models call for utilizing systems with dedicated hardware and software. High performance computing (HPC) systems provide dedicated and highly optimized resources necessary for large-scale training and deployment of DL models. Through a community effort by key players in this field, benchmarking of DL models is carried out regularly to quantify the advances of the deployment of such algorithms on HPC systems [26]. Despite the progress in these fields, several challenges still exist for the science community. Although additional RS datasets are becoming available, addressing important requirements in real EO applications is still challenging. These requirements include (i) the ability to exploit global-scale information while focusing on specific areas

[1] `https://eospso.nasa.gov/content/nasas-earth-observing-system-project-science-office`
[2] `https://scihub.copernicus.eu/twiki/do/view/SciHubWebPortal/AnnualReport2021`

of interest (AsoI), (ii) the use of data at multiple spatial resolutions acquired at different times from missions with heterogeneous sensors to (iii) enable cutting-edge research using continual streams of large volumes of data. To meet these requirements, a framework must be designed with flexibility in mind, allowing it to be adapted over time. Modularity plays a crucial role in this since new technologies must be continually integrated to provide access to a multitude of data sources and enhance performance. For example, engineering efforts can be made to optimize the utilization of available datasets on HPC machines by using file formats suitable for parallel input/output (I/O) without disrupting the functioning of subsequent data analysis modules in the pipeline. This will allow researchers to conduct studies on the fusion of multi-source data. Besides technical challenges posed by retrieving large datasets and their utilization in an HPC environment, scaling the training of DL models also requires a careful selection of training modalities and optimization algorithms. The two main approaches to speed up the training of the models in a DDL settings are data parallelism and model parallelism. While the former trains a copy of the model on each device feeding different chunks of the dataset, in the latter approach, the model is split among the devices [9]. Due to its more straightforward implementation, until recently, data parallelism libraries were more available (e.g., Horovod [3], Tensorflow MirroredStrategy [4] and PyTorch DistributedDataParallel [5]) than model parallelism ones. However, the landscape is rapidly changing, e.g., with the application programming interface (API) provided by DeepSpeed [6], that aims to increase access to training of large DL architecture through a combination of model and data parallelism, pipelining and CPU offloading. Developers have been publishing code to enable access to DDL. Still, strategies to compensate for the degradation of the models trained using large batch sizes must be adopted to achieve near-linear scalability [37]. The difficulty of training using large batch sizes is a known issue in the literature. It has been empirically observed in several studies [43, 64, 89, 113] as well as analyzed from a theoretical point of view [47, 108], and result in poor generalization capabilities that hinder the performance of the DL models at deployment. Not only can traditional ML methods utilized on RS data [69] in some specific settings still compete with advanced DL models [39], but they can also provide means of analyzing and visualizing data in a human-friendly way [59, 16]. Space agencies run concurrent missions that can share overall objectives, operating satellites equipped with sensors that are similar to some degree. As a concrete example, S2 and Landsat-8 (L8) both carry on board multi-spectral instruments for the observation of our planet. Still, the spectral and spatial characteristics of the instruments are not entirely identical. Commercial operators are now playing a significant role, offering products at high temporal and spatial resolution, such as the Planet Fusion products made available by Planet[7], emphasizing the importance of integrating multi-source information. It can be seen that various elements come into play and need to be combined: (i) the retrieval and extraction of informative content from large amounts of data, (ii) the DL libraries for scaling training, to reduce the time to convergence of the models as well as reduce the time to deployment at inference, and (iii) heterogeneous HPC systems with dedicated

[3] https://horovod.ai/

[4] https://www.tensorflow.org/api_docs/python/tf/distribute/MirroredStrategy

[5] https://pytorch.org/docs/stable/nn.html#torch.nn.parallel.DistributedDataParallel

[6] https://www.deepspeed.ai/

[7] https://assets.planet.com/docs/Planet_fusion_specification_March_2021.pdf

devices and environments. Such a complex set of hardware and software needs to operate in unison to fully realize the performance enhancement of DL models trained on large data sets in an HPC environment for addressing classification tasks in the EO domain.

1.2. Thesis Objectives

Considering the critical challenges introduced in Sect. 1.1, it follows that there is an urgent need for additional investigations on the adoption of DL at scale in the RS community. A number of questions naturally arise from the overview provided in Sect. 1.1, namely: *(i) is it possible to capitalize on the potential provided by the exploitation of available larger RS datasets, and if so, how? (ii) Can the usage of dedicated resources on HPC systems enable training of DL models for EO applications at scale? (iii) How can results obtained using such methods be validated?*

The questions above inspired the formulation of the main **thesis objectives (TOs)** that guided the realization of the research project:

- **TO 1**: Achieve near-linear scalability for DL models applied to RS data on HPC systems, utilizing distributed deep learning (DDL) frameworks.

- **TO 2**: Extract valuable information from complex RS datasets by utilizing scalable and parallel methods on HPC systems, with the objective of maximizing efficiency in large-scale EO applications.

- **TO 3**: Demonstrate the effectiveness of classical ML methods in addressing challenging applications in RS, and to provide examples of how classical data science methods can improve performance evaluation by providing validation and clear visualization of results.

- **TO 4**: Harness multi-source data to densify time series (TS) of RS measurements through harmonization and gain a deeper understanding of complex relationships among sources for more accurate insights.

To a large extent, **TO 1**, **TO 2** and **TO 3** were addressed in parallel since the need for utilization of DL models requires the exploitation of large datasets. Since the efficient training of such models is computationally demanding, dedicated HPC resources were also used, utilizing tools inherited from classical ML to validate the results. **TO 4** is part of an additional effort undertaken to increase the amount of informative data to be fed into ML/DL models. An abstraction of the methodological approach is depicted in Fig. 1.1 following the Business Process Modeling Notation (BPMN) notation [29], which clearly shows how the research activities to address the TOs were carried out. BPMN is a graphical representation technique that illustrates a complete sequence of planned business activities from start to finish. It plays a crucial role in Business Process Management by

visually depicting the various steps and information flows required to accomplish a process. The bottom layer, i.e., the foundation on which the work of the thesis is based, entails the set of technologies used as building blocks to develop scalable EO applications. To fully extract the information from the large volume of data acquired by RS missions (**TO 2**), the second layer relies on the underlying solutions from the bottom layer. Training DL models on benchmarking datasets such as BigEarthNet [95, 94] utilizing DDL algorithms on HPC systems was used to accelerate the analysis of various experimental set-ups for a variety of tasks in a shorter time. On top of the second layer, the third layer covers the activities to tackle the **TO 3**. Classical ML approaches validated the experimental results, providing a better understanding of the outcomes. The fourth layer concerns the research activities that utilize multi-source data to densify the TS of RS data. These activities were not performed throughout the entire project of the thesis, differently from those related to the first three layers. The fourth layer was thought of as a supplement to the existing applications.

1.3. Outline

This thesis was composed in a cumulative style. The significant findings are therefore presented in the form of peer-reviewed conference and journal publications and pending submissions, which are in Appendix A. Publications to which the thesis author only contributed to a lesser extent or are not directly addressing the TOs are deliberately excluded. For a complete list of all publications, however, please refer to both the **List of Publications** and **Additional Papers**. Appendix B lists the open-source code repositories the author contributed to during the thesis project.

1.3.1. Covering Paper

Chapter 1 — Introduction begins with Sect. 1.1 describes the motivation for this research project. The formulation of the TOs is presented in Sect. 1.2.

Chapter 2 — Background intends to familiarize the reader with basic concepts of RS, DL and HPC.

Chapter 3 — Related Work provides a review of SotA methods related to the work performed for this research project.

Chapter 4 — Summary of Papers and Contributions summarizes and explains the main contributions of the papers included in **Appendix A**.

Chapter 5 — Conclusions summarizes the thesis with final considerations on the work carried out during the project and presents perspectives for future research opportunities.

1.3.2. Appended Papers

PAPER I

R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. A. Benediktsson, "Remote Sensing Big Data Classification with High Performance Distributed Deep Learning", Remote Sensing (MDPI), vol. 11, no. 24: 3056, 2019, https://doi.org/10.3390/rs11243056.

PAPER II

R. Sedona, L. Hoffmann, R. Spang, G. Cavallaro, S. Griessbach, M. Höpfner, M. Book, and M. Riedel, "Exploration of Machine Learning Methods for the Classification of Infrared Limb Spectra of Polar Stratospheric Clouds", Atmospheric Measurement Techniques (Copernicus), vol. 13, no. 7, pp. 3661–3682, 2020. https://doi.org/10.5194/amt-13-3661-2020.

PAPER III

R. Sedona, G. Cavallaro, M. Riedel and M. Book, "Enhancing Large Batch Size Training of Deep Models for Remote Sensing Applications", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1583-1586, 2021, https://doi.org/10.1109/IGARSS47720.2021.9555136.

PAPER IV

R. Sedona, C. Paris, G. Cavallaro, L. Bruzzone and M. Riedel, "A High-Performance Multispectral Adaptation GAN for Harmonizing Dense Time Series of Landsat-8 and Sentinel-2 Images", in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10134-10146, 2021, https://doi.org/10.1109/JSTARS.2021.3115604.

PAPER V

R. Sedona, C. Paris, L. Tian, M. Riedel and G. Cavallaro, "An Automatic Approach for the Production of a Time Series of Consistent Land-Cover Maps Based on Long-Short Term Memory", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 203-206, 2022, https://doi.org/10.1109/IGARSS46834.2022.9883655.

PAPER VI

R. Sedona, C. Paris, J. Ebert, M. Riedel, G. Cavallaro, "Toward the Production of Spatiotemporally Consistent Annual Land Cover Maps using Sentinel-2 Time Series", IEEE Geoscience and Remote Sensing Letters vol. 20, pp. 1-5, 2023, https://doi.org/10.1

109/LGRS.2023.3329428.

1.4. Contribution

The primary contributions of this thesis are presented in this section. Table 1.1 provides an overview of the relationship between the TOs and papers (using the vocabulary presented in the List of Publications). Three of the publications, namely Paper I, V, and VI, present various use cases for LC classification. While they utilize different DL models, they have in common the adoption of data parallelism techniques for speeding up the training. In Paper I, the dataset for RS BigEarthNet [94] was used to reduce the training of a ResNet50 convolutional neural network (CNN) from approximately 38 hours using 1 GPUs to roughly 25 minutes using up to 96 GPUs. This was accomplished through efforts by the author to investigate available solutions and design an experimental set-up. After the selection of suitable strategies and their combination to address the problem of large global batch size, which causes increased model instability with severe consequences for model convergence, the author also contributed to the deployment of this approach and its benchmarking on HPC systems, achieving near linear scalability. For Paper V and Paper VI, an entire modular framework [73] for data query, download, and pre-processing was ported to HPC systems. This enabled flexible utilization of TS acquired by the S2 constellation. The first contribution of the author in Paper V was the implementation of a method based on random forest (RF) to retrieve reliable multi-temporal training sets, which were fed into the DL models. long short-term memory (LSTM) and Transformers were adopted to exploit the temporal dynamics of the input signal to enhance the consistency of the output multi-year LC maps. Additionally, in Paper VI, the author implemented modifications to the baseline Transformer model to retrieve an output with information at a finer temporal resolution (i.e., updating the LC map at each new observation), which was then analyzed and aggregated in post-processing to provide the user with a more informative LC map. In Paper III, the beneficial effects of utilizing a more advanced optimizer, developed explicitly for large batch sizes, were demonstrated on a RS dataset. The effort of the author in conceiving an experimental set-up with large batch size training for EO applications was beneficial to establish advantages and limitations of standard and specific optimization schema and laid the groundwork for further studies on hyperparameter tuning and model optimization in a scalable setting (not within the scope of this thesis but part of the Additional Papers). In Paper II, the author proposed the utilization of ML methods in the case study of the classification of limb spectra acquired by the Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) for the detection and prediction of polar stratospheric cloud (PSC) types. The author also investigated and suggested adopting tools for visualization and validation of the obtained results, which were used to demonstrate that the proposed methodology is physically sound, consistent with domain experts' knowledge, and can provide additional information as compared to SotA approaches. Finally, in Paper IV, the author adapted a generative adversarial network (GAN), which was used to successfully densify a TS of RS data and increase the accuracy in an LC classification task. Based on the expertise

paper / TO	Paper I	Paper II	Paper III	Paper IV	Paper V	Paper VI
TO 1	x		x	x	x	x
TO 2	x	x		x	x	x
TO 3	x	x	x	x	x	x
TO 4				x		

Table 1.1: Relationship between TOs and papers.

acquired in previous works, the author contributed to parallelizing the training of the DL model. For a more detailed discussion of the contributions of each publication, please refer to Chapter 4.

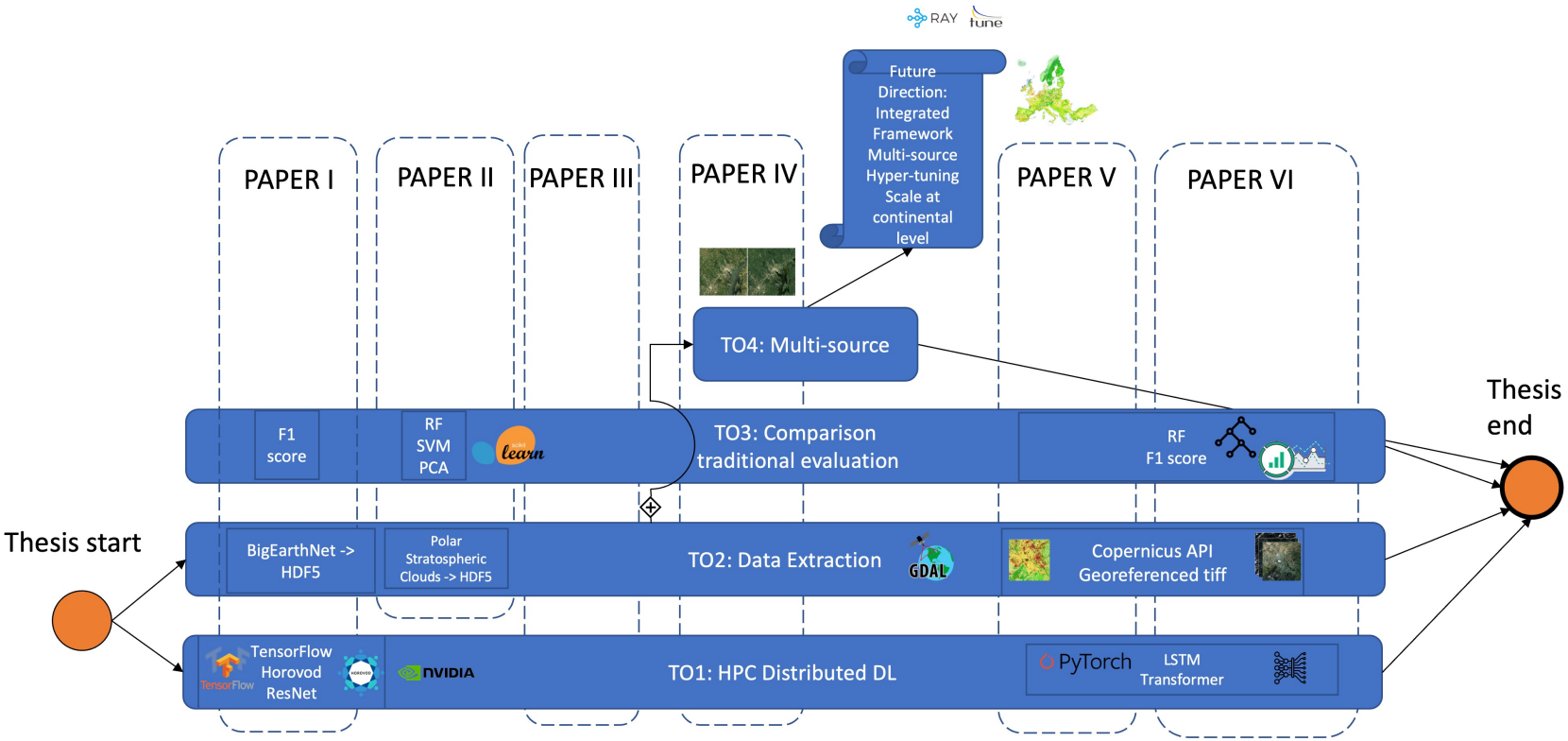


Figure 1.1: Methodological approach followed from the beginning to the end of the thesis using the BPMN notation.

2. Background

2.1. Remote Sensing

RS is the science of collecting information about an object at a distance [72]. Sensors used for EO applications measure the radiation that is backscattered by the object under analysis. The sensor is a device that, through the measurement of electromagnetic (EM) energy, can provide information on the features of the studied object. Sensors used for EO applications are often mounted on board satellites or aircraft. A top-level categorization of RS sensors can be made between passive and active ones. Passive sensors detect radiation emitted or reflected by an object under study. For example, the passive sensor operating at visible and infrared wavelengths measures the sunlight reflected by the Earth's surface, as illustrated in Fig. 2.1. Planck's radiation law describes the EM radiation emitted by physical bodies per wavelength depending on the body's temperature. An example of a sensor that measures EM energy emitted by the body under study is the passive microwave radiometer, which measures emitted radiation of the Earth's surface and is used in applications such as soil moisture monitoring and snow and ice detection [98]. On the other hand, active sensors utilize an active stimulus to illuminate the object and measure the amount of backscattered energy. An example of an active sensor is a laser altimeter, a device that emits a laser pulse toward the Earth's surface and computes the time it takes for the beam to reflect from the surface and return to the sensor [87]. The performance of passive sensors operating at visible wavelengths is strongly affected by the weather and illumination conditions. Passive sensors operating in the infrared and microwave spectra can provide additional information for a variety of applications, e.g, the estimation of snow depth, observation of the sea surface, and rainfall retrieval [13, 31, 92]. Active sensors such as laser and radar devices are widely used for EO applications. While some active sensors, such as synthetic aperture radars (SARs), measure the magnitude and phase change of EM radiation, others, such as a laser rangefinder or radar altimeter, only measure elapsed time between sending and receiving energy signals [98, 28, 27]. Reflectance of the EM energy is usually measured not only at a single wavelength but over a spectral range, or bandwidth. Spectral range or wavelength range is an interval of the EM spectrum used to measure average reflectance. Multispectral scanners, such as the Multispectral Instrument (MSI) mounted on-board satellites of the Sentinel-2 constellation, measure the reflectance at multiple wavelengths, while hyperspectral sensors can acquire up to hundreds of narrow bands [10].

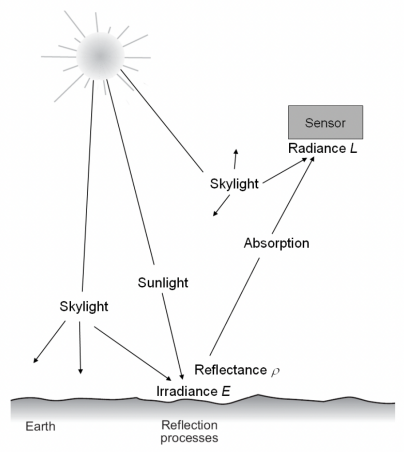


Figure 2.1: Illustration of visible and infrared passive RS sensor scheme [98].

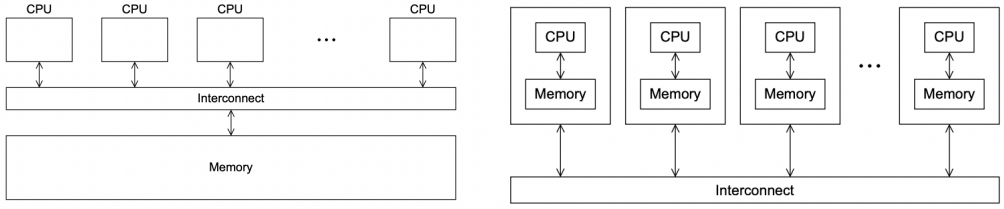
2.2. High Performance Computing

In this section, a background on HPC is provided to introduce the main technological aspects of supercomputers in Subsection 2.2.1 and to explain the distinction between shared and distributed memory systems in Subsection 2.2.2.

2.2.1. High Performance Computing Systems

In the modern HPC, supercomputers are based on a multitude of hardware designs, architectures, and platforms [75]. Heterogeneous systems equipped with a variety of specialized hardware are becoming more common. Partitions of an HPC system can be dedicated to specific tasks, as in the case of GPUs for DL applications [19]. The powerful resources provided by an HPC system and the advances in parallel computing allow a faster time-to-solution for highly complex problems in a wide range of fields, from computational fluid dynamics to health care. The power of HPC systems is typically measured in floating point operations per second (FLOPS). We are now entering the era of exascale as the first supercomputer Frontier set the new record at 1,102.00 PetaFLOPS in November 2022 [1]. Nowadays, not only specialized research centers provide HPC systems, but more and more are hosted on the cloud. Parallel computing executes multiple tasks concurrently on various servers or computer processors. Computation can then be carried out in parallel on dedicated resources, using multi- and many-core processors. The core elements of any HPC system are called nodes, which are computers equipped with high-performance multi-core processors or accelerators, depending on the intended application. Dedicated high-throughput, low-latency storage and memory devices enable high I/O performances. A key role is played by high-performance remote direct memory access (RDMA) technologies, such as Infiniband [44], that allow the interconnection of multiple nodes at high-speed [106].

[1] `https://www.top500.org/lists/top500/2022/11/`



(a) *(b)*

Figure 2.2: Representation of (a) shared memory and (b) distributed memory systems [71].

2.2.2. Shared and Distributed Systems

multiple instruction multiple data (MIMD) computers, in which the instructions can operate on different data, can be categorized into shared memory and distributed memory systems [6]. Shared memory systems can access only a single memory address space (shown in Fig. 2.2(a)), while the memory space of distributed memory systems are distributed among multiple nodes (shown in Fig. 2.2(b)). In a distributed memory system, the processor operating on one node cannot directly access the memory address of a processor on another node. Therefore, an exchange of messages between the two nodes is required. Since the rapid exchange of messages is of utmost importance in HPC clusters, the availability of high-throughput technology for the interconnection of the nodes is essential.

The most frequently used APIs for parallel programming on shared memory systems are Pthreads, which is low-level and allows fine-grained control over thread management, and OpenMP, which allows higher-level programming [71]. Message-Passing Interface (MPI) is a standard that defines specifications for exchanging messages among processors on distributed memory systems [66, 71]. MPI allows the programmer to implement a variety of topologies, among which is the ring, a concept upon which the Ring-AllReduce algorithm is based (see also Subsection 3.2.5).

2.2.3. Hierarchical Data Format

Hierarchical Data Format (HDF) is a collection of file formats intended to manage and store large data sets [99]. An HDF5 information set (infoset) serves as a container for annotated associations of array variables, groups, and types [30]. HDF datasets are array variables that hold data elements arranged logically as a multidimensional array. An HDF dataspace records the dataset's rank (number of dimensions) and current and maximum extent (number of elements) in the corresponding dimensions. An HDF array database lays out array elements as a single sequence. In contrast, for small HDF datasets (totaling less than 64 KB), all array elements are stored in the array variable's metadata or header (HDF datatype, dataspace, and other metadata) for easy retrieval. Adopting a storage particular layout strategy provides additional capabilities and potentially increases

performance for certain operations. For example, an HDF dataset with a contiguous layout strategy guarantees nearly constant access time to any element in the array, with zero overhead for locating elements in the dataset. For HDF datasets using a chunked layout strategy, HDF enables unrestricted extents for none, some, or all dimensions. HDF groups are similar to directories in a file system. An HDF group represents an explicit connection among zero or more HDF information items, such as HDF datasets, HDF groups, and HDF datatype objects. HDF data types provide a type system that offers nearly unlimited flexibility. The HDF array variable type (non-scalar) has two primary components: an HDF dataspace describing its shape and an HDF datatype describing the type of its data elements. Currently, ten families or categories of HDF datatypes are supported: integer, floating-point, string, bit field, opaque, compound, reference, enum, variable-length sequence, and array. Mostly, the type instances of these families are as their name implies. Several APIs exist to utilize HDFs with a variety of programming languages. h5py is a Python library that enables storing and manipulating very large amounts of data, which is useful in applications such as DL [20].

2.3. Deep Learning

DL is a subset of machine learning based on artificial neural networks (ANNs) with multiple layers that can model complex patterns based on the training data [55]. Inspired by the functional behavior of the human brain [65], ANNs is composed of stacks of layers. DL models consist of an input layer, an output layer, and at least one hidden layer in between. The interconnections among the neurons are associated with weights, and each neuron has an activation function.

An objective (or loss) function chosen depending on the task (e.g., the distance between predicted and real targets in a supervised classification setting) measures the discrepancy between the generated output and the desired target. Optimizing the DL model is performed by computing the gradient of the objective function with the backpropagation algorithm. Using the chain rule, the gradients of the loss based on the weights are calculated backward from the top of the network to the bottom. The weights of the model are then updated according to the utilized algorithm [35].

CNNs are primarily used in computer vision and image classification applications to detect features and patterns in images and to enable tasks such as object recognition [51], while recurrent neural networks (RNNs) work with sequential or time-series data and are therefore commonly used in natural language and speech recognition applications.

3. Related Work

DL models require training using large amounts of data to learn meaningful features. Therefore, the DL models need dedicated pipelines for extracting and handling such data, which can severely impact their performances HPC provide dedicated hardware accelerators to deploy and scale-up processing workflows efficiently and significantly enhance their computational performance (in terms of FLOPS) [63]. HPC systems are on the verge of entering the new era of exascale computing in the coming years as the most powerful computers have already reached the threshold of ExaFLOPS. HPC systems can help address challenging problems in applications from climatology to astrophysics, medicine, and industry [109] through massively scalable algorithms. The increase in the amount of RS data requires higher storage capacities. In addition, open challenges remain, such as the fact that near real-time EO applications need to be deployed on parallelized clusters. In the following sections, the current research that happens at the intersection between advances in the RS research for LC classification and DDL on HPC systems is discussed.

3.1. Land Cover

LC maps describe the physical properties of the coverage of Earth's surface, in which the areas that share similar semantic meaning are grouped into classes. In contrast, land usage (LU) provides information on how the surface is used and which activities are performed on the ground. Therefore, LU maps strongly depend on the interaction between humans and the environment [52].

3.1.1. Land Cover Classification

The availability of updated LC classification maps is fundamental to a wide range of applications since it enables the study of the dynamics of phenomena that occur on the Earth's surface. Field surveys and in-situ observations have been long complemented by automated schemes that exploit satellite imagery [97]. Frameworks that process large volumes of RS data need dedicated resources. In [11], the Google Earth Engine was used to access and process Sentinel-2 data and train a CNN to update the LC maps near real-time. A crucial element to fully exploit LC maps and extract information on the changes on the ground is their spatio-temporal consistency, i.e., observed changes among different

Figure 3.1: LC map generated by WorldCover for the Reykjavik area[1].

LC maps should be attributed to real change only, not to spurious change. [104] proposed a method to map global LC classes using multiyear TS from Moderate Resolution Imaging Spectroradiometer (MODIS) data, including data from the previous to ensure consistency of the new product. In [2], a hidden Markov models (HMM) was used to generate TSs of LC maps while reducing the number of unreal changes in the output products. The trade-off between reducing the occurrence of spurious changes in inter-annual LC maps and detecting real change is of the most significant importance when analyzing long TS of data. The adoption of SotA DL methods can further help us to increase the reliability of the LC classification maps, detecting the coverage on the ground and modeling the distinct components of the signals in the TS of LC products, allowing monitoring of ongoing environmental processes (e.g., desertification, urbanization, deforestation).

3.2. Deep Learning

A significant part of the effort regarding DL methods in this thesis was devoted to investigating the exploitation of spatial and temporal information provided by satellite imagery. At first, the utilization of CNN models was studied for the patch-based classification of LC classes (details on ResNet in Subsection 3.2.1). Secondly, the utilization of models capable of exploiting the temporal information of long TS of RS data was investigated (presented in Subsection 3.2.2 and 3.2.3).

3.2.1. ResNet

The ResNet is a deep residual CNN architecture developed by [41] to overcome difficulties in training networks with a very large number of layers (>20, up to 1000 layers, and more possible). AlexNet, a CNN model with eight layers [51], VGG with 16 layers [91], and

[1] `https://viewer.esa-worldcover.org/worldcover/`

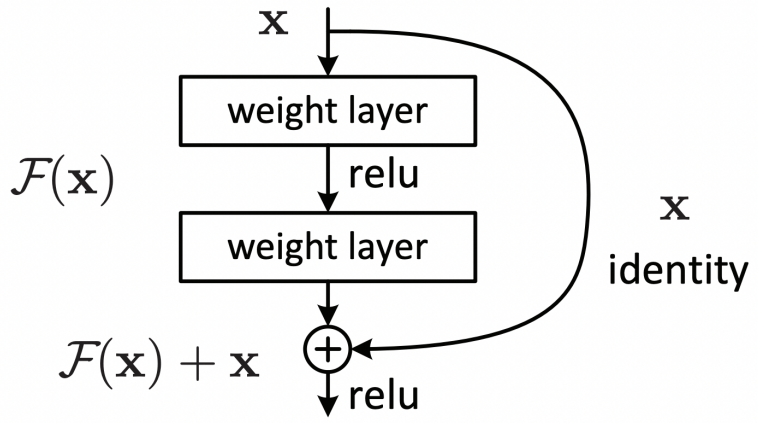


Figure 3.2: Depiction of skip connection [41].

GoogleNet with 22 layers [96] were the first DL models to consistently outperform traditional ML models for imaging tasks. An increasing number of processing layers further increased accuracy performance on ImageNet challenges regarding class recognition rates (the ImageNet-1k challenge has 1000 different object classes that need to be successfully learned during the training on 1.2 Million images [51, 22]). However, simply increasing the number of layers further by stacking more convolutional and other layers (pooling, etc) on top of each other was not functionally successful. The training of very deep networks resulted in worse accuracy, contrary to expectations set by previous results. It has been noted that the degradation of the training accuracy may be partly caused by a phenomenon known as vanishing (or exploding) gradients. ResNet architecture has been designed to overcome this issue by introducing residual blocks featuring skip connections. These connections implemented an explicit identity mapping for each successor layer in a deep network in addition to the learned operations that were applied to the input before it reached the next layer [41]. The network was forced to learn residual mappings corresponding to useful transformations and feature extraction on the image input. At the same time, loss gradients could still flow undisturbed during the backward pass via available skip connections through the whole depth of the network. Instead of directly fitting the underlying mapping $H(x)$, the residual mapping $F(x) := H(x) - x$ is learned [41]. In ResNet, the skip connections are implemented as identity mappings, resulting in the formulation $F(x) + x$ shown in Fig. 3.2. Various ResNet networks were shown to train successfully with several layers that were impossible to handle before while using a smaller number of parameters than previous, less deep architectures (e.g., VGG or Inception networks), allowing for faster training. ResNet-50 (where the number indicates the number of layers) has since established a strong baseline in terms of accuracy, representing a good trade-off between accuracy, depth, and number of parameters while being very suitable for parallelized, distributed training. Since it remains the strong baseline for object recognition tasks and is also widely used in scenarios for transfer learning ([23, 80, 50]), the ResNet-50 architecture was adopted for the work described in Sect. 4.1.

3.2.2. Long Short-Term Memory

LSTM are a type of RNN. The RNN dynamics can be described using deterministic transitions from previous to current hidden states:

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l$$

The central role of the LSTM model is held by the "cell state", a memory cell that maintains its state over time. The "cell state" is the horizontal line that passes through the top of the diagram below, and it can be visualized as a conveyor belt through which information remains unchanged [42].

Let $h_t^l \in \mathbb{R}^n$ be a hidden state in layer l in timestep t. Moreover, let $T_{n,m} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be an affine transform ($Wx + b$ for some W and b). Let $\odot$ be element-wise multiplication and h_t^0 be an input word vector at timestep k. The activations h_t^L are used to predict y_t, where L is the number of layers in the deep LSTM.

For classical RNNs, this function is given by:

$$h_t^l = f(T_{n,n}h_t^{l-1} + T_{n,n}h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \tanh\}$$

$$f_t = \sigma(W_f\dot{[}h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i\dot{[}h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\tilde{C}_t = \tanh(W_C\dot{[}h_{t-1}, x_t] + b_C) \tag{3.3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.4}$$

$$o_t = \sigma(W_o\dot{[}h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3.6}$$

In these equations, sigm and tanh are applied element-wise. Figure 3.3 illustrates the LSTM equations.

Lyu, Lu, and Mou [60]. LSTM networks trained on TS of S2 data outperform models trained on a single acquisition, i.e., a SVM and a CNN for LC classification tasks since a multi-temporal approach can learn the temporal pattern that is peculiar to LC classes [83].

3.2.3. Transformers

The attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, key, value, and output are all vectors. The output is calculated as the weighted sum of the values, where the weight assigned to each value is

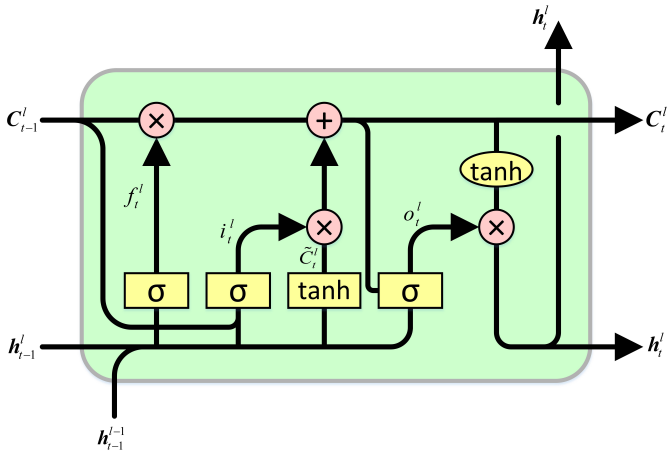


Figure 3.3: Depiction of cell of the LSTM adapted from [32].

calculated using the query's compatibility function with the corresponding key [102]. The Scaled Dot-Product Attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.7}$$

where $\sqrt{d_k}$ is the dimension of the key vector k and query vector q .

Multihead attention is defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{3.8}$$

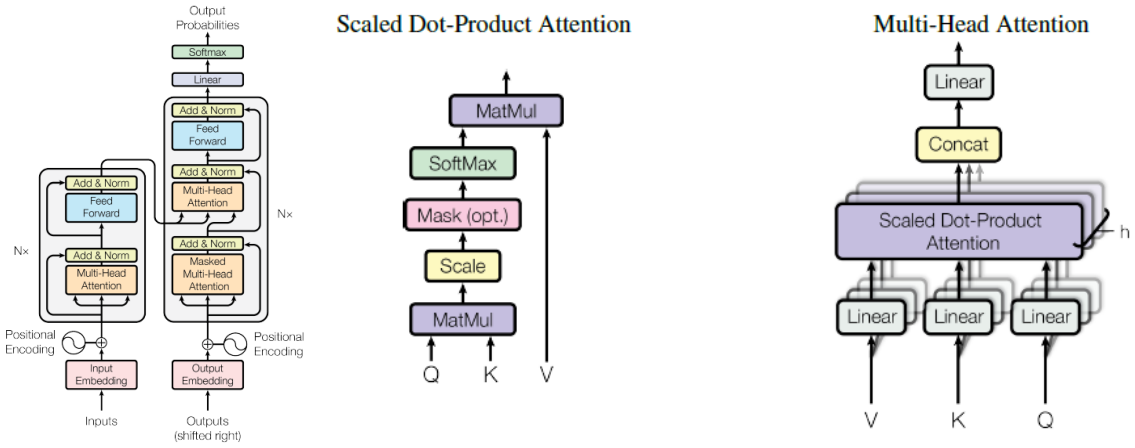
where:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3.9}$$

Rußwurm et al. [84] utilized a Transformer model alongside other methods, including RF and LSTM, for the supervised classification of field crops using TS of S2 data. In [14] a Transformer is used to refine the spatial features extracted by a CNN to enhance performances on change detection tasks. Visual Transformers [24], i.e., models that replace the convolutional layers of CNNs with attention layers, have also been adopted in the context of RS, matching performances of SotA methods based on CNNs [8].

3.2.4. Pix2pix

Pix2pix [81] is a conditional GAN, in that the two models that compose it, the generator and the discriminator, play one against the other. In the adversarial game, the generator tries to fool the discriminator, while the discriminator aims to maximize the probability of correctly detecting fake samples [36]. In a conditional GAN, additional information is fed



(a) *(b)*

Figure 3.4: (a) Transformer architecture, (b) attention mechanism [102].

into the generator to direct the data generation process [67]. The formula that describes Pix2pix is:

$$\min_G \max_D V(G,D) = \mathbb{E}_{X,Y}[\log D(X,Y)] + \\ + \mathbb{E}_{X,z}[\log(1 - D(X, G(X,z)))], \tag{3.10}$$

where $\mathbb{E}$ is the expected value, X and Y are the source and target images (having the same resolution), z is the input noise of the generator, and $V(G,D)$ is the value function. In particular, the generator G and the discriminator D of pix2pix are a U-net encoder-decoder architecture with skip connections and a PatchGAN, respectively. The first part contains several downsampling convolution layers in the U-net encoder-decoder generator [81]. The second part is a mirrored version of the first, with a transposed convolution for upsampling the data, which flows from the bottom to the top of the U-net through a bottleneck. The skip connections, which link the inner layers of the encoder and decoder, allow low-level information to pass directly from the first to the last layers of the U-net. The PatchGAN discriminator is designed to capture the patterns at the scale of the input image. Its objective is to classify $N \times N$ patches of $G(X,z)$ (the synthetic input patch created by the generator) and Y (the target patch) as fake or true, encouraging the generator to produce more accurate and realistic outputs. Pix2pix has been used in the domain adaptation tasks with RS data. Ao et al. [3] presented an approach to translate SAR imagery acquired by Sentinel-1 into the target domain of TerraSAR-X by adopting spatial Gram matrices to safeguard the structural information. In Lebedev et al. [54], an optimized version of pix2pix was designed to improve change detection compared to the original architecture.

3.2.5. Distributed Deep Learning

In recent years, the scale of distributed parallel training and the size of DL models have grown exponentially. The notorious Generative Pre-trained Transformer (GPT)-3 features 175 billion parameters and 96 layers of attention, using 499 billion tokens [12]. Using

frameworks to parallelize the training is essential for large DL models and even more crucial for the very large ones such as GPT-3 or DALL-E [79]. There are mainly two types of distributed parallel training: data parallelism and model parallelism [9]. Despite the increasing availability of dedicated efficient GPUs such as the NVidia A100 or the AMD MI250X, successfully training and deploying large DL models still poses several challenges. To overcome these limitations imposed by computationally expensive training, the DL community envisages a variety of methods that enable distributed training across multiple computing nodes of clusters or HPC machines equipped with accelerators like NVIDIA Collective Communication Library (NCCL) or highly specialized Tensor Processing Units (TPUs) [63, 111]. Using these methods, it became possible to perform distributed training of large network models without losing task performance and drastically reduce the time necessary for complete training. For example, the time required to fully train the object recognition DL model on ImageNet-1k (1.2 million images, about 100 epochs needed for convergence training) was reduced by several orders of magnitude, from days to minutes [109, 107]. Horovod is a popular software library that provides a convenient way to run training and supports TensorFlow and PyTorch [88]. By using Horovod, just a few modifications to the standard code used for rapid prototyping of a single-node model are needed to adapt it to a distributed execution across multiple nodes. To enable distributed training, Horovod adapts the parallel data schema. In data parallelism, it is assumed that a trained network model can fit in the memory of a GPU device. Multiple workers can be instantiated during training, each taking up one available GPU. Each worker contains a copy of the network that will be trained and receives a separate chunk of data for training. Each iteration updates the model, and the global batch is split into different local batches assigned to each worker. Working on its subset of the data, each worker performs a forward step to calculate the network activation and local loss based on its current input and a backward step to calculate the local gradient. To synchronize all network models among workers, Horovod uses a decentralized and synchronous update strategy based on Ring-AllReduce [88] operations, where the local gradients of all workers are collected and averaged. The global gradient computed in this way is used to update the weights of each local replica of the DL model. This contrasts centralized update strategies utilizing a parameter server (PS) to communicate model parameters to the workers. Machines hosted at Jülich Supercomputing Centre (JSC) do not support intercommunication among nodes using TCP/IP, which is required for the central PS. The distributed training approach employed by Horovod is known for its ability to provide high bandwidth and low latency communication among computing nodes. This is achieved through the use of MPI and NCCL libraries, which enable efficient interconnectivity among nodes. Decentralized updates better use the network topologies that connect the respective machines and use a more efficient communication strategy to perform distributed training. The decentralized approach can provide greater fault tolerance than strategies based on a central PS by not having a weak point in the communication chain, i.e., if the PS crashes, it is difficult to continue training. When a node fails, communication in a decentralized approach can continually be reconfigured without affecting training because all other workers have a complete copy of the model. Therefore, decentralized updates are a viable option for less reliable cluster systems. Centralized schemes can also be a successful option for robust HPC systems where grade failure is rare. Still, the communication bottleneck poses severe issues when scaling on many nodes [63]. Using a decentralized scheme for

the gradient exchange and weights update, such as the one implemented by Horovod, is a practical choice for simplicity and speed for distributed training on HPC. As a high-level framework at the top of DL libraries, Horovod uses well-established Compute Unified Device Architecture (CUDA) MPI processes and relies on the NCCL library[2] to implement robust and efficient communication among workers. The choice of Horovod as a library for effective distributed training was also motivated by the ease, clarity of structure, and transparency of the required code changes. A similar strategy can now also be implemented directly in TensorFlow[3] and PyTorch[4].

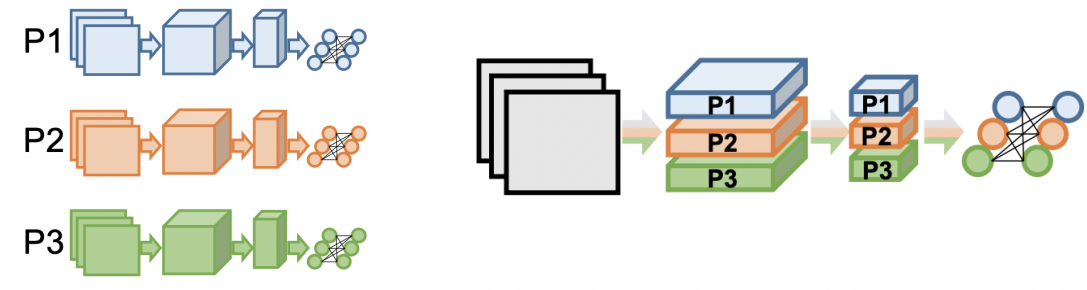
Large Batch Size Training

Besides the problems related to efficient intercommunication among nodes, the issue of degradation of the generalization capabilities of the model needs to be addressed in a large effective (global) batch size setting. Most optimization methods used to minimize the loss during training are derived from stochastic gradient descent (SGD). If training is deployed on a significant number of workers, the effective size of the global batch size increases. Therefore, the optimization handles larger batch sizes than those used to train the DL model on a single worker. Large batches (for ImageNet, on the order of a few thousand images per batch versus the standard batch size of several hundred for single-node training) cause significant performance degradation in terms of accuracy if used without additional countermeasures [37]. This may be partly due to the nature of SGD, which requires a certain amount of noise generated by the relatively small batch sizes used for update steps. Various solutions to ensure a similar level of performance with increasingly large effective batch sizes in a distributed learning setting are now available. The most utilized solutions consist of a combination of techniques such as setting up a learning rate scheduler that uses warm-up, scaling the learning rate by the number of workers, and scaling it down by a factor after a certain number of epochs [56, 109, 37]. More sophisticated strategies for handling very large batch sizes (e.g., for ImageNet, greater than $2^13 = 8192$) use adaptive learning rates adjusted based on the depth of the DL model, the magnitude of computed gradients, and training progression. This approach is used in Layer-wise Adaptive Rate Scaling (LARS), an adaptive optimizer designed explicitly for large-scale distributed training [110].

Distributed Deep Learning in Remote Sensing

The "big data" paradigm has been known in RS for over a decade. RS techniques provide large amounts of data from various sources daily at various spectral, spatial, and temporal resolutions. Consequently, the characteristic 3 Vs of "big data" (velocity, volume, variety

[2] `https://developer.nvidia.com/nccl`
[3] `https://www.tensorflow.org/api_docs/python/tf/distribute/MultiWorkerMirroredStrategy`
[4] `https://pytorch.org/docs/stable/notes/ddp.html`
[5] `https://www.uber.com/en-IT/blog/horovod/`



(a) (b)

Figure 3.5: (a) Data parallelism, (b) model parallelism [9].

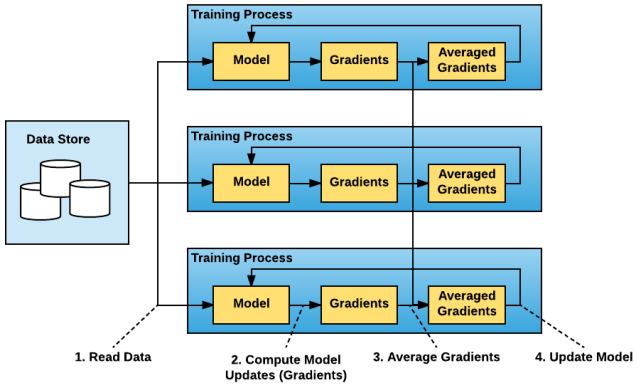


Figure 3.6: Data parallelism: exchange of the local gradients[5].

[53]) are not alien to RS [15]. HPC technologies have been used to enable computation of computationally demanding RS applications [76, 77]. [40] provides an overview of recent advances in the usage of cloud computing and HPC for RS data applications based on DL. In [5], a model parallelism approach is adopted to reduce the amount of communication with the central PS. The methodology is compared with a data parallelism approach for a LC classification task.

3.3. Harmonization

The continual monitoring of the surface of our planet is of utmost importance, with space agencies operating missions towards that goal. A variety of sensors (optical, radar, lidar, etc.) are used onboard satellites, and the increased temporal coverage provided by their combination (an example is shown in Fig. 3.7) justifies the concrete need for data fusion. Data fusion includes tools and algorithms to utilize data originating from various sources, aiming to obtain information of greater quality [103]. [18]. The Harmonized Landsat Sentinel-2 (HLS) utilizes a series of processing blocks to atmospherically correct and co-register the L8 and S2 data and finally perform band-pass filtering to harmonize the radiometric properties of the MSI (S2) and the Operational Land Imager (OLI) sensors (L8) [18]. Sen2Like [85] follows a similar approach for spatial co-registration and spectral

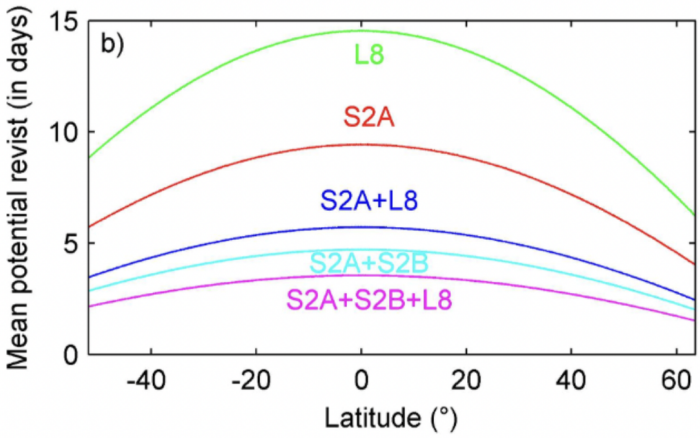


Figure 3.7: Revisit time decreases combining multiple satellites [18].

adjustment, for which knowledge of the calibration of the sensors is critical. Relying on the HLS framework, [49] goes in the direction of providing harmonized, inter-operable, analysis-ready TS of RS data, utilizing imagery acquired by S2, L8 and PlanetScope [6]. Through a case study of crop monitoring, they provided additional insights into the potential benefits of near-daily observations.

[6] https://www.planet.com/products/planet-imagery/

4. Summary of Papers and Contributions

In this Chapter, the publications enumerated in the List of Publications and included in the thesis as Appendix are introduced, and the respective contributions are highlighted.

4.1. PAPER I

R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. A. Benediktsson, "Remote Sensing Big Data Classification with High Performance Distributed Deep Learning", Remote Sensing (MDPI), vol. 11, no. 24: 3056, 2019, `https://doi.org/10.3390/rs11243056`.

*This publication contributes to the **TO 1** in that it adopts a strategy for DDL based on the utilization of Horovod on top of TensorFlow to train a multi-spectral CNN. It fulfills **TO 2** since the DL models are fed with a customized parallel loader for large data from HDF5 files. Performances were evaluated using classical ML metrics, serving **TO 3**.*

Here, the author effectively trained a multi-spectral CNNs on the Jülich Research on Exascale Cluster Architectures (JURECA) and Jülich Wizard for European Leadership Science (JUWELS) HPC systems hosted at the JSC on the dedicated partitions equipped with GPUs. The chosen dataset was a large benchmarking RS archive, BigEarthNet[1] [95], that posed several challenges, namely:

- Large number of patches (~500.000), originally available in data format not optimized for usage on HPC systems.
- Multi-label classification problem, i.e., more than one label assigned to each patch.

The experimental set-up consisted of a patch-based classification task based on LC classes. The author conducted a survey of the existing frameworks for DDL and decided to use Horovod [88]. This framework was chosen because it enables efficient scaling on HPC machines with a significant number of GPUs through the Ring-AllReduce algorithm [74]. The main technical difficulty was posed by the large size of the original dataset, which had

[1] `https://bigearth.net/`

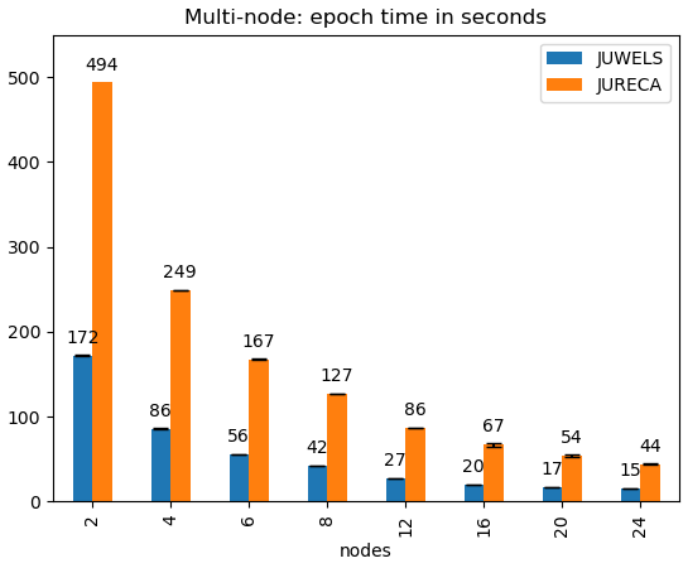


Figure 4.1: Time per epoch during training with respect to the number of nodes.

to be converted to a file format suitable for utilization on HPC and the implementation of a custom parallel data loader to streamline the process from the retrieval of data from disk to feeding it into the model. To allow the model to output multiple labels per patch, the softmax activation function of the original classification layer of the ResNet50 was substituted with a sigmoid. A combination of strategies was adopted to (i) stabilize the training of the ResNet50 in a large batch size setting (batch_size >= 8192 samples) and (ii) maintain performances in terms of test accuracy. A gradual warm-up was employed alongside a heuristic for the scaling of the learning rate with respect to the number of the utilized GPUs. A step-wise decay policy was applied to reduce the learning rate and stabilize the training at later stages. In this work, the main scientific contributions were to:

- Effectively train a ResNet-50 CNN model on BigEarthNet, reducing training instabilities and maintaining an acceptable generalization capability of the model on test data.

- Perform near-linear speed-up of the training on up to 96 GPUs on multiple HPC systems for benchmarking, reducing the total training time from ~38 hours to ~25 minutes (epoch time is shown in Fig. 4.1).

4.2. PAPER II

R. Sedona, L. Hoffmann, R. Spang, G. Cavallaro, S. Griessbach, M. Höpfner, M. Book, and M. Riedel, “Exploration of Machine Learning Methods for the Classification of Infrared Limb Spectra of Polar Stratospheric Clouds”, Atmospheric Measurement Tech-

niques (Copernicus), vol. 13, no. 7, pp. 3661–3682, 2020. https://doi.org/10.5194/amt-13-3661-2020.

*In this publication, a data pre-processing workflow was implemented to extract informative features from the original dataset, serving partially to **TO 2**. **TO 3** is addressed by comparing various classical ML methods with well established methods based on domain knowledge.*

The potential of applying ML methods to classify polar stratospheric clouds (PSCs) observed by infrared limb sounders was explored in this work. Two datasets were considered in this study, the MIPAS real data and the Cloud Scenario Database (CSDB) synthetic data. The first task was to access, convert, and process TSs of RS data in a format suitable for utilization with ML methods. An initial analysis was performed to assess the essential characteristics of the CSDB and then used ML methods to reduce the dimensionality of the feature space. Samples from the real MIPAS dataset were finally classified, comparing the performances of RF and SVM with the well-established Bayesian method [93]. The principal component analysis (PCA) was utilized to reduce the feature space's dimensionality and evaluate how well the principal components could be distinguished. Feature importance matrices were used to extract which characteristics of the signals contributed the most to the classification output and compared with physical knowledge to demystify the "black box" represented by ML models [82].

The main scientific challenges were to:

- Extract information from real and synthetic datasets by adopting tools from classical ML to classify PSC.

- Show that obtained results were physically sound through a comparison with a well-established method (e.g., Fig. 4.2 evaluates the consistency of the SVM and the Bayesian method).

The contributions were to prove that automatic ML methods can (i) achieve results in line with methods based on apriori knowledge of domain experts for the task of PSC classification and (ii) provide additional insights on the classified samples.

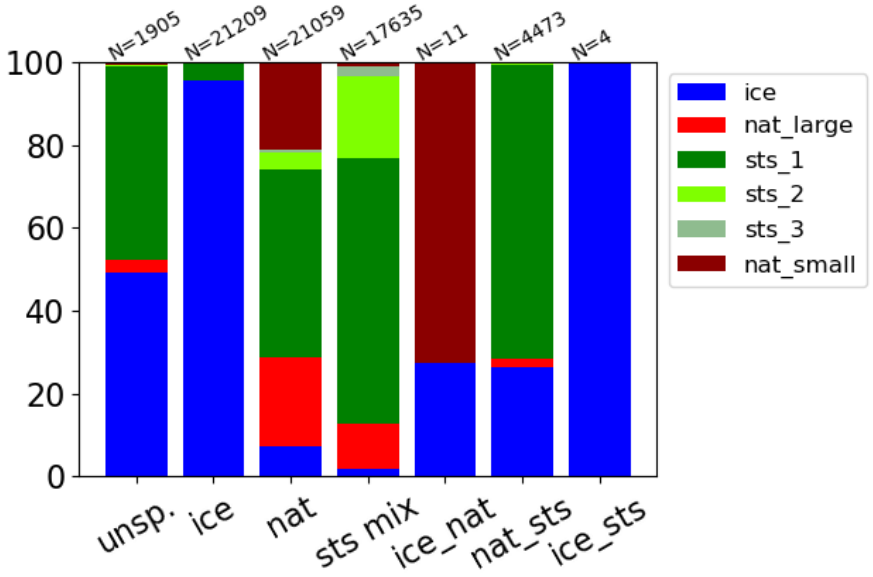


Figure 4.2: Agreement of results provided by the Bayesian classifier and SVM.

4.3. PAPER III

R. Sedona, G. Cavallaro, M. Riedel and M. Book, "Enhancing Large Batch Size Training of Deep Models for Remote Sensing Applications", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1583-1586, 2021, `https://doi.org/10.1109/IGARSS47720.2021.9555136`.

The research conducted for this conference paper was intended as an extension to work performed for Paper I to gain additional insights on promising directions towards the adoption of algorithms designed for DDL training in a large batch size setting, using benchmarking RS datasets, thus serving the ***TO 1****,* ***TO 2*** *and* ***TO 3****.*

In the paper described in Sect. 4.1 DL models were trained with large effective batch sizes. Since the experimental set-up based on large batch sizes posed several challenges [37], in this work, the Layer-wise Adaptive Moments Optimizer for Batch Training (LAMB) [112] optimizer was used to train a CNNs on HPC systems, aiming at a model with good generalization for a classification task based on RS imagery, namely the DeepSat SAT-4 and SAT-6 datasets [7]. The importance of such studies lies in the fact that a large volume of data needs to be utilized to monitor the dynamics of LC classes at multiple locations on the Earth's surface with high temporal resolution. To this end, the boost in speed-up provided by DDL is of excellent value, with an acceleration both during the prototyping of the models and at deployment.

The main contribution was to investigate the adoption of the LAMB optimizer on RS data in a large effective batch size setting while also exposing its limitation. The utilization of LAMB enabled training with a batch size of up to 65.000 samples, while the SGD

and Adam [48] optimizers did not converge. However, this study does not address the exploration of the sizeable hyper-parameter space of the experimental set-up (e.g., learning rate, decay policy, momentum, etc.), which could help reach convergence and increase the test accuracy of the model [38, 70].

4.4. PAPER IV

R. Sedona, C. Paris, G. Cavallaro, L. Bruzzone and M. Riedel, "A High-Performance Multispectral Adaptation GAN for Harmonizing Dense Time Series of Landsat-8 and Sentinel-2 Images", in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10134-10146, 2021, `https://doi.org/10.1109/JSTARS.2021.3115604`.

For this research project, the author developed an approach for the utilization of TS of RS data based on a DL model aiming to harmonize multi-source data, optimizing it to scale on HPC resources, serving ***TO 1****,* ***TO 2****,* ***TO 3*** *and* ***TO 4****.*

This work focused on the development of a framework (depicted in Fig. 4.3) aiming to densify TS through the harmonization of imagery acquired by S2 and L8 to produce analysis ready data (ARD) [25], i.e., requiring no additional effort from the user (no data filtering required, outliers already removed). A DL model based on pix2pix [45] (introduced in Sect. 3.2.4) was developed to spectrally and spatially harmonize a subset of the S2 and L8 data (i.e., the six overlapping bands). The generator of the model was modified from the original U-Net encoder-decoder, which learns to generate target L8 images from input S2 images. At the same time, the discriminator aims at detecting fake vs. real images. To increase the stability of the training, spectral normalization layers [68] were inserted after the instance normalization in the discriminator, and the original adversarial loss was substituted with a relativistic adversarial loss [46]. The densified output was evaluated in a crop type classification case study and through spectral agreement metrics. The main contributions were:

- The demonstration that an automated approach based on DL can generate ARD competitive with those provided by physical methods, e.g. the HLS.

- The development of a method that scales efficiently on multiple GPUs.

The main limitation of this work, however, lies in the study of a single AoI due to the technical difficulties encountered during data retrieval. Among other scientific motivations (e.g., increase the informative content of the samples), the aforementioned technical aspect also drove the adoption and porting of an existing framework [73], further described in Sect. 4.5.

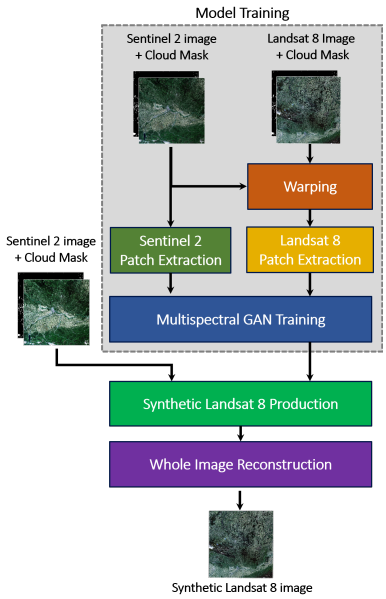


Figure 4.3: Block scheme of the proposed method.

4.5. PAPER V

R. Sedona, C. Paris, L. Tian, M. Riedel and G. Cavallaro, "An Automatic Approach for the Production of a Time Series of Consistent Land-Cover Maps Based on Long-Short Term Memory", in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 203-206, 2022, `https://doi.org/10.1109/IGARSS46834.2022.9883655`.

*This conference paper contributes to **TO 1** and **TO 2** since DL models were trained on multi-year RS data. A classical ML model was adopted to increase the consistency of the input data to the DL models, relating also to **TO 3**.*

In preparation for this work, a pre-existing framework [73] for the acquisition and processing of large volumes of informative RS data was ported to HPC systems hosted at JSC. The framework utilizes k-means clustering to retrieve the most reliable samples from the considered tile, and stratified random sampling is performed to collect the points, creating a clean dataset [78]. Using the Sentinelsat [2] API, S2 images were queried from the Copernicus Open Access Hub for the years 2018, 2019, and 2020, within the AoI of Trentino, Italy. This area was selected because ground truth data for the validation of the proposed approach was available due to the impact of the Vaia storm [101]. A RF was used to classify the samples for the three years individually. New training, validation, and test sets were collected utilizing samples associated with the same label throughout the three years to increase the stability of the dataset. The multi-year training dataset, consisting of ~10.000 points and 20 acquisitions per year, was then fed into a LSTM model during training. At inference time, the matrix with the reflectances covering the AoI (total number of pixels is $> 6.000.000$) was used as input to predict the new LC map. The

[2] `https://sentinelsat.readthedocs.io/en/stable/`

paper's main contribution lies in implementing an approach that produces a reliable and informative multi-year training set. This was used as input data for an LSTM, showing that a longer TS of informative data can help increase the consistency of the output LC map, the importance of which is explained in Sect. 3.1.1.

4.6. PAPER VI

R. Sedona, C. Paris, J. Ebert, M. Riedel, G. Cavallaro, "Toward the Production of Spatiotemporally Consistent Annual Land Cover Maps using Sentinel-2 Time Series", IEEE Geoscience and Remote Sensing Letters vol. 20, pp. 1-5, 2023, `https://doi.org/10.1109/LGRS.2023.3329428`.

Since this work was designed as an extension of the previous Paper V, ***TO 1*** *and* ***TO 2*** *are addressed by the utilization of an attention-based DL model that exploits multi-year RS for increased consistency of the output LC classification maps, using classical methods for the evaluation of the obtained results (****TO 3****).*

Following the footsteps of Paper V, described in the previous Sect. 4.5, a scalable and semi-automatic method was proposed to generate annual LC maps using a Transformer model. The data were TSs of satellite imagery acquired by S2. For the tile of the AoI covering Trentino, 15 acquisitions per year for 2018, 2019, and 2020 were downloaded. The averaging before the final classification layer of the baseline Transformer [84] for RS data was removed to retrieve multi-temporal attention weights and, consequently, a multi-temporal classified output sequence. We employed a convolution of a binary intrayear LC label sequence with a step function to detect permanent changes in the LC time series, differentiating them from seasonal variations by assessing their stability, recurrence, and temporal occurrence. These post-processing steps were crucial in discriminating between stable and non-stable patterns in the output LC maps, thereby providing insight into the spatial and temporal locations of acquisitions where abrupt changes occurred. The main contributions are:

- The adaptation of a Transformer for EO applications to extract multi-temporal output TS of LC to generate the stable component from the real change and spurious change components in the output signals describing the evolution of the LC maps.

- Estimation of occurrence of real changes to provide more accurate temporal information to the end-user on top of the updated LC map (as shown in Fig. 4.4).

This line of work allowed gaining insights into the way the attention mechanism of the Transformer model responds to TS of RS data, fostering further developments. (i) On one side, the visualization of the attention weights could guide the selection of more informative training sets, (ii) on the other side, the impact of global vs. causal attention masks should be carefully analyzed to understand the impact of the selection of the

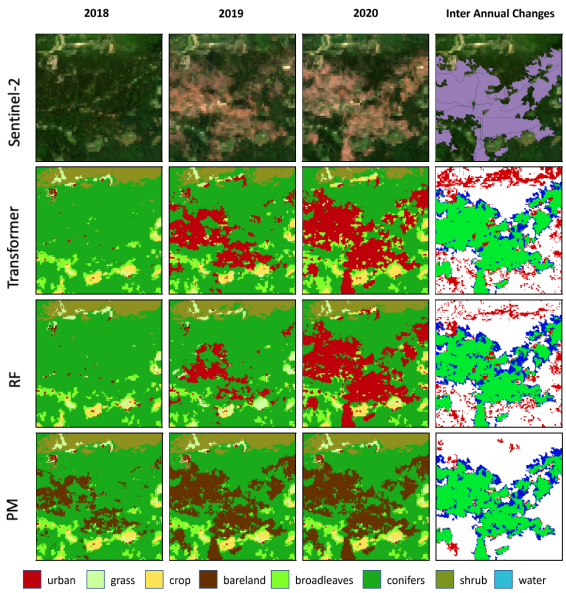


Figure 4.4: Examples of multiyear maps and changes obtained for a small portion of the considered study area for 2018–2020 using the: 1) Transformer, 2) random forest, and 3) proposed method. The Sentinel-2 images acquired in 2018–2020 are reported with the changes reference map.

attention window on the final output.

5. Conclusions

5.1. Summary

In recent decades, EO has benefitted from increasingly large volumes of multi-source RS data, particularly in the area of LC maps. Scientists and policymakers rely heavily on such data to address pressing environmental issues. As the amount of data increases, classical ML and, more recently, DL methods have been developed to extract information from these large datasets. However, the increased size of the datasets needed to train DL models has increased the time to deployment, which can hinder their effective utilization. DDL on HPC systems can speed up the training of the models, reducing the time to deployment and development. Space agencies operate a variety of missions with a variety of sensors that can be utilized to increase the temporal resolution at which a particular area is observed, with potential benefits in DL model accuracy.

The use of RS data and HPC systems provide excellent potential for large-scale EO applications. Several scientific and technical challenges need to be addressed to maximize the benefit of this potential. The main research objectives of this project are to achieve near-linear scalability for DL models on HPC systems, extract valuable information from complex RS datasets, demonstrate the effectiveness of classical ML methods in addressing challenging applications, and harness multi-source data to densify TS of RS measurements. This research project is addressing these objectives in parallel, utilizing both DL models and classical ML methods while also making use of large datasets and HPC resources to improve computational efficiency. By addressing these challenges and objectives, the project aims to increase the amount of informative data available for DL models and gain a deeper understanding of complex relationships among sources for more accurate insights.

Throughout the thesis work, multiple aspects that play a crucial role in the development of a RS framework for EO applications were addressed by the author. DL models need a large quantity of data for effective training. The availability of RS datasets has been continually increasing over the years. Still, the flexibility to carry out large-scale experiments, including specific study areas, comes at the cost of increased efforts for the user. **TO 2** (information extraction) was fulfilled through a significant effort devoted to simplifying the extraction of informative samples to feed the DL models. Motivated by the need for increased temporal resolution of multispectral imagery, an investigation on approaches for the densification of TS of RS data was performed. **TO 4** (multi-source data) was reached

by the utilization of methodologies to harmonize data from S2 and L8 missions, which were shown to enhanceLC classification performances. **TO 3** (evaluation of obtained results) was achieved by borrowing tools from classical ML, conducting assessments of results obtained with DL that helped the validation of the scientific approaches developed to carry out a variety of tasks for EO applications. Since the computationally demanding iterative optimization of DL on large datasets hinders the execution of the experiments by the researcher and results in a long time to deployment for actual use cases, to tackle these issues, a multifold strategy relying on data parallelism was adopted, achieving a significant speed-up of such time-consuming process.

Throughout the project, various research and development activities were conducted to tackle the challenges above in scientific applications. Notably, the first objective (**TO 1**) (near-linear scaling on HPC systems) was accomplished by implementing data parallelism techniques, which enabled near-linear scaling of DL models. As a result, the time to convergence was significantly reduced, thereby addressing one of the significant obstacles in the field. The rapidly evolving technological landscape, situated at the intersection of dynamic advancements in DL architectures, cutting-edge libraries for DDL, and the ever-changing HPC landscape, demands a continuous and unwavering focus to fully leverage the adoption of these diverse technological solutions to tackle pressing scientific challenges. Paper II demonstrates that ML methods can compete with classical methods in the case study of PSC classification while also utilizing tools for visualization and validation of the obtained results. Paper III shows the benefits of using a more advanced optimizer on a RS dataset, which lays the foundation for further studies on hyperparameter tuning and model optimization in a scalable setting.

The use of data parallelism as a strategy for DDL is a common theme in Paper I, V and VI, which present different use cases for LC classification. Paper I utilizes the RS BigEarthNet dataset, with up to 96 GPUs and a combination of strategies to address the problem of large global batch size. This reduces the time to convergence of a ResNet50 CNN. Paper V and VI, on the other hand, employ a whole modular framework for data query, download, and pre-processing on HPC systems, enabling flexible utilization of TS acquired by the Sentinel-2 constellation. DL models that exploit the temporal dynamics of the input signal are employed to increase the consistency of the output multi-year LC maps. Paper IV, a method based on pix2pix is used to successfully densify a TS of RS data, resulting in increased accuracy in an LC classification task.

5.2. Future Directions

Integrating EO data with other modalities, such as climatic data records, national surveys, or social-media information, holds great potential for advancing the field. By combining data from various sources, researchers can gain a more complete and nuanced understanding of the phenomena they study. Many research paths have become available with large amounts of data and using the DL models.

Workflows serve as a method for organizing and automating a sequence of computational and data manipulation tasks. These tasks can be visually represented as a series of building blocks, and formalizing them enables the code to be reused and transported across various platforms. Workflow managers are software tools that aid in designing and executing workflows, and they can optimize processing using mathematical principles like scheduling and parallelization. One promising direction is the deployment of workflows on HPC systems to enable continual ingestion of new data, which can be achieved through tools such as Apache Airflow [1]. Adopting such tools could help researchers keep their models up-to-date with the latest information, improve the accuracy of their predictions, and deploy the applications at scale.

Another critical area of research is hyperparameter tuning of DL models, which can significantly improve model performance [1]. To achieve high performance in deep neural networks, it is necessary to set appropriate hyperparameters before training. However, determining the best hyperparameters for a specific task can be challenging and time-consuming. Typically, this involves a lot of manual tuning, which can lead to significant improvements in performance but may require a considerable amount of computational resources. One of the main challenges in finding the optimal set of hyperparameters is the expensive evaluation of different configurations. Running an entire training for each configuration can be computationally expensive and time-consuming. To overcome this challenge, researchers have proposed two main strategies for reducing the overall computational costs:

- Improving the choice of hyperparameters with optimization algorithms: optimization algorithms can help automate the hyperparameter tuning process by efficiently searching for their best combination. Examples of such algorithms include grid search, random search, and Bayesian optimization.

- Reducing the runtime of the training runs: researchers have proposed several techniques to reduce the time required to train deep neural networks, such as using faster optimization algorithms, reducing the network size, and using parallel processing.

In addition, including multi-source data holds great potential for improving the quality of model outputs. By combining data from multiple sources, researchers can gain a more comprehensive view of the phenomena they study. The exponential growth of the volume of data collected by various RS sensors mounted onboard satellites and airborne has significantly increased in recent years [114]. Ancillary data from sources like social media, crowdsourcing, and web scraping are also gaining importance as sources of information. The sensors on these platforms vary significantly in terms of the properties they sense and their data's spatial and spectral resolution. The challenges posed by the vast amounts of data from different sources include processing it effectively and efficiently, especially when combining datasets to extract maximum utility [33]. Multi-source data fusion has recently received significant attention in addressing these challenges. Advances in computing power, data interoperability, and data fusion methods have enabled effectively

[1] `https://airflow.apache.org/`

managing and analyzing these large and complex datasets.

Finally, the spatial information, exploited and well understood for a long time [100, 21], could be coupled with the temporal information [34] to learn a better representation of the patterns in the data. The use of DL models that exploit spatio-temporal information, such as ViVit [4] or ConvLSTM [90], is an area of active research in RS [58, 57]. These models are particularly well-suited for analyzing EO data since they can capture its spatial and temporal dimensions. As such, they promise to advance the field and unlock the full potential of EO data.

References

[1] Marcel Aach et al. "Accelerating Hyperparameter Tuning of a Deep Learning Model for Remote Sensing Image Classification". In: *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*. 2022, pp. 263–266. DOI: 10.1109/IGARSS46834.2022.9883257.

[2] S. Parker Abercrombie and Mark A. Friedl. "Improving the Consistency of Multitemporal Land Cover Maps Using a Hidden Markov Model". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.2 (2016), pp. 703–713. DOI: 10.1109/TGRS.2015.2463689.

[3] D. Ao et al. *Dialectical GAN for SAR Image Translation: From Sentinel-1 to TerraSAR-X*. en. Oct. 2018. DOI: 10.3390/rs10101597. URL: http://dx.doi.org/10.3390/rs10101597.

[4] Anurag Arnab et al. *ViViT: A Video Vision Transformer*. 2021. DOI: 10.48550/ARXIV.2103.15691. URL: https://arxiv.org/abs/2103.15691.

[5] Maria Aspri, Grigorios Tsagkatakis, and Panagiotis Tsakalides. *Distributed Training and Inference of Deep Learning Models for Multi-Modal Land Cover Classification*. en. Aug. 2020. DOI: 10.3390/rs12172670. URL: http://dx.doi.org/10.3390/rs12172670.

[6] Eric Aubanel. *Elements of Parallel Computing*. Chapman and Hall/CRC, Dec. 2016. DOI: 10.1201/b21979. URL: http://dx.doi.org/10.1201/b21979.

[7] Saikat Basu et al. *DeepSat - A Learning framework for Satellite Imagery*. 2015. DOI: 10.48550/ARXIV.1509.03602. URL: https://arxiv.org/abs/1509.03602.

[8] Yakoub Bazi et al. *Vision Transformers for Remote Sensing Image Classification*. en. Feb. 2021. DOI: 10.3390/rs13030516. URL: http://dx.doi.org/10.3390/rs13030516.

[9] Tal Ben-Nun and Torsten Hoefler. "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis". In: *ACM Comput. Surv.* 52.4 (Aug. 2019). ISSN: 0360-0300. DOI: 10.1145/3320060. URL: https://doi.org/10.1145/3320060.

[10] Jose Bioucas-Dias et al. "Hyperspectral Remote Sensing Data Analysis and Future Challenges". In: *Geoscience and Remote Sensing Magazine, IEEE* 1 (June 2013), pp. 6–36. DOI: 10.1109/MGRS.2013.2244672.

[11] Christopher F. Brown et al. *Dynamic World, Near real-time global 10m land use land cover mapping*. en. June 2022. DOI: 10.1038/s41597-022-01307-4. URL: http://dx.doi.org/10.1038/s41597-022-01307-4.

[12] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

[13] Tao Che et al. "Snow depth derived from passive microwave remote-sensing data in China". In: *Annals of Glaciology* 49 (2008), pp. 145–154. DOI: 10.3189/172-75-640-8787-814690.

[14] Hao Chen, Zipeng Qi, and Zhenwei Shi. "Remote Sensing Image Change Detection With Transformers". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–14. DOI: 10.1109/tgrs.2021.3095166. URL: https://doi.org/10.1109%2Ftgrs.2021.3095166.

[15] Mingmin Chi et al. "Big Data for Remote Sensing: Challenges and Opportunities". In: *Proceedings of the IEEE* 104.11 (2016), pp. 2207–2219. DOI: 10.1109/JPROC.2016.2598228.

[16] Jaegul Choo and Shixia Liu. *Visual Analytics for Explainable Deep Learning*. 2018. DOI: 10.48550/ARXIV.1804.02527. URL: https://arxiv.org/abs/1804.02527.

[17] Dan Claudiu Cireşan et al. "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". In: *Neural Computation* 22.12 (2010), pp. 3207–3220. DOI: 10.1162/NECO_a_00052.

[18] Martin Claverie et al. "The Harmonized Landsat and Sentinel-2 surface reflectance data set". In: *Remote Sensing of Environment* 219 (2018), pp. 145–161. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2018.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0034425718304139.

[19] Adam Coates et al. "Deep Learning with COTS HPC Systems". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, III–1337–III–1345.

[20] Andrew Collette. *Python and HDF5*. O'Reilly Media, 2013. ISBN: 978-1-4493-6783-1.

[21] Alexis Comber and Michael Wulder. *Considering spatiotemporal processes in big data analysis: Insights from remote sensing of land cover and land use*. en. July 2019. DOI: 10.1111/tgis.12559. URL: http://dx.doi.org/10.1111/tgis.12559.

[22] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[23] Jeff Donahue et al. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, June 2014, pp. 647–655. URL: https://proceedings.mlr.press/v32/donahue14.html.

[24] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929.

[25] Alexey Egorov et al. *Demonstration of Percent Tree Cover Mapping Using Landsat Analysis Ready Data (ARD) and Sensitivity with Respect to Landsat ARD Processing Level*. en. Jan. 2018. DOI: 10.3390/rs10020209. URL: http://dx.doi.org/10.3390/rs10020209.

[26] Steven Farrell et al. *MLPerf HPC: A Holistic Benchmark Suite for Scientific Machine Learning on HPC Systems*. 2021. DOI: 10.48550/ARXIV.2110.11466. URL: https://arxiv.org/abs/2110.11466.

[27] Laurent Ferro-Famil and Eric Pottier. "1 - Synthetic Aperture Radar Imaging". In: *Microwave Remote Sensing of Land Surface*. Ed. by Nicolas Baghdadi and Mehrez Zribi. Elsevier, 2016, pp. 1–65. ISBN: 978-1-78548-159-8. DOI: https://doi.org/10.1016/B978-1-78548-159-8.50001-3. URL: https://www.sciencedirect.com/science/article/pii/B9781785481598500013.

[28] Africa Flores et al. *The SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation*. Apr. 2019. DOI: 10.25966/nr2c-s697.

[29] Robin Flowers and Charles Edeki. "Business Process Modeling Notation". In: 2013.

[30] Mike Folk et al. *An overview of the HDF5 technology suite and its applications*. Mar. 2011. DOI: 10.1145/1966895.1966900. URL: http://dx.doi.org/10.1145/1966895.1966900.

[31] Chelle L. Gentemann et al. *Passive Microwave Remote Sensing of the Ocean: An Overview*. 2010. DOI: 10.1007/978-90-481-8681-5_2. URL: http://dx.doi.org/10.1007/978-90-481-8681-5_2.

[32] F.A. Gers and J. Schmidhuber. "Recurrent nets that time and count". In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Vol. 3. 2000, 189–194 vol.3. DOI: 10.1109/IJCNN.2000.861302.

[33] Pedram Ghamisi et al. "Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art". In: *IEEE Geoscience and Remote Sensing Magazine* 7.1 (2019), pp. 6–39. DOI: 10.1109/MGRS.2018.2890023.

[34] Cristina Gómez, Joanne C. White, and Michael A. Wulder. *Optical remotely sensed time series data for land cover classification: A review*. en. June 2016. DOI: 10.1016/j.isprsjprs.2016.03.008. URL: http://dx.doi.org/10.1016/j.isprsjprs.2016.03.008.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[36] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[37] Priya Goyal et al. *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*. 2017. DOI: 10.48550/ARXIV.1706.02677. URL: https://arxiv.org/abs/1706.02677.

[38] Klaus Greff et al. "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2222–2232. DOI: 10.1109/tnnls.2016.2582924. URL: https://doi.org/10.1109%2Ftnnls.2016.2582924.

[39] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. *Why do tree-based models still outperform deep learning on tabular data?* 2022. DOI: 10.48550/ARXIV.2207.08815. URL: https://arxiv.org/abs/2207.08815.

[40] Juan M. Haut et al. "Distributed Deep Learning for Remote Sensing Data Interpretation". In: *Proceedings of the IEEE* 109.8 (2021), pp. 1320–1349. DOI: 10.1109/JPROC.2021.3063258.

[41] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 770–778.

[42] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory*. en. Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[43] Elad Hoffer, Itay Hubara, and Daniel Soudry. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks". In: (2017). DOI: 10.48550/ARXIV.1705.08741. URL: https://arxiv.org/abs/1705.08741.

[44] *Infiniband architecture specification*$_2$004. 2004.

[45] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2016. DOI: 10.48550/ARXIV.1611.07004. URL: https://arxiv.org/abs/1611.07004.

[46] Alexia Jolicoeur-Martineau. *The relativistic discriminator: a key element missing from standard GAN*. 2018. arXiv: 1807.00734 [cs.LG].

[47] Nitish Shirish Keskar et al. *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*. 2016. DOI: 10.48550/ARXIV.1609.04836. URL: https://arxiv.org/abs/1609.04836.

[48] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[49] Lukas Kondmann et al. "DENETHOR: The DynamicEarthNET dataset for Harmonized, inter-Operable, analysis-Ready, daily crop monitoring from space". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021. URL: https://openreview.net/forum?id=uUa4jNMLjrL.

[50] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. *Do Better ImageNet Models Transfer Better?* 2018. DOI: 10.48550/ARXIV.1805.08974. URL: https://arxiv.org/abs/1805.08974.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[52] *Land-Use and Land-Cover Change*. Springer Berlin Heidelberg, 2006. DOI: 10.1007/3-540-32202-7. URL: http://dx.doi.org/10.1007/3-540-32202-7.

[53] Doug Laney et al. "3D data management: Controlling data volume, velocity and variety". In: *META group research note* 6.70 (2001), p. 1.

[54] M. A. Lebedev et al. *CHANGE DETECTION IN REMOTE SENSING IMAGES USING CONDITIONAL ADVERSARIAL NETWORKS*. en. May 2018. DOI: 10.5194/isprs-archives-xlii-2-565-2018. URL: http://dx.doi.org/10.5194/isprs-archives-XLII-2-565-2018.

[55] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. *Deep learning*. en. May 2015. DOI: 10.1038/nature14539. URL: http://dx.doi.org/10.1038/nature14539.

[56] Haifeng Li et al. *RSI-CB: A Large Scale Remote Sensing Image Classification Benchmark via Crowdsource Data*. 2017. DOI: 10.48550/ARXIV.1705.10450. URL: https://arxiv.org/abs/1705.10450.

[57] Jiaxin Li et al. "Deep learning in multimodal remote sensing data fusion: A comprehensive review". In: *International Journal of Applied Earth Observation and Geoinformation* 112 (2022), p. 102926.

[58] Weisheng Li et al. "Spatiotemporal fusion of remote sensing images using a convolutional neural network with attention and multiscale mechanisms". In: *International Journal of Remote Sensing* 42.6 (2021), pp. 1973–1993.

[59] Shixia Liu et al. *Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective*. 2017. DOI: 10.48550/ARXIV.1702.01226. URL: https://arxiv.org/abs/1702.01226.

[60] Haobo Lyu, Hui Lu, and Lichao Mou. *Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection*. en. June 2016. DOI: 10.3390/rs8060506. URL: http://dx.doi.org/10.3390/rs8060506.

[61] Lei Ma et al. *Deep learning in remote sensing applications: A meta-analysis and review*. en. June 2019. DOI: 10.1016/j.isprsjprs.2019.04.015. URL: http://dx.doi.org/10.1016/j.isprsjprs.2019.04.015.

[62] Yan Ma et al. "Remote sensing big data computing: Challenges and opportunities". In: *Future Generation Computer Systems* 51 (2015). Special Section: A Note on New Trends in Data-Aware Scheduling and Resource Provisioning in Modern HPC Systems, pp. 47–60. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2014.10.029. URL: https://www.sciencedirect.com/science/article/pii/S0167739X14002234.

[63] Ruben Mayer and Hans-Arno Jacobsen. "Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques, and Tools". In: *ACM Comput. Surv.* 53.1 (Feb. 2020). ISSN: 0360-0300. DOI: 10.1145/3363554. URL: https://doi.org/10.1145/3363554.

[64] Sam McCandlish et al. *An Empirical Model of Large-Batch Training*. 2018. DOI: 10.48550/ARXIV.1812.06162. URL: https://arxiv.org/abs/1812.06162.

[65] Warren S. McCulloch and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. en. Dec. 1943. DOI: 10.1007/bf02478259. URL: http://dx.doi.org/10.1007/BF02478259.

[66] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 4.0*. June 2021. URL: https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf.

[67] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. DOI: 10.48550/ARXIV.1411.1784. URL: https://arxiv.org/abs/1411.1784.

[68] Takeru Miyato et al. *Spectral Normalization for Generative Adversarial Networks*. 2018. arXiv: 1802.05957 [cs.LG].

[69] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. *Support vector machines in remote sensing: A review*. en. May 2011. DOI: 10.1016/j.isprsjprs.2010.11.001. URL: http://dx.doi.org/10.1016/j.isprsjprs.2010.11.001.

[70] Zachary Nado et al. *A Large Batch Optimizer Reality Check: Traditional, Generic Optimizers Suffice Across Batch Sizes*. 2021. DOI: 10.48550/ARXIV.2102.06356. URL: https://arxiv.org/abs/2102.06356.

[71] Frank Nielsen. *A Glance at High Performance Computing (HPC)*. 2016. DOI: 10.1007/978-3-319-21903-5_1. URL: http://dx.doi.org/10.1007/978-3-319-21903-5_1.

[72] P.C. "T. M. Lillesand R. W. Kiefer 1979. Remote Sensing and Image Interpretation. xii 612 pp., numerous illustrations. New York, Chichester: John Wiley. Price £12.50. ISBN 0 471 02609 3." In: *Geological Magazine* 117.3 (1980), pp. 305–306. DOI: 10.1017/S0016756800030636.

[73] Claudia Paris and Lorenzo Bruzzone. "A Novel Approach to the Unsupervised Extraction of Reliable Training Samples From Thematic Products". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.3 (2021), pp. 1930–1948. DOI: 10.1109/TGRS.2020.3001004.

[74] Pitch Patarasuk and Xin Yuan. "Bandwidth Optimal All-Reduce Algorithms for Clusters of Workstations". In: *J. Parallel Distrib. Comput.* 69.2 (Feb. 2009), pp. 117–124. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2008.09.002. URL: https://doi.org/10.1016/j.jpdc.2008.09.002.

[75] Carpenter Paul et al. "Heterogeneous High Performance Computing Heterogeneity is here to stay: Challenges and Opportunities in HPC". In: (Feb. 2022).

[76] Antonio Plaza and Chein-I Chang. *High Performance Computing in Remote Sensing*. Oct. 2007. DOI: 10.1201/9781420011616. URL: http://dx.doi.org/10.1201/9781420011616.

[77] Antonio Plaza et al. "Parallel Hyperspectral Image and Signal Processing [Applications Corner]". In: *IEEE Signal Processing Magazine* 28.3 (2011), pp. 119–126. DOI: 10.1109/MSP.2011.940409.

[78] Erhard Rahm and Hong Hai Do. “Data Cleaning: Problems and Current Approaches”. In: *IEEE Data Eng. Bull.* 23 (2000), pp. 3–13.

[79] Aditya Ramesh et al. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. DOI: 10.48550/ARXIV.2204.06125. URL: https://arxiv.org/abs/2204.06125.

[80] Ali Sharif Razavian et al. *CNN Features off-the-shelf: an Astounding Baseline for Recognition*. 2014. DOI: 10.48550/ARXIV.1403.6382. URL: https://arxiv.org/abs/1403.6382.

[81] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[82] Cynthia Rudin. *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead*. en. May 2019. DOI: 10.1038/s42256-019-0048-x. URL: http://dx.doi.org/10.1038/s42256-019-0048-x.

[83] Marc Rußwurm and Marco Körner. “Temporal Vegetation Modelling Using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-spectral Satellite Images”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1496–1504. DOI: 10.1109/CVPRW.2017.193.

[84] Marc Rußwurm et al. *BreizhCrops: A Time Series Dataset for Crop Type Mapping*. 2019. DOI: 10.48550/ARXIV.1905.11893. URL: https://arxiv.org/abs/1905.11893.

[85] Sébastien Saunier et al. “Sen2like, A Tool To Generate Sentinel-2 Harmonised Surface Reflectance Products - First Results with Landsat-8”. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. 2019, pp. 5650–5653. DOI: 10.1109/IGARSS.2019.8899213.

[86] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (Jan. 2015), pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003. URL: https://doi.org/10.1016%2Fj.neunet.2014.09.003.

[87] G. Schubert. *Treatise on Geophysics*. Jan. 2007. DOI: 10.1016/C2009-1-28330-4.

[88] Alexander Sergeev and Mike Del Balso. *Horovod: fast and easy distributed deep learning in TensorFlow*. 2018. DOI: 10.48550/ARXIV.1802.05799. URL: https://arxiv.org/abs/1802.05799.

[89] Christopher J. Shallue et al. “Measuring the Effects of Data Parallelism on Neural Network Training”. In: (2018). DOI: 10.48550/ARXIV.1811.03600. URL: https://arxiv.org/abs/1811.03600.

[90] Xingjian Shi et al. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. 2015. DOI: 10.48550/ARXIV.1506.04214. URL: https://arxiv.org/abs/1506.04214.

[91] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

[92] E. A. Smith et al. *Design of an inversion-based precipitation profile retrieval algorithm using an explicit cloud model for initial guess microphysics*. en. 1994. DOI: 10.1007/bf01030052. URL: http://dx.doi.org/10.1007/BF01030052.

[93] Reinhold Spang et al. *A multi-wavelength classification method for polar stratospheric cloud types using infrared limb spectra*. en. Aug. 2016. DOI: 10.5194/amt-9-3619-2016. URL: http://dx.doi.org/10.5194/amt-9-3619-2016.

[94] Gencer Sumbul et al. *BigEarthNet-MM: A Large-Scale, Multimodal, Multilabel Benchmark Archive for Remote Sensing Image Classification and Retrieval [Software and Data Sets]*. Sept. 2021. DOI: 10.1109/mgrs.2021.3089174. URL: http://dx.doi.org/10.1109/MGRS.2021.3089174.

[95] Gencer Sumbul et al. "Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding". In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, July 2019. DOI: 10.1109/igarss.2019.8900532. URL: https://doi.org/10.1109%2Figarss.2019.8900532.

[96] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. DOI: 10.48550/ARXIV.1409.4842. URL: https://arxiv.org/abs/1409.4842.

[97] Swapan Talukdar et al. "Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review". In: *Remote Sensing* 12.7 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12071135. URL: https://www.mdpi.com/2072-4292/12/7/1135.

[98] K. Tempfli et al. *Principles of remote sensing : an introductory textbook*. English. ITC Educational Textbook Series. Netherlands: International Institute for Geo-Information Science and Earth Observation, 2009. ISBN: 978-90-6164-270-1.

[99] The HDF Group. *Hierarchical Data Format, version 5*. 1997-NNNN. URL: https://www.hdfgroup.org/HDF5/.

[100] W. R. Tobler. *A Computer Movie Simulating Urban Growth in the Detroit Region*. June 1970. DOI: 10.2307/143141. URL: http://dx.doi.org/10.2307/143141.

[101] Gaia Vaglio Laurin et al. *Satellite open data to monitor forest damage caused by extreme climate-induced events: a case study of the Vaia storm in Northern Italy*. en. Dec. 2020. DOI: 10.1093/forestry/cpaa043. URL: http://dx.doi.org/10.1093/forestry/cpaa043.

[102] Ashish Vaswani et al. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[103] L. Wald. "Some terms of reference in data fusion". In: *IEEE Transactions on Geoscience and Remote Sensing* 37.3 (1999), pp. 1190–1193. DOI: 10.1109/36.763269.

[104] Jie Wang et al. *Mapping global land cover in 2001 and 2010 with spatial-temporal consistency at 250m resolution*. en. May 2015. DOI: 10.1016/j.isprsjprs.2014.03.007. URL: http://dx.doi.org/10.1016/j.isprsjprs.2014.03.007.

[105] Mark D. Wilkinson et al. *The FAIR Guiding Principles for scientific data management and stewardship*. en. Mar. 2016. DOI: 10.1038/sdata.2016.18. URL: http://dx.doi.org/10.1038/sdata.2016.18.

[106] Timothy S. Woodall et al. “High Performance RDMA Protocols in HPC”. In: *PVM/MPI*. 2006.

[107] Masafumi Yamazaki et al. *Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds*. 2019. DOI: 10.48550/ARXIV.1903.12650. URL: https://arxiv.org/abs/1903.12650.

[108] Dong Yin et al. “Gradient Diversity: a Key Ingredient for Scalable Distributed Learning”. In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, Sept. 2018, pp. 1998–2007. URL: https://proceedings.mlr.press/v84/yin18a.html.

[109] Chris Ying et al. *Image Classification at Supercomputer Scale*. 2018. DOI: 10.48550/ARXIV.1811.06992. URL: https://arxiv.org/abs/1811.06992.

[110] Yang You, Igor Gitman, and Boris Ginsburg. *Large Batch Training of Convolutional Networks*. 2017. DOI: 10.48550/ARXIV.1708.03888. URL: https://arxiv.org/abs/1708.03888.

[111] Yang You et al. “Fast Deep Neural Network Training on Distributed Systems and Cloud TPUs”. In: *IEEE Trans. Parallel Distrib. Syst.* 30.11 (Nov. 2019), pp. 2449–2462. ISSN: 1045-9219. DOI: 10.1109/TPDS.2019.2913833. URL: https://doi.org/10.1109/TPDS.2019.2913833.

[112] Yang You et al. *Large Batch Optimization for Deep Learning: Training BERT in 76 minutes*. 2019. DOI: 10.48550/ARXIV.1904.00962. URL: https://arxiv.org/abs/1904.00962.

[113] Guodong Zhang et al. *Which Algorithmic Choices Matter at Which Batch Sizes? Insights From a Noisy Quadratic Model*. 2019. DOI: 10.48550/ARXIV.1907.04164. URL: https://arxiv.org/abs/1907.04164.

[114] Jixian Zhang. *Multi-source remote sensing data fusion: status and trends*. en. Mar. 2010. DOI: 10.1080/19479830903561035. URL: http://dx.doi.org/10.1080/19479830903561035.

[115] Xiao Xiang Zhu et al. “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources”. In: *IEEE Geoscience and Remote Sensing Magazine* 5.4 (Dec. 2017), pp. 8–36. DOI: 10.1109/mgrs.2017.2762307. URL: https://doi.org/10.1109%2Fmgrs.2017.2762307.

A. Appended Papers

The publications indexed in the List of Publications, further summarized and explained in Chapter 4, are included as Appendix hereby.

remote sensing

MDPI

Article

Remote Sensing Big Data Classification with High Performance Distributed Deep Learning

Rocco Sedona [1,2,3,4,*,†], Gabriele Cavallaro [2,3,4,†], Jenia Jitsev [2,4,†], Alexandre Strube [2], Morris Riedel [1,2,3,4] and Jón Atli Benediktsson [1]

1 School of Engineering and Natural Sciences, University of Iceland, Dunhagi 5, 107 Reykjavík, Iceland; morris@hi.is (M.R.); benedikt@hi.is (J.A.B.)
2 Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich (FZJ), Wilhelm-Johnen-Strasse 1, 52425 Jülich, Germany; g.cavallaro@fz-juelich.de (G.C.); j.jitsev@fz-juelich.de (J.J.); a.strube@fz-juelich.de (A.S.)
3 High Productivity Data Processing Research Group, JSC, 52425 Jülich, Germany
4 Cross-Sectional Team Deep Learning (CST-DL), JSC, 52425 Jülich, Germany
* Correspondence: r.sedona@fz-juelich.de; Tel.: +49-2461-61-1497
† These authors contributed equally to this work.

Received: 16 October 2019; Accepted: 11 December 2019; Published: 17 December 2019

check for updates

Abstract: High-Performance Computing (HPC) has recently been attracting more attention in remote sensing applications due to the challenges posed by the increased amount of open data that are produced daily by Earth Observation (EO) programs. The unique parallel computing environments and programming techniques that are integrated in High-Performance Computing (HPC) systems are able to solve large-scale problems such as the training of classification algorithms with large amounts of Remote Sensing (RS) data. This paper shows that the training of state-of-the-art deep Convolutional Neural Networks (CNNs) can be efficiently performed in distributed fashion using parallel implementation techniques on HPC machines containing a large number of Graphics Processing Units (GPUs). The experimental results confirm that distributed training can drastically reduce the amount of time needed to perform full training, resulting in near linear scaling without loss of test accuracy.

Keywords: distributed deep learning; high performance computing; residual neural network; convolutional neural network; classification; sentinel-2

1. Introduction

Modern Earth Observation (EO) programs have an open data policy and provide a massive volume of free multisensor data every day. Their systems have substantially advanced in recent decades due to the technological evolution integrated into Remote Sensing (RS) optical and microwave instruments [1]. NASA's Landsat [2] (i.e., the longest running EO program) and ESA's Copernicus [3] provide data with high spectral–spatial coverage at high revisiting time, which enables global monitoring of the Earth in a near real-time manner. Copernicus, with its fleet of Sentinel satellites, is now the World's largest single EO program (https://sentinel.esa.int/web/sentinel/missions). These programs are showing that the vast amount of raw data available call for re-definition of the challenges within the entire RS life cycle (i.e., data acquisition, processing, and application phases). It is not by coincidence that RS data are now described under the big data terminology, with characteristics such as volume (increasing scale of acquired/archived data), velocity (rapidly growing data generation rate and real-time processing needs), variety (data acquired from multiple satellites' sensors that have different spectral, spatial, temporal, and radiometric resolutions), veracity (data uncertainty/accuracy),

and value (extracted information) [4,5]. The Sentinel-2 mission, for instance, has been operating since June 2017 with a constellation of two polar orbiting satellite platforms, which allow a temporal resolution of 5 days at the equator (and even less for areas covered by more than one orbit). Both Sentinel-2A and Sentinel-2B are equipped with a Multispectral (MS) instrument which acquires 13 optical narrow bands in moderate-to-high spatial resolution (10, 20, and 60 m) and generates 23 TB/day of MS data. The freely available imagery from Sentinel-2 received major attention within the research community. From 1 December 2017 to 30 November 2018, the Sentinel Data Access System had a publication rate of over 26,500 products/day with an average daily download volume of 166 TB (https://sentinels.copernicus.eu/web/sentinel/news/-/article/2018-sentinel-data-access-annual-report). The large-scale, high-frequency monitoring of the Earth requires robust and scalable Machine Learning (ML) models trained over annotated (i.e., not raw) time series of multisensor images at global level [6,7] (e.g., acquired by Landsat 8 and Sentinel-2). However, these data do not exist yet. This is largely due to the inherent interpretation complexity of RS data (e.g., hyperspectral and RADAR data) and the effort and cost involved in the collection of training samples. This remains a key limiting factor in the RS community for the research and development of successfully operational Deep Learning (DL) classifiers for RS data.

Nevertheless, DL has already brought crucial achievements in solving RS image classification problems, working on raw multispectral satellite image data [8–10]. The state-of-the-art results have been achieved via deep networks with backbones based on convolutional transformations (e.g., Convolutional Neural Networks (CNNs) [11,12], Recurrent Neural Networks (RNNs) [13], and Generative Adversarial Networks (GANs) [14]). Their hierarchical architecture composed of stacked repetitive operations enables the extraction of useful image features from raw pixel data and modeling high-level semantic content of RS images. However, DL architectures have a much larger number of parameters to estimate than classic ML methods (e.g., shallow classifiers based on handcrafted features) [15]. Thus, their performance and generalization capabilities are considerably dependent on the amount and quality of available training data. That is, to train these networks, a very large annotated training set of sufficient diversity is needed in order to learn effective models.

Table 1 shows the main free annotated remote sensing datasets (i.e., for classification of RGB and MS images) that are currently available for benchmarking DL classifiers. The gap in terms of data size with the computer vision domain (e.g., ImageNet with 14,197,122 images (http://www.image-net.org/)) is still considerably high. Nonetheless, there is an evident trend towards datasets with a higher number of annotated samples and degree of classification complexity (e.g., BigEarthNet [16], a multiclass classification task of 590,326 images). Consequently, the computational intensity and memory demands of DL will continuously increase in the future. In this scenario, approaches relying on local workstation machines (i.e., using MATLAB, R, SAS, SNAP, and ENVI for data analysis and interpretation), can provide only limited capabilities. Despite modern commodity computers and laptops becoming more powerful in terms of multicore configurations and GPUs, the limitations with regard to computational power and memory are always an issue when it comes to fast training of large high-accuracy models from correspondingly large amounts of data. Therefore, the use of highly scalable and parallel distributed architectures (such as clusters [17], grids [18], or clouds [19]) is a necessary solution to train DL classifiers in a reasonable amount of time, which can then also provide users with a high-accuracy performance in the recognition tasks. High-Performance Computing (HPC) systems can reach a performance in the order of petaflops (i.e., 10^{15} floating point operations per second) and are already delivering unprecedented breakthroughs [20]. It is important to observe that ML and DL algorithms have transformed the workloads and workflows that run on these systems, especially when compared to classic HPC simulation problems. DL algorithms require higher memory and networking bandwidth throughput capabilities, as well as optimized software and libraries to deliver the required performance. On the one hand, DL can lead to more accurate classification results of land cover classes when networks are trained over large RS annotated datasets. On the other hand,

deep networks pose challenges in terms of training time. In fact, the use of a large datasets for training a DL model requires the availability of non-negligible time resources.

Table 1. Non-exhaustive list of open remote sensing datasets for image classification.

Datasets	Image Type	Image Per Class	Scene Classes	Annotation Type	Total Images	Spatial Resolution (m)	Image Sizes	Year	Ref.
UC Merced	Aerial RGB	100	21	Single/Multi label	2100	0.3	256 × 256	2010	[21]
WHU-RS19	Aerial RGB	∼50	19	Single label	1005	up to 0.5	600 × 600	2012	[22]
RSSCN7	Aerial RGB	400	7	Single label	2800	–	400 × 400	2015	[23]
SAT-6	Aerial MS	–	6	Single label	405,000	1	28 × 28	2015	[24]
SIRI-WHU	Aerial RGB	200	12	Single label	2400	2	200 × 200	2016	[25]
RSC11	Aerial RGB	100	11	Single label	1323	0.2	512 × 512	2016	[26]
Brazilian Coffee	Satellite MS	1438	2	Single label	2876	–	64 × 64	2016	[27]
RESISC45	Aerial RGB	700	45	Single label	31500	30 to 0.2	256 × 256	2016	[28]
AID	Aerial RGB	∼300	30	Single label	10,000	0.6	600 × 600	2016	[29]
EuroSAT	Satellite MS	∼2500	10	Single label	27,000	10	64 × 64	2017	[30]
RSI-CB128	Aerial RGB	∼800	45	Single label	36,000	0.3 to 3	128 × 128	2017	[6]
RSI-CB256	Aerial RGB	∼690	35	Single label	24,000	0.3 to 3	256 × 256	2017	[6]
PatternNet	Aerial RGB	∼800	38	Single label	30,400	0.062∼4.693	256 × 256	2017	[31]
BigEarthNet	Satellite MS	328 to 217,119	43	Multi label	590,326	10,20,60	120 × 120 60 × 60 20 × 20	2018	[16]

The objective of this contribution is to show that HPC systems speed up the training of DL networks through distributed training frameworks, which can exploit the parallel environment of HPC clusters. The distribution of the model among multiple nodes can considerably speed up the process of training it. This enables deployment of various models and comparison of their performances in a reasonable amount of time. The training of the model is performed via a multimachine data parallelism strategy that allows minimizing the time required to finish full training: The processing is distributed across multiple machines connected by a fast dedicated network (i.e., InfiniBand). This paper proposes a high-performance distributed implementation of the Residual Network (ResNet) [32] type of deep convolutional networks (so-called deep residual networks) for the multiclass RS image classification problem. The experiments are performed with the BigEarthNet [16] dataset over the HPC systems that are based at the Jülich Supercomputing Centre. The experimental results attest that distributed deep neural network training can extremely reduce the amount of time that is required to complete the training step without affecting prediction accuracy.

2. Deep Learning

2.1. The ResNet

ResNet is a deep residual CNN architecture developed by He et al. in [32] to overcome difficulties in training networks with a very large number of layers (>20, up to 1000 layers and more possible [33]), winning the ImageNet competition in 2015 [34]. The first instantiations of deep feed-forward CNNs were the ones providing groundbreaking advances in the field of computer vision on tasks like object detection and object recognition, outperforming previous state-of-the-art ML methods by large margins, e.g., AlexNet with 8 layers [35], VGG with 16 layers [36] or GoogleNet (Inception) with 22 layers [37]. An increasing number of processing layers resulted in further increasing accuracy performance on ImageNet challenges in terms of class recognition rates (the ImageNet-1k challenge has 1000 different object classes that have to be successfully learned during the training on 1.2 Million images [35,38]).

However, simply increasing the number of layers further by stacking more and more convolutional and other layers (pooling, etc) on top of each other was not functionally successful. The training of very deep networks resulted in worse accuracy, contrary to expectations set by previous results. It has been noted that degradation of the training accuracies may be partly caused by a phenomenon known as vanishing (or exploding) gradients. ResNet architecture has been designed to overcome this issue by introducing so-called residual blocks featuring skip connections. These connections implemented an explicit identity mapping for each successor layer in a deep network in addition to the learned operations that were applied to the input before it reaches the next layer [32,33]. The network was thus forced to learn residual mappings corresponding to useful transformations and feature extraction on the image input, while loss gradients could still flow undisturbed during the backward pass via available skip connections through the whole depth of the network. Different ResNet networks were shown to train successfully with a number of layers that was impossible to handle before, while using a smaller number of parameters than previous, less deep architectures (e.g, VGG or Inception networks), thus allowing for faster training.

ResNet-50 (where the number indicates the number of layers) has since then established a strong baseline in terms of accuracy, representing good trade-off between accuracy, depth, and number of parameters, in the same time being very suitable for parallelized, distributed training. As it still remains the strong baseline for object recognition tasks and is also widely used in scenarios for transfer learning ([39–41]), the ResNet-50 architecture is adopted for experiments to show successful distributed training for multiclass, multilabel classification from RS multispectral images.

2.2. Distributed Frameworks

Despite the permanently increasing computational power of Central Processing Unit (CPU)- and Graphics Processing Unit (GPU)-based hardware and essential improvements in efficiency of deep

neural network architectures like ResNet, it remains still a computationally very demanding procedure to train a particular deep neural network to successfully perform a challenging task like object recognition. Even with state-of-the-art hardware like NVIDIAs V100, full training of a ResNet-50 object recognition network on ImageNet-1k dataset of 1.2 Million images using a single GPU can still take more than one day on a single workstation machine (also when taking into account possible acceleration via more efficient mixed-precision (fp16 and fp32) training or special optimized computational graph compilers like TensorFlow's XLA). To conduct a multitude of experiments with various network architectures on large datasets, training therefore constitutes a prohibitively time-expensive procedure.

To overcome these limitations imposed by computationally expensive training, the DL community envisages different methods that enable distributed training across multiple computing nodes of clusters or HPC machines equipped with accelerators like GPUs or highly specialized TPUs [42,43]. Using these methods, it became possible to perform distributed training of large network models without loss of task performance and drastically reduce the amount of time necessary for a complete training. For instance, the time to fully train an object recognition network model on ImageNet-1k (1.2 Millions of images, ca. 80–100 epochs necessary for training to converge) was reduced by orders of magnitude only within a few years from almost one day to few minutes without substantial loss in recognition accuracy [44,45].

This work relies on a certain type of distributed training to conduct scaling experiments and make use of Horovod—a software library that offers a convenient way to execute training and supports TensorFlow and Keras [46]. Using Horovod, only a few modifications in the standard code used for quick single node model prototyping are necessary to adapt it for distributed execution across many nodes.

To enable distributed training, Horovod adapts a data parallel scheme. In the data parallel scheme, it is assumed that a network model to be trained can fit into the memory of a single GPU device. Many so-called workers can be then instantiated during the training, each occupying one available GPU. Each worker contains a clone of the network to train and gets a separate portion of data to train on, so that for each model update iteration, the global data mini-batch is split into different portions that are assigned to each worker. Working on their own portion of the mini-batch, each worker performs a forward pass to compute the network activations and the local loss given their current input, and a backward pass to compute the local gradients.

To keep all the network models across workers in sync, Horovod employs a decentralized, synchronous update strategy based on Ring-AllReduce operations [46,47], where gradients of all workers are collected, averaged, and applied to every clone model network to update their parameter weights. This is in contrast to centralized update strategies that usually require so-called parameter servers (PS) to communicate model parameters to the workers.

However, those implementations rely on TCP/IP internode communication, which is not available on our machines. On the other hand, Horovod relies on operations based on MPI and NCCL libraries, thus being our preferred choice.

The decentralized update makes better use of network topologies connecting the respective machines and thus usually employs a more efficient, homogeneous communication strategy to perform distributed training. On the one hand, the centralized parameter server-based update strategy offers the flexibility to add or remove the workers, which requires only reconfiguration of a parameter server. On the other hand, the decentralized approach may offer higher fault tolerance in terms of not having one weak spot in the communication chain—when a parameter server fails, it is hard to resume training; when a worker node fails, communication in the decentralized approach can still be reconfigured without affecting training, as every other working node possesses a full copy of the model.

For less reliable cluster systems, decentralized updates are therefore a viable option. For robust HPC systems, where note failure is rare, centralized schemes can be a performant choice as well. However, to avoid bottlenecks in communication during large-scale distributed training on HPC,

the setup of many PS is required, which complicates resource allocation, increases complexity of the necessary code, and makes proper training implementation difficult [42]. Thus, using a decentralized update scheme as employed by Horovod is an efficient choice in terms of simplicity and speed for distributed training on HPC.

As a high-level framework at the top of deep learning libraries, Horovod uses well-established MPI CUDA-aware routines and relies on the NCCL library [46,48] for efficient and robust implementation of communication between workers that makes the best out of the available network topology and bandwidth. The choice for Horovod as library for efficient distributed training is also motivated by the ease, clear structure, and transparency of the necessary code modifications. The corresponding strategy can be as well implemented in pure TensorFlow via the distributed strategies framework [49]; however, the effort to rewrite a single node prototype code is still considerably more when compared to modifications required by Horovod. Horovod also supports a unified scheme for using it with other libraries (PyTorch, MxNet), which again minimizes the effort to deal with specific details of each respective framework when implementing distributed training.

Apart from issues regarding efficient communication of information necessary for model updates during distributed training across multiple nodes, there is a further aspect to be dealt with in the algorithmic challenge to perform distributed training. This aspect is rooted in the nature of the optimization procedure that performs actual loss minimization. The majority of the optimization methods used to minimize loss during training are different variations of Stochastic Gradient Descent (SGD). If training has to be distributed across a substantial amount of workers, the effective size of the global mini-batch has to grow. Optimization thus has to cope with mini-batch sizes that are substantially larger that those used for training on a single node. Large mini-batches (for ImageNet, in the order of a few thousand images per batch as compared to the standard mini-batch size of a few hundreds for single node training) lead to substantial degradation of performance, e.g., recognition accuracy, if used without any additional countermeasures [50]. This may be partly due to the very nature of SGD, which requires a certain amount of noise produced by the rather small sizes of mini-batches used for update steps.

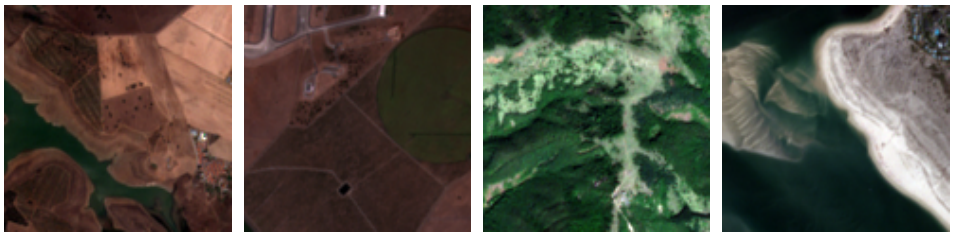
Currently, there are different solutions to secure the same performance level achieved on a single node with small mini-batch sizes despite the essential increase of the effective mini-batch size during distributed training. In the core of the simplest solutions is the tuning of the learning rate schedule that uses warm-up phases before the training, scales the learning rate with the number of distributed workers, and reduces the rate according to a fixed factor after a fixed number of epochs [6,44,50]. More sophisticated strategies to deal with very large batch sizes (for ImageNet, for instance, greater than $2^{13} = 8192$) use adaptive learning rates that are tuned dependent on network layer depth and the value of computed gradients and progress of training, such as that employed in LARS (Layer-wise Adaptive Rate Scaling)—an adaptive optimizer dedicated to large-scale distributed training setting [45,51].

3. Experimental Setup

3.1. Data

The training of the models was carried out using the list of patches provided by BigEarthNet (http://bigearth.net/). BigEarthNet is an archive consisting of 590,326 patches extracted from 125 Sentinel-2 tiles (Level 2A) acquired from June 2017 to May 2018 [16]. A number of labels is associated with each patch. The 43 labels originate from the CORINE Land Cover (CLS) inventory of 2018, available for 10 European countries. According to [16], the number of labels for each patch varies between 1 and 12, being in 95% of the cases at most 5. The patches have 12 spectral bands: (a) the 3 RGB bands and band 8 at 10 m resolution (120 $\times$ 120 pixels), (b) bands 5, 6, 7, 8a, 11, and 12 at 20 m resolution (60 $\times$ 60 pixels), and (c) band 1 and 9 at 60 m resolution (20 $\times$ 20 pixels). Band 10 has been excluded since it is used mainly for cirrus detection [52]. BigEarthNet also provides a list of the patches

with a significant amount of the area covered by snow or clouds, making it possible to exclude them from the analysis [53]. Figure 1 shows an example of the patches.



(**a**) (**b**) (**c**) (**d**)

Figure 1. Example of patches: (**a**) agro-forestry areas, complex cultivation patterns, non-irrigated arable land, transitional woodland/shrub, water bodies, (**b**) airports, olive groves, permanently irrigated land, (**c**) broad-leaved forest, burnt areas, transitional woodland/shrub, (**d**) beaches/dunes/sands, estuaries, sea and ocean, and sport and leisure facilities.

3.2. Environment

The experiments were carried on two HPC sytems installed at the Jülich Supercomputing Centre: the Jülich Wizard for European Leadership Science (JUWELS) [54], and the Jülich Research on Exascale Cluster Architectures (JURECA) [55] supercomputers. In both machines, GPUs partitions were used: JUWELS consists of 46 nodes, with each having four NVIDIA V100 GPUs (with 16 GB of memory each), while JURECA has 75 nodes, each equipped with four NVIDIA K80 GPUs (with 24 GB of memory each). The available benchmark for the experiments relies on a maximum of 24 nodes (i.e., 96 GPUs) for each system.

For the evaluation, the following Python libraries were used: `TensorFlow 1.13.1, Keras 2.2.4, Horovod 0.16.2, Mpi4py 3.0.1 and Scikit-learn 0.20.3`.

In order to upsample the Sentinel-2 bands at a lower resolution to the maximum resolution of 10 m, we use two different upscaling methods. One is based on the super-resolution deep network approach proposed by Lanaras et al. in [56]. Using super-resolved images, we can obtain the same high resolution across different bands. The authors provide a pretrained CNN model (i.e., `DSen2` (https://github.com/lanha/DSen2)) that was trained over a large Sentinel-2 training set which covers a wide range of geographical locations across different climate zones and land-cover types [56]. Another is based on simple standard bilinear interpolation. The simple upscaling is there to check whether there is any advantage in using an advanced super-resolution technique in our case.

The extraction of the patches was carried out with the Geospatial Data Abstraction Library `GDAL 2.3.2` through its Python API. `GDAL` [57] is an open source programming library and set of utilities that facilitates the manipulation of raster data: It helps with data translation from different file formats, data types, and map projections.

3.3. Preprocessing Pipeline

One of the aims of this work is to evaluate models' performance that take Sentinel-2 patches as input, with all the multispectral bands upsampled to the resolution of 10 m for the RGB bands. The original BigEarthNet archive was used as a basis to extract the information for generating a new dataset, one that includes super-resolved patches, as well as the original ones (i.e., publicly available (http://hdl.handle.net/21.11125/921dbc5e-5948-4453-90c0-40b399ffa418)). In order to extract bands at a higher resolution, and to study whether those could help in enhancing the performances of the classification scheme, the DSen2 framework was employed to obtain patches in which the bands originally at a lower resolution (20 and 60 m) were super-resolved: In this way,

all bands become available at the maximum resolution of 10 m. DSen2 consists of two CNNs to perform the trained enhancement of the lower resolution bands into the highest resolution [56].

As shown in Figure 2, the first step in the preprocessing pipeline was to download the freely-available 125 Level 2A tiles from Copernicus Data Hub (https://scihub.copernicus.eu/). After that, the tiles were given as input to DSen2, and in this way, the upsampled tiles were computed. To extract BigEarthNet's original 519,226 patches with a low percentage of snow or cloud coverage, the approach described by the parallel Algorithm 1 was adopted. The algorithm computes the number of patches belonging to each Sentinel-2 tile and creates a matrix with the indices of the tile to be processed by each CPU, in such a way that the total amount of total patches is similar among all processors. With this strategy, idle time is avoided (e.g., a process that already extracted a small number of patches has to wait until other processes to finish their task). The algorithm was executed in parallel using 72 CPUs on JURECA.

The patches were saved in a single Hierarchical Data Format 5 (HDF5) [58] file. This format can be written and read in parallel. It has been organized by associating the data with different keys: "data_super" is the key of the datacube with the 12 upsampled multispectral bands, "data_10m", "data_20m" and "data_60m" stands for datacubes of bands at the original resolution of 10, 20, and 60 m. respectively, and "classes" includes the labels of each patch already binarized.

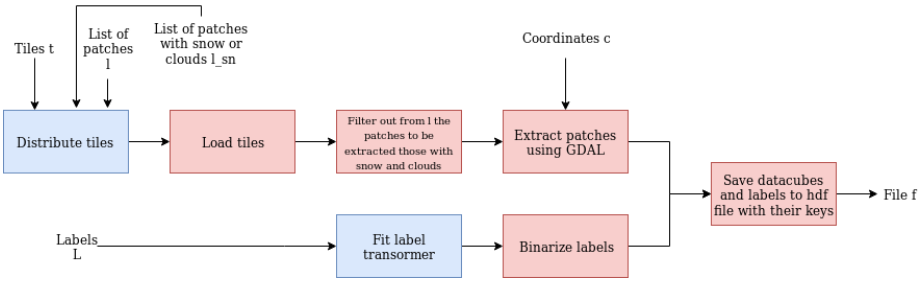


Figure 2. Preprocessing pipeline: extraction of the Sentinel-2 patches and their corresponding classes. Patches covered in snow and clouds are excluded.

Algorithm 1 Distribution of tiles

Input: input parameters n number of CPUs and t tiles

Output: matrix M with indices of tiles per processor

$M \leftarrow range(n_{proc})$
if $n - size(t) > 0$ **then**
  for $i \leftarrow 1$ to $len(t)/n$ **do**
    $arr \leftarrow zeros([n_{proc}])$
    $arr \leftarrow t.values[0 : n_{proc}]$
    $M \leftarrow vstack([M, flip(pad(range((i+1) \times n_{proc}, (i+1) \times n_{proc} + len(b), 1), (0, len(arr) - len(b)))$
else
  $M \leftarrow t$
return M

3.4. Multilabel Classification

Even though in some RS applications, the use of a single label per sample of a scene may be sufficient for a correct classification, there are cases where it might not be sufficient. As stated in [59],

an image of a beach could be correctly classified with a single label, without the need for separated labels such as "sand", "sea", or "buildings". Multilabel classification is defined as that type of classification where classes associated to each sample are not mutually exclusive [60].

According to [59], however, more complex scenarios require finer-grain labeling. For instance, distinguishing between images of urban areas with different building densities would require specific classes, which may also occur in combination with the presence of other classes, such as "road" or "green area". A characteristic of multilabeling classification is, in fact, that the occurrence of some class could be correlated with those of others appearing in similar scenarios.

The standard approach when it comes to the computation of the loss function in the multilabel classification case is the binary cross entropy. A vector of dimension equal to the number of classes is associated to each sample, where every vector cell represents the presence or absence of a specific class. In this way, the problem can be dealt with as a binary classification problem for each of the classes, hence treating them independently. The activation function used for multilabel classification is the sigmoid function, squashing all the elements of the label vector between 0 and 1. This is different from using the softmax activation function, which transforms the probabilities so that they sum up to 1. Instead, using the sigmoid function, it is possible to assure that the labels are not mutually exclusive in the multilabel case, but more than one can be associated to each sample.

3.5. Restricted RGB and Original Multispectral ResNet-50

Two configurations have been considered for the experiments to establish baselines for successful training. They differ according to the data in input: (a) input is limited to three RGB bands only, and (b) input contains 12 multispectral bands. The motivation is, on the one hand, to prepare grounds for transfer learning experiments using ImageNet pretraining on the data that contain RGB channels only. On the other hand, RGB configuration serves as a minimal baseline to check whether a full multispectral input can provide any additional boost for classification performance within standard ResNet architecture.

The classification scheme used in this paper is based on a slightly modified version of ResNet-50. In the present work, some changes to the model have been made to better adapt it to the land cover classification problem. The output layer has been modified to output the prediction probabilities for the 43 CLC classes. The input size has been changed from the original size of 224 $\times$ 224 pixels for each image to the size of the patches (i.e., 120 $\times$ 120 for the 10 m, 60 $\times$ 60 for the 20 m, and 20 $\times$ 20 for the 60 m resolution). Two different kinds of regularization have been adopted to reduce the risk of overfitting: (1) an L2 regularization has been applied to all convolutional layers to penalize large weights, and (2) a dropout with probability equal to 0.5 has been placed before the model's last dense layer.

Two data augmentation techniques were used. The first one is a simple rotation of 90, 180 or 270° and a flip operation, applied randomly to the patches.The second method is called a mix-up and consists in taking a batch and subtracting from it a shuffled version of itself, with a probability drawn from a beta distribution for each patch [61]. The use of these virtual augmented data created with a simple linear combination of the original samples encourages the model to learn smoother decision boundaries, making it more robust when unseen samples are fed into the network. An SGD with Nesterov momentum was selected as an optimizer [62]. The initial learning rate was computed as $\eta = 0.1\frac{kn}{256}$ [50], where k is the number of workers (i.e., GPUs) and n is the batch size for each worker, which in this paper is set to 64. In our work, a step decay learning annealing schedule was used: The actual learning rate was computed multiplying by 0.1 the original learning rate after 30 epochs, by 0.01 after 60 epochs. and by 0.001 after 80 epochs. In our work, we trained the models for a total of 100 epochs. This technique is used to reduce the probability of the model to get stuck in a plateau using a too small learning rate, while on the other hand, a learning rate which is too high may cause an instability in the optimization process [63]. A warm-up of 5 epochs was applied at the start of the training process.

4. Results

4.1. Classification

The classification results are presented for the RGB and the multispectral models. Both models were adapted to the problem of multilabel classification from the original ResNet-50 [32]. For the performance metric of the experiment, we employed the F1 score, which is widely used for multiabel image classification problems. In Tables 2 and 3, the prediction results for a single experiment performed over 1 node of JUWELS (i.e., 4 NVIDIA V100 GPUs) are reported. For this proposed ResNet-50 architecture, the model trained on RGB bands performs almost as well as the multispectral model (see Table 2 that shows the global scores). The prediction scores of each individual class are reported by Table 3. It can be seen that some classes have a very high F1 score: e.g., the class "Sea and ocean" has a high F1 score. This is not surprising due to to the specific distinguishable spectral signature of water. For the same reason, the class "Coastal lagoons" is also easily detected by the model, despite heavy imbalance—this class has a much smaller number of samples compared, for instance, to "Sea and ocean".

Table 2. Classification results for the RGB and multispectral model: P precision, R recall and F1 score.

	P	R	F1
RGB	0.82	0.71	0.77
multispectral	0.83	0.75	0.79

Table 3. Classification results of each class for the RGB and multispectral model: F1 score and support for each class considering the test set.

	Support	F1 (Multispectral)	F1 (RGB)
Agro-forestry areas	5611	0.803621	0.795872
Airports	157	0.300518	0.374384
Annual crops associated with permanent crops	1275	0.457738	0.442318
Bare rock	511	0.604819	0.620192
Beaches, dunes, sands	319	0.695810	0.608964
Broad-leaved forest	28,090	0.791465	0.771761
Burnt areas	66	0.029851	0
Coastal lagoons	287	0.884758	0.880294
Complex cultivation patterns	21,142	0.722448	0.698238
Coniferous forest	33,583	0.874152	0.866716
Construction sites	244	0.234482	0.213058
Continuous urban fabric	1975	0.784672	0.517737
Discontinuous urban fabric	13,338	0.780262	0.722825
Dump sites	181	0.287037	0.268518
Estuaries	197	0.699088	0.585034
Fruit trees and berry plantations	875	0.452648	0.417887
Green urban areas	338	0.387750	0.369477
Industrial or commercial units	2417	0.552506	0.556856
Inland marshes	1142	0.408505	0.364675
Intertidal flats	216	0.635097	0.584126
Land principally occupied by agriculture	26,447	0.686677	0.667633
Mineral extraction sites	835	0.507598	0.490980
Mixed forest	35,975	0.834221	0.797793
Moors and heathland	1060	0.561134	0.430953
Natural grassland	2273	0.569581	0.512231
Non-irrigated arable land	36,562	0.865387	0.839924
Olive groves	2372	0.621071	0.541914

Table 3. *Cont.*

	Support	F1 (Multispectral)	F1 (RGB)
Pastures	20,770	0.780565	0.771802
Peatbogs	3411	0.535477	0.690319
Permanently irrigated land	2505	0.675662	0.643835
Port areas	93	0.503597	0.522388
Rice fields	709	0.669542	0.604770
Road and rail networks and associated land	671	0.300785	0.268623
Salines	75	0.608000	0.517857
Salt marshes	264	0.568578	0.532299
Sclerophyllous vegetation	2114	0.762123	0.671300
Sea and ocean	13,964	0.909013	0.979917
Sparsely vegetated areas	261	0.483460	0.380681
Sport and leisure facilities	996	0.367029	0.406827
Transitional woodland/shrub	29,671	0.664189	0.639412
Vineyards	1821	0.564012	0.545454
Water bodies	11,545	0.858107	0.823858
Water courses	1914	0.803948	0.737060

4.2. Processing Time

The processing times of the JURECA and JUWELS systems are reported only for the multispectral model. Due to the limited amount of computing time (i.e., core hours) allocated for this project, each experiment has been run only twice. Figures 3 and 4 report the mean and standard deviation values. It can be observed that the training time using two nodes (i.e., 8 GPUs) is half (172 s for an epoch on JUWELS) of the time required to execute the same training with one node (i.e., 4 GPUs) (347 s). The same can be said in the cases where 2 vs. 4 (172 s vs. 86 s), 4 vs. 8 (86 s vs. 42 s) and 8 vs. 16 (42 s vs. 20 s) nodes are considered. However, the scaling between 12 and 24 nodes seems to be less than linear (27 s vs. 15 s).

The use of this distribution approach has allowed us to reduce the total time for a full training on JUWELS from almost 35,000 s using 4 GPUs on 1 node to less than 2500 s using 96 GPUs on 24 nodes. The results on JURECA shown in Figure 4 confirm this observation. Although it can be seen that the full run on JURECA (on 2 nodes approximately 14 h, as can be seen in Figure 5) takes almost 3 times more time than those run on JUWELS (on 2 nodes in less than 5 h) due to the available GPUs (K80 vs. V100), on the other hand, taking advantage of this parallelization framework has enabled the full training of the model using older GPUs in a reasonable amount of time.

5. Discussion

The class imbalance poses a serious caveat on the performances of the models. In fact, it can be observed that there are classes which are heavily under-represented compared to others—e.g., in the test subset considered for this work, there are more than 30,000 patches associated with the label "Coniferous forest" but just 93 with label "Port areas". Thus, it comes as no surprise that the F1 score obtained for the classes with a low support (i.e., low number of samples) is on average less than the F1 score of the most populated classes of the dataset, since it is known that class imbalance can cause a bias towards the majority class [64]. As reported in Section 3.5 in this work, two simple data augmentation techniques were applied. However, the problem of class imbalance may require the use of of different techniques, e.g., upsampling of the under-represented samples [65] or loss weighting to let the model give more importance to samples associated with classes present in a lesser amount [64]. These methods should be implemented and tested in future work.W Another limitation that stems from the imbalance problem is that the spectral signature (i.e., the radiation reflected by the surface as a function of the wavelength) of areas associated with some classes could change over time, causing low classification results. That may be the case for the class "Burnt areas" (an example in Figure 1c), showing a very low F1 score. An approach to deal with such a class could be the adoption

of a multitemporal analysis, implementing, for instance, a change detection method to identify when a significant change in the spectral signature of a patch (such as the one caused by a fire) occurs. Moreover, CLS classes may be semantically too stringent for the purpose of classification of land cover using optical data alone. As an example, CLS has two different classes for "Discontinuous urban fabric" and "Green urban areas", which may represent patches with a similar information content. One last point which could be considered is the fact that the presence of some classes may be correlated with those of another one. For instance, it is reasonable to assume that "Beaches, dunes, sands" is correlated with the presence of classes associated with water, as can be observed in Figure 1d, or that a cultivation pattern is present at the same time of arable land as in in Figure 1a. In this work, it has not been used a method to explicitly take this information into account, as, e.g., it was done in [16], where the local descriptors generated by a CNN were then updated using an LSTM network on subtiles of the patches.

In Section 3.2, we stated that our work makes use of DSen2 to upsample the patches to the same resolution of 10 m across the different bands. We used DSen2 since it is a well-established method for super-resolution. However, an experiment in which we used a simple bilinear interpolation, run on 8 nodes on JUWELS, showed a very similar F1 score to those obtained using DSen2 (shown in Figure 6). Further studies should be conducted to investigate whether different DL models could take advantage of the enhanced spectral characteristics provided by DSen2.

Section 4.1 mentions that the model trained on RGB bands obtains a slightly lower average F1 score to the one achieved by the multispectral model. However, for the class airports, bare rock, peatbogs, port areas, sea ocean, and sport and leisure facilities, the F1 score of RGB is higher. For these classes, the model that is trained with the multispectral data is not able to isolate the RGB information from the other bands. Generally, a correct network architecture should deliver at least the same classification results (since multispectral data include the same RGB bands). As we explain in Section 2.1, we selected the ResNet-50 since it is a well-established baseline architecture in terms of accuracy, represents a good trade-off between depth and number of parameters, and is very suitable for parallelization. According to the current results, we established that ResNet-50 is not suitable to deal properly with the information provided by all the multispectral bands. However, a more detailed study (i.e., out of the scope of this work) should be conducted by considering different experimental classification settings (e.g., compare the classification result obtained with one band against RGB).

As has been stated at the introduction of this paper, DL poses challenging questions in terms of time required for the training of a model due to the large number of parameters. The results presented in Section 4.2, confirmed that the Horovod distributed training framework enables the achievement of near linear scaling. However, when dealing with distributed training, the consistency of the classification results has to be constantly monitored. The reason is that when the size of the batch is increased (defined as $b_e = b_g \times k$, where b_e is the effective batch size, b_g is the batch size per GPU, and k is the number of GPUs) a degradation of the accuracy often occurs. At first glance in Figure 6, a slow trend of a decrease in the fscore is apparent when a larger number of nodes is employed. The results obtained using JUWELS are confirmed also by those from JURECA (please note that the fscore of 1 node is not reported, since the computation time has exceeded the limit of the system). Without further special mechanisms, stable training with SGD is possible only for a total batch size of $<$8192 [66]. During training, an explosion of the loss during the first epochs with a high learning rate was typically observed, which does not occur at a more advanced stage of the training when a lower learning rate is used. This phenomenon is particularly noticeable when a large number of nodes is used. The initial learning rate is in fact dependent on the number of nodes in the formula shown in Section 3.5. As a direct consequence, if a large number of nodes is used, the initial learning rate is large. The step decay learning rate scheduler used in the present work is the one defined by Goyal et al. [50]; however, different schemes such as the polynomial decay scheduler could be employed to make the loss less prone to explosion during the training process. The use of different types of optimizers could as well be studied further in detail as a workaround to overcome this known problem.

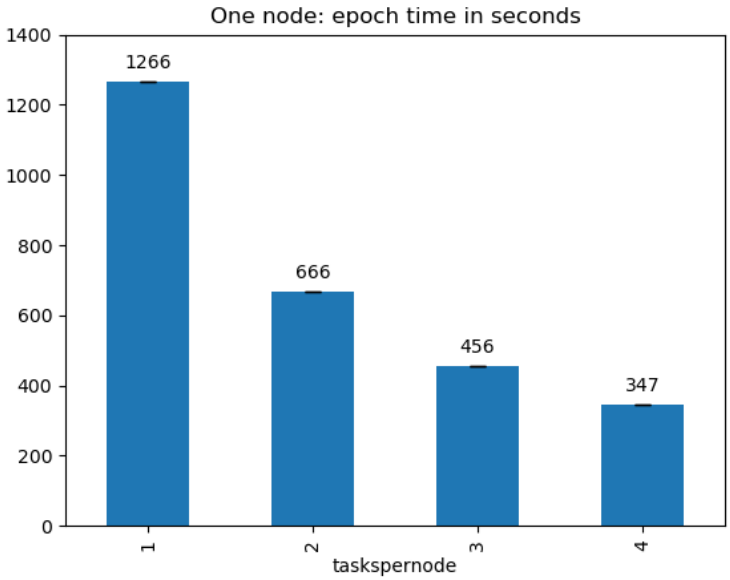


Figure 3. JUWELS: One node, 1, 2, 3 and 4 GPUs, time per epoch, multispectral model.

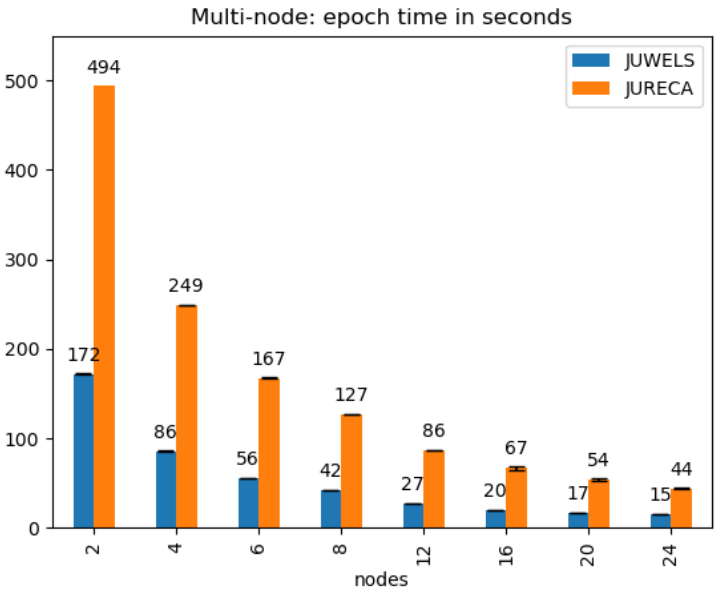


Figure 4. Multinode, time per epoch, multispectral model.

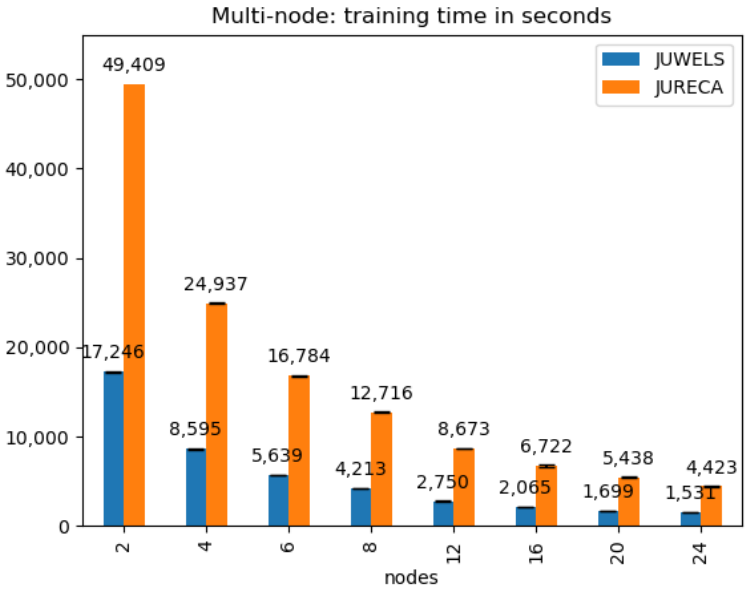


Figure 5. Training time, multispectral model.

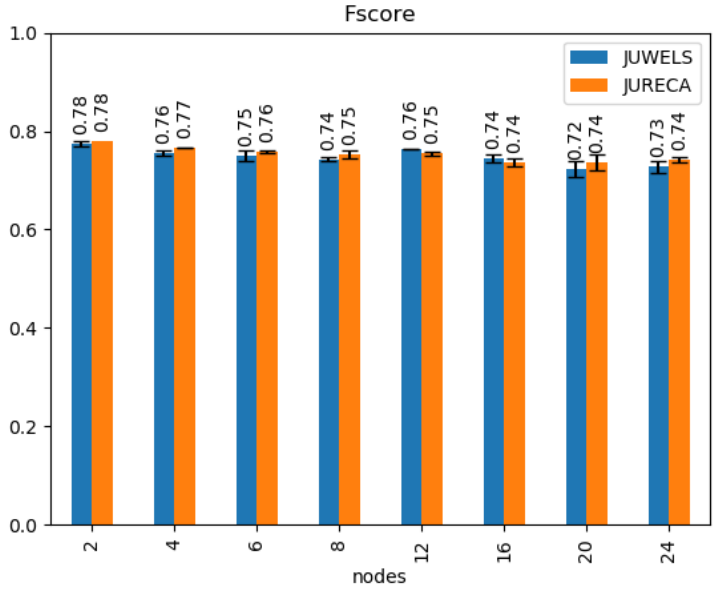


Figure 6. Fscore, multispectral model.

6. Conclusions

Large-scale deep neural networks have millions of weights and require large amounts of data to optimize these parameters to converge to a satisfactory testing accuracy. With the size of the learning networks and annotated remote sensing datasets growing, it becomes possible to automatically extract useful features and representations suitable for high-accuracy classification tasks, but at the cost of higher computation time necessary for the full training. The experimental results of this paper confirm that distributed training over HPC systems can drastically reduce the amount of time needed to complete the training step, resulting in near linear scaling without significant loss of test accuracy. The publication of this paper includes the availability of the dataset and the Python implementation of the models (https://gitlab.com/rocco.sedona/mdpi-paper-bigearth).

Author Contributions: Data curation, R.S. and G.C.; Conceptualization, investigation, formal analysis, methodology, writing—original draft preparation, R.S., G.C., and J.J.; Experiment adjustment, R.S., G.C., J.J., and A.S; Supervision, writing—review and editing, G.C., J.J., and A.S.; Project administration and funding acquisition, M.R. and J.A.B.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EO	Earth Observation
RS	Remote Sensing
DL	Deep Learning
ML	Machine Learning
HPC	High-Performance Computing
MPI	Message Passing Interface
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GAN	Generative Adversarial Network
MS	Multispectral
ResNet	Residual Network
JUWELS	Jülich Wizard for European Leadership Science
JURECA	Jülich Research on Exascale Cluster Architectures
GPU	Graphics Processing Unit
CPU	Central Processing Unit
SGD	Stochastic Gradient Descent
CLS	CORINE Land Cover

References

1. Emery, W.; Camps, A. Basic Electromagnetic Concepts and Applications to Optical Sensors. In *Introduction to Satellite Remote Sensing*; Elsevier: Amsterdam, the Netherlands, 2017. [CrossRef]
2. Wulder, M.A.; Masek, J.G.; Cohen, W.B.; Loveland, T.R.; Woodcock, C.E. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote. Sens. Environ.* **2012**. [CrossRef]
3. Aschbacher, J. ESA's earth observation strategy and Copernicus. In *Satellite Earth Observations and Their Impact on Society and Policy*; Springer: Singapore, 2017. [CrossRef]
4. Ma, Y.; Wu, H.; Wang, L.; Huang, B.; Zomaya, A.; Jie, W. Remote sensing big data computing: Challenges and opportunities. *Future Gener. Comput. Syst.* **2015**, *51*, 47–60. [CrossRef]
5. Chi, M.; Plaza, A.; Benediktsson, J.A.; Sun, Z.; Shen, J.; Zhu, Y. Big Data for Remote Sensing: Challenges and Opportunities. *Proc. IEEE* **2016**. [CrossRef]
6. Li, H.; Tao, C.; Wu, Z.; Chen, J.; Gong, J.; Deng, M. RSI-CB: Large Scale Remote. Sens. Image Classif. Benchmark Via Crowdsource Data. *arXiv* **2017**, arXiv:1705.10450.

7. Stoian, A.; Poulain, V.; Inglada, J.; Poughon, V.; Derksen, D. Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems. *Remote. Sens.* **2019**, *11*, 1986. [CrossRef]
8. Ball, J.E.; Anderson, D.T.; Chan, C.S. A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community. *SPIE J. Appl. Remote. Sens. (JARS) Spec. Sect. Feature Deep. Learn. Remote. Sens. Appl.* **2017**, *11*, 042609. [CrossRef]
9. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote. Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
10. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *152*, 166–177. [CrossRef]
11. Romero, A.; Gatta, C.; Camps-valls, G.; Member, S. Unsupervised Deep Feature Extraction for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2015**, *54*, 1–14 [CrossRef]
12. Zhang, L.; Zhang, L.; Du, B. Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art. *IEEE Geosci. Remote. Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
13. Ienco, D.; Gaetano, R.; Dupaquier, C.; Maurel, P. Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks. *IEEE Geosci. Remote. Sens. Lett.* **2017**. [CrossRef]
14. Lin, D.; Fu, K.; Wang, Y.; Xu, G.; Sun, X. MARTA GANs: Unsupervised Representation Learning for Remote Sensing Image Classification. *IEEE Geosci. Remote. Sens. Lett.* **2017**. [CrossRef]
15. Cavallaro, G.; Falco, N.; Dalla Mura, M.; Benediktsson, J.A. Automatic Attribute Profiles. *IEEE Trans. Image Process.* **2017**. [CrossRef] [PubMed]
16. Sumbul, G.; Charfuelan, M.; Demir, B.; Markl, V. Bigearthnet: Large-Scale Benchmark Arch. Remote. Sens. Image Underst. *arXiv* **2019**, arXiv:1902.06148.
17. Plaza, A.; Valencia, D.; Plaza, J.; Martinez, P. Commodity cluster-based parallel processing of hyperspectral imagery. *J. Parallel Distrib. Comput.* **2006**. [CrossRef]
18. Gorgan, D.; Bacu, V.; Stefanut, T.; Rodila, D.; Mihon, D. Grid based satellite image processing platform for Earth Observation application development. In Proceedings of the 5th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2009, Rende, Italy, 21–23 September 2009. [CrossRef]
19. Foster, I.; Zhao, Y.; Raicu, I.; Lu, S. Cloud computing and grid computing 360-degree compared. In Proceedings of the Grid Comput. Environ. Workshop 2008 (GCE '08) Austin, TX, USA, 12–16 November 2008; pp. 1–10. [CrossRef]
20. McKinney, R.; Pallipuram, V.K.; Vargas, R.; Taufer, M. From HPC performance to climate modeling: Transforming methods for HPC predictions into models of extreme climate conditions. In Proceedings of the 11th IEEE International Conference on eScience, eScience 2015, Munich, Germany, 31 August–4 September 2015. [CrossRef]
21. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010. [CrossRef]
22. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Trans. Geosci. Remote. Sens.* **2013**. [CrossRef]
23. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep Learning Based Feature Selection for Remote Sensing Scene Classification. *IEEE Geosci. Remote. Sens. Lett.* **2015**. [CrossRef]
24. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; DiBiano, R.; Karki, M.; Nemani, R. DeepSat—A learning framework for satellite imagery. In Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, Seattle, WD, USA, 3–6 November 2015. [CrossRef]
25. Zhao, B.; Zhong, Y.; Xia, G.S.; Zhang, L. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote. Sens.* **2016**. [CrossRef]
26. Zhao, L.; Tang, P.; Huo, L. Feature significance-based multibag-of-visual-words model for remote sensing image scene classification. *J. Appl. Remote. Sens.* **2016**, *10*, 1–21. [CrossRef]
27. Penatti, O.A.; Nogueira, K.; Dos Santos, J.A. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015. [CrossRef]

28. Cheng, G.; Han, J.; Lu, X. Remote. Sens. Image Scene Classif. Benchmark State Art. *Proc. IEEE* **2017**, 1865–1883. [CrossRef]
29. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote. Sens.* **2017**. [CrossRef]
30. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *Comput. Res. Repos.* **2017**. [CrossRef]
31. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS J. Photogramm. Remote. Sens.* **2018**. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vegas, NV, USA, 27–30 June 2016. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
34. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; others. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, NIPS Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, NV, USA, 3–6 December 2012*; The MIT Press: London, UK, 2012; pp. 1097–1105.
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
37. Szegedy, C.; Wei, L.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
38. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Li, F.-F.. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
39. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In Proceedings of the International Conference on Machine Learning, Bejing, China, 22–24 June 2014; pp. 647–655.
40. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 512–519. [CrossRef]
41. Kornblith, S.; Shlens, J.; Le, Q.V. Do better imagenet models transfer better? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Beach, CA, USA, 16–20 Junary 2019; pp. 2661–2671.
42. Mayer, R.; Jacobsen, H.A. Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques and Tools. *arXiv* **2019**, arXiv:1903.11314.
43. You, Y.; Zhang, Z.; Hsieh, C.; Demmel, J.; Keutzer, K. Fast Deep Neural Network Training on Distributed Systems and Cloud TPUs. *IEEE Trans. Parallel Distrib. Syst.* **2019**. [CrossRef]
44. Ying, C.; Kumar, S.; Chen, D.; Wang, T.; Cheng, Y. Image classification at supercomputer scale. *arXiv* **2018**, arXiv:1811.06992.
45. Yamazaki, M.; Kasagi, A.; Tabuchi, A.; Honda, T.; Miwa, M.; Fukumoto, N.; Tabaru, T.; Ike, A.; Nakashima, K. Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds. *arXiv* **2019**, arXiv:1903.12650.
46. Sergeev, A.; Balso, M.D. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv* **2018**, arXiv:1802.05799.
47. Gibiansky, A. Bringing HPC Techniques to Deep Learning. Available online: http://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/ (accessed on 1 April 2019).
48. NVIDIA Collective Communications Library (NCCL). Available online: https://developer.nvidia.com/nccl (accessed on 15 October 2019).
49. TensorFlow Distributed Strategy Documentation. Available online: https://www.tensorflow.org/guide/distributed_training (accessed on 1 April 2019).

50. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv* **2017**, arXiv:1706.02677.
51. You, Y.; Gitman, I.; Ginsburg, B. Scaling sgd batch size to 32k for imagenet training. *arXiv* **2017**, arXiv:1708.03888.
52. Sentinel2 B10: High Atmospheric Absorption Band. Available online: https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-1c/cloud-masks (accessed on 15 May 2019).
53. Scripts to Remove Cloudy and Snowy Patches Provided by BigEarthNet Archive Creators from the Remote Sensing Image Analysis (RSiM) Group at the TU Berlin. Available online: http://bigearth.net/ (accessed on 1 April 2019).
54. Jülich Supercomputing Centre. JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre. *J. Large-Scale Res. Facil.* **2019**, *5*. [CrossRef]
55. Jülich Supercomputing Centre. JURECA: Modular supercomputer at Jülich Supercomputing Centre. *J. Large-Scale Res. Facil.* **2018**, *4*. [CrossRef]
56. Lanaras, C.; Bioucas-Dias, J.; Galliani, S.; Baltsavias, E.; Schindler, K. Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS J. Photogramm. Remote. Sens.* **2018**. [CrossRef]
57. GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction Software Library*; Open Source Geospatial Foundation: Chicago, IL, USA, 2019.
58. The HDF Group. Hierarchical Data Format, Version 5, 1997-NNNN. Available online: http://www.hdfgroup.org/HDF5/ (accessed on 1 April 2019) .
59. Chaudhuri, B.; Demir, B.; Chaudhuri, S.; Bruzzone, L. Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method. *IEEE Trans. Geosci. Remote. Sens.* **2018**, *56*, 1144–1158. [CrossRef]
60. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier Chains for Multi-label Classification. In *Machine Learning and Knowledge Discovery in Databases*; Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 254–269.
61. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv* **2017**, arXiv:1710.09412.
62. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013.
63. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2012**. [CrossRef]
64. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**. [CrossRef]
65. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**. [CrossRef]
66. Parikh, N. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *Found. Trends Optim.* **2014**. [CrossRef]

Atmos. Meas. Tech., 13, 3661–3682, 2020
https://doi.org/10.5194/amt-13-3661-2020

Atmospheric Measurement Techniques
Open Access
EGU

Exploration of machine learning methods for the classification of infrared limb spectra of polar stratospheric clouds

Rocco Sedona[1,2], Lars Hoffmann[1], Reinhold Spang[3], Gabriele Cavallaro[1], Sabine Griessbach[1], Michael Höpfner[4], Matthias Book[2], and Morris Riedel[1,2]

[1]Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, Jülich, Germany
[2]School of Engineering and Natural Sciences, University of Iceland, Reykjavík, Iceland
[3]Institut für Energie- und Klimaforschung (IEK-7), Forschungszentrum Jülich, Jülich, Germany
[4]Institut für Meteorlogie und Klimaforschung, Karlsruher Institut für Technologie, Karlsruhe, Germany

Correspondence: Rocco Sedona (r.sedona@fz-juelich.de)

Received: 10 December 2019 – Discussion started: 30 January 2020
Revised: 20 May 2020 – Accepted: 15 June 2020 – Published: 8 July 2020

Abstract. Polar stratospheric clouds (PSCs) play a key role in polar ozone depletion in the stratosphere. Improved observations and continuous monitoring of PSCs can help to validate and improve chemistry–climate models that are used to predict the evolution of the polar ozone hole. In this paper, we explore the potential of applying machine learning (ML) methods to classify PSC observations of infrared limb sounders. Two datasets were considered in this study. The first dataset is a collection of infrared spectra captured in Northern Hemisphere winter 2006/2007 and Southern Hemisphere winter 2009 by the Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) instrument on board the European Space Agency's (ESA) Envisat satellite. The second dataset is the cloud scenario database (CSDB) of simulated MIPAS spectra. We first performed an initial analysis to assess the basic characteristics of the CSDB and to decide which features to extract from it. Here, we focused on an approach using brightness temperature differences (BTDs). From both the measured and the simulated infrared spectra, more than 10 000 BTD features were generated. Next, we assessed the use of ML methods for the reduction of the dimensionality of this large feature space using principal component analysis (PCA) and kernel principal component analysis (KPCA) followed by a classification with the support vector machine (SVM). The random forest (RF) technique, which embeds the feature selection step, has also been used as a classifier. All methods were found to be suitable to retrieve information on the composition of PSCs. Of these, RF seems to be the most promising method, being less prone to overfitting and producing results that agree well with established results based on conventional classification methods.

1 Introduction

Polar stratospheric clouds (PSCs) typically form in the polar winter stratosphere between 15 and 30 km of altitude. PSCs can be observed only at high latitudes, as they exist only at very low temperatures ($T < 195$ K) found in the polar vortices. PSCs are known to play an important role in ozone depletion caused by heterogeneous reactions under cold conditions, while denitrification of the stratosphere extends the ozone destruction cycles into springtime, as the absence of NO_y limits the deactivation process of the reactive ozone-destroying substances (Solomon, 1999; Salawitch et al., 1993). The presence of man-made chlorofluorocarbons (CFCs) in the stratosphere, which have been used for example in industrial compounds present in refrigerants, solvents, blowing agents for plastic foam, affects ozone depletion. CFCs are inert compounds in the troposphere but get transformed under stratospheric conditions to the chlorine reservoir gases HCl and $ClONO_2$. PSC particles are involved in the release of chlorine from the reservoirs.

The main constituents of PSCs are three, i.e., nitric acid trihydrate (NAT), supercooled ternary solution (STS), and ice (Lowe and MacKenzie, 2008). Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) measurements have been used to study PSC processes (Arnone et al., 2012;

Khosrawi et al., 2018; Tritscher et al., 2019). The infrared spectra acquired by MIPAS are rather sensitive to optically thin clouds due to the limb observation geometry. This is particularly interesting for NAT and STS PSCs, as ice PSCs are in general optically thicker than NAT and STS (Fromm et al., 2003). As ice clouds form at a lower temperature than NAT and STS, they are mainly present in the Antarctic, while their presence in the Arctic (where the stratospheric temperature minimum in polar winter is higher) is only notable for extremely cold winter conditions (e.g., Campbell and Sassen, 2008; Pawson et al., 1995).

Besides using MIPAS measurements, classification has been carried out with different schemes based on the optical properties of PSCs by lidar measurements. A review of those methods is available in Achtert and Tesche (2014). Classification schemes are based on two features, namely the backscatter ratio and the depolarization ratio. As exposed in Biele et al. (2001), particles with large backscatter ratio and depolarization are likely to be composed of ice (type II). Type I particles are characterized by a low backscatter ratio. The subtype Ia particles show a large depolarization and are composed of NAT, whereas subtype Ib particles have low depolarization and consist of STS. The threshold to classify the PSC types varies among different works such as Browell et al. (1990), Toon et al. (1990), Adriani (2004), Pitts et al. (2009), and Pitts et al. (2011). The nomenclature presented above is a simplification of real case scenarios, since PSCs can occur also with mixtures of particles with different composition (Pitts et al., 2009). Other methods that are used to measure PSCs are in situ optical and nonoptical measurements from balloon or aircraft as well as microwave observations (Buontempo et al., 2009; Molleker et al., 2014; Voigt, 2000; Lambert et al., 2012).

The use of machine learning (ML) algorithms increased dramatically during the last decade. ML can offer valuable tools to deal with a variety of problems. In this paper, we used ML methods for two different tasks: first, for the selection of informative features from the simulated MIPAS spectra; second, to classify the MIPAS spectra depending on the composition of the PSC. In this work we significantly extended the application of ML methods for the analysis of MIPAS PSC observations. Standard methods that exploit infrared limb observation to classify PSCs are based on empirical approaches. Given physical knowledge of the properties of the PSC, some features have been extracted from the spectra, for example the ratio of the radiances between specific spectral windows. These approaches have been proven to be capable of detecting and discriminating between different PSC classes (Spang et al., 2004; Höpfner et al., 2006).

The purpose of this study is to explore the use of ML methods to improve the PSC classification for infrared limb satellite measurements and to potentially gain more knowledge on the impact of the different PSC classes on the spectra. We compare results from the most advanced empirical method, the Bayesian classifier of Spang et al. (2016), with three automatic approaches. The first one relies on principal component analysis (PCA) and kernel principal component analysis (KPCA) for feature extraction, followed by classification with the support vector machine (SVM). The second one is similar to the first, but uses kernel principal component analysis (KPCA) for feature extraction instead of PCA. The third one is based on the random forest (RF), a classifier that directly embeds a feature selection (Cortes and Vapnik, 1995; Breiman, 2001; Jolliffe and Cadima, 2016). A common problem of ML is the lack of annotated data. To overcome this limitation, we used a synthetic dataset for training and testing, the cloud scenario database (CSDB), especially developed for MIPAS cloud and PSC analyses (Spang et al., 2012). As a ground truth for PSC classification is largely missing, we evaluate the ML results by comparing them with results from existing methods and show that they are consistent with established scientific knowledge.

In Sect. 2, we introduce the MIPAS and synthetic CSDB datasets. A brief description of the ML methods used for feature reduction and classification is provided in Sect. 3. In Sect. 4, we compare results of PCA + SVM, KPCA + SVM, and RF for feature selection and classification. We present three case studies and statistical analyses for the 2006/2007 Arctic and 2009 Antarctic winter season. The final discussion and conclusions are given in Sect. 5.

2 Data

2.1 MIPAS

The MIPAS instrument (Fischer et al., 2008) was an infrared limb emission spectrometer on board the European Space Agency's (ESA) Envisat satellite to study the thermal emission of the Earth's atmosphere constituents. Envisat operated from July 2002 to April 2012 in a polar low Earth orbit with a repeat cycle of 35 d. MIPAS measured up to 87° S and 89° N latitude and therefore provided nearly global coverage at day- and nighttime. The number of orbits of the satellite per day was equal to 14.3, resulting in a total of about 1000 limb scans per day.

The wavelength range covered by the MIPAS interferometer was about 4 to 15 µm. From the beginning of the mission to spring 2004, the instrument operated in the full resolution (FR) mode (0.025 cm^{-1} spectral sampling). Later on, this has to be changed to the optimized resolution (OR) mode (0.0625 cm^{-1}) due to a technical problem of the interferometer (Raspollini et al., 2006, 2013). The FR measurements were taken with a constant 3 km vertical and 550 km horizontal spacing, while for the OR measurements the vertical sampling depended on altitude, varying from 1.5 to 4.5 km, and a horizontal spacing of 420 km was achieved. The altitude range of the FR and OR measurements varied from 5–70 km at the poles to 12–77 km at the Equator.

Table 1. Infrared spectral regions considered for PSC classification.

Spectral region	Index range	Wavenumber range (cm^{-1})
R1	0–57	782–840
R2	58–83	940–965
R3	84–98	1224–1250
R4	99–106	1404–1412
R5	107–112	1930–1935
R6	113–125	1972–1985
R7	126–130	2001–2006
R8	131–136	2140–2146
W1	137	788.2–796.2
W2	138	832–834.4
W3	139	819–821
W4	140	832.3–834.4
W5	141	947.5–950

For our analyses, we used MIPAS Level 1B data (version 7.11) acquired at 15–30 km of altitude between May and September 2009 at 60–90° S and between November 2006 and February 2007 at 60–90° N. The 2009 Southern Hemisphere winter presents a slightly higher than average PSC activity, especially for ice in June and August. The 2006/2007 Northern Hemisphere winter is characterized by a large area covered by NAT, with an exception made for early January, and some ice is present in late December (this analysis was obtained from NASA Ozone Watch from their website at https://ozonewatch.gsfc.nasa.gov, last access: 20 April 2020). The high-resolution MIPAS spectra were averaged to obtain 136 spectral windows of 1 cm^{-1} width, because PSC particles are expected to typically cause only broader-scale features. The 1 cm^{-1} window data used in this study comprise the eight spectral regions reported in Table 1. In addition to these, five windows (W1–W5) larger than 1 cm^{-1} have been considered, as used in the study of Spang et al. (2016).

From the 1 cm^{-1} windows and the five additional larger windows, more than 10 000 brightness temperature differences (BTDs) were extracted using a two-step preprocessing. At first, the infrared spectra were converted from radiance intensities to brightness temperatures (BTs). This approach is considered helpful, as variations in the signals are more linear in BT compared to radiances. Then, the BTDs were computed by subtracting the BT of each window with respect to the remaining ones. The main motivation for using BTDs rather than BTs for classification is to try to remove background signals from interfering instrument effects such as radiometric offsets.

Other wavelength ranges covered by MIPAS have been excluded here as they are mainly sensitive to the presence of trace gases. The interference of cloud and trace gas emissions makes it more difficult to analyze the effects of the PSC particles (Spang et al., 2016). As an example, Fig. 1 shows

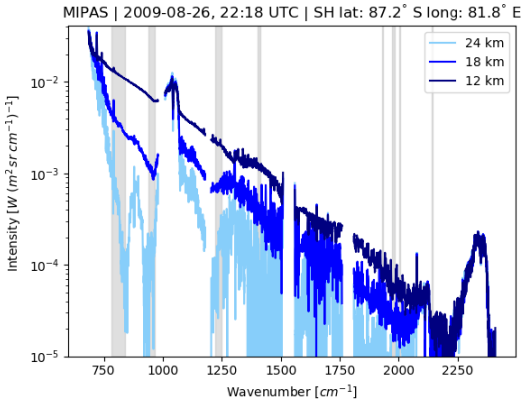


Figure 1. MIPAS measurements in Southern Hemisphere polar winter at three tangent altitudes from the same profile showing clear-air (light blue), optically thin (blue), and optically thick (dark blue) conditions. The gray bars indicate the wavenumber regions considered for PSC classification in this study.

Table 2. PSC constituents, particle concentrations, and sizes covered by the CSDB.

PSC constituents	Volume density ($\mu m^3\ cm^{-3}$)	Median radius (μm)
ice	10, 50, 100	1.0, 2.0, 3.0, 4.0, 5.0, 10.0
NAT	0.1, 0.5, 1.0, 5.0, 10.0	0.5, 1.0, 2.0, 3.0, 4.0, 5.0
STS	0.1, 0.5, 1.0, 5.0, 10.0	0.1, 0.5, 1.0

MIPAS spectra of PSC observations acquired in late August 2009 in Southern Hemisphere polar winter conditions, with the spectral regions used for PSC detection and classification being highlighted.

2.2 Cloud scenario database

A synthetic dataset consisting of simulated radiances for the MIPAS instrument provides the training and testing data for this study. The CSDB was generated by considering more than 70 000 different cloud scenarios (Spang et al., 2012). The CSDB spectra were generated using the Karlsruhe Optimized and Precise Radiative Transfer Algorithm (KOPRA) model (Stiller et al., 1998). Limb spectra were simulated from 12 to 30 km tangent height, with 1 km vertical spacing. Cloud top heights were varied between 12.5 and 28.5 km, with 0.5 km vertical spacing. The cloud vertical extent varies between 0.5, 1, 2, 4, and 8 km. The spectral features selected from the CSDB are the same as those for MIPAS (Sect. 2.1, Fig. 1).

As described in Spang et al. (2016), the CSDB was calculated with typical particle radii and volume densities of PSCs (Table 2). Five different PSC compositions have been con-

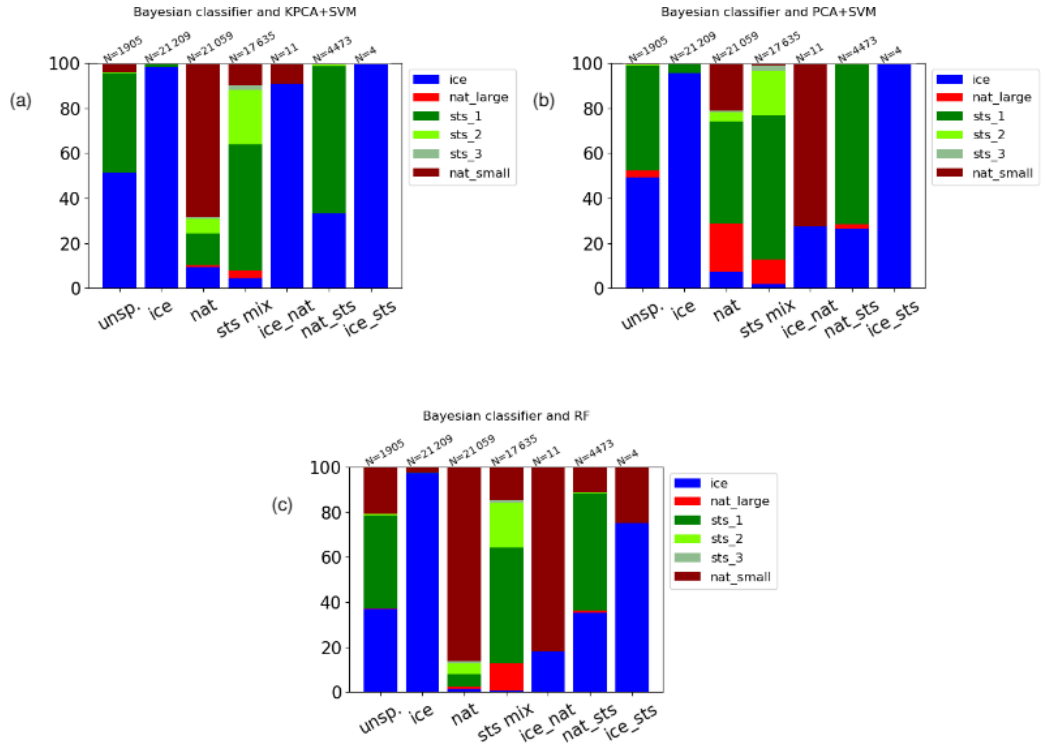


Figure 2. Intercomparison of ML and Bayesian classifiers for Southern Hemisphere winter (May to September 2009). Ticks on the x axis represent the classes of the BC. The y axis indicates the fraction of the classes as predicted by the KPCA + SVM **(a)**, PCA + SVM classifier **(b)**, and the RF classifier **(c)**. N is the number of samples belonging to each class of the Bayesian classifier.

sidered: ice; NAT; STS with 2 % H_2SO_4, 48 % HNO_3, and 50 % H_2O (called later on STS 1); STS with 25 % H_2SO_4, 25 % HNO_3, and 50 % H_2O (STS 2); and STS with 48 % H_2SO_4, 2 % HNO_3, and 50 % H_2O (STS 3). These values are derived from the model by Carslaw et al. (1995) and span over all possible compositions. The CSDB does not give any representative frequency of real occurrences in the atmosphere. For this study, we decided to split the set of NAT spectra into two classes, large NAT (radius $> 2\,\mu m$) and small NAT (radius $<= 2\,\mu m$). This decision was taken to assess the capability of the classifiers to correctly separate between the two classes. It is well known that small NAT particles (radius $<= 2\,\mu m$) produce a specific spectral signature at 820 cm^{-1} (Spang and Remedios, 2003; Höpfner et al., 2006). Spectra for large NAT particles are more prone to overlap with those of ice and STS.

To prepare both the real MIPAS and the CSDB data for PSC classification, we applied the cloud index (CI) method of Spang et al. (2004) with a threshold of 4.5 to filter out clear-air spectra. In optimal conditions a CI < 6 detects clouds with extinction coefficients down to about $2 \times 10^{-5}\,km^{-1}$ in the midinfrared (Sembhi et al., 2012). However, in the polar winter regions these optimal conditions do not persist over an entire winter season. Hence, we selected a threshold of 4.5 that reliably discriminates clear air from cloudy air in the Southern and Northern Hemisphere polar winter regions as it is sensitive to extinctions down to $5 \times 10^{-4}\,km^{-1}$ (Griessbach et al., 2020).

3 Methods

3.1 Conventional classification methods

Spang et al. (2016) provide an overview on various conventional methods used to classify Envisat MIPAS PSC observations. Furthermore, a Bayesian approach has been introduced in their study to combine the results of individual classification methods. This approach is used as a benchmark for the new classifiers introduced in the present paper. The Bayesian classifier considers a total of 13 features, including corre-

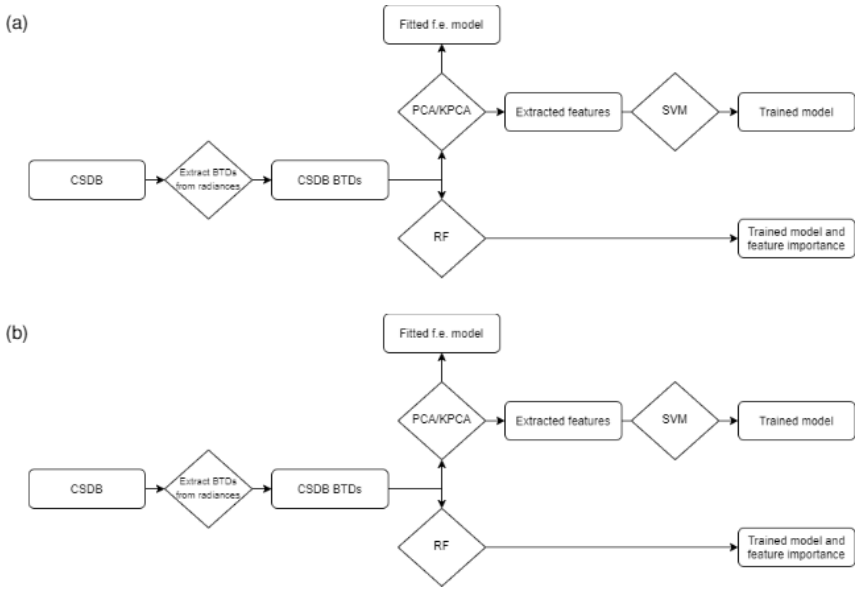


Figure 3. (a) Flowchart of the training process and **(b)** prediction. "F.e." stands for feature extraction.

lations between the cloud index (CI) (Spang et al., 2004), the NAT index (NI) (Spang and Remedios, 2003; Höpfner et al., 2006), and another five additional BTDs. Each feature has been assigned individual probabilities $p_{i,j}$ in order to discriminate between the different PSC composition classes. The output of the Bayesian classifier is calculated according to $P_j = \prod_i p_{i,j} / \sum_j \left(\prod_i p_{i,j}\right)$, where the indices $i = 1, \ldots, 13$ and $j = 1, 2, 3$ refer to the individual feature and the PSC constituent, respectively. The normalized probabilities P_j per PSC constituent are used for final classification applying the maximum a posteriori principle. The BC composition classes are the following: unknown, ice, NAT, STS_mix, ICE_NAT, NAT_STS, and ICE_STS. A stepwise decision criterion is applied to classify each spectrum. If the maximum of P_j (with $j = 1 \ldots 3$) is greater than 50 %, then the spectrum is assigned a single PSC composition label. If two P_j values are between 40 % and 50 %, then a mixed composition class, for example ICE_STS for $j = 1$ and $j = 3$, is attributed. If the classification results in P1, P2, or P3 < 40 %, then the spectrum is labeled as "unknown". Considering the Southern Hemisphere 2009 case, the NAT_STS mixed composition class is populated with more than 4000 spectra, while ICE_STS and ICE_NAT predictions are negligible (Fig. 2). The analysis of the complete MIPAS period (9 Southern Hemisphere and 10 Northern Hemisphere winters in Spang et al., 2018) showed that ICE_STS and ICE_NAT classes are generally only in the subpercentage range and statistically not relevant. The Bayesian classifier requires a priori information and detailed expert knowledge on the selection of the features to be used as discriminators and in assigning the individual probabilities $p_{i,j}$ for classification. In this work, we aim at investigating automatic ML approaches instead of the manual or empirical methods applied for the Bayesian classifier. Nevertheless, being carefully designed and evaluated, the results of the Bayesian classifier are used for further reference and comparison in this study.

3.2 Feature extraction using PCA and KPCA

In a first step, we calculated BTDs from the 1 cm^{-1} downsampled radiances of the CSDB. Calculating the BTDs between the 142 spectral windows resulted in 10 011 BTDs for a total of 70 000 spectra. In a second step, in order to reduce the number of data, we applied a variance threshold to exclude BTD features with relatively low variance ($\sigma^2 < 10\,\mathrm{K}^2$), as this indicates that the corresponding windows have rather similar information content. In order to further reduce the difficulties and complexity of the classification task, we decided to even further reduce the number of BTD features before training of the classifiers by means of feature extraction.

Feature selection methods are used for picking subsets of an entire set of features while keeping the information content as high as possible. The methods help to reduce the training time of the classifier and to reduce the risk of overfitting. Feature selection methods typically belong to three

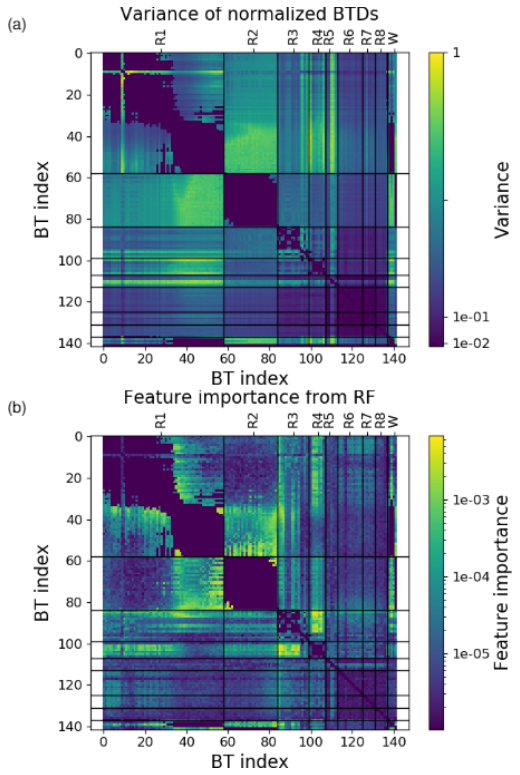


Figure 4. Variance of normalized BTDs **(a)** and feature importance as estimated by the RF classifier **(b)**. The BT index numbers on the x and y axis correspond to the spectral regions as listed in Table 1.

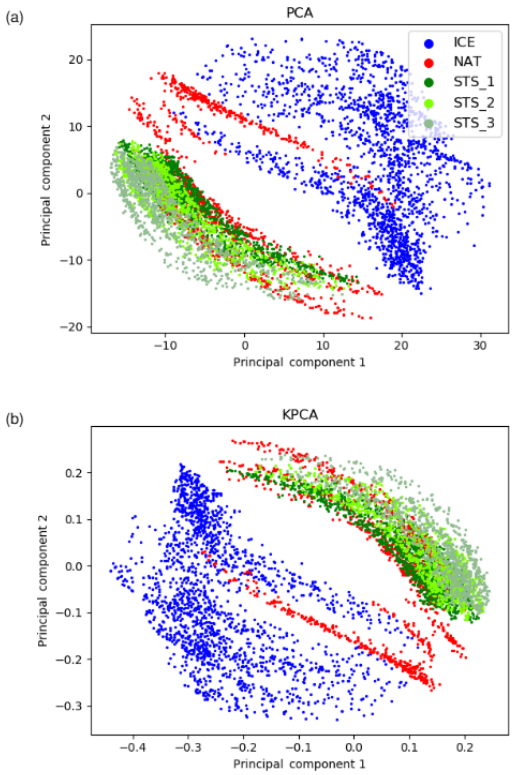


Figure 5. Correlations of the first two principal components from the PCA **(a)** and KPCA **(b)** analysis applied to the CSDB.

main families (Bolón-Canedo et al., 2016): (i) filter methods, where the importance of the feature is derived from intrinsic characteristics of it; (ii) wrapper methods, where the features are selected by optimizing the performances of a classifier; and (iii) embedded methods, where classification and selection happen at the same time. Here, we used a more advanced approach to dimensionality reduction, which goes under the name of feature extraction. In this case, instead of simply selecting a subset of the original features, the set of features itself is transformed to another space where the selection takes place.

Principal component analysis (PCA) is among the most popular feature extraction methods (Jolliffe and Cadima, 2016). The main idea of the PCA is to reproject the data to a space where the features are ranked on the variance that they account for. At first a centering of the data through the subtraction of the mean is performed. Then, the covariance matrix is calculated and its eigenvectors and eigenvalues are computed. At this point, selecting the eigenvectors whose eigenvalues are largest, it is possible to pick the components on which most of the variance of the data lays. PCA already found applications in the analysis of atmospheric midinfrared spectra, in particular for the compression of high-resolution spectra and for accelerating radiative transfer calculations (e.g., Huang and Antonelli, 2001; Dudhia et al., 2002; Fauvel et al., 2009; Estornell et al., 2013). PCA has been used in this study for two main purposes, dimensionality reduction and visualization of the data.

Kernel PCA (KPCA) is an extension of the PCA where the original data $\mathbf{x}$ are first transformed using a mapping function $\phi(\mathbf{x})$ to a higher dimensional feature space. The main advantage of using KPCA relies in the fact that it can capture nonlinear patterns, which PCA, being a linear method, may fail to represent well. However the construction of the kernel matrix K for mapping can be expensive in terms of memory. This latter problem undermines severely the possibility of using this algorithm for large datasets. At this point the kernel trick comes into play (Schölkopf et al., 1997). It helps to avoid the inconvenience of having to compute the covari-

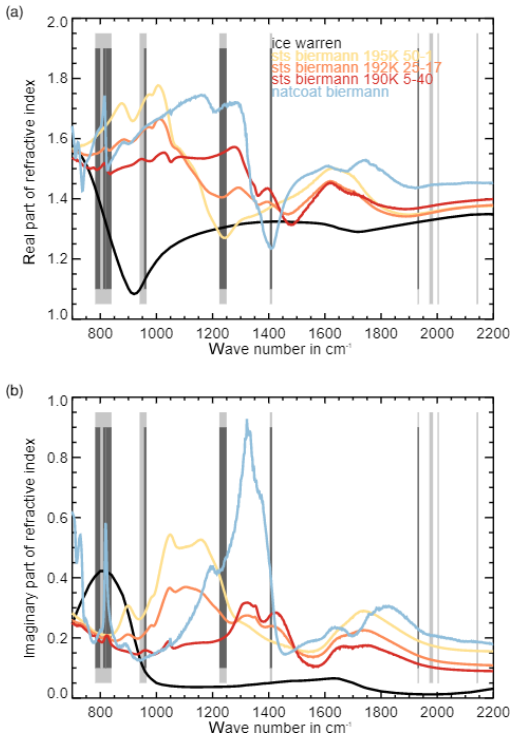


Figure 6. Real **(a)** and imaginary **(b)** part of PSC particle refractive indices. The gray bars represent the eight spectral regions considered in this study.

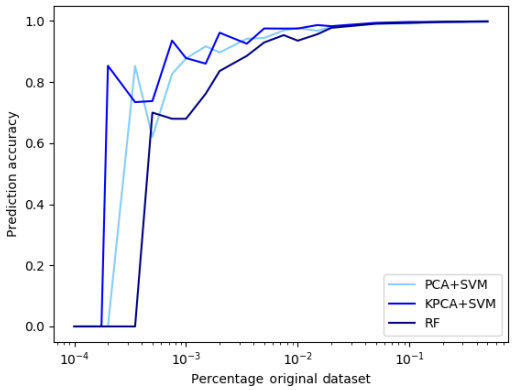


Figure 7. Prediction accuracy using subsets of the CSDB of different size.

ance matrix in a large transformed space. Instead of translating each data point to the transformed feature space using the mapping function $\phi(\mathbf{x})$, the inner product can be calculated as $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)$, resulting in a much less demanding computational task. Among the most common kernels are the radial basis function (RBF) and the polynomial (Genton, 2002), which we also considered in this study.

3.3 Classification using support vector machines and random forests

Supervised classification is a ML task in which the classes or labels of unknown samples are predicted by making use of a large dataset of samples with already known labels. In order to do that, the classification algorithm first has to be trained; i.e., it has to learn a map from the input data to its target values. After a classifier is trained, one can give it as input an unlabeled set of data points with the aim of predicting the labels. The training of a classifier is usually a computationally demanding task. However, the classification of unknown samples using an already trained classifier is computationally cheap.

A large number of classifiers exist based on rather different concepts. Bayesian classifiers follow a statistical approach. Support vector machines (SVMs) are based on geometrical properties. Random forests (RF) are based on the construction of multiple decision trees. Neural networks try to emulate the behavior of the human brain by stacking a number of layers composed of artificial neurons (Zeiler and Fergus, 2014). According to the "no free lunch" theorem, it is not possible to state safely which algorithm is expected to perform best for any problem (Wolpert, 1996). In this study, we selected two well-established methods, RFs and SVMs, to test their performance.

Random forest is an algorithm that learns a classification model by building a set of decision trees. A decision tree is composed of decision nodes, which lead to further branches and leaf nodes, which finally represent classification results. RFs are nonparametric models that do not assume any underlying distribution in the data (Breiman, 2001). RF builds a number of decision trees selecting a random subset of the original features for each tree. In this way the model becomes more robust against overfitting. The classification result of the RF model will be the label of the class that has been voted for by the majority of decision trees (Liu et al., 2012). An interesting characteristic of the RF classifier is that it can give by calculating the Gini index (Ceriani and Verme, 2012) also a measure of the feature importance. In this way, the RF classifier can also be exploited for performing feature selection.

The performance of a RF classification model depends on a number of hyperparameters, which must be defined before training. (i) The "number of estimators" or decision trees of the forest needs to be defined. (ii) A random subset of the features is selected by each decision tree to split a node. The dimension of the subset is controlled by the hyperparameter

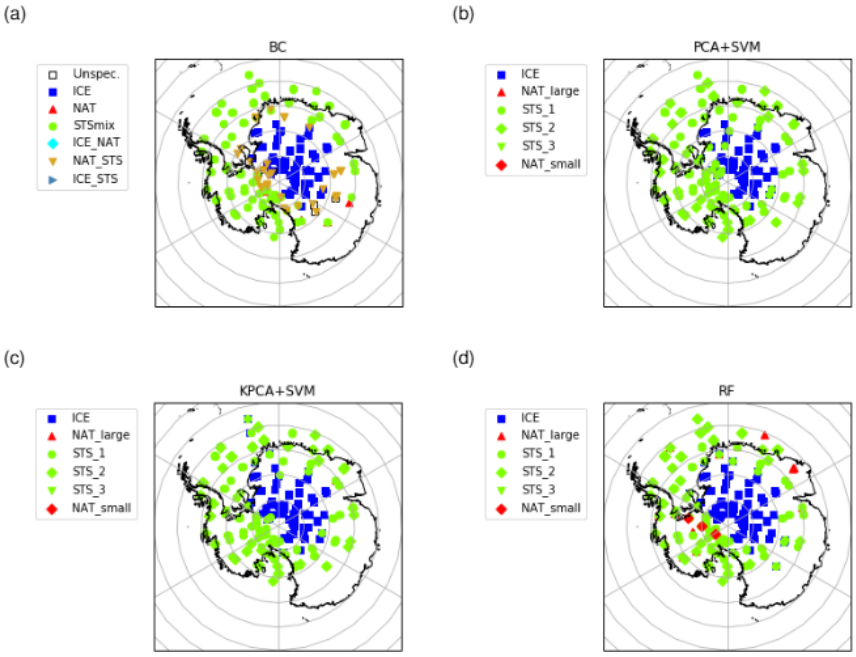


Figure 8. MIPAS observations of PSCs on 14 June 2009 in the Southern Hemisphere at tangent altitudes between 18 and 22 km. The classification was performed with **(a)** the Bayesian classifier, **(b)** the SVM based on PCA features, **(c)** the SVM based on KPCA features, and **(d)** the RF classifier.

"maximum number of features". (iii) The "maximum depth", i.e., the maximum number of levels in each decision tree, controls the complexity of the decision trees. In fact, the deeper a decision tree is, the more splits can take place in it. (iv) The "minimum number of samples before split" that has to be present in a node before it can be split also needs to be defined. (v) A node without a further split has to contain a "minimum number of samples per leaf" to exist. (vi) Finally, we have to decide whether to use "bootstrapping" or not. Bootstrapping is a method used to select a subset of the available data points, introducing further randomness to increase robustness (Probst et al., 2019).

SVMs became popular around the 1990s (Cortes and Vapnik, 1995). The method is based on the idea of identifying hyperplanes, which best separate sets of data points into two classes. In particular, SVM aims at maximizing the margin, which is the distance between few points of the data, referred to as "support vectors", and the hyperplane that separates the two classes. The "soft margin" optimization technique takes into account the fact that misclassification can occur due to outliers. For that reason a tuning parameter C is included in order to allow for the presence of misclassified samples during the optimization of the margin to a given extent. The choice of the parameter C is a trade-off between minimizing the error on the training data and finding a hyperplane that may generalize better (Brereton and Lloyd, 2010).

SVM had been originally developed to find linear decision boundaries. However, the introduction of the kernel trick (cf., Sect. 3.2) enables the possibility for nonlinear decision boundaries. Kernel functions, e.g., radial basis functions or polynomials, are mapping from the original space to a nonlinearly transformed space, where the linear SVM is applied (Patle and Chouhan, 2013). In the case of a nonlinear kernel, the parameter γ is used to define how much a support vector has influence on deciding the class of a sample. A small value of γ implies that this support vector also has impact on samples far in the feature space, and a large value of γ has an influence only on samples that are close in the feature space.

We recap in Fig. 3 the entire pipeline for training and prediction. The BTDs extracted from the CSDB dataset are given as input to the PCA or KPCA methods, and the extracted features are fed to the SVM classifier for model training (PCA + SVM and KPCA + SVM). On the other hand, the RF classifier is given as input BTDs directly, without prior feature extraction. The input samples (BTDs) are annotated with a label as explained in Sect. 2.2. In prediction (Fig. 3b), the BTDs extracted from the MIPAS measurements are the input to the three methods PCA + SVM, KPCA + SVM, and

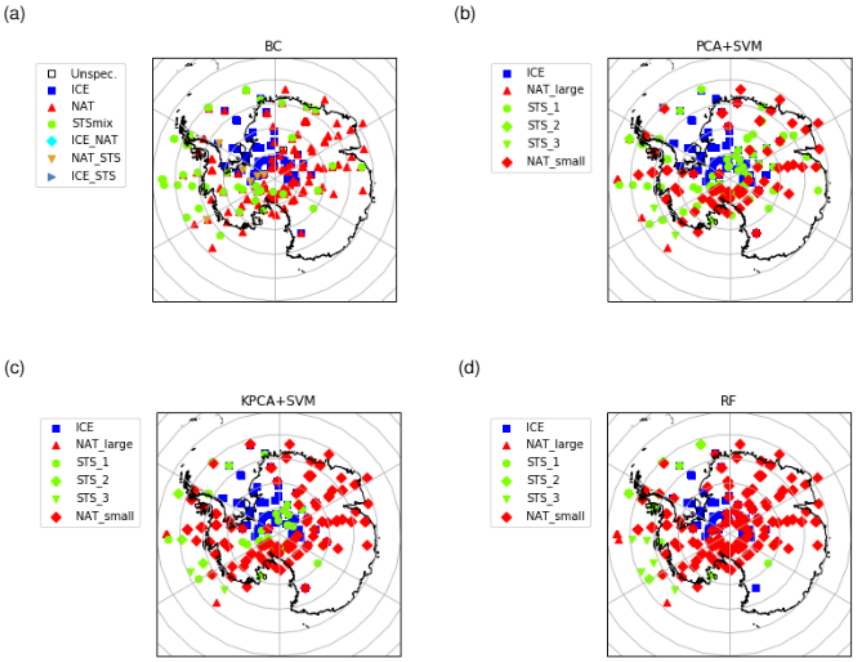


Figure 9. Same as Fig. 8 but for 26 August 2009.

RF, where the outputs are the predicted label for each sample. The RF classifier provides a feature importance measure as well. During prediction, the sample is assigned to one of the following classes representing the main constituent: ice, small NAT, large NAT, STS 1, STS 2, and STS 3. Compared to the NAT class of the Bayesian classifier, in the proposed ML methods NAT particles are assigned to small and large NAT subclasses. The STS_mix class of the BC overlaps with STS 1, STS 2, and STS 3. There are no directly corresponding classes to the mixed composition ones of the BC. As discussed above in the text, only a few spectra are classified by the BC as ICE_STS or ICE_NAT. Samples belonging to the NAT_STS class of the BC, characterized by a non-negligible population, are labeled by the new ML classes mostly as STS 1 (Fig. 2).

4 Results

4.1 Feature extraction

In this study, we applied PCA and KPCA for feature extraction from a large set of BTDs. Both PCA and KPCA are reprojecting the original BTD features to a new space, where the eigenvectors are ordered in such a way that they maximize variance contributions of the data. Figure 4a shows a matrix of the normalized variances of the individual BTDs considered here. The matrices in Fig. 4 are symmetric; thus the reader can either focus on the location (i.e., the indices of the BTs from which the BTD feature has been computed) of the maximum values in the upper or lower triangular part. A closer inspection shows that the largest variances originate from BTDs in the range from 820 to 840 cm^{-1} (indicated as spectral region R1 in Table 1) and 956 to 964 cm^{-1} (R2). BTDs close to 790 cm^{-1} (R1, BT index $\sim$ 10) also show high variance. Another region with high variances originates from BTDs between 820 and 840 cm^{-1} (part of R1) and between 1404 and 1412 cm^{-1} (R4) as well as between 1930 and 1935 cm^{-1} (R5). Around 820, 1408, and 1930 cm^{-1} the imaginary part (absorption contribution) of the complex refractive index of NAT has pronounced features (Höpfner et al., 2006), whereas around 960 cm^{-1} the real part (scattering contribution) of the complex refractive index of ice has a pronounced minimum (e.g., Griessbach et al., 2016). Even though in our work the ML classifiers are given BTDs (computed from radiance) as input and refractive indices are not directly used in the classification process, the latter can provide insights on microphysical properties of the different PSC particles and additional information on the features used by the ML methods.

Table 3. Top ten list of BTDs providing maximum feature importance as estimated by the RF classifier.

Feature importance	BTD indices	BTD wavenumbers (cm^{-1})
0.006815	85–105	1225.5–1410.5
0.005798	61–83	942.5–964.5
0.004334	57–76	839.5–957.5
0.003233	37–56	819.5–838.5
0.002649	86–139	1226.5–820
0.002633	58–139	840.5–820
0.002272	40–87	822.5–1227.5
0.001677	26–139	808.5–820
0.001592	27–101	809.5–1406.5
0.001033	102–137	1407.5–792.2

The first and second principal components, which capture most of the variance in the data, are shown in Fig. 5. Comparing PCA and KPCA, we note that they mostly differ in terms of order and amplitude. This means that the eigenvalues change, but the eigenvectors are rather similar in the linear and nonlinear case. For this dataset, the nonlinear KPCA method (using a polynomial kernel) does not seem to be very sensitive to nonlinear patterns that are hidden to the linear PCA method. However, it should be noted that the SVM classifier is sensitive to differences in scaling of the input features as they result from the use of PCA and KPCA for feature selection. Therefore, classification results of PCA + SVM and KPCA + SVM can still be expected to differ and are tested separately.

As discussed in Sect. 3.3, RF itself is considered to be an effective tool not only for classification but also for feature selection. It is capable of finding nonlinear decision boundaries to separate between the classes. However, the method does not group the features together in components like PCA or KPCA. It is rather delivering a measure of importance of all of the individual features. Figure 4b shows the feature importance matrix provided by the RF. Note that the values are normalized; i.e., the feature importance values of the upper triangular matrix sum up to 1. We can observe that this approach highlights clusters similar to Fig. 4a.

Similarly to PCA and KPCA, BTDs between windows in the range from 820 to 840 cm^{-1} (R1) and from 956 to 964 cm^{-1} (R2) are considered to be important by the RF algorithm. BTDs between 1224 and 1250 cm^{-1} (R3) and between 1404 and 1412 cm^{-1} (R4) are also regarded as important. The importance of the RF features located in this cluster is in contrast with the relatively low BTD variance in the same area. A similar observation can be done regarding BTDs between 782 and 800 cm^{-1} and between 810 and 820 cm^{-1} (both belonging to R1). This region is in the range of values of the NAT feature, providing a possible explanation of the capability of the RF to detect the characteristic peak of small NAT as well as its shift with the increase in the radius. BTDs between 960 cm^{-1} (R2) and 1404 to 1412 cm^{-1} (R4) are also quite important. Table 3 specifically provides the most important BTDs between the different regions. Actually, Fig. 6 shows that all the windows or BTDs found here by the RF are associated with physical features of the PSC spectra, namely a peak in the real and imaginary part of the complex refractive index of NAT around 820 cm^{-1} or a minimum in the real part of the complex refractive index of ice around 960 cm^{-1}. STS can be identified based on the absence of these features. Considering the larger windows W, the matrices of the variance and of the RF feature importance seem to agree, with the exception of W3 ($\sim$ 820 cm^{-1}) that is regarded as important by the RF scheme but is not characterized by high variance, confirming the capability of the RF for detecting the NAT feature.

A closer inspection reveals an interesting difference between PCA and KPCA on the one hand and RF on the other hand. Two additionally identified windows around $\sim$ 790 (BT index $\sim$ 10) and $\sim$ 1235 cm^{-1} (BT index $\sim$ 90) are located at features in the imaginary part of the refractive index of ice and NAT, respectively (Höpfner et al., 2006). This latter set of BTDs is considered to have a large feature importance by the RF method but does not show a particularly large variance. This suggests that a supervised method like RF can capture important features where unsupervised methods like PCA and KPCA may fail.

4.2 Hyperparameter tuning and cross-validation accuracy

Concerning classification, we compared two SVM-based classifiers that take as input the features from PCA and KPCA and the RF that uses the BTD features without prior feature selection. The first step in applying the classifiers is training and tuning of the hyperparameters. Cross validation is a standard method to find optimal hyperparameters and to validate a ML model (Kohavi, 1995). For cross validation the dataset is split into a number of subsets, called folds. The model is trained on all the folds, except for one, which is used for testing. This procedure is repeated until the model has been tested on all the folds. The cross-validation accuracy refers to the mean error of the classification results for the testing datasets. Cross validation is considered essential to avoid overfitting while training a ML model. Selecting the best hyperparameters that maximize the cross-validation accuracy of a ML model is of great importance to exploit the models' capabilities at a maximum.

In this study, we applied 5-fold cross validation on the CSDB dataset. For the SVM models we decided to utilize a grid-search approach to find the hyperparameters. As the parameter space of the RF model is much larger, a random-search approach was adopted (Bergstra and Bengio, 2012). The test values and optimum values of the hyperparameters for the SVM and RF classifiers are reported in Tables 4 and 5, respectively. For the optimum hyperparameter values, all

Table 4. Hyperparameter choices considered for the SVM classifier.

Hyperparameter	Test values	Optimal value
Kernel	linear, RBF, polynomial	RBF
C	1, 10, 100, 1000	1000
γ	0.0001, 0.001, 0.01, 0.1, 1, 10	1 (PCA) / 10 (KPCA)

Table 5. Hyperparameter choices considered for the RF classifier.

Hyperparameter	Test values	Optimal value
Number of estimators	200, 210, ..., 2000	1000
Maximum number of features	auto, sqrt	auto
Maximum depth	10, 20, ..., 110	50
Minimum number of samples before split	2, 5, 10	2
Minimum number of samples per leaf	1, 2, 4	1
Bootstrapping	true, false	false

Table 6. Scores of the RF classifier on a small subset of CSDB samples.

Class	Precision	Recall	F1 score	Support
Ice	1.00	1.00	1.00	56
NAT_large	1.00	0.91	0.95	23
NAT_small	1.00	1.00	1.00	33
STS_1	0.96	0.76	0.85	34
STS_2	0.78	0.97	0.86	33
STS_3	0.94	0.97	0.96	34
Total	0.95	0.94	0.94	210

classification methods provided an overall prediction accuracy close to 99 %. Also, our tests showed that the ML methods considered here for the PSC classification problem are rather robust against changes in the hyperparameters.

During the training of the classifiers, we conducted two experiments. In the first experiment, we checked how large the amount of synthetic samples from the CSDB needs to be in order to obtain good cross-validation accuracy. For this experiment, we performed the training with subsets of the original CSDB data, using randomly sampled fractions of 50 %, 20 %, 10 %, 5 %, 2 %, 1 %, 0.05 %, 0.02 %, 0.01 %, 0.005 %, 0.002 %, and 0.001 % of the full dataset. This experiment was run for all three ML models (PCA + SVM, KPCA + SVM, and RF) using the optimal hyperparameters found during the cross-validation step. The results in Fig. 7 show that using even substantially smaller datasets ($> 0.02\,\%$ of the original data or about 1200 samples) would still result in acceptable prediction accuracy ($> 80\,\%$). This result is surprising and points to a potential limitation of the CSDB for the purpose of training ML models that will be discussed in more detail in Sect. 5.

In the second experiment, we intentionally performed and analyzed the training and testing of the RF method with a rather small subset of data. Although the results from this procedure are less robust, they can help pinpoint potential issues that cannot be detected using the full dataset. We computed different scores to assess the quality of the prediction for the RF classifier in the case of 600 randomly selected samples used for training and around 200 samples used for testing. As shown in Table 6, also using a limited number of samples for training leads to very high classification accuracy. The metrics used in Table 6 are precision $P = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, recall $R = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, and F1 score $F1 = 2(R \times P)/(R+P)$, where TP is the number of true positives, FP the number of false positives, FN the number of false negatives, and support is the number of samples (Tharwat, 2018). As reported in Table 6, it is found that ice and small NAT accuracies are higher than the ones of STS. This is a hint to the fact that distinguishing small NAT and ice from the other classes is an easier task than separating spectra of PSCs containing larger NAT particles from those populated with STS, which is consistent with previous studies (Höpfner et al., 2009).

An additional experiment was performed on the CSDB spectra labeled as large NAT. The BC misclassifies a large amount of those spectra (99 % of them classified as STS_mix), whereas the proposed ML methods correctly classify them as large NAT (Table 7). This experiment suggests that the new classification schemes can help in overcoming the inability of the BC in discriminating between large NAT and STS.

Table 7. Predicted labels vs. CSDB classes, with analysis restricted to NAT large (radius >2 µm).

NAT large, CSDB					
BC class	Pred. by BC	Proposed ML class	Pred. by PCA + SVM	Pred. by KPCA + SVM	Pred. by RF
ICE	0	ICE	0	0	0
NAT	0.0012	NAT_small	0	0	0
		NAT_large	1	1	1
STS_mix	0.9988	STS_1	0	0	0
		STS_2	0	0	0
		STS_3	0	0	0
NAT \ STS	0				
ICE \ NAT	0				
ICE \ STS	0				

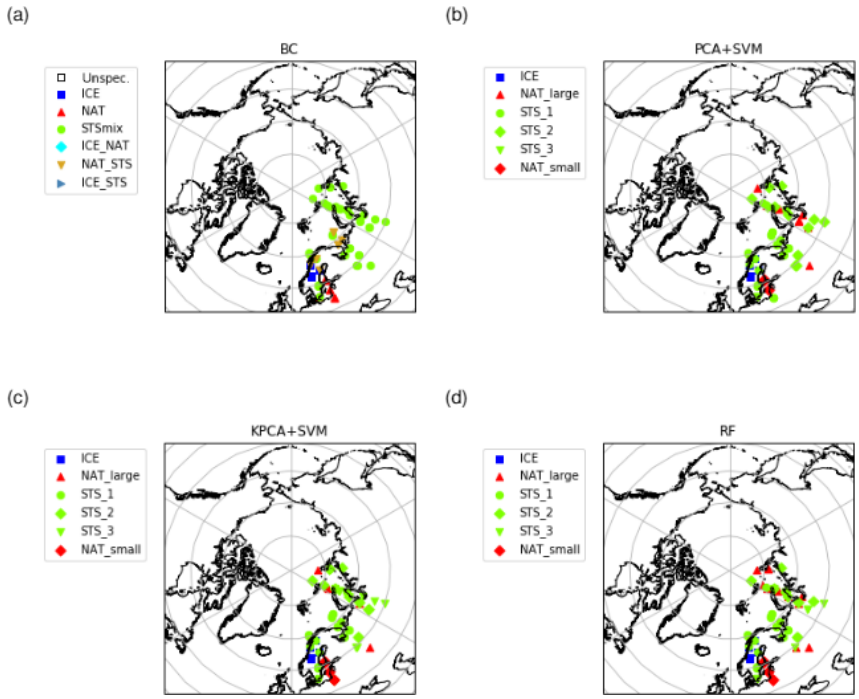


Figure 10. Same as Fig. 8 but for 25 January 2007 and the Northern Hemisphere.

4.3 Classification using real MIPAS data

4.3.1 Case studies

For three case studies looking at individual days of MIPAS observations, two in the Southern Hemisphere and one in the Northern Hemisphere winter season, we compared the results of the different classification methods (Figs. 8 to 10). Early in the Southern Hemisphere PSC season, on 14 June 2009 (Fig. 8), we found that the classification results are mostly coherent among all the classifiers, not only from a quantitative point of view but also geographically, especially concerning the separation of ice and STS PSCs. Further, we found that most of the PSCs, which were labeled as NAT by the Bayesian classifier, were classified as STS by the ML classification methods. While both SVM classification schemes did not indicate the presence of NAT, the RF found some NAT, but mostly at different places than the Bayesian classi-

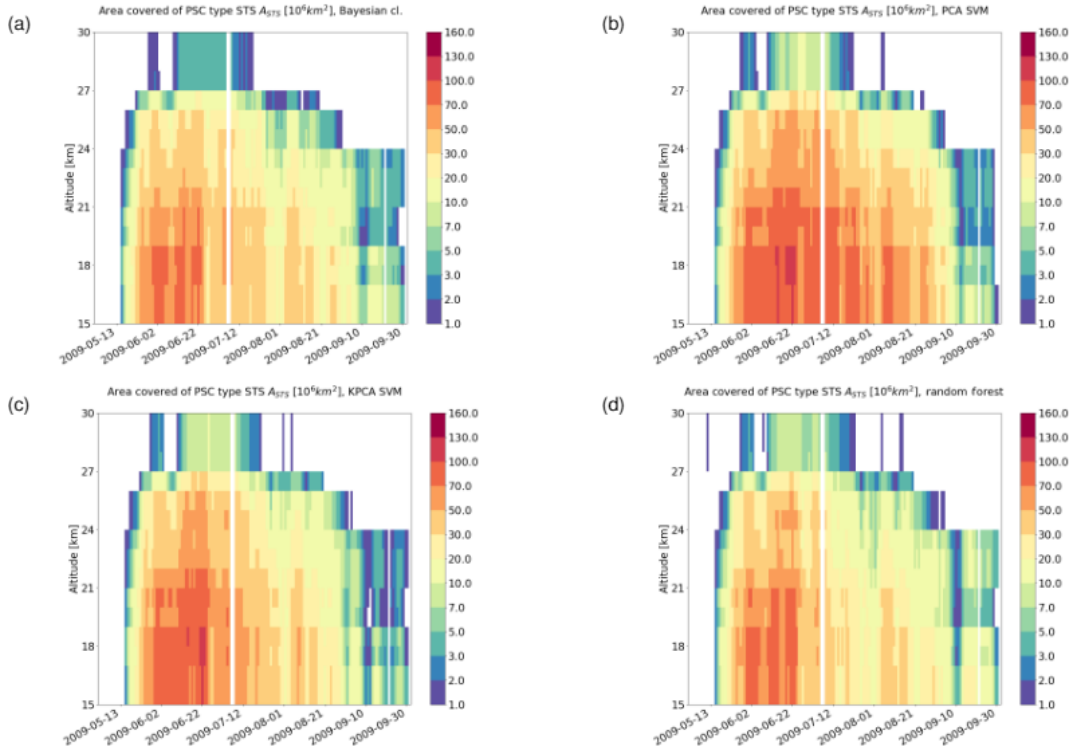


Figure 11. Area covered by STS clouds from May to September 2009 in the Southern Hemisphere based on results of **(a)** the Bayesian classifier, **(b)** the PCA + SVM classifier, **(c)** the KPCA + SVM classifier, and **(d)** the RF classifier. The bins span a length of 1 d in time and 1 km in altitude. A horizontal (3 d) and vertical (3 km) moving average has been applied for the sake of a smoother representation.

fier. Note that from a climatological point of view, NAT PSCs are not expected to be the dominant PSC type until the middle to end of June for the Southern Hemisphere (Pitts et al., 2018).

Later in the Southern Hemisphere PSC season, on 26 August 2009 (Fig. 9), it is again found that the separation between ice and nonice PSCs is largely consistent for all the classifiers. The NAT predictions by the RF classifier tend to agree better with the Bayesian classifier than the NAT classifications by the SVM method. Overall, the Southern Hemisphere case studies seem to suggest that the SVM classifiers (using PCA or KPCA) underestimate the presence of NAT PSCs compared to the BC and the RF classifiers. We note that separating the NAT and STS classes from limb infrared spectra presents some difficulties.

As a third case study, we analyzed classification results for 25 January 2007 for the Northern Hemisphere (Fig. 10). This case was already analyzed to some extent by Hoffmann et al. (2017). It is considered to be particularly interesting, as ice PSCs have been detected over Scandinavia at synoptic-scale temperatures well above the frost point. Hoffmann et al. (2017) provided evidence that the PSC formation in this case was triggered by orographic gravity waves over the Scandinavian Mountains. Also in this case study the classification of ice PSCs over Scandinavia shows a good agreement for the new ML methods with the Bayesian classifier. Further, we see that the two SVM and the RF methods identified small NAT where the Bayesian classifier also found NAT. However, at the locations where the Bayesian classifier indicates a mixture of NAT and STS, the ML methods indicate STS, and the ML methods indicate large NAT at locations where the Bayesian classifier found STS.

4.3.2 Seasonal analyses

For a seasonal analysis, we first considered MIPAS observations during the months from May to September 2009. Figures 11 to 13 show the area coverage for each class of PSC along time and altitude. Comparing the time series of the classification results, we can assess the agreement quantita-

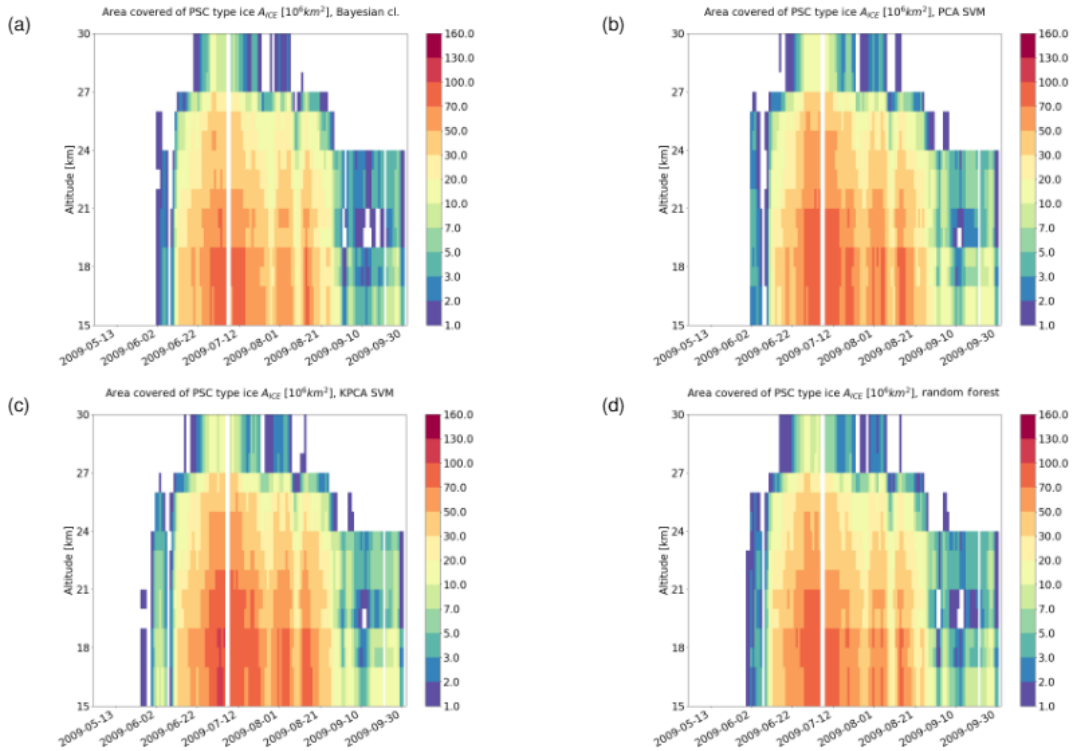


Figure 12. Same as Fig. 11 but for ice.

tively. The mixed composition classes of the Bayesian classifier (NAT_STS, ICE_STS, and ICE_NAT) are not considered in this analysis. Taking a look at STS (Fig. 11), all the classifiers predict an early season appearance. While the RF predicts a time series that resembles quite closely the one predicted by the Bayesian classifier, the other two ML methods (PCA + SVM and KPCA + SVM) predict a significantly larger coverage of STS clouds over the winter. Regarding the ice PSCs (Fig. 12), the patterns in the time series are similar between all classifiers. However, we can observe that, even if the spatiotemporal characteristics are similar, both SVM methods predict a notably larger area covered by ice clouds. Moreover, the KPCA + SVM classifier predicts an earlier emergence of ice with respect to the other classifiers. Considering the NAT time series (Fig. 13), all the classifiers predict a late appearance during the season. The classification schemes based on SVM predict a much lower presence of NAT with respect to the RF and the Bayesian classifier. Furthermore, most of the bins with a high value of NAT coverage in the Bayesian classification scheme are predicted as small NAT particles. This result confirms that the spectral features of small NAT are strong enough to find a good decision boundary, as explained in Sect. 2.2.

Figure 14 shows the overall percentages of the PSC classes for May to September 2009 for the Southern Hemisphere. The occurrence frequencies of ice PSCs are quite consistent, ranging from 32 % for the Bayesian classifier to 39 % for KPCA + SVM. It is found that the approaches based on SVM slightly overestimate the presence of ice with respect to the RF (35 %) and the Bayesian classifier. However, the main differences that were encountered are in the separation between STS and NAT. The two classification schemes using SVM predict a much smaller amount of NAT PSCs (17 and 26 % taking small and large NAT together) compared to the RF (33 % considering only small NAT, 37 % taking small and large NAT together) and the Bayesian classifier (32 % NAT). The RF and the Bayesian classifier are more coherent between themselves. Other interesting findings are related to the classification between small and large NAT. Indeed, the vast majority of the NAT predictions in the KPCA + SVM and RF methods belong to the small NAT class. PCA + SVM diverges significantly from the other methods, largely underestimating small NAT and overestimating large NAT. This

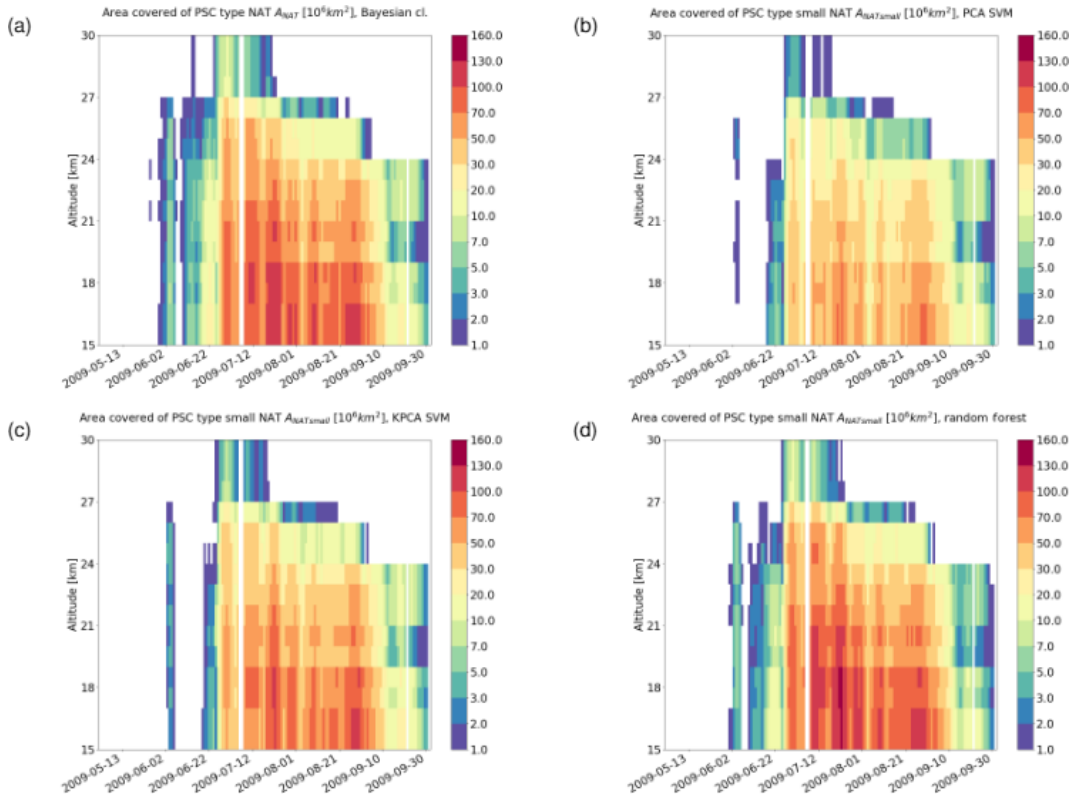


Figure 13. Same as Fig. 11 but for NAT.

suggests once more that the discrimination between small NAT and STS PSCs is more easily possible using midinfrared spectra for classification, while larger NAT PSCs are harder to separate.

In addition to the results presented above, we conducted the seasonal analyses also for MIPAS observations acquired in the months from November 2006 to February 2007 in the Northern Hemisphere (Fig. 15). As expected, a much smaller fraction of ice PSCs (4–6 %) was found compared to the Southern Hemisphere. As in the Southern Hemisphere winter, the SVM classifiers taking as input the PCA and KPCA features found significantly less NAT (both 6 %) than the Bayesian classifier (15 %), whereas the RF classifier identified a significantly larger fraction of large NAT spectra (30 %) that resulted in a significantly higher NAT detection rate (37 %). This finding may point to a potential improvement of the RF classifier compared to the Bayesian classifier. In fact, it had been already reported by Spang et al. (2016) that the Bayesian classifier for MIPAS underestimated the fraction of NAT clouds compared to Cloud-Aerosol Lidar with Orthogonal Polarization (CALIOP) observations. Further, the STS partitioning between the three STS subclasses is different between the Southern and Northern Hemisphere winters. While in the Southern Hemisphere STS 1 is dominant, in the Northern Hemisphere STS 2 is dominant and the fraction of STS 3 is significantly increased. This result is plausible, because the Northern Hemisphere winters are warmer than the Southern Hemisphere winters, and STS 1 forms at lower temperatures (e.g., $\sim$ 189 K) than STS 2 ($\sim$ 192 K) and STS 3 ($\sim$ 195 K at 50 hPa, Carslaw et al., 1995).

Figures 16 and 2 show cross tabulations between the classification results of the Bayesian classifier and the three ML methods. They allow us to directly assess how much the different classification schemes agree in terms of their predictions for the different classes. For instance, considering the ice class of the PCA + SVM and KPCA + SVM classifiers, it can be seen that around 80 % of the samples were classified consistently with the Bayesian method, while this percentage is above 90 % for the RF (Fig. 16). Concerning NAT, the

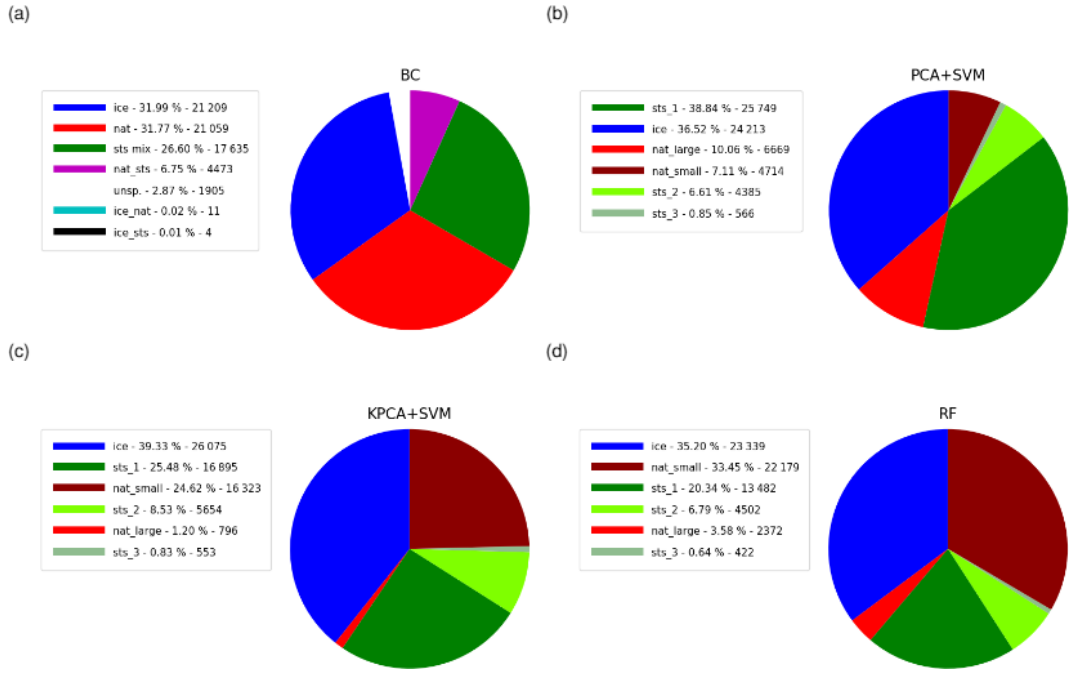


Figure 14. Partitioning of the PSC composition classes for the Southern Hemisphere winter (May to September 2009) derived by **(a)** the Bayesian classifier, **(b)** the PCA + SVM classifier, **(c)** the KPCA + SVM classifier, and **(d)** the RF classifier. Percentage values and number of events are reported in the legends.

RF classifier predicts as small NAT more than 80 % of what had been classified as NAT class by the Bayesian classifier (Fig. 2). The PCA + SVM and KPCA + SVM methods predict a smaller fraction of small NAT for the NAT class of the Bayesian classifier, around 30 % and 70 %, respectively. The PCA + SVM in particular predicts a significantly smaller amount of samples belonging to the small NAT class than the other methods (Fig. 16), while it predicts a larger number of samples of the STS subclasses. This result may suggest that PCA + SVM and KPCA + SVM are not as sensitive as BC for small NAT detection, while RF is. Considering the STS subclasses of the RF and KPCA + SVM classifiers altogether, they seem to mostly agree with the STS_mix predictions of the Bayesian classifier. On the other hand, the total number of samples predicted by the PCA + SVM scheme as belonging to the STS subclasses is notably larger than the predictions of the Bayesian classifier (Fig. 16). This finding is in line with what has been discussed a few lines above and in Sect. 4.3.2. There is a large percentage of spectra predicted as large NAT by the proposed ML methods that are instead classified as STS by the BC, especially in the results of the RF scheme. This is probably caused by the fact that the BC misclassifies spectra of large NAT, as discussed in Sect. 4.2 for the CSDB.

5 Summary and conclusions

In this study, we investigated whether ML methods can be applied for the PSC classification of infrared limb spectra. We compared the classification results obtained by three different ML methods – PCA + SVM, KPCA + SVM, and RF – with those of the Bayesian classifier introduced by Spang et al. (2016). First, we discussed PCA, KPCA, and RF as methods for feature extraction from midinfrared spectral regions and showed that the selected features correspond with distinct features in the complex refractive indices of NAT and ice PSCs. Then we compared classification results obtained by the ML methods with respect to previous work using conventional classification methods combined with a Bayesian approach.

We presented three case studies as well as seasonal analyses for the validation and comparison of the classification results. Based on the case studies, we showed that there is spatial agreement of the ML method predictions between ice and nonice PSCs. However, there is some disagreement be-

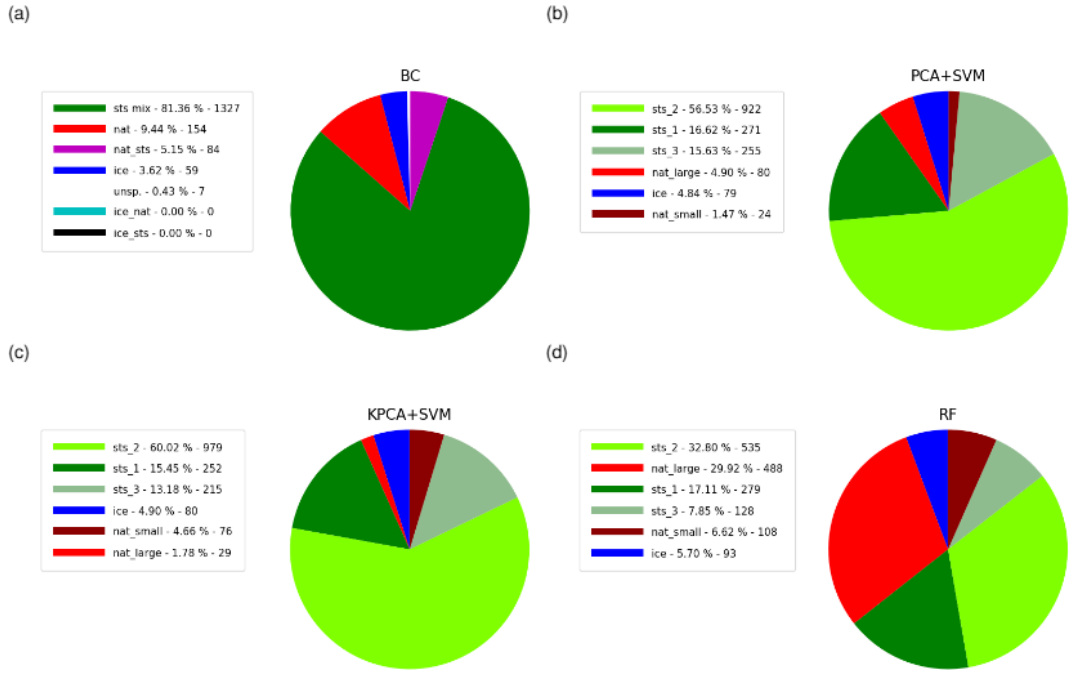


Figure 15. Same as Fig. 14 but for November 2006 to February 2007 for the Northern Hemisphere.

tween NAT and STS. We evaluated time series and pie charts of cloud coverage for the Southern Hemisphere polar winter 2009 and the Northern Hemisphere polar winter 2006/2007, showing that all methods are highly consistent with respect to the classification of ice. For the NAT and STS predictions, RF and the Bayesian classifier tend to agree best, whereas the SVM methods yielded larger differences. The agreement between the different classification schemes was further quantified by means of cross tabulation. While the SVM methods found significantly less NAT than the Bayesian classifier, the RF classifier found slightly more NAT than the Bayesian classifier. The RF results might be more realistic, because the Bayesian classifier is known to find less NAT for MIPAS compared to CALIOP satellite observations, especially for Northern Hemisphere winter conditions (Spang et al., 2016). A practical advantage of RF, presented in Sect. 3.3 and further discussed in Sect. 4.1, is that it enables a better control on the importance of the features it selects to train the model. Moreover, RF is a fully supervised method, from feature selection to training, whereas the feature extraction methods PCA and KPCA are unsupervised methods and may fail to capture important features if they do not show high variance. From the user point of view, RF is also simpler to deploy since it embeds feature selection and does not require a two-step process of feature extraction and training (unlike PCA + SVM and KPCA + SVM). Parallel implementations of the ML methods presented in this paper are also available, enabling significant acceleration of model training and prediction with a large number of data (Cavallaro et al., 2015; Genuer et al., 2017).

The Bayesian method developed by Spang et al. (2016) requires a priori knowledge of a domain expert to select the decision boundaries and to tune the probabilities used for classification for different areas in the feature space. The ML schemes proposed in this work are more objective in the premises and rely only on the available training data without additional assumptions. Models have been trained on the CSDB, a simulation dataset that has been created systematically sampling the parameter space, not reflecting the natural occurrence frequencies of parameters. This point is in our opinion of great importance, as we demonstrated that ML methods are capable of predicting PSC composition classes without the need of substantial prior knowledge, providing a means for consistency checking of subjective assessments. Although the lack of ground truth narrows the assessment down to comparison with other classification schemes, we found that the classification results of the ML methods are consistent with spectral features of the PSC particles, in particular, the features found in the real and imaginary part of their refractive indices. Another important benefit of the pro-

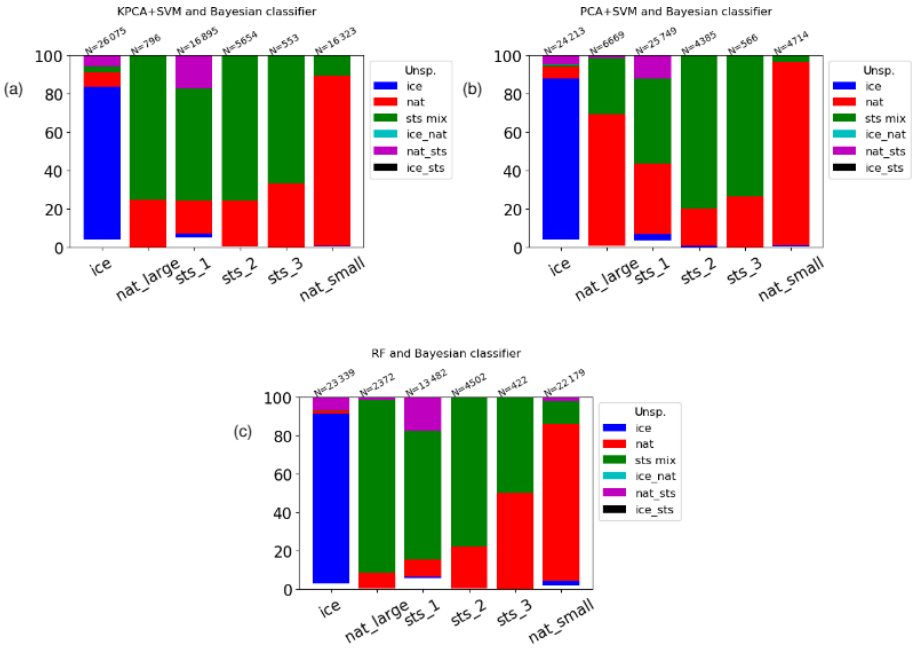


Figure 16. Intercomparison of ML and Bayesian classifiers for Southern Hemisphere winter (May to September 2009). Ticks on the x axis represent the classes of the KPCA + SVM classifier **(a)**, the PCA + SVM classifier **(b)**, and the RF classifier **(c)**. The y axis indicates the fraction of the classes as predicted by the Bayesian classifier. N is the number of samples belonging to each class of the ML classifiers.

posed ML methods is that they have shown the potential of extending the prediction to NAT particles with large radius, which was not possible with the BC scheme. This aspect has been successfully tested on the synthetic CSDB dataset and might be a promising path for future research.

However, there are still some limitations to the proposed ML approach. First, the feature selection methods found the highest variance and feature importance at spectral windows where ice and NAT have pronounced features in the complex refractive indices, whereas the main features of STS are located at wavenumbers not covered by the CSDB. Since the classification of STS is therefore based on the absence of features in the optical properties and for the large NAT particles the features in the optical properties vanish as well, the discrimination between STS and large NAT is more complicated than the identification of ice. Hence, we suppose that the inclusion of more spectral windows, especially regions where the optical properties of STS have features, may bear the potential to improve the separation between STS and NAT. Second, we showed that using a much smaller subset of the original CSDB for training of the ML methods would have been sufficient to achieve similar classification results. This suggests that the information provided by the CSDB is largely redundant, at least in terms of training of the ML methods. Despite the fact that the CSDB contains many training spectra, it was calculated only for a limited number of PSC volume densities, particle sizes, and cloud layer heights and depths as well as fixed atmospheric background conditions. It could be helpful to test the ML methods using a training dataset providing better coverage of the micro- and macrophysical parameter space and more variability in the atmospheric background conditions. Third, in the CSDB and the ML classification schemes we assumed only pure constituent (ice, NAT, STS 1, STS 2, and STS 3) PSCs, whereas in the atmosphere mixed clouds are frequently observed (e.g., Deshler et al., 2003; Pitts et al., 2018). In future work, mixed PSCs should be included, as an investigation of mixed PSCs could be beneficial to assess how far the ML methods applied to limb infrared spectra agree with predictions from CALIOP measurements that already comprise mixed-type scenarios.

In general, the presented classification methods are straightforward to adopt on spectrally resolved measurements of other infrared limb sensors like the Cryogenic Infrared Spectrometers and Telescopes for the Atmosphere (CRISTA) (Offermann et al., 1999) or the GLObal limb Radiance Imager for the Atmosphere (GLORIA) (Riese et al., 2005, 2014; Ungermann et al., 2010) space- or airborne instruments. It could be of interest to extend the methods to

combine different observational datasets, even with different types of sensors providing different spectral and geometrical properties of their acquisitions. This study has assessed the potential of ML methods in predicting PSC composition classes, which may be a starting point for new classification schemes for different aerosol types in the upper troposphere and lower stratosphere region (Sembhi et al., 2012; Griessbach et al., 2014, 2016), helping to answer open questions about the role of these particles in the atmospheric radiation budget.

Code and data availability. The MIPAS Level 1B IPF version 7.11 data can be accessed via ESA's Earth Online portal at https://earth.esa.int/web/guest/-/mipas-localized-calibrated-emission-spectra-1541 (ESA, 2019). The CSDB database can be obtained by contacting Michael Höpfner, Karlsruhe. The software repository containing the ML codes developed for this study is available at https://gitlab.com/rocco.sedona/psc_mipas_classification (Sedona, 2020).

Author contributions. GC, LH, and ReS developed the concept for this study. RoS developed the software and conducted the formal analysis of the results. SG, MH, and ReS provided expertise on the MIPAS measurements. MH prepared and provided the CSDB. GC and MR provided expertise on the ML methods. RoS wrote the manuscript with contributions from all coauthors.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank the European Space Agency (ESA) for making the Envisat MIPAS data available. We found the scikit-learn software package (https://scikit-learn.org/, last access: 10 December 2019) of great importance for the development of the code for this study.

Financial support. The article processing charges for this open-access publication were covered by a Research Centre of the Helmholtz Association.

Review statement. This paper was edited by Christian von Savigny and reviewed by two anonymous referees.

References

Achtert, P. and Tesche, M.: Assessing lidar-based classification schemes for polar stratospheric clouds based on 16 years of measurements at Esrange, Sweden, J. Geophys. Res.-Atmos., 119, 1386–1405, https://doi.org/10.1002/2013jd020355, 2014.

Adriani, A.: Climatology of polar stratospheric clouds based on lidar observations from 1993 to 2001 over McMurdo Station, Antarctica, J. Geophys. Res., 109, D24, https://doi.org/10.1029/2004jd004800, 2004.

Arnone, E., Castelli, E., Papandrea, E., Carlotti, M., and Dinelli, B. M.: Extreme ozone depletion in the 2010–2011 Arctic winter stratosphere as observed by MIPAS/ENVISAT using a 2-D tomographic approach, Atmos. Chem. Phys., 12, 9149–9165, https://doi.org/10.5194/acp-12-9149-2012, 2012.

Bergstra, J. and Bengio, Y.: Random search for hyper-parameter optimization, J. Mach. Learn. Res., 13, 281–305, 2012.

Biele, J., Tsias, A., Luo, B. P., Carslaw, K. S., Neuber, R., Beyerle, G., and Peter, T.: Nonequilibrium coexistence of solid and liquid particles in Arctic stratospheric clouds, J. Geophys. Res.-Atmos., 106, 22991–23007, https://doi.org/10.1029/2001jd900188, 2001.

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A.: Feature selection for high-dimensional data, Progress in Artificial Intelligence, Springer-Verlag, Berlin, Heidelberg, https://doi.org/10.1007/s13748-015-0080-y, 2016.

Breiman, L.: Machine Learning, 45, 5–32, https://doi.org/10.1023/a:1010933404324, 2001.

Brereton, R. G. and Lloyd, G. R.: Support Vector Machines for classification and regression, The Analyst, 135, 230–267, https://doi.org/10.1039/b918972f, 2010.

Browell, E. V., Butler, C. F., Ismail, S., Robinette, P. A., Carter, A. F., Higdon, N. S., Toon, O. B., Schoeberl, M. R., and Tuck, A. F.: Airborne lidar observations in the wintertime Arctic stratosphere: Polar stratospheric clouds, Geophys. Res. Lett., 17, 385–388, https://doi.org/10.1029/gl017i004p00385, 1990.

Buontempo, C., Cairo, F., Di Donfrancesco, G., Morbidini, R., Viterbini, M., and Adriani, A.: Optical measurements of atmospheric particles from airborne platforms: In situ and remote sensing instruments for balloons and aircrafts, Ann. Geophys., 49, 57–64, https://doi.org/10.4401/ag-3149, 2009.

Campbell, J. R. and Sassen, K.: Polar stratospheric clouds at the South Pole from 5 years of continuous lidar data: Macrophysical, optical, and thermodynamic properties, J. Geophys. Res., 113, D20204, https://doi.org/10.1029/2007jd009680, 2008.

Carslaw, K. S., Luo, B., and Peter, T.: An analytic expression for the composition of aqueous HNO_3-H_2SO_4 stratospheric aerosols including gas phase removal of HNO_3, Geophys. Res. Lett., 22, 1877–1880, https://doi.org/10.1029/95gl01668, 1995.

Cavallaro, G., Riedel, M., Richerzhagen, M., Benediktsson, J. A., and Plaza, A.: On Understanding Big Data Impacts in Remotely Sensed Image Classification Using Support Vector Machine Methods, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens., 8, 4634–4646, 2015.

Ceriani, L. and Verme, P.: The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini, J. Econ. Inequal., 10, 421–443, https://doi.org/10.1007/s10888-011-9188-x, 2012.

Cortes, C. and Vapnik, V.: Support-Vector Networks, Mach. Learn., 20, 273–297, https://doi.org/10.1023/A:1022627411411, 1995.

Deshler, T., Larsen, N., Weissner, C., Schreiner, J., Mauersberger, K., Cairo, F., Adriani, A., Di Donfrancesco, G., Ovarlez, J., Ovarlez, H., Blum, U., Fricke, K. H., and Dornbrack, A.: Large nitric acid particles at the top of an Arctic stratospheric cloud, J. Geo-

phys. Res., 108, 4517, https://doi.org/10.1029/2003JD003479, 2003.

Dudhia, A., Morris, P. E., and Wells, R. J.: Fast monochromatic radiative transfer calculations for limb sounding, J. Quant. Spectrosc. Ra. T., 74, 745–756, 2002.

ESA: MIPAS geo-located and calibrated atmospheric spectra (ENVISAT.MIP.NL_1P), available at: https://earth.esa.int/web/guest/-/mipas-localized-calibrated-emission-spectra-1541 last access: 10 December 2019.

Estornell, J., Martí-Gavliá, J. M., Sebastiá, M. T., and Mengual, J.: Principal component analysis applied to remote sensing, Model. Sci. Educ. Learn., 6, 83–89, https://doi.org/10.4995/msel.2013.1905, 2013.

Fauvel, M., Chanussot, J., and Benediktsson, J. A.: Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas, Eurasip J. Adv. Sign. Process., 2009, 783194, https://doi.org/10.1155/2009/783194, 2009.

Fischer, H., Birk, M., Blom, C., Carli, B., Carlotti, M., von Clarmann, T., Delbouille, L., Dudhia, A., Ehhalt, D., Endemann, M., Flaud, J. M., Gessner, R., Kleinert, A., Koopman, R., Langen, J., López-Puertas, M., Mosner, P., Nett, H., Oelhaf, H., Perron, G., Remedios, J., Ridolfi, M., Stiller, G., and Zander, R.: MIPAS: an instrument for atmospheric and climate research, Atmos. Chem. Phys., 8, 2151–2188, https://doi.org/10.5194/acp-8-2151-2008, 2008.

Fromm, M., Alfred, J., and Pitts, M.: A unified, long-term, high-latitude stratospheric aerosol and cloud database using SAM II, SAGE II, and POAM II/III data: Algorithm description, database definition, and climatology, J. Geophys. Res., 108, 4366, https://doi.org/10.1029/2002jd002772, 2003.

Genton, M.: Classes of kernels for machine learning: a statistics perspective, J. Mach. Learn. Res., 2, 299–312, 2002.

Genuer, R., Poggi, J.-M., Tuleau-Malot, C., and Villa-Vialaneix, N.: Random Forests for Big Data, Big Data Res., 9, 28–46, https://doi.org/10.1016/j.bdr.2017.07.003, 2017.

Griessbach, S., Hoffmann, L., Spang, R., and Riese, M.: Volcanic ash detection with infrared limb sounding: MIPAS observations and radiative transfer simulations, Atmos. Meas. Tech., 7, 1487–1507, https://doi.org/10.5194/amt-7-1487-2014, 2014.

Griessbach, S., Hoffmann, L., Spang, R., von Hobe, M., Müller, R., and Riese, M.: Infrared limb emission measurements of aerosol in the troposphere and stratosphere, Atmos. Meas. Tech., 9, 4399–4423, https://doi.org/10.5194/amt-9-4399-2016, 2016.

Griessbach, S., Hoffmann, L., Spang, R., Achtert, P., von Hobe, M., Mateshvili, N., Müller, R., Riese, M., Rolf, C., Seifert, P., and Vernier, J.-P.: Aerosol and cloud top height information of Envisat MIPAS measurements, Atmos. Meas. Tech., 13, 1243–1271, https://doi.org/10.5194/amt-13-1243-2020, 2020.

Hoffmann, L., Spang, R., Orr, A., Alexander, M. J., Holt, L. A., and Stein, O.: A decadal satellite record of gravity wave activity in the lower stratosphere to study polar stratospheric cloud formation, Atmos. Chem. Phys., 17, 2901–2920, https://doi.org/10.5194/acp-17-2901-2017, 2017.

Höpfner, M., Larsen, N., Spang, R., Luo, B. P., Ma, J., Svendsen, S. H., Eckermann, S. D., Knudsen, B., Massoli, P., Cairo, F., Stiller, G., v. Clarmann, T., and Fischer, H.: MIPAS detects Antarctic stratospheric belt of NAT PSCs caused by mountain waves, Atmos. Chem. Phys., 6, 1221–1230, https://doi.org/10.5194/acp-6-1221-2006, 2006.

Höpfner, M., Luo, B. P., Massoli, P., Cairo, F., Spang, R., Snels, M., Di Donfrancesco, G., Stiller, G., von Clarmann, T., Fischer, H., and Biermann, U.: Spectroscopic evidence for NAT, STS, and ice in MIPAS infrared limb emission measurements of polar stratospheric clouds, Atmos. Chem. Phys., 6, 1201–1219, https://doi.org/10.5194/acp-6-1201-2006, 2006.

Höpfner, M., Pitts, M. C., and Poole, L. R.: Comparison between CALIPSO and MIPAS observations of polar stratospheric clouds, J. Geophys. Res., 114, D00H05, https://doi.org/10.1029/2009JD012114, 2009.

Huang, H.-L. and Antonelli, P.: Application of Principal Component Analysis to High-Resolution Infrared Measurement Compression and Retrieval, J. Appl. Meteorol., 40, 365–388, https://doi.org/10.1175/1520-0450(2001)040<0365:AOPCAT>2.0.CO;2, 2001.

Jolliffe, I. T. and Cadima, J.: Principal component analysis: a review and recent developments, Philos. Trans. Roy. Soc. A-Math., 374, 20150 202, https://doi.org/10.1098/rsta.2015.0202, 2016.

Khosrawi, F., Kirner, O., Stiller, G., Höpfner, M., Santee, M. L., Kellmann, S., and Braesicke, P.: Comparison of ECHAM5/MESSy Atmospheric Chemistry (EMAC) simulations of the Arctic winter 2009/2010 and 2010/2011 with Envisat/MIPAS and Aura/MLS observations, Atmos. Chem. Phys., 18, 8873–8892, https://doi.org/10.5194/acp-18-8873-2018, 2018.

Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference of Artificial Intelligence, 14, 1137–1145, 1995.

Lambert, A., Santee, M. L., Wu, D. L., and Chae, J. H.: A-train CALIOP and MLS observations of early winter Antarctic polar stratospheric clouds and nitric acid in 2008, Atmos. Chem. Phys., 12, 2899–2931, https://doi.org/10.5194/acp-12-2899-2012, 2012.

Liu, Y., Wang, Y., and Zhang, J.: New Machine Learning Algorithm: Random Forest, in: Information Computing and Applications, pp. 246–252, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-34062-8_32, 2012.

Lowe, D. and MacKenzie, A. R.: Polar stratospheric cloud microphysics and chemistry, J. Atm. Sol.-Terr. Phys., 70, 13–40, https://doi.org/10.1016/j.jastp.2007.09.011, 2008.

Molleker, S., Borrmann, S., Schlager, H., Luo, B., Frey, W., Klingebiel, M., Weigel, R., Ebert, M., Mitev, V., Matthey, R., Woiwode, W., Oelhaf, H., Dörnbrack, A., Stratmann, G., Grooß, J.-U., Günther, G., Vogel, B., Müller, R., Krämer, M., Meyer, J., and Cairo, F.: Microphysical properties of synoptic-scale polar stratospheric clouds: in situ measurements of unexpectedly large HNO_3-containing particles in the Arctic vortex, Atmos. Chem. Phys., 14, 10785–10801, https://doi.org/10.5194/acp-14-10785-2014, 2014.

Offermann, D., Grossmann, K.-U., Barthol, P., Knieling, P., Riese, M., and Trant, R.: Cryogenic Infrared Spectrometers and Telescopes for the Atmosphere (CRISTA) experiment and middle atmosphere variability, J. Geophys. Res., 104, 16311–16325, 1999.

Patle, A. and Chouhan, D. S.: SVM kernel functions for classification, in: 2013 International Conference on Advances in Technology and Engineering (ICATE), Mumbai, pp. 1–9, IEEE, https://doi.org/10.1109/icadte.2013.6524743, 2013.

Pawson, S., Naujokat, B., and Labitzke, K.: On the polar stratospheric cloud formation potential of the northern stratosphere, J.

Geophys. Res., 100, 23215, https://doi.org/10.1029/95jd01918, 1995.

Pitts, M. C., Poole, L. R., and Thomason, L. W.: CALIPSO polar stratospheric cloud observations: second-generation detection algorithm and composition discrimination, Atmos. Chem. Phys., 9, 7577–7589, https://doi.org/10.5194/acp-9-7577-2009, 2009.

Pitts, M. C., Poole, L. R., Dörnbrack, A., and Thomason, L. W.: The 2009–2010 Arctic polar stratospheric cloud season: a CALIPSO perspective, Atmos. Chem. Phys., 11, 2161–2177, https://doi.org/10.5194/acp-11-2161-2011, 2011.

Pitts, M. C., Poole, L. R., and Gonzalez, R.: Polar stratospheric cloud climatology based on CALIPSO spaceborne lidar measurements from 2006 to 2017, Atmos. Chem. Phys., 18, 10881–10913, https://doi.org/10.5194/acp-18-10881-2018, 2018.

Probst, P., Wright, M. N., and Boulesteix, A.-L.: Hyperparameters and tuning strategies for random forest, WIRES Data Mining Knowledge Discovery, 9, e1301, https://doi.org/10.1002/widm.1301, 2019.

Raspollini, P., Belotti, C., Burgess, A., Carli, B., Carlotti, M., Ceccherini, S., Dinelli, B. M., Dudhia, A., Flaud, J.-M., Funke, B., Höpfner, M., López-Puertas, M., Payne, V., Piccolo, C., Remedios, J. J., Ridolfi, M., and Spang, R.: MIPAS level 2 operational analysis, Atmos. Chem. Phys., 6, 5605–5630, https://doi.org/10.5194/acp-6-5605-2006, 2006.

Raspollini, P., Carli, B., Carlotti, M., Ceccherini, S., Dehn, A., Dinelli, B. M., Dudhia, A., Flaud, J.-M., López-Puertas, M., Niro, F., Remedios, J. J., Ridolfi, M., Sembhi, H., Sgheri, L., and von Clarmann, T.: Ten years of MIPAS measurements with ESA Level 2 processor V6 – Part 1: Retrieval algorithm and diagnostics of the products, Atmos. Meas. Tech., 6, 2419–2439, https://doi.org/10.5194/amt-6-2419-2013, 2013.

Riese, M., Friedl-Vallon, F., Spang, R., Preusse, P., Schiller, C., Hoffmann, L., Konopka, P., Oelhaf, H., von Clarmann, T., and Höpfner, M.: GLObal limb Radiance Imager for the Atmosphere (GLORIA): Scientific objectives, Adv. Space Res., 36, 989–995, 2005.

Riese, M., Oelhaf, H., Preusse, P., Blank, J., Ern, M., Friedl-Vallon, F., Fischer, H., Guggenmoser, T., Höpfner, M., Hoor, P., Kaufmann, M., Orphal, J., Plöger, F., Spang, R., Suminska-Ebersoldt, O., Ungermann, J., Vogel, B., and Woiwode, W.: Gimballed Limb Observer for Radiance Imaging of the Atmosphere (GLORIA) scientific objectives, Atmos. Meas. Tech., 7, 1915–1928, https://doi.org/10.5194/amt-7-1915-2014, 2014.

Salawitch, R., Wofsy, S., Gottlieb, E., Lait, L., Newman, P., Schoeberl, M., Loewenstein, M., Podolske, J., Strahan, S., Proffitt, M., Webster, C., May, R., Fahey, D., Baumgardner, D., Dye, J., Wilson, J., Kelly, K., Elkins, J., Chan, K., and Anderson, J.: Chemical Loss of Ozone in the Arctic Polar Vortex in the Winter of 1991–1992, Science, 261, 1146–1149, https://doi.org/10.1126/science.261.5125.1146, 1993.

Schölkopf, B., Smola, A., and Müller, K. R.: Kernel principal component analysis, in: Artificial Neural Networks – ICANN'97, edited by: Gerstner, W., Germond, A., Hasler, M., Nicoud, J. D., ICANN 1997, Lecture Notes in Computer Science, vol. 1327, Springer, Berlin, Heidelberg, pp. 583–588, https://doi.org/10.1007/BFb0020217, 1997.

Sedona, R.: PSC MIPAS classification, available at: https://gitlab.com/rocco.sedona/psc_mipas_classification, last access: 19 May 2020.

Sembhi, H., Remedios, J., Trent, T., Moore, D. P., Spang, R., Massie, S., and Vernier, J.-P.: MIPAS detection of cloud and aerosol particle occurrence in the UTLS with comparison to HIRDLS and CALIOP, Atmos. Meas. Tech., 5, 2537–2553, https://doi.org/10.5194/amt-5-2537-2012, 2012.

Solomon, S.: Stratospheric ozone depletion: A review of concepts and history, Rev. Geophys., 37, 275–316, https://doi.org/10.1029/1999RG900008, 1999.

Spang, R. and Remedios, J. J.: Observations of a distinctive infrared spectral feature in the atmospheric spectra of polar stratospheric clouds measured by the CRISTA instrument, Geophys. Res. Lett., 30, 1875, https://doi.org/10.1029/2003GL017231, 2003.

Spang, R., Remedios, J. J., and Barkley, M. P.: Colour indices for the detection and differentiation of cloud type in infra-red limb emission spectra, Adv. Space Res., 33, 1041–1047, 2004.

Spang, R., Arndt, K., Dudhia, A., Höpfner, M., Hoffmann, L., Hurley, J., Grainger, R. G., Griessbach, S., Poulsen, C., Remedios, J. J., Riese, M., Sembhi, H., Siddans, R., Waterfall, A., and Zehner, C.: Fast cloud parameter retrievals of MIPAS/Envisat, Atmos. Chem. Phys., 12, 7135–7164, https://doi.org/10.5194/acp-12-7135-2012, 2012.

Spang, R., Hoffmann, L., Höpfner, M., Griessbach, S., Müller, R., Pitts, M. C., Orr, A. M. W., and Riese, M.: A multi-wavelength classification method for polar stratospheric cloud types using infrared limb spectra, Atmos. Meas. Tech., 9, 3619–3639, https://doi.org/10.5194/amt-9-3619-2016, 2016.

Spang, R., Hoffmann, L., Müller, R., Grooß, J.-U., Tritscher, I., Höpfner, M., Pitts, M., Orr, A., and Riese, M.: A climatology of polar stratospheric cloud composition between 2002 and 2012 based on MIPAS/Envisat observations, Atmos. Chem. Phys., 18, 5089–5113, https://doi.org/10.5194/acp-18-5089-2018, 2018.

Stiller, G. P., Hoepfner, M., Kuntz, M., von Clarmann, T., Echle, G., Fischer, H., Funke, B., Glatthor, N., Hase, F., Kemnitzer, H., and Zorn, S.: Karlsruhe optimized and precise radiative transfer algorithm. Part I: requirements, justification, and model error estimation, in: Optical Remote Sensing of the Atmosphere and Clouds, Proc. SPIE, 3501, https://doi.org/10.1117/12.317754, 1998.

Tharwat, A.: Classification assessment methods, Appl. Comput. Inf., in press, https://doi.org/10.1016/j.aci.2018.08.003, 2018.

Toon, O. B., Browell, E. V., Kinne, S., and Jordan, J.: An analysis of lidar observations of polar stratospheric clouds, Geophys. Res. Lett., 17, 393–396, https://doi.org/10.1029/gl017i004p00393, 1990.

Tritscher, I., Grooß, J.-U., Spang, R., Pitts, M. C., Poole, L. R., Müller, R., and Riese, M.: Lagrangian simulation of ice particles and resulting dehydration in the polar winter stratosphere, Atmos. Chem. Phys., 19, 543–563, https://doi.org/10.5194/acp-19-543-2019, 2019.

Ungermann, J., Kaufmann, M., Hoffmann, L., Preusse, P., Oelhaf, H., Friedl-Vallon, F., and Riese, M.: Towards a 3-D tomographic retrieval for the air-borne limb-imager GLORIA, Atmos. Meas. Tech., 3, 1647–1665, https://doi.org/10.5194/amt-3-1647-2010, 2010.

Voigt, C.: Nitric Acid Trihydrate (NAT) in Polar Stratospheric Clouds, Science, 290, 1756–1758, https://doi.org/10.1126/science.290.5497.1756, 2000.

Wolpert, D. H.: The Lack of A Priori Distinctions Between Learning Algorithms, Neural Comput., 8, 1341–1390, https://doi.org/10.1162/neco.1996.8.7.1341, 1996.

Zeiler, M. D. and Fergus, R.: Visualizing and understanding convolutional networks, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), available at: https://doi.org/10.1007/978-3-319-10590-1_53, 2014.

ENHANCING LARGE BATCH SIZE TRAINING OF DEEP MODELS FOR REMOTE SENSING APPLICATIONS

Rocco Sedona[1,2]*, Gabriele Cavallaro*[2]*, Morris Riedel*[1,2]*, and Matthias Book*[2]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] School of Engineering and Natural Sciences, University of Iceland, Iceland

IGARSS 2021 - 2021 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-0369-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/IGARSS47720.2021.9555136

ABSTRACT

A wide variety of Remote Sensing (RS) missions are continuously acquiring a large volume of data every day. The availability of large datasets has propelled Deep Learning (DL) methods also in the RS domain. Convolutional Neural Networks (CNNs) have become the state of the art when tackling the classification of images, however the process of training is time consuming. In this work we exploit the Layer-wise Adaptive Moments optimizer for Batch training (LAMB) optimizer to use large batch size training on High-Performance Computing (HPC) systems. With the use of LAMB combined with learning rate scheduling and warm-up strategies, the experimental results on RS data classification demonstrate that a ResNet50 can be trained faster with batch sizes up to 32K.

***Index Terms*—** Distributed deep learning, high performance computing, residual neural network, convolutional neural network, classification, deepsat

1. INTRODUCTION

Deep Learning (DL) is emerging as the leading Artificial Intelligence (AI) technique owing to the current convergence of scalable computing capability (i.e., HPC and Cloud computing), easy access to large volumes of data, and the emergence of new algorithms enabling robust training of large-scale deep CNNs [1].

Recent HPC architectures and parallel programming have been influenced by the rapid advancement of DL and hardware accelerators (e.g., GPUs). The classical workloads that run on HPC systems (e.g., numerical methods based on physical laws in various scientific fields) are becoming more heterogeneous. They are being transformed by DL algorithms that require higher memory, storage, and networking capabilities, as well as optimized software and libraries, to deliver the required performance [2].

HPC systems are an effective solution that deals with the challenges posed by big RS data. Modern Earth Observation (EO) programs (e.g., ESA's Copernicus) provide continuous streams of massive volumes of multi-sensor RS data on a daily basis [1].

While DL has provided numerous breakthrough for many RS applications, some challenges are still unsolved. The deployment of a deep model can produce a neural network architecture with a significant number of tunable parameters (i.e., millions for a ResNet50 architecture [3]), which requires a large amount of time to complete its training.

To achieve high scalability performance over a large number of GPUs the main approach is to increase the effective batch size (i.e., the batch size per worker multiplied by the number of workers)[4]. However, it was noted that the use of the popular the Stochastic Gradient Descent (SGD) optimizer in a setting with batch sizes larger than 8K can lead to substantial degradation of performance, e.g., classification accuracy, if used without any additional countermeasures [5].

One mechanism to avoid this difficulty is tuning the learning rate schedule that uses warm-up phases before the training, scales learning rate with the number of distributed workers, and reduces the rate according to a fixed factor after a fixed number of epochs [6]. More sophisticated strategies to deal with very large batch sizes use adaptive learning rates that are tuned dependent on layer depth, value of computed gradients and progress of training. In [7] the authors showed that the use of a learning rate with linear scaling w.r.t. the number of GPUs, step dacay, and warm-up allowed training DL models on a RS dataset with batch sizes up to 8K.

In this study, we propose to use the recently presented LAMB [8] optimizer with a multifold strategy consisting of a learning rate scheduler with polynomial decay that calculates the initial learning rate with a non-linear rule. We utilized also a warm-up phase at the beginning of the training with length proportional to the batch size [4] [5] [8]. We demonstrate that this training strategy and the adoption LAMB optimizer can scale the training of a ResNet50 for the classification of two RS datasets, using batch sizes up to 32K without a significant degradation of the accuracy.

The results of this research were achieved through the support of HELMHOLTZ AI CONSULTANTS @ FZJ https://www.helmholtz.ai/themenmenue/our-research/consultant-teams/helmholtz-ai-consultants-fzj/index.html

[1] https://sentinels.copernicus.eu/web/sentinel/news/-/article/2018-sentinel-data-access-annual-report

2. PROBLEM FORMULATION

Distributed computing frameworks such as the TensorFlow native mirrored and parameter server strategies, PyTorch Distributed, Horovod [9] have gained visibility lately, enabling a faster trainining of deep neural networks on large datasets [4]. There are two approaches for distributed training, the model distribution and the data distribution ((i.e., data parallelism)) [10]. In this work we used data distribution and run the experiments on one of the HPC systems hosted at the Jülich Supercomputing Centre (JSC). Data distributed frameworks are more straightforward to implement and require less hand-tuning. In data parallelism the DL model is replicated on each worker and data are divided in different chunks among the workers. The training of the models is then executed in parallel, where each replica performs backpropagation on different data. At the end of each iteration the models exchange their local parameters between each other in a synchronized way. In this work, we adopted the Horovod library due to its flexible API that can be used on top of the most popular DL libraries such as TensorFlow, Keras, PyTorch and MXNet. Horovod relies on Message Passing Interface (MPI) and NVIDIA Collective Communication Library (NCCL) libraries for the synchronization of the model parameters among the different workers, which is performed using a decentralized ring-allreduce algorithm [9].

3. METHODOLOGY

3.1. ResNet50

ResNet50 [3] was presented in 2015 and it is still among the most widely used CNNs for solving various computer vision tasks. Although stacking a large number of layers to create a deep neural network would intuitively provide very powerful and expressive models, in practice the training becomes more difficult due to the so called vanishing gradient problem. It was noted that in deep neural networks the gradient becomes small as a function of the depth, preventing the model from updating the weights [11]. ResNet50 aims at overcoming this issue by adopting the skip connections: instead of directly fitting the underlying mapping $H(x)$, the residual mapping $F(x) := H(x) - x$ is learned [3]. Implementing the skip connections as identity mappings ($F(x) + x$), [3] creates a deep CNN solving the vanishing gradient problem.

3.2. LAMB optimizer

With the data distribution parallel strategy the effective batch size is the result of the multiplication of the per-worker batch size by the number of workers. The adoption of the SGD optimizer was shown to help tackling optimization problems with batch size up to 8K, used in combination with a strategy that computes the initial learning rate according to a linear scaling rule and a warm-up phase [4]. However, above the threshold of 8K, this solution is not sufficient to train a model without degradation of the results during testing. In [5], the authors found that if the ratio of the L2-norm of weights and gradients is high, the training can become unstable. The LAMB optimizer has been specifically proposed to improve the training stability and generalization performance [8]. LAMB is based on the popular optimization algorithm adaptive learning rate optimization algorithm (ADAM) [12]. In contrast to SGD, LAMB is a layer-wise adaptive algorithm that adopts a per dimension normalization with respect to the square root of the second moment and a layer-wise normalization. The general rule for updating the parameters with iterative algorithms such as ADAM and SGD is:

$$x_{t+1} = x_t + \eta_t u_t, \tag{1}$$

where x are the parameters of the model, η is the learning rate and u is the update of the parameters. For the layer-wise adaptive strategies the formula becomes:

$$x^i_{t+1} = x^i_t - \eta_t \frac{\Phi(\|x^i_t\|)}{\|g^i_t\|} g^i_t, \tag{2}$$

where x^i are the parameters at layer i, g^i the gradient at layer i, η_t is the learning rate at step t and Φ is a scaling function. Comparing the classical rule for the update of the weights (eq. 1) with the formula of the layer-wise adaptive strategy (eq. 2), we can observe that the two changes are the following: (i) the update is scaled to unit l_2-norm and (ii) an additional scaling Φ is applied [8].

4. EXPERIMENTAL RESULTS

4.1. Dataset

The experiments were carried out on the SAT-4 and SAT-6 airborne datasets [13]. The patches were created using the National Agriculture Imagery Program (NAIP) dataset, which consists of 330,000 scenes covering the Continental United States. The size of each patch is $28 \times 28 \times 4$. Each patch has 4 channels (i.e., RGB with near infrared) with 1m spatial resolution. Each patch is associated to one class. The SAT-4 dataset contains 500,000 patches ans includes annotations of four land cover classes, which are barren land, trees, grassland and a class that groups together everything that is not the three aforementioned. The dataset was split in a training set of 400.000 patches and a test set of 100.000 patches. Similarly to SAT-4, SAT-6 contains patches with size $28 \times 28 \times 4$, but the total number of image patches is 405.000 and is annotated with six landcover classes that are barren land, trees, grassland, roads, buildings and water bodies. The training set consists of 324.000 patches and the test set of 81.000 patches.

4.2. Experimental Setup

We used the Dynamical Exascale Entry Platform (DEEP), that is an European pre-exascale platform which incorporates

Batch size	8K	16K	32K	65K
Learning rate	0.02	0.028	0.04	0.05
Warm-up	5	10	20	40

Table 1. Hyper-parameters of LAMB optimizer as in the experimental setting of [8].

heterogeneous HPC systems. DEEP is being developed by the European project Dynamical Exascale Entry Platform – Extreme Scale Technologies (DEEP-EST) [2]. The Extreme Scale Booster (ESB) partition hosts 75 nodes, each equipped with 1 Nvidia V100 Tesla Graphics Processing Unit (GPU) (each with 32 GB of memory). To test whether the data distributed algorithm with large batch sizes can scale. we used up to 32 GPUs. We used `Python 3.8.5` and the following libraries for DL and the data distributed framework: `TensorFlow 2.3.1`, `Horovod 0.20.3`, `Scikit-learn 0.20.3` and `Scipy 1.5.2`. SAT-4 and SAT-6 are saved as MATLAB .mat files and were read using the Scipy library. We trained the models with the Keras API and built the input pipeling with the TensorFlow data API. We trained a ResNet-50 models from scratch, i.e. without loading pre-trained weights, on the datasets SAT4 and SAT6 [13]. Each patch of the dataset is associated to one of the classes, making this a patch-based multi-class classification problem. Thus, we stacked a fully connected layer (with 6 neurons for SAT-6 and 4 neurons for SAT-4) on top of the model activated with the softmax function. We selected a number of epochs equal to 100 for the training of the models. The initial learning rate was set using a heuristics that computes the learning rate proportionally to the root square of the effective batch size. We also adopted a polynomial scheduler of order 2 for the learning rate as shown in [8], as well as a warm-up that gradually ramps up the value of the learning rate at the beginning of the training. The combination of these techniques helps solving the problem of instability that can cause exploding gradients. The hyper-parameters were selected based on [8] and are shown in Tab. 1. As a baseline for comparison we also used the ADAM optimizer with a fixed learning rate set to 0.001 as in [12] and batch size equal to 64. We tested also the SGD optimizer with hyper-parameters as explained in [4], but the training did not converge in 100 epochs, thus a further exploration of the hyper-parameter space should be performed and results could be reported in future works. We implemented a simple data augmentation with random flips and rotation of the patches, which helps reducing the overfitting.

4.3. Evaluation

The accuracy and loss metrics shown in Tab. 2, 3 and 4 are the average of 3 runs for each set of hyper-parameters. Using LAMB and batch sizes up to 32K we could obtain results that are comparable to those obtained using small batch sizes and consistent with state of the art results [14]. In particular, we can see that the accuracy obtained by using LAMB with a batch size of 8K is very similar to that obtained using ADAM with a much smaller batch size equal to 64 on both datasets (shown in Tab. 2, 3 and 4). We did not test the ADAM optimizer since it is known that it tends not to generalize well on test data when large batch sizes are employed [15]. As stated above, results remain acceptable with batch sizes of 32K using the new LAMB approach, while they diverge with batch size equal to 65K. We can observe that as the batch size increases, the test losses and accuracies tend to increase and decrease respectively, up to the point where they significantly diverge from baseline results. This happens even though we did not observe training difficulties such as exploding gradient, a behaviour that was observed using SGD with large batches [4]. As the batch size grows, the generalization gap becomes non negligible [15] due to the fact that optimizers in the large batch size regime converge to sharp instead of flat minimizers [16].

Batch size	N. GPUs	Accuracy	Loss	Time [s]
8K	4	0.99	0.02	34
16K	8	0.98	0.07	18
32K	16	0.96	0.11	9
65K	32	diverges		5

Table 2. Accuracy and test loss, training time per epoch epoch with LAMB optimizer, dataset SAT4.

Batch size	N. GPUs	Accuracy	Loss	Time [s]
8K	4	0.99	0.05	41
16K	8	0.98	0.11	22
32K	16	0.94	0.17	11
65K	32	diverges		6

Table 3. Accuracy and test loss, training time per epoch epoch with LAMB optimizer, dataset SAT6.

The scaling in terms of time required to complete an epoch is slightly less than linear w.r.t. the number of GPUs that are employed for the training (Tab. 2 and 3). We hypothesize that the use of large per-worker batch size is beneficial for scaling. In fact, using large batch sizes the communication time (time spent to exchange the gradients between the workers) remains smaller than the computation time (time spent to propagate the batches back and forth in the CNN), reducing the GPUs idle time. A conclusive and thorough study of the possible set-ups could be beneficial also to other researchers in the field.

[2] https://www.deep-est.eu/

Dataset	Batch size	Accuracy	Loss	Time [s]
SAT-4	64	0.98	0.02	263
SAT-6	64	0.98	0.04	214

Table 4. Accuracy and test loss, training time per epoch with ADAM optimizer, dataset SAT4 and SAT6. These experiments were carried out on a single GPU.

5. CONCLUSIONS

In this work the LAMB optimizer was used to train a ResNet-50 model with large batch sizes up to 32K. The results obtained with the SAT-4 and SAT-6 RS datasets showed that the training performance remained unaffected and that processing speed up was achieved. Training the model with batch sizes above the threshold of 32K is still problematic, as it was shown in the results using a batch size equal to 65K. An additional consideration is that in the present work we used two RS datasets with a simple multi-class classification problem, but the question whether this approach could be extended to more complex classification problems is still without an answer. We are currently planning to work on a comparison with the Layer-wise Adaptive Rate Scaling (LARS) optimizer [5], which might be included in future publications. However, this work should be considered as a preliminary assessment and a systematic analysis that includes also other training strategies and algorithms to deal with large batch size should be undertaken, such as the adoption of a cyclical learning rate scheduler [17] and the dynamic increase of the batch size during the training [18]. A quantitative analysis that takes into consideration the optimal configuration for distributed DL such as the per-worker batch size or specific parameters of Horovod is also lacking at the moment and is in the future plans of the authors. The repository with the Python code is publicly available [3].

6. REFERENCES

[1] G. Fox, J. A. Glazier, and et al., "Learning Everywhere: Pervasive Machine Learning for Effective High-Performance Computation," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 422–429.

[2] T. Ben-Nun and T. Hoefler, "Demystifying Parallel and Distributed Deep Learning: An in-Depth Concurrency Analysis," *CoRR*, vol. abs/1802.09941, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[3] https://gitlab.version.fz-juelich.de/sedona3/igarss2021_sat6

[4] P. Dollár P. Goyal and et al., "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," *arXiv:1706.02677*.

[5] Y. You, I. Gitman, and B. Ginsburg, "Large Batch Training of Convolutional Networks," 2017.

[6] M. Yamazaki, A. Kasagi, and et al., "Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds," *arXiv preprint arXiv:1903.12650*, 2019.

[7] R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J.A. Benediktsson, "Remote Sensing Big Data Classification with High Performance Distributed Deep Learning," *Remote Sensing*, vol. 11, no. 24, pp. 3056, Dec 2019.

[8] Y. You, J. Li, and et al., "Large Batch Optimization for Deep Learning: Training BERT in 76 minutes," 2020.

[9] A. Sergeev and M. Del Balso, "Horovod: Fast and Easy Distributed Deep Learning in TensorFlow," *arXiv preprint arXiv:1802.05799*, 2018.

[10] T. Ben-Nun and T. Hoefler, "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis," *ACM Computing Surveys*, 2019.

[11] B. Hanin and D. Rolnick, "How to Start Training: The Effect of Initialization and Architecture," 2018.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[13] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat - A Learning Framework for Satellite Imagery," 2015.

[14] M. A. Kadhim and M. H. Abed, *Convolutional Neural Network for Satellite Image Classification*, p. 165–178, Springer International Publishing, Mar 2019.

[15] E. Hoffer, I. Hubara, and D. Soudry, "Train Longer, Generalize Better: Closing the Generalization Gap in Large Batch Training of Neural Networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, NIPS'17, p. 1729–1739, Curran Associates Inc.

[16] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," 2017.

[17] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," 2017.

[18] S. L. Smith, P.J. Kindermans, C. Ying, and Q.V. Le, "Don't Decay the Learning Rate, Increase the Batch Size," 2018.

A High-Performance Multispectral Adaptation GAN for Harmonizing Dense Time Series of Landsat-8 and Sentinel-2 Images

Rocco Sedona, *Student Member, IEEE*, Claudia Paris, *Member, IEEE*, Gabriele Cavallaro, *Member, IEEE*, Lorenzo Bruzzone, *Fellow, IEEE*, and Morris Riedel, *Member, IEEE*

***Abstract*—The combination of data acquired by Landsat-8 and Sentinel-2 earth observation missions produces dense time series (TSs) of multispectral images that are essential for monitoring the dynamics of land-cover and land-use classes across the earth's surface with high temporal resolution. However, the optical sensors of the two missions have different spectral and spatial properties, thus they require a harmonization processing step before they can be exploited in remote sensing applications. In this work, we propose a workflow-based on a deep learning approach to harmonize these two products developed and deployed on an high-performance computing environment. In particular, we use a multispectral generative adversarial network with a U-Net generator and a PatchGan discriminator to integrate existing Landsat-8 TSs with data sensed by the Sentinel-2 mission. We show a qualitative and quantitative comparison with an existing physical method [National Aeronautics and Space Administration (NASA) Harmonized Landsat and Sentinel (HLS)] and analyze original and generated data in different experimental setups with the support of spectral distortion metrics. To demonstrate the effectiveness of the proposed approach, a crop type mapping task is addressed using the harmonized dense TS of images, which achieved an overall accuracy of 87.83% compared to 81.66% of the state-of-the-art method.**

***Index Terms*—Deep learning (DL), dense time series (TSs), generative adversarial network (GAN), harmonization, high performance computing (HPC), Landsat-8, remote sensing (RS), sentinel-2, virtual constellation.**

I. Introduction

THE availability of multispectral images systematically acquired by remote sensing (RS) satellites is pivotal for the observation of land surface change and dynamic processes [1], such as changes resulting from natural calamities [2], expansion of urban areas [3], vegetation anomaly and phenology changes [4], distribution of surface water resources [5], deforestation [6], etc. The time series (TS) of multispectral images acquired by the NASA/United States Geological Survey (USGS)'s Landsat-8 [7] and the European Space Agency (ESA)'s Sentinel-2 [8] missions are the most widely accessible moderate-to-high spatial resolution RS satellite images.

Landsat-8 was launched in 2013 and carries the operational land imager (OLI) and the thermal infrared sensor (TIRS). It acquires multispectral images at 30 m spatial resolution, which is suitable for a wide variety of tasks. However, Landsat-8 can only revisit the same area every 16 days, which is not sufficient in applications requiring more frequent observations (e.g., near real-time monitoring of continuous processes [9]). The Sentinel-2 A and Sentinel-2B are the two polar orbiting satellites of the Sentinel-2 constellation that were launched in 2015 and 2017, respectively. This constellation can reach a revisit time of 5 days at the equator (and even less for areas covered by more than one orbit) and acquire 13 optical bands with 10, 20, and 60 m spatial resolution.

The starting of the Sentinel-2 mission has opened potential opportunities for combining its data with the ones acquired by Landsat-8 to achieve more dense observations. In particular, their integration can densify the acquired TSs and increase the revisit time up to 3–5 days [10] and obtain more frequent cloud-free surface observations. Furthermore, the spatial resolution and spectral configuration (i.e., placement and number of spectral bands) of the Sentinel-2 sensor were designed to be compatible to analogous bands in Satellite Pour l'Observation de la Terre (SPOT) and Landsat sensors [11]. Consequently, many research works have exploited virtual constellations of Sentinel-2 and Landsat-8 for addressing different types of applications, for example to assess winter wheat yields at regional scale [12], estimate number and timing of mowing events of grasslands [13], monitor aquatic systems [14], retrieve the temporal variations in biochemical and structural vegetation properties [15], estimate inland water quality [16], detect irrigated areas [17], analyze land productivity and yield assessment [18], map land surface phenology at continental scale [19], determine the spatial distribution of evergreen forest in cloudy and rainy areas [20], etc.

Manuscript received July 14, 2021; revised September 2, 2021; accepted September 22, 2021. Date of publication September 27, 2021; date of current version October 15, 2021. This work was supported in part by the CoE RAISE project from the European Union's Horizon 2020 Research and Innovation Framework Programme under Grant agreement 951733, in part by the ADMIRE project from the European Union's Horizon 2020 JTI-EuroHPC research and innovation programme under Grant agreement 956748, and in part by the DEEP-EST project, from the European Union's Horizon 2020 research and innovation programme under Grant agreement 754304. (*Corresponding author: Gabriele Cavallaro.)*

Rocco Sedona and Morris Riedel are with the Jülich Supercomputing Centre, 52428 Jülich, Germany, and also with the University of Iceland, 107 Reykjavik, Iceland (e-mail: r.sedona@fz-juelich.de; morris@hi.is).

Claudia Paris and Lorenzo Bruzzone are with the Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy (e-mail: claudia.paris@unitn.it; lorenzo.bruzzone@ing.unitn.it).

Gabriele Cavallaro is with the Jülich Supercomputing Centre, 52428 Jülich, Germany (e-mail: g.cavallaro@fz-juelich.de).

Digital Object Identifier 10.1109/JSTARS.2021.3115604

Despite the similarity between Sentinel-2 and Landsat-8 observations, the two missions have different spatial resolution, field of view spectral bandwidth, and spectral response function. Consequently, before using together Sentinel-2 and Landsat-8 images, it is necessary to apply models for cross-sensor data integration [21]–[23]. Linear regression is the most widely used approach to reduce the spectral differences between the two sensors. The authors in [24] used bidirectional reflectance distribution function (BRDF) correction and data resampling to attenuate the difference introduced by the different field of view and spatial resolution, respectively. Other studies designed regional fixed per-band transformation coefficients for applying reflectance adjustment in Australia [25], Southern Africa [26], and United States [27].

Since 2018, NASA is producing a Harmonized Landsat and Sentinel (HLS) dataset[1] to further improve the temporal resolution of the combined product [28]. NASA proposed a method that creates global fixed per-band transformation coefficients to reduce the reflectance difference between Landsat-8 and Sentinel-2 and generate smooth spectral TSs. In particular, the approach takes into account the differences in spatial resolution, atmospheric correction approaches, view geometry and radiometric characteristics of spectral bands. ESA has considered this approach as a reference work for the definition of the Sen2Like framework [29]. The objective of Sen2Like is to generate Sentinel-2 like harmonized/fused surface reflectances with higher periodicity by integrating additional compatible optical mission sensors. The current implementation (November 2020) can harmonize Landsat-8 and Sentinel-2 data products[2]. The authors in [30] observed that these methods can reduce the reflectance difference to only some degree. It is possible that the regional or global scale fixed per-band transformation coefficients may not be suitable for all land cover types and at all geographical locations. To mitigate this problem, they proposed a time-series-based approach[3] to improve the consistency of the HLS datasets, which uses the TSs of matched Landsat-8 and Sentinel-2 observations to build linear regression models for each pixel. They then conducted the reflectance adjustment for each individual pixel separately.

Instead of using a physical method or fitting the transformation coefficients of a linear regression, in our work we developed an approach based on machine learning (ML), and more specifically on a generative adversarial network (GAN) architecture to harmonize the Sentinel-2 and Landsat-8 products, transforming the data acquired by the Sentinel-2 multispectral instrument (MSI) sensor into Landsat-8 OLI-like data. In the last decade deep learning (DL) has enabled a leap in the quality of a wide variety of applications in remote sensing (RS) [31]. In particular, generative adversarial networks (GANs) were first presented by [32] in 2014 and are based on the training with the backpropagation algorithm of two submodels, a generator, and a discriminator. An extension of GANs are the conditional GANs [33], in which the generator is given additional information to better approximate the distribution of the real samples. The competitive game of one model against the other pushes the generator to create new fake examples that are indistinguishable from real ones. While the generator creates new data from an input distribution, the discriminator is devoted to discern the real and generated examples looking at their distribution. For these reasons, GAN have attracted much research efforts to computer-vision-related tasks [34].

GANs have been employed also in different RS applications. Among those, a promising application is super-resolution, where GANs offer the ability to retrieve high-frequency components that seem not to be captured by existing convolutional neural networks (CNNs) [35], thanks to the contribution of the adversarial loss [36]–[38]. Chen *et al.* [39] proposed a GAN-based approach to super-resolve Landsat-8 images and reconstruct them to be Sentinel-2-like using the true color composite of RGB bands. In our approach we propose the opposite direction of the data flow, from Sentinel-2 to Landsat-8 data, as our proposed method focuses on radiometric consistency rather than spatial resolution. Moreover, we also use the near infrared (NIR) and the short wave infrared (SWIR) bands, which are extremely important to perform environmental monitoring (e.g., vegetation biophysical and biochemical variable retrieval, ice detection, etc.). In particular, the NIR and SWIR spectral channels provide key information on vegetation and crops status. GANs have been applied also to other tasks, such as to enhance the detection of small objects in RS data with an adaptation of the enhanced super-resolution generative adversarial network (ESRGAN) [40], or to change detection with multi-sensor data with the use of a CycleGAN [41]. Conditional GANs were used also for the fusion of acquisitions from synthetic aperture radar (SAR) and optical sensors, e.g., in [42] optical data were reconstructed from SAR and in [43] a GAN was used to fuse SAR and optical multispectral data for cloud removal.

A well known bottleneck of employing DL models is the large amount of computational resources that are needed for the training phase. DL models require to be fed with large amounts of data in order to learn meaningful features, thus implying the need for dedicated pipelines for extraction and handling of such data, which can impact severely the performances of the methods. Despite the great success of CNNs, their deployment on commodity hardware (e.g., desktop computers, laptops) is often challenging, given their computational power and memory constraints. High-performance computing systems can come at aid in that regard, offering dedicated hardware accelerators to efficiently deploy and scale-up processing workflows and significantly enhancing their computational performance (i.e, reported as floating point operations per second (FLOPS)). HPC systems are on the verge of entering into the new era of exascale computing in the coming years, as currently the most powerful computers can reach hundreds of PetaFLOPS[4]. A large number of fields of research use HPC systems for addressing data storage challenges and developing scalable data processing workflows:

[1] [Online]. Available: https://hls.gsfc.nasa.gov/
[2] [Online]. Available: https://github.com/senbox-org/sen2like
[3] [Online]. Available: https://github.com/GERSL/TRA
[4] [Online]. Available: https://www.top500.org/lists/top500/2020/11/

from climatology to astrophysics, medicine and industrial applications [44]. In RS, HPC has been an essential component from the very beginning in the field of EO since its technology and applications include unique data processing, storage or transmission requirements [45], [46]. In the current era of artificial intelligence (AI) supercomputers (i.e., HPC systems equipped with specialized hardware accelerators [47]), applications from RS also use them to speed-up the processing of DL models that include a high number of trainable parameters [48].

From this brief analysis of the literature, it turns out that the integration of the multispectral images acquired by Landsat-8 and Sentinel-2 is extremely interesting from the operational view point due to the complementary properties of the two sensors. While Landsat satellites are approaching 50 years of continuous global data collection with a temporal revisit of 16 days, the recent launch of Sentinel-2 allows for the acquisition of images having a very high revisit time (i.e., 5 days at the equator with 2 satellites which results in 2–3 days at mid-latitudes). In this context, Sentinel-2 images can be used to generate Landsat-8 like images (from the spectral and spatial view point) with the aim of having dense TSs of images compatible with the TSs of real Landsat-8 available in the past. Such long and dense TSs of images allow for long-term environmental analyzes, which are extremely important for several applications (i.e., climate change, deforestation analysis, desertification, urban monitoring, etc.).

In the literature, the integration of Landsat-8 and Sentinel-2 images has been mainly addressed by the RS community considering physical methods or regression models due to their capability of properly handling the harmonization problem from the physical view point, and their low computational burden. The latter is particularly important when working at country or continental scale, where the optical preprocessing has to be applied over a hundred of images. However, such methods can only partially mitigate the reflectance difference and may fail in heterogeneous areas where complex nonlinear harmonization problems have to be solved. In this framework, it is necessary to define an automatic system suitable for all land cover types and at all geographical locations, which is able perform the integration of these data in a fast and efficient way.

This article presents an automatic work-flow which aims to facilitate the integration of the optical satellite images acquired by Landsat-8 and Sentinel-2 spectral sensors at operational level. Differently from the literature, the proposed system architecture takes advantage from the capability of the GAN to accurately learn and model the considered nonlinear problem, while preserving the spectral and spatial properties of the two satellite sensors. To mitigate the computational cost of the required DL models, we take advantage of HPC systems to deploy a parallel and scalable processing workflow that encompasses the extraction of the features from the input tiles, the training of the model and the reconstruction of the harmonized Landsat-8 and Sentinel-2 data product. The speed-up of the training of the DL model is obtained thanks to the adoption of a data parallel strategy, which distributes the training of the GAN on multiple GPUs.

The main contributions of this work are the following: 1) the definition of a multispectral adaptation GAN tailored to the peculiar properties of Sentinel-2 and Landsat-8 in terms of spatial resolution, spectral bandwidth, and spectral response function; 2) the implementation of a fully automatic and unsupervised dedicated pipeline, ready-to-use, being able to ingest Sentinel-2 and Landsat-8 data and to produce a dense TS of optical satellite images; and 3) the efficient implementation of a parallel and scalable processing workflow developed and deployed on an HPC environment on up to 16 GPUs, thanks to the adoption of a data distributed strategy, which contributes to mitigate the computational burden of the training.

II. Proposed Multispectral Adaptation GAN

The aim of this work is to generate harmonized dense time series (TSs) of Landsat-8 and Sentinel-2 images. To this end, we propose a multispectral adaptation GAN (MGAN) model tailored to the specific properties of the considered satellite optical data. Our objective is to model the spatial and spectral properties (point spread function) of the two sensors in order to adapt the Sentinel-2 data to be Landsat-8 like. Indeed, the proposed GAN is tailored to the specific spectral and spatial properties of the considered sensors to facilitate the adaptation of the Sentinel-2 images to the Landsat-8 ones. In particular, the proposed architecture is build upon the established pix2pix conditional GAN [49] that was designed for color and grayscale image-to-image translation. Based on the GAN concept, the adversarial game played by the two models of the original pix2pix architecture [49] can be represented by the formula

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[\log D(\mathbf{X},\mathbf{Y})] + \mathbb{E}_{\mathbf{X},z}[\log(1 - D(\mathbf{X}, G(\mathbf{X},z)))] \quad (1)$$

where $\mathbb{E}$ is the expected value, $\mathbf{X}$ and $\mathbf{Y}$ are the source and target images (having the same resolution), z the input noise of the generator and $V(G,D)$ is the value function. In particular, the generator G and the discriminator D of pix2pix are a U-net encoder–decoder architecture with skip connections and a PatchGAN, respectively. In the U-net encoder–decoder generator [50], the first part contains a number of downsampling convolution layers. The second part is a mirrored version of the first, with a transposed convolution for upsampling the data, which flows from the bottom to the top of the U-net through a bottleneck. The skip connections, which link the inner layers of the encoder and decoder, allow low-level information to pass directly from the first to the last layers of the U-net.

Differently from the original implementation of the pix2pix, the input data are no more RGB natural images, but multiresolution and multiband images with different spectral properties. To handle the peculiarities of the considered RS data, we trained the proposed MGAN from scratch using paired Landsat-8 and Sentinel-2 images. Table I reports the properties of the considered spectral bands in terms of spatial and spectral resolutions for both the considered optical sensors. According to the spectral

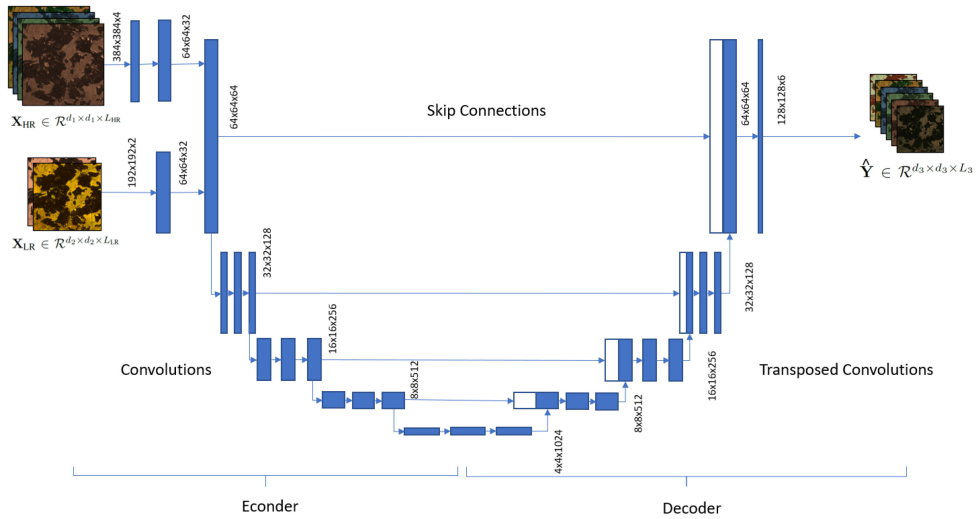


Fig. 1. Flowchart of the modified U-Net tailored to the peculiar spectral and spatial properties of Sentinel-2 and Landsat-8.

TABLE I
SPECTRAL BANDS OF LANDSAT-8 AND SENTINEL-2 SELECTED ACCORDING TO THE SPECTRAL AGREEMENT OF THE OPTICAL SENSORS

Landsat-8			**Sentinel-2**		
Band	Wavelenght (µm)	Res. (m)	Band	Wavelenght (µm)	Res. (m)
2	0.450-0.515	30	2	0.458-0.523	10
3	0.525-0.600	30	3	0.543-0.578	10
4	0.630-0.680	30	4	0.650-0.680	10
5	0.845-0.885	30	8	0.785-0.900	10
6	1.560-1.660	30	11	1.565-1.655	20
7	2.100-2.300	30	12	2.100-2.280	20

characteristic of Sentinel-2 and Landsat-8, we focused the attention on the four 10 m bands and the two shortwave infrared spectral channels acquired at 20 m by Sentinel-2 (i.e., the spectral bands consistent with the Landsat-8 ones). Let us focus on the multiresolution Sentinel-2 images. Let $\mathbf{X}_{\text{HR}} \in \mathcal{R}^{d_1 \times d_1 \times L_{\text{HR}}}$ and $\mathbf{X}_{\text{LR}} \in \mathcal{R}^{d_2 \times d_2 \times L_{\text{LR}}}$ be the set of high resolution (10 m) and low resolution (20 m) spectral channels of Sentinel-2, respectively, where $\mathbf{X}_{\text{HR}}$ has $d_1 \times d_1$ pixels and L_{HR} bands while $\mathbf{X}_{\text{LR}}$ has $d_2 \times d_2$ pixels and L_{LR} bands. Let $\mathbf{Y} \in \mathcal{R}^{d_3 \times d_3 \times L_3}$ be the real Landsat-8 image contemporary to the Sentinel-2 one, having $d_3 \times d_3$ pixels and a number of bands equal to $L_3 = L_{\text{HR}} + L_{\text{LR}}$.

In the considered implementation of the proposed MGAN, the bottom of the generator has been modified to take as input the patches of Sentinel-2 at original resolution $\mathbf{X}_{\text{HR}}$ and $\mathbf{X}_{\text{LR}}$ (i.e., 10 and 20 m). To this end, we added one convolutional layer for each initial resolution, concatenating their output before entering into the encoder–decoder structure. Fig. 1 illustrates the modified U-Net tailored to the peculiar spectral and spatial properties of Sentinel-2 and Landsat-8 for facilitating the sensor adaptation performed by the proposed MGAN. The patches of the high-resolution Sentinel-2 spectral channels X_{HR} have size $384 \times 384 \times 4$, while the low-resolution ones X_{LR} have size $192 \times 192 \times 2$. The different convolutions and transposed convolutions lead to the direct production of a Landsat-8 like image having size $128 \times 128 \times 6$, which implicitly includes the 2 channels of X_{LR} and the 4 channels X_{HR} having spatial resolution of 30 m. This condition allows us to keep the same number of inner layers of the generator and the discriminator as in the original implementation. Let $\hat{\mathbf{Y}} \in \mathcal{R}^{d_3 \times d_3 \times L_3}$ be the downsampled Sentinel-2 image having all the spectral bands at the spatial resolution of the desired target image. Please note that the downsampling convolution layer allows us to directly handle the spatial resolutions of the different spectral bands of Sentinel-2 without the need of performing any preprocessing interpolation step.

The PatchGAN discriminator is designed to capture the patterns at the scale of the input image. Its objective is to classify $N \times N$ patches of $G(\mathbf{X}, z)$ (the input synthetic patch created by the generator) and $\mathbf{Y}$ (the target Landsat-8 patch) as fake or true, encouraging the generator to produce more accurate and realistic outputs. Differently from the standard pix2pix implementation, the generator of the considered MGAN does not perform the instance normalization [51], since it is not suited to multispectral images. Indeed, similarly to the case of the standard batch normalization typically used in computer vision, the patches may not be consistent from the spectral view point. For this reason, in the model we added the spectral normalization right after the instance normalization in the downsampling blocks of the discriminator [52]. The addition of those layers in the discriminator is beneficial for the stability of the training and the spectral content of the obtained synthetic Landsat-8 images. In greater details, we train the generator and discriminator jointly, employing two losses. The L_1 loss is used in for the training of

Fig. 2. True color representation of the Sentinel-2 image acquired on the 21/04/2018 over the considered study area (coordinates are reported in the UTM WGS84 33 N system). An example of the reference data used to perform the crop type classification task is reported in the zoom area highlighted in red.

the generator to learn a low-frequency representation

$$L_1 = \mathbb{E}_{\mathbf{X}} \left\| \hat{\mathbf{Y}} - \mathbf{Y} \right\| \tag{2}$$

where $\hat{\mathbf{Y}} = G(\mathbf{X}, z)$ is the generated image obtained considering as input the Sentinel-2 image $\mathbf{X}$ and $\mathbf{Y}$ is the target Landsat-8 image. We adopted a relativistic adversarial loss for the discriminator [shown in (3)] as a replacement of the original adversarial loss employed in pix2pix. Using the relativistic loss, instead of the absolute probability that one input image is real or fake, the relative probability that a real image is more realistic than a fake one is computed [53]. The adoption of the relativistic adversarial loss increases for the stability of the training [54]. The discriminator loss is

$$\begin{aligned} L_D = & -\mathbb{E}_{\mathbf{Y}}[\log(D_{R_a}(\mathbf{Y}, \hat{\mathbf{Y}}))] \\ & -\mathbb{E}_{\hat{\mathbf{Y}}}[\log(1 - D_{R_a}(\hat{\mathbf{Y}}, \mathbf{Y}))] \end{aligned} \tag{3}$$

and the generator loss

$$\begin{aligned} L_G = & -\mathbb{E}_{\mathbf{Y}}[\log(1 - D_{R_a}(\hat{\mathbf{Y}}, \mathbf{Y}))] \\ & -\mathbb{E}_{\hat{\mathbf{Y}}}[\log(D_{R_a}(\mathbf{Y}, \hat{\mathbf{Y}}))] \end{aligned} \tag{4}$$

where $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ are the real and the fake generated images, respectively, and D_{R_a} is the output of the discriminator. To properly train the considered MGAN from scratch, we implemented data augmentation. The lack of large amount of data is known to pose several challenges during the training of GAN, since in that setting the discriminator tends to fool the generator easily, which in turn gets stuck and cannot improve anymore. This is particularly true when dealing with RS data [55] recently introduced a data augmentation technique specifically designed to work with GANs. Differentiable augmentation addresses this issue by applying the same set of transformations on both the generated and real images, regularizing the discriminator and reducing training instability. We adopted the color (contrast, brightness, saturation), translation (the images are translated and zero padded) and cutout (masking a region of the images) policies.

III. Dataset Description and Design of Experiments

In this section, we present the considered study area and the RS data employed to test the proposed approach. Then, we describe in detail the procedure designed to generate the harmonized TS of Sentinel-2 and Landsat-8 images.

A. Dataset Description

Fig. 2 presents the considered study area, which covers the valley of the Donau in the proximities of Linz, Austria (tile 33UVP of Sentinel-2, tile 191/026 of Landsat-8). Such area is characterized by a heterogeneous landscape typical of the Alpine region, where the topography ranges from high mountain to lowlands areas. The land cover is characterized by the presence of many crop types, which model a complex scenario since crops rapidly change their textural and spectral features. Moreover, the study area is heavily affected by cloud and snow coverage. Due to high temporal resolution of Sentinel-2, several pairs of real Landsat-8 and Sentinel-2 images acquired at the same date (or a one day of distance) are used to train the GAN network from scratch. Table II reports the acquisition dates of the considered images collected in Spring and Autumn. Only images having low cloud coverage (smaller than 30%) were used to train the MGAN.

TABLE II
LANDSAT-8 (TILE 191/026) AND SENTINEL 2 (TILE 33UVP) IMAGES USED IN THE EXPERIMENTS.

	Landsat-8 images	Sentinel-2 images
Training	04/04/2018	04/04/2018
	20/04/2018	21/04/2018
	06/05/2018	06/05/2018
	27/09/2018	26/09/2018
	13/10/2018	13/10/2018
Prediction	-	03/07/2018

Five images were used to train the MGAN, while the sentinel 2 image acquired on the 03/07/2018 was used for prediction only.

TABLE III
NUMBER OF SAMPLES FOR EACH CROP TYPE

Crop Type	# Samples
Grassland	2600
Maize	1668
Winter Barley	2400
Winter Caraway	400
Rapeseed	868
Beet	972
Spring Cereal	766
Winter Wheat	600

To assess the capability of the trained MGAN to correctly generate synthetic Landsat-8 data from Sentinel-2 images, the Sentinel-2 data acquired on 03/07/2018 was not involved in the training but used for prediction only. Indeed, the Landsat-8 acquisition available in July 2018 are all strongly affected by cloud coverage; thus, they cannot be used to train the model. This real test case demonstrates the importance of the proposed method from the operational view point. The use of Sentinel-2 data to generate synthetic Landsat-8 images having a good temporal sampling of the whole year. These TSs are extremely important to correctly handle multitemporal tasks such as crop type mapping. To this end, a 2018 reference dataset of crop types of the considered study area is used to accomplish this peculiar classification task. Table III reports the set of crop types of the considered classification problems together with the number of samples per class. The training and test sets are statistically independent, since training and test samples have been extracted from spatially disjoint portions of the considered study area. An example of ground reference data used to perform the crop type mapping task is reported in Fig. 2, where in the zoom the different crop types are highlighted in different colors.

B. Design of the Experiments

To train the considered MGAN, both the Landsat-8 and Sentinel-2 images are split into patches. Fig. 3 reports the different stages of our method, from the training of the model to the prediction and reconstruction of the entire tiles. First, the Landsat-8 images are warped to extract the region overlapping

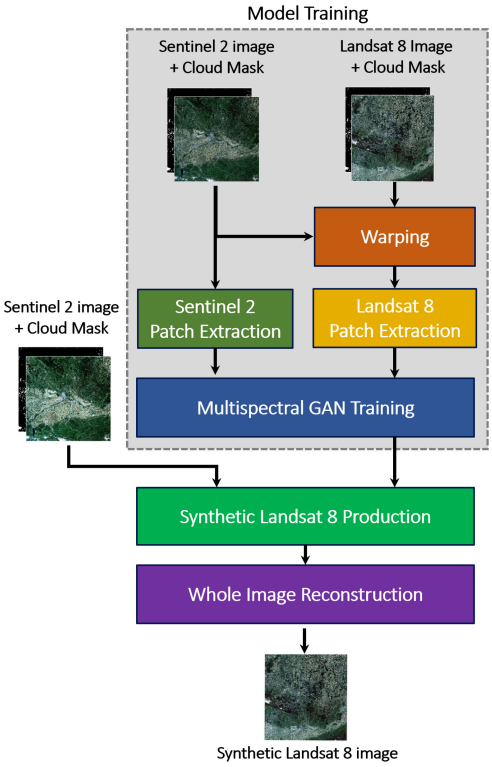


Fig. 3. Flowchart of the different stages of the proposed method. It receives as input TSs of Landsat-8 and Sentinel-2 acquired over the same geographical area. Firstly, the warping step aligns the two TSs. Then, the patch extractor generates paired and overlapping patches (i.e., training samples) that are trained by the proposed MGAN. The final stage reconstructs the whole synthetic Landsat-8 image.

with the Sentinel-2 tiles, by applying a nearest neighbor resampling strategy that does not affect the spectral content of the image. Then, possible spectral outliers are removed from both images. To this end, we considered the standard procedure of saturating the pixel values below and above the 1 and 99 percentiles of the spectral distribution computed per band. Finally, the paired patches were extracted from the two original TSs of Landsat-8 and Sentinel-2. For the Landsat-8 data, the dimension of the patches is 128 x 128 px (30 m resolution), while for Sentinel-2 they are 192 x 192 px (for the 20 m resolution bands) and 394 x 394 px (for the 10 m resolution bands). In particular, a stride of half the dimension of the patch is considered to generate overlapping patches, thus increasing the number of samples. Patches with a significant cloud or snow coverage are not used during training and are excluded with the usage of the available masks. The information provided by the cloud masks of Landsat-8 (i.e., pixel_qa band) and Sentinel-2 (i.e., SCL band) are used to define the valid patches for training. The pixel values of each patch are normalized per band by subtracting the mean and dividing by the maximum value. Once that the GAN is trained, it can be

used to predict synthetic Landsat-8 images by using Sentinel-2 data. During prediction, each original Sentinel-2 patch is fed into the generator and the corresponding synthetic Landsat-8 patch is produced. The final step is the reconstruction of the entire image from the predicted patches. We applied a buffer equal to $1/4$ of the dimension of the patch when fusing them. The tile is then reconstructed using only the central part of the patches, skipping the buffers to limit distortions caused by the convolution operations at the edges of the patches.

IV. Implementation and Experimental Setup

In this section, details are given on the implementation and computational setup. Moreover, the quality indexes used to quantitatively evaluate the proposed method are reported in the experimental setup section.

A. Implementation Setup

Of the two main families to distribute the training of a model [56], we used the data distribution approach (i.e., data parallelism). Among the different frameworks that exist to integrate a data distributed strategy into existing code we adopted Horovod [57], a library that offers a flexible API that works on top of most DL libraries, i.e., TensorFlow, Keras, PyTorch, and MXNet. Horovod makes use of Message Passing Interface (MPI) and the NVIDIA Collective Communication Library (NCCL) to implement a decentralized and efficient ring-allreduce algorithm [57], which allows the computation of the gradients in a distributed fashion. We used ADAM with base learning rate $lr = 0.0001$ for the optimization of both the generator and the discriminator, which we scaled linearly w.r.t. the number of graphics processing units (GPUs), without warm-up phase and learning rate schedulers. The training was performed for 100 epochs, as after that point the L_1 loss begins to diverge and the quality of the predicted patches deteriorates. The weights of the U-Net and of the PatchGAN were initialized with the default Glorot uniform distribution [58]. The local batch size used for each GPU is 16, therefore the resulting maximum global batch size used in the present work, computed as global _ batch_size = number _ gpus $\times$ local _ batch_size, is equal to 256.

B. Experimental Setup

The experiments were carried out on the extreme scale booster (ESB) partition of the of the dynamic exascale entry platform–extreme scale technologies (DEEP-EST) and on the booster partition of the Jülich Wizard for European Leadership Science (JUWELS) supercomputers at the Jülich Supercomputing Centre (JSC) [59]. The training was scaled on up to 16 Nvidia Tesla V100 and A100 graphics processing unit (GPU). We used Horovod data-parallel framework on top of TensorFlow2, with a custom made training loop. The data preprocessing was deployed on the Jülich Wizard for European Leadership Science (JUWELS) system [47]. We used the Geospatial Data Abstraction Library `GDAL 2.3.2` through its Python API.

To quantitatively evaluate the results obtained we considered several spectral distortion metrics typically used in the literature. In particular, we considered the relative dimensionless global error (ERGAS), the spectral angle mapper (SAM), the root-mean-square error (rmse), the universal image quality index (UIQI), and the peak signal-to-noise ratio (PSNR) measures on the valid patches (i.e., low cloud coverage). Spectral angle mapper (SAM) [60] measures the spectral distortion in terms of angle between the vectors of the reference image and generated image

$$\mathrm{SAM}(\mathbf{Y}, \hat{\mathbf{Y}}) \triangleq \arccos\left(\frac{\langle \mathbf{Y}, \hat{\mathbf{Y}} \rangle}{\|\mathbf{Y}\|_2 \cdot \|\hat{\mathbf{Y}}\|_2}\right) \tag{5}$$

where $\mathbf{Y}$ is the real input and $\hat{\mathbf{Y}}$ the predicted input. The lower is the value of SAM, the lower the presence of spectral deviations between the two images. Relative dimensionless global error (ERGAS) measures the quality of the generated image compared to the reference image as a normalized mean square error between each band of the two images [61]

$$\mathrm{ERGAS}(\mathbf{Y}, \hat{\mathbf{Y}}) \triangleq 100\frac{1}{S}\sqrt{\frac{1}{L_3}\sum_{l=1}^{L_3}\frac{\mathrm{MSE}(\mathbf{Y}_l, \hat{\mathbf{Y}}_l)}{\mu^2_{\hat{\mathbf{Y}}_l}}} \tag{6}$$

where $\frac{1}{S}$ is the ratio between the pixel sizes (i.e., equal to one in our case), $\mathbf{Y}_l$ and $\hat{\mathbf{Y}}_l$ are the lth bands of the generated image and of the reference image, respectively; the $\mathrm{MSE}(\mathbf{Y}_l, \hat{\mathbf{Y}}_l)$ is the mean squared error between $\mathbf{Y}_l$ and $\hat{\mathbf{Y}}_l$ and $\mu_{\hat{\mathbf{Y}}_l}$ is the mean of $\hat{\mathbf{Y}}_l$. As for SAM, a low value of ERGAS implies a low presence of distortion in the generated image compared to the reference.

The RMSE is defined as

$$\mathrm{RMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) \triangleq \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|}{d_3^2} \tag{7}$$

where $\mathbf{Y}$ is the original input, $\hat{\mathbf{Y}}$ the predicted input.

The universal image quality index (UIQI) [62] has been computed on a sliding window of size 32×32 pixels, and averaged over all window positions per band. Let $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$ denote the jth windowed segment of a single band of the reference and the simulated images, respectively. The UIQI is given by

$$\mathrm{Q}(\mathbf{y}, \hat{\mathbf{y}}) \triangleq \frac{1}{W}\sum_{j=1}^{W}\frac{\sigma_{\mathbf{y}_j\hat{\mathbf{y}}_j}}{\sigma_{\mathbf{y}_j}\sigma_{\hat{\mathbf{y}}_j}} \times \frac{2\mu_{\mathbf{y}_j}\mu_{\hat{\mathbf{y}}_j}}{\mu^2_{\mathbf{y}_j} + \mu^2_{\hat{\mathbf{y}}_j}} \times \frac{2\sigma_{\mathbf{y}_j}\sigma_{\hat{\mathbf{y}}_j}}{\sigma^2_{\mathbf{y}_{i,j}} + \sigma^2_{\hat{\mathbf{y}}_j}} \tag{8}$$

where $\sigma_{\mathbf{y}_j\hat{\mathbf{y}}_j}$ is the covariance between $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$, $\sigma_{\mathbf{y}_j}$ and $\mu_{\mathbf{y}_j}$ are the standard deviation and the mean value of $\mathbf{y}_j$, while $\sigma_{\hat{\mathbf{y}}_j}$ and $\mu_{\hat{\mathbf{y}}_j}$ are the standard deviation and the mean value of $\hat{\mathbf{y}}_j$, respectively. This index has a range of $[-1, 1]$, being equal to 1 when $\mathbf{y} = \hat{\mathbf{y}}$. To extend the UIQI index to the multiband case, we average the band indexes as follows:

$$\mathrm{Q}(\mathbf{Y}, \hat{\mathbf{Y}}) \triangleq \frac{1}{L_3}\sum_{l=1}^{L_3}\mathrm{Q}(\mathbf{Y}_l, \hat{\mathbf{Y}}_l) \tag{9}$$

TABLE IV
SPECTRAL DISTORTION METRICS BETWEEN THE ORIGINAL LANDSAT-8 DATA AND: 1) THE SYNTHETIC LANDSAT-8 IMAGES GENERATED USING THE PROPOSED MGAN, 2) THE HARMONIZED LANDSAT-8 IMAGES GENERATE USING THE BASELINE METHOD HLS, AND 3) THE ORIGINAL CONTEMPORARY SENTINEL-2 IMAGES.

Data	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7	Overall
				SAM			
Synthetic Landsat-8 (MGAN)	**0.22**	**0.18**	**0.21**	**0.16**	**0.17**	**0.19**	**0.19**
Harmonized Landsat-8 (HLS)	0.46	0.37	0.48	0.29	0.30	0.39	0.38
Original Sentinel-2	0.32	0.26	0.31	0.23	0.20	0.22	0.26
				ERGAS			
Synthetic Landsat-8 (MGAN)	**719**	**674**	**731**	**647**	**669**	**696**	**1933**
Harmonized Landsat-8 (HLS)	1321	924	1051	1020	882	930	2903
Original Sentinel-2	875	780	884	727	685	740	2180
				RMSE			
Synthetic Landsat-8 (MGAN)	**185**	**305**	**313**	**1229**	**806**	**520**	**668**
Harmonized Landsat-8 (HLS)	275	390	475	1381	946	692	799
Original Sentinel-2	285	371	393	1344	843	607	744
				UIQI			
Synthetic Landsat 8 (MGAN)	**0.66**	**0.67**	**0.66**	**0.67**	**0.67**	**0.66**	**0.67**
Harmonized Landsat-8 (HLS)	0.58	0.63	0.55	0.60	0.64	0.60	0.60
Original Sentinel 2	0.62	0.65	0.64	0.66	0.66	0.65	0.65
				PSNR			
Synthetic Landsat-8 (MGAN)	**337**	**333**	**333**	**322**	**325**	**329**	**327**
Harmonized Landsat-8 (HLS)	331	328	326	317	321	323	321
Original Sentinel 2	331	330	329	319	324	326	324

The obtained results are the average values over the 5 images of the considered dataset. Results are provided per spectral band and overall. The best results are highlighted in bold.

The peak signal-to-noise ratio (PSNR) is defined as

$$\text{PSNR}(\mathbf{Y}, \hat{\mathbf{Y}}) \triangleq 20 \log_{10} \left(\frac{\lambda^2}{\frac{1}{d_3^2} \|\mathbf{Y} - \hat{\mathbf{Y}}\|} \right) \tag{10}$$

where λ is the number of levels of the images.

V. EXPERIMENTAL RESULTS

In this section, first we present the quantitative results obtained in terms of spectral distortion metrics. The quantitative evaluation is provided together with qualitative examples of the obtained synthetic Landsat-8 images. Finally, the generated TS of synthetic Landsat-8 images is used to perform a crop type mapping task to assess the capability of the network to accurately reproduce the spectral properties of the data. The proposed approach is compared with the physical method HLS [28] developed for reducing the reflectance differences between Landsat-8 and Sentinel-2, thus generating smooth spectral TSs. Please note that such method is widely used from the operational view point [29].

A. Quantitative and Qualitative Results

Table IV reports the results obtained for different spectral distortion metrics comparing the original Landsat-8 images and: 1) the synthetic Landsat-8 images produced by the proposed MGAN; 2) the harmonized Landsat-8 images generated using the baseline method HLS; and 3) the original contemporary Sentinel-2 images. The best results are highlighted in bold. Please note that the evaluation of the spectral difference between real Landsat-8 data and Sentinel-2 data is reported to evaluate the capability of the methods to reduce the spectral difference of these data.

From the results obtained, one can notice that the metrics computed between Landsat-8 and Sentinel-2 images demonstrate the need of harmonizing these data from the spectral view point. The HLS reduces the spectral distortion for some spectral bands. However, for all the metrics, the best results are achieve by the synthetic Landsat images generated with the proposed MGAN. In particular, the MGAN is able to correctly reproduce the spectral properties of Landsat-8 regardless of the spectral bands. Indeed, similar error metrics are achieved in both the RGB spectral channels (i.e., Band2, Band3, and Band4) as well as the near infrared (Band5) and shortwave infrared bands (Band6 ad Band7).

The results obtained from the quantitative view point are confirmed from the qualitative ones. In order to assess the consistency between the generated and the target data, Fig. 4 reports some portions of the: 1) original Landsat-8 image (target); 2) syntethic Landsat-8 data produced by the MGAN; and 3) harmonized Landsat-8 data produced by the baseline method (HLS); and 4) contemporary Sentinel-2 image used to generate

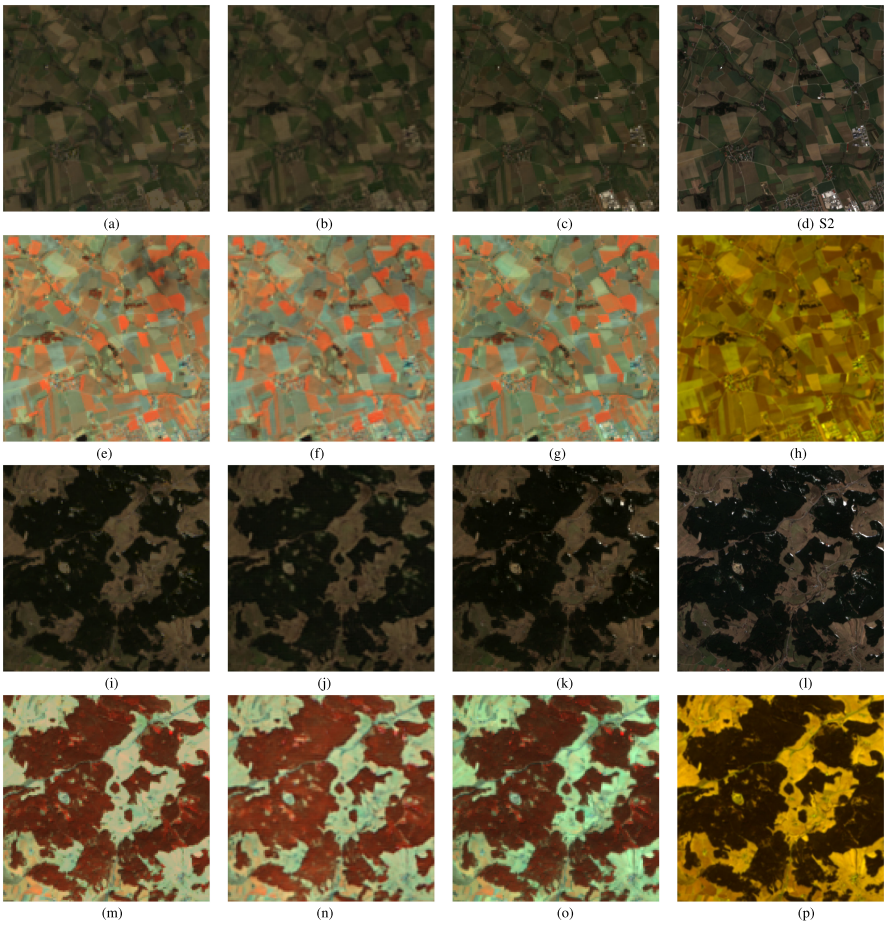


Fig. 4. Qualitative examples of the obtained Landsat-8 images. The true color composite (RGB) is reported for the (a), (i) target Landsat-8 image, (b), (j) generated Landsat 8 with the proposed MGAN, (c), (k) generated Landsat 8 with the HLS, and (d), (l) original Sentinel-2 image. The false color composite is reported for the NIR and SWIR bands for the (e), (m) target Landsat-8 image, (f), (n) generated Landsat-8 with the proposed MGAN, (g), (o) generated Landsat-8 with the HLS, and (h), (p) original Sentinel-2 image.

the Landsat-8 data. The synthetic image produced by the MGAN looks more similar to the original Landsat-8 image than the original Sentinel-2 input data and the harmonized Landsat8 data produced by the HLS method. These results also confirm that the quality of the generated images is good and does not suffer from significant distortions and artifacts. From the results obtained, one can notice that the generated data looks more similar to the original Landsat 8 image than the original Sentinel-2 input data and the harmonized Landsat-8 data produced by the HLS method. For instance, the presence of bright buildings absent in the real Landasat-8 images [see Fig. 4(a)] is visible in the harmonized data produced by the HLS method [Fig. 4(c)] but not present in the synthetic data produced my the MGAN [see Fig. 4(b)].

B. Crop Type Mapping Results

To assess the capability of the proposed MGAN to accurately model the spectral information of Landsat-8, a crop type mapping task was carried out using the obtained TS of produced synthetic images. This peculiar classification task requires the availability of accurate multitemporal and multispectral information to properly retrieve the crop types present in the scene. Indeed, differently from other land-cover classification tasks that can be performed using mono-temporal data, the temporal information is fundamental to accurately model the phenological trend of the crop types.

Table V reports the classification results obtained by considering TSs of: 1) 5 synthetic Landsat-8 images produced by the proposed MGAN; 2) 5 harmonized Landsat-8 images obtained

TABLE V
CROP TYPE MAPPING RESULTS OBTAINED BY CONSIDERING TSS OF: 1) 5 SYNTHETIC LANDSAT-8 IMAGES PRODUCED BY THE PROPOSED MGAN, 2) 5 HARMONIZED LANDSAT-8 IMAGES OBTAINED BY USING THE BASELINE METHOD (HLS), AND 3) 6 SYNTHETIC LANDSA- 8 IMAGES PRODUCED BY THE PROPOSED MGAN. THE TS OF 5 IMAGES WERE PRODUCED BY THE PROPOSED AND THE BASELINE METHODS USING BOTH THE ORIGINAL SENTINEL 2 AND LANDSAT 8 IMAGES. TO GENERATE THE TS OF 6 IMAGES, A SENTINEL-2 IMAGE ACQUIRED ON THE 03/07/2018 FOR WHICH NO CORRESPONDING CLOUDLESS LANDSAT-8 DATA ARE AVAILABLE WAS USED. PA IS THE PRODUCER'S ACCURACY OR RECALL, UA IS THE USER'S ACCURACY OR PRECISION, F1 IS THE F1-SCORE AND OA IS THE OVERALL ACCURACY

Crop Type		TS of 5 images						TS of 6 images		
		Synthetic Landsat-8 (MGAN)			Harmonized Landsat-8 (HLS)			Synthetic Landsat-8 (MGAN)		
		PA%	UA%	F1%	PA%	UA%	F1%	PA%	UA%	F1%
	Grassland	96.38	88.24	92.13	95.46	76.60	85.00	96.85	90.25	93.43
	Maize	87.05	88.00	87.52	79.50	79.31	79.40	92.93	92.70	92.81
	Winter Barley	88.17	86.37	87.26	82.83	82.08	82.45	93.92	90.30	92.07
	Winter Caraway	71.00	96.60	81.84	64.50	94.16	76.56	70.00	95.89	80.92
	Rapeseed	88.25	98.46	93.08	81.57	94.91	87.74	88.71	96.49	92.44
	Beet	93.62	90.46	92.01	85.19	87.71	86.43	95.68	94.90	95.29
	Spring Cereal	77.28	80.65	78.93	65.54	83.39	73.40	82.77	87.81	85.22
	Winter Wheat	64.67	74.33	69.16	49.67	79.68	61.19	78.00	89.31	83.27
	OA%	87.83			81.66			91.53		

by using the baseline methods (HLS); 3) 6 synthetic Landsat-8 images produced by the proposed MGAN. The TSs of 5 images were produced by the proposed and the baseline methods using both the original Sentinel-2 and Landsat-8 images. To generate the TS of 6 images, we considered the Sentinel-2 image acquired on 03/07/2018 for which no corresponding cloudless Landsat-8 data are available. Since no cloud-less images were acquired by the Landsat-8 sensor in July 2018 for the considered tile, no quantitative evaluation can be performed in terms of spectral distortion metrics. However, the PA%, UA%, F1%, and OA% confirm the quality of the added image. The classification is performed by training a standard Support Vector Machine (SVM) with RBF kernels [63]. The optimal kernel parameters (i.e., the regularization parameter C and the spread of the kernel γ) were selected by a fivefold cross-validation.

This test case demonstrates the need to densify existing TSs of satellite data. The temporal and spectral information provided by the satellite acquisition of July 2018 sharply increases the classification results by improving the modelling of the phenological trends of the considered crop types. This increases the OA% from 87.83% (TS of 5 synthetic Landsat-8 images) to 91.53 % (TS of 6 synthetic Landsat-8 images). From these results, we can conclude that the proposed MGAN can be used to generate harmonized dense TSs of Landsat-8 and Sentinel-2 images.

C. Scaling Efficiency

The adoption of Horovod allowed us to distribute the training on multiple GPUs and significantly reduce the time required to complete the optimization of the model. The maximum number of GPUs used in the present work is 16, a configuration with which we obtained a speed-up of 14x on the JUWELS-BOOSTER and 12x on the DEEP-ESB partitions compared to the use of a single GPU (shown in Fig. 5). The scaling efficiency was close to 90% on the JUWELS-BOOSTER and

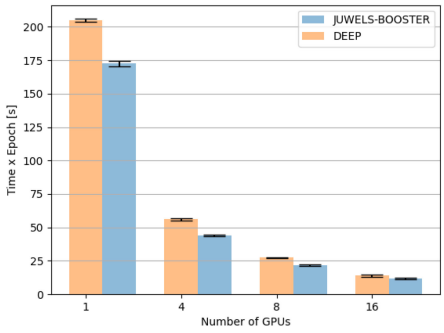


Fig. 5. Training time per epoch w.r.t. the number of GPUs on the JUWELS-BOOSTER and DEEP-ESB partitions.

above 75% on the DEEP-ESB partitions, respectively. In both cases, the scaling efficiency declined more steeply with 8 and more GPUs, possibly due to the increased communication time (time spent to synchronize the gradient among the GPUs) w.r.t. the computation time (time spent to optimize the model on each local GPU, which decreases proportionally to the increase of the number of GPUs, since each GPU is fed a smaller portion of the entire dataset). It can be noted that the efficiency shrinks more prominently on the DEEP-ESB partition. This behaviour could be explained by the fact that on the DEEP-ESB partition each node is equipped with only one V100 GPU, while each node of the JUWELS-BOOSTER partition has 4 GPUs. This means that when using the DEEP-ESB partition the communication is only inter-node (the nodes are connected through InfiniBand), while on the JUWELS-BOOSTER partition the communication takes place both inter- and intra-node (faster NVLink connections). We performed 3 runs for each experiment, and the reported results are the average and standard deviation. Fig. 6 shows the training time that was reduced from 175 and more than

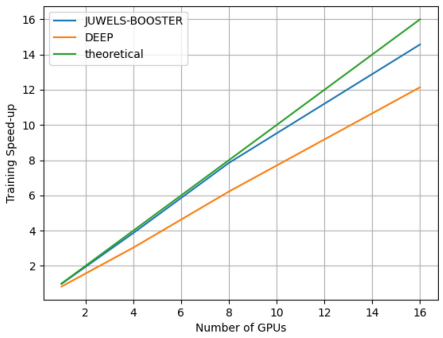


Fig. 6. Time per epoch w.r.t. the number of GPUs on the JUWELS-BOOSTER and DEEP-ESB partitions.

200 seconds using 1 GPU to 12 and 14 seconds per epoch (16 GPUs) on the DEEP-ESB and JUWELS-BOOSTER partitions, respectively. The JUWELS-BOOSTER, which features the newer A100 GPUs, allowed us to obtain a 20% increase in performances in terms of training time compared to the V100 installed on the DEEP-ESB partition.

VI. Conclusion

In this article, we introduced a method to densify and harmonize TSs of images acquired by Landsat-8 and Sentinel-2 satellite. The proposed method, which is based on a multispectral adaptation GAN, was applied to a TS which covers 6 acquisitions in 2018. We designed an experimental setup to validate our approach by comparing it with the well established HLS. The results obtained demonstrate that the proposed GAN is able to accurately reconstruct the spectral properties of Landsat-8 by using the Sentinel-2 images. Moreover, the qualitative comparison with the baseline method confirms the quantitative evaluation of the spectral distortion metrics. Although the physical model employed to harmonize Sentinel-2 and Landsat-8 is a powerful tool to generate long and dense TSs of optical satellite images, the proposed method achieves more accurate results from the spectral view point. Another important result is provided by the classification accuracy obtained when considering the TS of 6 images, which allow us to test the capability of the network to accurately predict synthetic Landsat-8 images never used to train the MGAN. The OA% was increased from 87.83% (TS of 5 synthetic Landsat-8 images) to 91.53 % (TS of 6 synthetic Landsat-8 images). Moreover, we deployed the entire workflow in an HPC environment, and with the utilization of Horovod we could make an efficient use of the resources provided by such system, reducing the time required for the training of the model.

Although in this work we demonstrated that our approach can successfully densify TSs of Landsat-8 images, several challenges remain open. We focused our attention on one single region where we could validate our method also in terms of classification; however, our approach should be also extended to include different areas in the future. A strategy to ingest new data from different TSs and scale the training should be drawn up, in order to make the training of the models with larger amount of data feasible in a reasonable amount of time. Further effort should be also put on finding the optimal hyperparameters of the training, such as the optimizers, learning rate, scheduler. Neural architecture search could be employed to optimize the structure of the model, i.e., the number and type of layers, the activation functions, etc. Further loss functions should be also added, although this would significantly increase the space of the hyperparameters search, and a tradeoff with available computational resources should be found. A repository with the code is available at.[5]

Acknowledgment

The authors would like to thank M. Maskey and B. Freitag from NASA Interagency Implementation and Advanced Concepts Team (IMPACT) for the provision of the HLS datasets.

References

[1] Z. Zhu, "Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 370–384, 2017.

[2] B. Feizizadeh, T. Blaschke, D. Tiede, and M. H. R. Moghaddam, "Evaluating fuzzy operators of an object-based image analysis for detecting landslides and their changes," *Geomorphology*, vol. 293, pp. 240–254, 2017.

[3] W. Zhang, X. Lu, and X. Li, "A coarse-to-fine semi-supervised change detection for multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3587–3599, Jun. 2018.

[4] M. Meroni, D. Fasbender, F. Rembold, C. Atzberger, and A. Klisch, "Near real-time vegetation anomaly detection with MODIS NDVI: Timeliness vs. accuracy and effect of anomaly computation options," *Remote Sens. Environ.*, vol. 221, pp. 508–521, 2019.

[5] J.-F. Pekel *et al.*, "A near real-time water surface detection method based on HSV transformation of MODIS multi-spectral time series data," *Remote Sens. Environ.*, vol. 140, pp. 704–716, 2014.

[6] X. Tang, E. L. Bullock, P. Olofsson, S. Estel, and C. E. Woodcock, "Near real-time monitoring of tropical forest disturbance: New algorithms and assessment framework," *Remote Sens. Environ.*, vol. 224, pp. 202–218, 2019.

[7] M. Wulder, J. Masek, W. Cohen, T. Loveland, and C. Woodcock, "Opening the archive: How free data has enabled the science and monitoring promise of landsat," *Remote Sens. Environ.*, vol. 122, pp. 2–10, 2012.

[8] J. Aschbacher, "ESA's earth observation strategy and copernicus," in *Proc. Satell. Earth Observ. Impact Soc. Policy*, 2017, pp. 81–86.

[9] P. Defourny *et al.*, "Near real-time agriculture monitoring at national scale at parcel resolution: Performance assessment of the Sen2-Agri automated system in various cropping systems around the world," *Remote Sens. Environ.*, vol. 221, pp. 551–568, 2019.

[10] J. Li and D. P. Roy, "A global analysis of Sentinel-2A, Sentinel-2B and Landsat-8 data revisit intervals and implications for terrestrial monitoring," *Remote Sens.*, vol. 9, no. 9, 2017, Art. no. 902.

[11] M. Drusch *et al.*, "Sentinel-2: ESA's optical high-resolution mission for GMES operational services," *Remote Sens. Environ.*, vol. 120, pp. 25–36, 2012.

[12] B. Huang and H. Song, "Spatiotemporal reflectance fusion via sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3707–3716, Oct. 2012.

[13] S. Skakun *et al.*, "Winter wheat yield assessment from Landsat 8 and Sentinel-2 data: Incorporating surface reflectance, through phenological fitting, into regression yield models," *Remote Sens.*, vol. 11, no. 15, 2019, Art. no. 1768.

[14] N. Pahlevan, S. K. Chittimalli, S. V. Balasubramanian, and V. Vellucci, "Sentinel-2/Landsat-8 product consistency and implications for monitoring aquatic systems," *Remote Sens. Environ.*, vol. 220, pp. 19–29, 2019.

[15] R. Raj, B. Bayat, P. Lukeš, L. šigut, and L. Homolová, "Analyzing daily estimation of forest gross primary production based on harmonized Landsat-8 and Sentinel-2 product using SCOPE process-based model," *Remote Sens.*, vol. 12, no. 22, 2020, Art. no. 3773.

[5][Online]. Available: https://gitlab.jsc.fz-juelich.de/sedona3/mgan

[16] K. T. Peterson, V. Sagan, and J. J. Sloan, "Deep learning-based water quality estimation and anomaly detection using Landsat-8/Sentinel-2 virtual constellation and cloud computing," *GIScience Remote Sens.*, vol. 57, no. 4, pp. 510–525, 2020.

[17] P. Griffiths, C. Nendel, J. Pickert, and P. Hostert, "Towards national-scale characterization of grassland use intensity from integrated Sentinel-2 and landsat time series," *Remote Sens. Environ.*, vol. 238, 2020, Art. no. 111124.

[18] N. Kussul, M. Lavreniuk, A. Kolotii, S. Skakun, O. Rakoid, and L. Shumilo, "A workflow for sustainable development goals indicators assessment based on high-resolution satellite data," *Int. J. Digit. Earth*, vol. 13, no. 2, pp. 309–321, 2020.

[19] D. K. Bolton, J. M. Gray, E. K. Melaas, M. Moon, L. Eklundh, and M. A. Friedl, "Continental-scale land surface phenology from harmonized Landsat 8 and Sentinel-2 imagery," *Remote Sens. Environ.*, vol. 240, 2020, Art. no. 111685.

[20] T. Wu, Y. Zhao, S. Wang, H. Su, Y. Yang, and D. Jia, "Improving the accuracy of fractional Evergreen forest cover estimation at subpixel scale in cloudy and rainy areas by harmonizing Landsat-8 and Sentinel-2 time-series data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3373–3385, 2021, doi: 10.1109/JSTARS.2021.3064580.

[21] P. D'Odorico, A. Gonsamo, A. Damm, and M. E. Schaepman, "Experimental evaluation of Sentinel-2 spectral response functions for NDVI time-series continuity," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 3, pp. 1336–1348, Mar. 2013.

[22] E. Mandanici and G. Bitelli, "Preliminary comparison of Sentinel-2 and Landsat 8 imagery for a combined use," *Remote Sens.*, vol. 8, no. 12, 2016, Art. no. 1014.

[23] L. Korhonen, P. HadiPackalen, and M. Rautiainen, "Comparison of Sentinel-2 and Landsat 8 in the estimation of Boreal forest canopy cover and leaf area index," *Remote Sens. Environ.*, vol. 195, pp. 259–274, 2017.

[24] D. Roy *et al.*, "A general method to normalize landsat reflectance data to nadir BRDF adjusted reflectance," *Remote Sens. Environ.*, vol. 176, pp. 255–271, 2016.

[25] N. Flood, "Comparing Sentinel-2 A and Landsat 7 and 8 using surface reflectance over Australia," *Remote Sens.*, vol. 9, no. 7, 2017, Art. no. 659.

[26] H. K. Zhang *et al.*, "Characterization of Sentinel-2A and Landsat-8 top of atmosphere, surface, and nadir BRDF adjusted reflectance and NDVI differences," *Remote Sens. Environ.*, vol. 215, pp. 482–494, 2018.

[27] R. Chastain, I. Housman, J. Goldstein, M. Finco, and K. Tenneson, "Empirical cross sensor comparison of Sentinel-2 A and 2B MSI, Landsat-8 OLI, and Landsat-7 ETM top of atmosphere spectral characteristics over the conterminous United States," *Remote Sens. Environ.*, vol. 221, pp. 274–285, 2019.

[28] M. Claverie *et al.*, "The harmonized landsat and Sentinel-2 surface reflectance data set," *Remote Sens. Environ.*, vol. 219, pp. 145–161, 2018.

[29] S. Saunier *et al.*, "Sen2like, a tool to generate Sentinel-2 harmonised surface reflectance products – first results with Landsat-8," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 5650–5653.

[30] R. Shang and Z. Zhu, "Harmonizing Landsat 8 and Sentinel-2: A time-series-based reflectance adjustment approach," *Remote Sens. Environ.*, vol. 235, 2019, Art. no. 111439.

[31] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 152, pp. 166–177, 2019.

[32] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Adv. Neural Inf. Process. Syst.*, Curran Associates, Inc., vol. 27, 2014.

[33] M. Mirza and S. Osindero, "Conditional generative adversarial Nets," *CoRR*, 2014, *arXiv:1411.1784*.

[34] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[35] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 105–114.

[36] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced Gan for remote sensing image superresolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019.

[37] Y. Xiong *et al.*, "Improved SRGAN for remote sensing image super-resolution across locations and sensors," *Remote Sens.*, vol. 12, no. 8, Apr. 2020, Art. no. 1263, [Online]. Available: http://dx.doi.org/10.3390/rs12081263

[38] R. Zhang, G. Cavallaro, and J. Jitsev, "Super-resolution of large volumes of Sentinel-2 images with high performance distributed deep learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 617–620.

[39] B. Chen, J. Li, and Y. Jin, "Deep learning for feature-level data fusion: Higher resolution reconstruction of historical landsat archive," *Remote Sens.*, vol. 13, no. 2, 2021, Art. no. 167. [Online]. Available: https://www.mdpi.com/2072-4292/13/2/167

[40] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network," *Remote Sens.*, vol. 12, no. 9, May 2020, Art. no. 1432. [Online]. Available: http://dx.doi.org/10.3390/rs12091432

[41] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 5033–5036. [Online]. Available: http://dx.doi.org/10.1109/IGARSS.2019.8900173

[42] M. Fuentes Reyes, S. Auer, N. Merkle, C. Henry, and M. Schmitt, "Sar-to-optical image translation based on conditional generative adversarial networks-optimization, opportunities and limits," *Remote Sens.*, vol. 11, no. 17, 2019, Art. no. 2067. [Online]. Available: https://www.mdpi.com/2072-4292/11/17/2067

[43] C. Grohnfeldt, M. Schmitt, and X. Zhu, "A conditional generative adversarial network to fuse sar and multispectral optical data for cloud removal from Sentinel-2 images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 1726–1729. [Online]. Available: http://dx.doi.org/10.1109/IGARSS.2018.8519215

[44] "The scientific case for high performance computing in Europe 2012-2020," 2020. [Online]. Available: https://exdci.eu/sites/all/themes/exdci_theme/images/prace_-_the_scientific_case_-_full_text_-.pdf

[45] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.

[46] A. Plaza, Q. Du, Y.-L. Chang, and R. King, "High performance computing for hyperspectral remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing,* vol. 4, no. 3, pp. 528–544, Sep. 2011.

[47] D. Krause, "Juwels: Modular Tier-0/1 supercomputer at the Jülich supercomputing centre," *J. Large-Scale Res. Facilities*, vol. 5, p. 135, 2019.

[48] R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. Benediktsson, "Remote sensing Big Data classification with high performance distributed deep learning," *Remote Sens.*, vol. 11, no. 24, Dec. 2019, Art. no. 3059. [Online]. Available: http://dx.doi.org/10.3390/rs11243056

[49] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[51] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2017, pp. 4105–4113.

[52] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 6th Int. Conf. Learning Representations*, 2018.

[53] X. Wang *et al.*, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proc. Computer Vis. Workshops*, 2019, pp. 63–79.

[54] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," in *Proc. 7th Int. Conf. Learning Representations*, 2019.

[55] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient GAN training," in *Proc. Adv. Neural Inf. Process. Syst.*, Curran Associates, Inc., vol. 33, 2020, pp. 7559–7570.

[56] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," *ACM Comput. Sur.*, vol. 52, no. 4, Aug. 2019, Art. no. 65.

[57] A. Sergeev and M. Del Balso, "Horovod: Fast and easy distributed deep learning in TensorFlow," 2018, *arXiv:1802.05799*.

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist., Proc. Mach. Learn. Res.*, 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[59] E. Suarez, N. Eicker, and T. Lippert, *Modular Supercomputing Architecture: From Idea to Production*, vol. 3, Boca Raton, FL, USA: CRC Press, 2019, pp. 223–251.

[60] P. Jagalingam and A. V. Hegde, "A review of quality metrics for fused image," *Aquatic Procedia*, vol. 4, pp. 133–142, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.aqpro.2015.02.019
[61] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data-fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3012–3021, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2007.904923
[62] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
[63] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

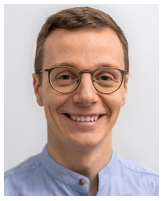
Rocco Sedona (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in information and communications engineering from the University of Trento, Trento, Italy, in 2016 and 2019, respectively. He is currently working toward the Ph.D. degree in computational engineering with the University of Iceland, Reykjavik, Iceland.

He is member of the "High Productivity Data Processing" (HPDP) research group with the Jülich Supercomputing Centre, Jülich, Germany. His research interests include machine learning methods for remote sensing applications, with a particular focus on distributing deep learning models on multiple GPUs of HPC systems.

Claudia Paris (Member, IEEE) received the B.S. and M.S. (summa cum laude) degrees in telecommunication engineering and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2010, 2012, and 2016, respectively. She accomplished the Honors Master Program in Research within the Master Degree in Telecommunication Engineering in 2012.

Since 2014 she is a Teaching Assistant with the Department of Information Engineering and Computer Science of the University of Trento, where she is currently an Assistant Professor. Her main research includes image and signal processing, machine learning, and deep learning with applications to remote sensing image analysis. Her main research interests include remote sensing single date and TSs image classification, land cover map update and fusion of multisource remote sensing data for the estimation of biophysical parameters. She conducts research on these topics within the frameworks of national and international projects.

Dr. Paris was the recipient of the very prestigious Symposium Prize Paper Award (SPPA) at the 2016 International Symposium on Geoscience and Remote Sensing (Beijing, China, 2016) and at the 2017 International Symposium on Geoscience and Remote Sensing (Fort Worth, Texas, USA, 2017). She has been a member of the program and scientific committee of several international conferences and workshops.



Gabriele Cavallaro (Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Trento, Trento, Italy, in 2011 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2016.

He is currently the Deputy Head of the "High Productivity Data Processing" (HPDP) research group, Jülich Supercomputing Centre, Jülich, Germany. Since 2019, he gives lectures on scalable machine learning for remote sensing Big Data with the Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany. His research interests include remote sensing data processing with parallel machine learning algorithms that scale on high performance and distributed systems.

Dr. Cavallaro was the recipient of the IEEE GRSS Third Prize in the Student Paper Competition of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2015 (Milan, Italy). He is also the Chair of the High-Performance and Disruptive Computing in Remote Sensing (HDCRS) Working Group of the IEEE GRSS ESI Technical Committee. He serves on the scientific committees of several international conferences and he is a referee for numerous international journals.

Lorenzo Bruzzone (Fellow Member, IEEE) received the Laurea (M.S.) degree in electronic engineering (*summa cum laude*) and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

He is currently a Full Professor of Telecommunications with the University of Trento, Trento, Italy, where he teaches remote sensing, radar, and digital communications. He is the founder and the Director of the Remote Sensing Laboratory with the Department of Information Engineering and Computer Science, University of Trento. He is the Principal Investigator of many research projects. Among the others, he is currently the Principal Investigator of the Radar for icy Moon exploration (RIME) instrument in the framework of the JUpiter ICy moons Explorer (JUICE) mission of the European Space Agency (ESA) and the Science Lead for the High Resolution Land Cover project in the framework of the Climate Change Initiative of ESA. He is the author (or coauthor) of 294 scientific publications in referred international journals (221 in IEEE journals), more than 340 papers in conference proceedings, and 22 book chapters. His current research interests include remote sensing, radar and SAR, signal processing, machine learning, and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects.

Dr. Bruzzone is the Editor/Co-editor of 18 books/conference proceedings and 1 scientific book. His papers are highly cited, as proven from the total number of citations (more than 37 000) and the value of the h-index (89) (source: Google Scholar). He was invited as keynote speaker in more than 40 international conferences and workshops. Since 2009 he has been a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS), where since 2019 he is the Vice-President for Professional Activities. He ranked first place in the Student Prize Paper Competition of the 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Seattle, July 1998. Since that he was recipient of many international and national honors and awards, including the recent IEEE GRSS 2015 Outstanding Service Award, the 2017 and 2018 IEEE IGARSS Symposium Prize Paper Awards and the 2019 WHISPER Outstanding Paper Award. Dr. Bruzzone was a Guest Co-Editor of many Special Issues of international journals. He is the co-founder of the IEEE International Workshop on the Analysis of Multitemporal Remote-Sensing Images (MultiTemp) series and is currently a member of the Permanent Steering Committee of this series of workshops. Since 2003, he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He has been the founder of the *IEEE Geoscience and Remote Sensing Magazine* for which he has been Editor-in-Chief between 2013 and 2017. Currently he is an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has been Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society between 2012 and 2016.

Morris Riedel (Member, IEEE) received the Ph.D. degree in computer engineering from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.

He worked in data-intensive parallel and distributed systems since 2004. He is currently a Full Professor of High-Performance Computing with an emphasis on parallel and scalable machine learning with the School of Natural Sciences and Engineering, University of Iceland, Reykjavik, Iceland. Since 2004, He held various positions with the Juelich Supercomputing Centre of Forschungszentrum Juelich, Germany. In addition, he is the Head of the joint High Productivity Data Processing research group between the Juelich Supercomputing Centre and the University of Iceland. Since 2020, he is also the EuroHPC Joint Undertaking governing board member for Iceland. His online YouTube and university lectures include high-performance computing - advanced scientific computing, cloud computing and big data - parallel and scalable machine and deep learning, as well as statistical data mining. In addition, he has performed numerous hands-on training events in parallel and scalable machine and deep learning techniques on cutting-edge HPC systems. His research interests include high-performance computing, remote sensing applications, medicine and health applications, pattern recognition, image processing, and data sciences, and he has authored extensively in those fields.

AN AUTOMATIC APPROACH FOR THE PRODUCTION OF A TIME SERIES OF CONSISTENT LAND-COVER MAPS BASED ON LONG-SHORT TERM MEMORY

Rocco Sedona[1,2]*, Claudia Paris*[3]*, Liang Tian*[2]*, Morris Riedel*[1,2]*, Gabriele Cavallaro*[1]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] School of Engineering and Natural Sciences, University of Iceland, Iceland
[3] Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, The Netherlands

IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-2792-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IGARSS46834.2022.9883655

ABSTRACT

This paper presents an approach that aims to produce a Time-Series (TS) of consistent Land-Cover (LC) maps, typically needed to perform environmental monitoring. First, it creates an annual training set for each TS to be classified, leveraging on publicly available thematic products. These annual training sets are then used to generate a set of preliminary LC maps that allow for the identification of the unchanged areas, i.e., the stable temporal component. Such areas can be used to define an informative and reliable multi-year training set, by selecting samples belonging to the different years for all the classes. The multi-year training set is finally employed to train a unique multi-year Long Short Term Memory (LSTM) model, which enhances the consistency of the annual LC maps. The preliminary results carried out on three TSs of Sentinel 2 images acquired in Italy in 2018, 2019 and 2020 demonstrates the capability of the method to improve the consistency of the annual LC maps. The agreement of the obtained maps is $\approx 78\%$, compared to the $\approx 74\%$ achieved by the LSTM models trained separately.

Index Terms— Deep Learning (DL) Models, Long Short Term Memory (LSTM), Time-Series (TS) of Consistent Land-Cover (LC) Maps, Multi-year training set.

1. INTRODUCTION

The dense Time-Series (TS) of images with a worldwide coverage provided by Sentinel 1 and Sentinel 2 allow the production of large-scale Land-Cover (LC) maps in a timely manner [1]. For this reasons, several methods have been recently proposed to produce maps at country, continental or global scale [2, 3]. However, when the maps are produced separately there is the risk of showing unrealistic year-to-year LC changes. The map consistency is crucial when monitoring complex environmental processes such as desertification, arctic greening or soil erosion. While a lot of effort has been devoted to generating annual maps, little has been done for the production of consistent thematic products.

In [4] a method for mapping global LC types from 2001 to 2010 at 250 m resolution with multiple year TSs of MODIS data is proposed. The strategy is to generate a map for each single year by using the data acquired in the preceding and subsequent years as well. In [5], the authors propose to apply a Hidden Markov Model (HMM) as a post-processing step to a TS of LC maps to help distinguish real LC change from spurious changes arising from errors in classification. On the one hand, these methods have demonstrated improvements on the temporal consistency of classification maps. On the other hand, these strategies may lead to the risk of losing inter-annual LC changes, especially when the analysis includes long TSs of data.

In this paper, we propose a novel approach which aims to produce a TS of consistent annual LC products based on the Long Short Term Memory (LSTM) multitemporal Deep Learning (DL) model. Contrary to the above-mentioned approaches, our method uses only the TS of the year under study and does not impose constraints with the application of the post-processing analysis. Furthermore, instead of separately classifying the TSs of Earth Observation (EO) data acquired in different years, our method trains one LSTM model with the multi-year training set to produce a TS of consistent LC maps. First, it extracts a training set per year leveraging on publicly available thematic product. The annual training sets are used to separately train different Random Forest (RF) models to detect unchanged area, which can be used to produce a reliable multi-year training set. Finally, the multi-year LSTM model is trained to produce a set of annual LC maps. The advantages of the proposed approach are: (1) the automatic production of a multi-year training set, (2) the use of a LSTM for capturing the temporal trends, and (3) the training of a unique LSTM model using multi-year TSs of EO data, which enhances the consistency of the annual LC maps.

Part of this work was performed in the CoE RAISE project which has received funding from the European Union's Horizon 2020 Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733. The authors gratefully acknowledge the computing resources from the DEEP-EST project, which received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement no. 754304.

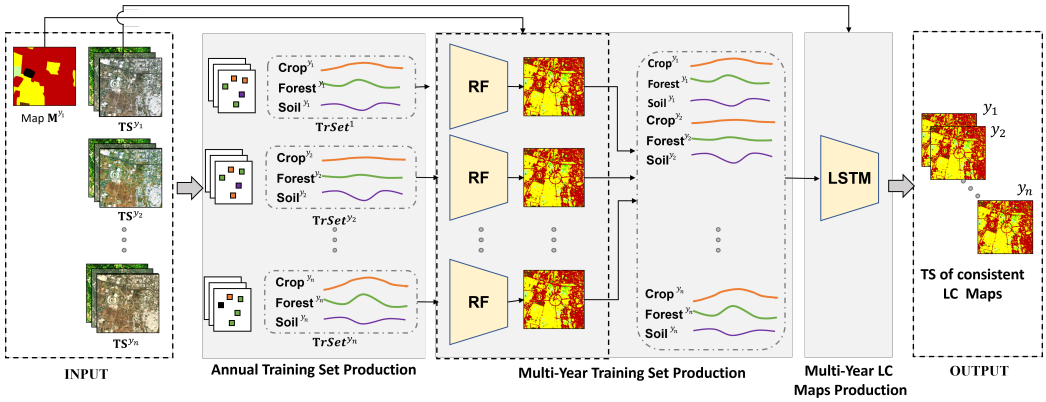


Fig. 1: Flow chart of the proposed approach which aims to produce a TS of consistent LC maps.

2. PROPOSED MULTI-YEAR MAPPING METHOD

Figure 1 shows the block-scheme of the proposed method made up of three main steps: (i) the annual training set production, (ii) the multi-year training set production, and (iii) the multi-year LC maps production.

2.1. Problem Formulation and Notation

In this section we formalize the multi-year LC mapping problem and define the notation used in the paper. Let $\mathbf{TS}^{y_i} = (\mathbf{X}_1^{y_i}, \mathbf{X}_2^{y_i}, \cdots, \mathbf{X}_q^{y_i})$ be the TS made up of the q images acquired in the ith year, where $\mathbf{X}_j^{y_i} \in \mathbb{R}^{m \times n \times b}$ is a multispectral image having $m \times n$ pixels and b spectral channels, with $j = [1, \cdots, q]$ and $i = [1, \cdots, k]$. The proposed method assumes that the k TSs are atmospherically corrected, are made up of the same number of q images, and are consistent from the temporal view point (the acquisition dates of the images in all the TSs are the same) [6]. Let $\mathbf{M}^{y_1} \in \mathbb{R}^{m \times n}$ be a publicly available thematic product contemporary to one TS of EO data considered and having LC classes $\Omega = \{\omega_u\}_{u=1}^{U}$. Here for simplicity we assume that the map is contemporary to the first TS, i.e., $\mathbf{TS}^{y_1}$, however this is not a strict requirement. The map is assumed to be co-registered to the EO data and to have the same spatial resolution.

The goal of the method is to generate a TS of consistent annual LC maps $\{\hat{\mathbf{M}}^{y_1}, \hat{\mathbf{M}}^{y_2}, \cdots, \hat{\mathbf{M}}^{y_k}\}$ leveraging on: (i) the publicly available thematic product $\mathbf{M}^{y_1}$ to support the production of a multi-year training set, (ii) the temporal correlation existing between multi-year TSs acquired in the same study area $\{\mathbf{TS}^{y_1}, \mathbf{TS}^{y_2}, \cdots, \mathbf{TS}^{y_k}\}$ and, (iii) the capability of the LSTM network to capture the temporal dynamic. Please note that the proposed method can be applied to any EO data without geographical constrains due to the availability of many global LC maps.

2.2. Annual Training Set Production

This step aims to automatically generate an annual training set for each TS of EO data that have to be classified, i.e., $\{\mathbf{TS}^{y_i}\}_{i=1}^{k}$, which will be used in the next step for the production of the multi-year training set. To this end, we considered an approach similar to the one presented in [7]. The method uses the information provided by the EO data to automatically detect and extract the most reliable map labeled units. In greater detail, for each LC class ω_u, we first select all the samples in the ith $\mathbf{TS}^{y_i}$ associated to this label. Then, an automatic clustering analysis is performed to remove spurious samples not correctly associated to that label (i.e., possible changes occurred on the ground or there are classification errors present in the map). To this end, the class samples are partitioned into t_{ω_u} clusters $\{\mathbf{C}^1_{\omega_u,y_i}, \mathbf{C}^2_{\omega_u,y_i}, \ldots, \mathbf{C}^{t_{\omega_u}}_{\omega_u,y_i}\}$ according to their spectral similarity. Based on the majority decision rule, it is reasonable to assume that the dominant cluster is made up of pixels having the highest probability to be correctly associated to ω_u. The clustering is applied in a feature space made up of a set of robust spectral indices strictly connected to the physical meaning of the LC classes, i.e., Vegetation Indices (i.e., NDVI and EVI), Water Index (NDWI), Snow Index (NDSI) and Soil Index as computed in the Sen2Cor processor[1].

Once the most reliable map units are identified per class, a stratified random sampling strategy is applied to generate training sets having LC prior probabilities proportionate to those reported in $\mathbf{M}^{y_1}$. At the end of this step, the method generates a set of annual training sets $\{\mathbf{TrSet}^{y_i}\}_{i=1}^{k}$, where $\mathbf{TrSet}^{y_i} = \{(\mathbf{x}_b^{y_i}, l_b^{y_i})\}_b$ is the training set associated to the ith year having $\mathbf{x}_b^{y_i} \in \mathbb{R}^{1 \times s}$ and $l_b^{y_i} \in \Omega$, with $s = q \times b$ is the number of spectral features per number of images in the TS.

[1] https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm

2.3. Multi-year Training Set Production

The goal of this step is the production of a multi-year training set by selecting samples belonging to different years for all the classes. Such training set aims to help the LSTM model in learning the different spectral signatures of the same LC class in the different years. First, the obtained annual training sets $\{\mathbf{TrSet}^{y_i}\}_{i=1}^{k}$ are used to separately train k RF classifiers, i.e., a RF per TS. Due to the ensemble-learning strategy that combines a large set of classification trees, such models are able to deal with the presence of noise in the training set. The RF classifiers generate a set of preliminary LC maps $\{\dot{\mathbf{M}}^{y_1}, \dot{\mathbf{M}}^{y_2}, \cdots, \dot{\mathbf{M}}^{y_k}\}$ that can be used to determine the non-changed areas, i.e., where all the classified maps agree with the original thematic product $\mathbf{M}^{y_1}$. Such map agreement allows us to select the most reliable unchanged pixels across the different years. Indeed, even though the maps may also provide useful information on possible changed areas, in the considered preliminary implementation of the method we aim to include only the most reliable samples in the multi-year training set. At the end of this step, we obtain the multi-year training set, $\mathbf{TrSet} = \{(\mathbf{x}_h^{y_i}, l_h^{y_i})\}_h$, with $i = [1, \cdots, k]$ and $h \approx k \times b$.

2.4. Multi-Year LC Maps Production

The last step of the proposed method aims to produce a set of consistent TS of LC maps. To this end, we train a unique LSTM model considering the multi-year training set generated in the previous step. That model is then used to produce the TS of LC maps by classifying the corresponding annual TSs. Please note that, differently from the literature, the method does not perform any temporal smoothing step to the TS of annual LC maps. Although effective, such post-processing step may lead to the loss of changes actually occurring on the ground. For the same reason, the classification map of each year is generated considering only the TS of that specific year, instead of considering images acquired in the preceding and subsequent years. In contrast, the developed multi-year training set helps the LSTM model to better capture the different behaviours of the pixels belonging to the same class in different years, thus implicitly reinforcing the temporal consistency. In particular, the use of the same LSTM to generate the TS of LC maps allows the reduction of pixel noise across the LC maps produced for the different years, while not hampering the detection of changes occurring on the ground. At the end of the proposed approach, we obtain the set of LC maps having Ω classes will be generated, i.e., $\{\hat{\mathbf{M}}^{y_1}, \hat{\mathbf{M}}^{y_2}, \cdots, \hat{\mathbf{M}}^{y_k}\}$.

3. DATASET DESCRIPTION

In our study, we make use of acquisitions taken by the Sentinel-2 satellites. The tile that we analyze is the T32TPS, covering the area of the Trentino region, Italy. We downloaded and pre-processed 20 paired acquisitions for each of the years 2018, 2019, 2020, with a maximum difference between the date of acquisition of $\delta = 6$ days between the different years. We excluded observations with considerable cloud coverage, namely where more than 40% of the pixels are assigned to clouds in the Scene Classification Layer (SCL) map. For each year, we extracted two sets of samples randomly selected from the unchanged areas of the output maps produced by the RF classifiers. The first set is used to train the multi-year LSTM considering approximately 24.000 samples, while the second set made up of 12.000 samples was used to evaluate the loss and accuracy scores on unseen data at training time. In particular, to extract the annual training sets we relied on the CORINE Land Cover (CLC) map [8] available on the European level, considering 10 widespread LC classes, i.e., "Artificial", "Grass", "Crops", "Mineral", "Rocks", "Sand", "Broadleaves", "Conifers", "Water", "Snow" .

4. EXPERIMENTAL SETUP AND RESULTS

To assess the effectiveness of the proposed approach, we compared the maps obtained with the ones generated by the LSTM separately trained per year, i.e., the standard baseline approach. Such single-year LSTM models were trained considering only the pseudo labels representing that year considering the corresponding TSs of 20 acquisitions. In contrast, the proposed approach was trained considering training samples extracted from all the TSs of images. To have a fair comparison, we considered the same LSTM model. In particular, we base our work on a *PyTorch* implementation of a LSTM with 4 layers and hidden dimension equal to 128 [2]. We trained the LSTM models for 100 epochs with the Adam optimizer on sequences of 20 observations, each with 10 features (the pixel values of the 10 considered bands), activating the dropout option. For the prediction, we split each acquisition in 25 patches of dimension 2560×2560px, and stack the features of each pixel for every observation of the year to compose the TS. We applied a median filter with kernel dimension equal to 11 to reduce the noise of the output thematic maps.

Table 1 shows the LC map agreement achieved per class and overall considering the standard baseline method, i.e., single-year LSTM and the proposed multi-year LSTM. One can notice that the proposed method is able to increase the consistency of the results obtained regardless of the LC class by increasing the overall map agreement of almost $\approx 4\%$ for both years. Figure 2 shows the number of LC changes per pixel for a portion of the considered study area when using: (a) the multi-year proposed method, (b) the single-year baseline method. Moreover, one image of 2018 and the corresponding LC map obtained with the proposed method is reported. From this result, one can notice that the proposed approach is able to reduce the classification errors at pixel level

[2] https://github.com/dl4sits/BreizhCrops

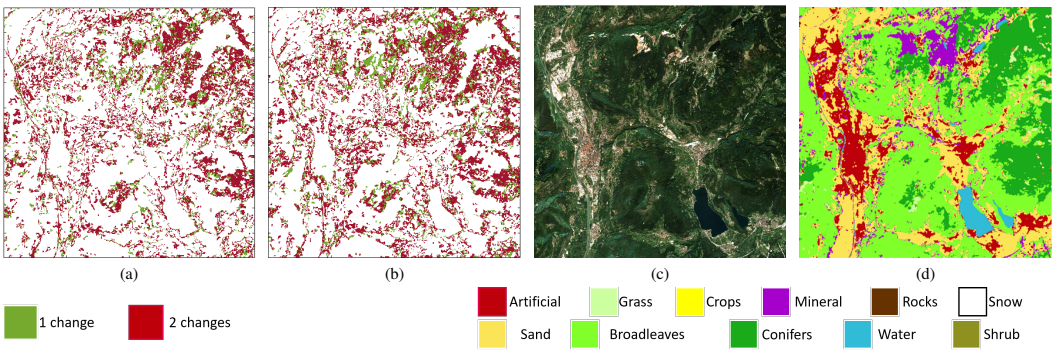


Fig. 2: Portion of the considered study area: (a) number of LC changes per pixel obtained with the proposed multi-year method, (b) number of LC changes per pixel obtained with the single year baseline method, (c) the true color composition of a Sentinel 2 image acquired in 2018, and (d) corresponding LC map obtained.

Table 1: LC map agreement achieved per class and overall considering the standard baseline method, i.e., single-year LSTM and the proposed multi-year LSTM.

Class	**Multi-Year LSTM**		**Single-Year LSTM**	
	2018-2019	2019-2020	2018-2019	2019-2020
Artificial	0.89	0.68	0.93	0.62
Grass	0.69	0.60	0.61	0.42
Crops	0.76	0.81	0.65	0.85
Mineral	0.33	0.51	0.34	0.43
Rocks	0.43	0.46	0.45	0.43
Sand	0.14	0.31	0.05	0.16
Broadleaves	0.85	0.87	0.79	0.84
Conifers	0.81	0.78	0.79	0.77
Shrubland	0.64	0.66	0.63	0.62
Water	0.97	0.98	0.98	0.95
Overall	0.78	0.78	0.74	0.74

as well as to reduce the detection of false changes with respect to the baseline method.

5. CONCLUSION

This paper presents a novel approach for the production of a TS of consistent LC maps by taking advantage of the temporal correlation existing between TSs acquired in different years in the same study area. The method first extracts an annual training set per TS to generate a set of preliminary LC maps. Then, it exploits the unchanged areas to define a reliable and informative multi-year training set to train a unique LSTM model. The preliminary results obtained demonstrate the effectiveness of the proposed approach. As future development, We plan to delve into the analysis of training methods to make use of longer TSs. A possible approach that can be investigated is the stateful LSTM to retain the cell state among prediction of sequences from the same pixel. The adoption of models that rely on both temporal and spatial correlation should also be considered for the creation of more consistent output maps.

6. REFERENCES

[1] K. D. Ngo, A. M. Lechner, and T. T. Vu, "Land cover mapping of the mekong delta to support natural resource management with multi-temporal sentinel-1a synthetic aperture radar imagery," *Remote Sensing Applications: Society and Environment*, vol. 17, pp. 100272, 2020.

[2] R. Malinowski, S. Lewiński, M. Rybicki, E. Gromny, M. Jenerowicz, M. Krupiński, A. Nowakowski, C. Wojtkowski, M. Krupiński, E. Krätzschmar, et al., "Automated production of a land cover/use map of europe based on sentinel-2 imagery," *Remote Sensing*, vol. 12, no. 21, pp. 3523, 2020.

[3] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes, "Operational high resolution land cover map production at the country scale using satellite image time series," *Remote Sensing*, vol. 9, no. 1, 2017.

[4] J. Wang, Y. Zhao, C. Li, L. Yu, D. Liu, and P. Gong, "Mapping global land cover in 2001 and 2010 with spatial-temporal consistency at 250m resolution," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 103, pp. 38–47, 2015, Global Land Cover Mapping and Monitoring.

[5] S. P. Abercrombie and M. A. Friedl, "Improving the consistency of multitemporal land cover maps using a hidden markov model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 2, pp. 703–713, 2015.

[6] S. M. Vicente-Serrano, F. Pérez-Cabello, and T. Lasanta, "Assessment of radiometric correction techniques in analyzing vegetation variability and change using time series of landsat images," *Remote Sensing of Environment*, vol. 112, no. 10, pp. 3916–3934, 2008.

[7] C. Paris and L. Bruzzone, "A novel approach to the unsupervised extraction of reliable training samples from thematic products," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1930–1948, 2020.

[8] M. Bossard, J. Feranec, and J. Otahel, "CORINE land cover technical guide – Addendum 2000," Tech. Rep. 40, European Environment Agency, Copenhagen, 2000.

IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 20, 2023 2505805

Toward the Production of Spatiotemporally Consistent Annual Land Cover Maps Using Sentinel-2 Time Series

Rocco Sedona, *Member, IEEE*, Claudia Paris, *Senior Member, IEEE*, Jan Ebert, Morris Riedel, *Member, IEEE*, and Gabriele Cavallaro, *Senior Member, IEEE*

***Abstract*— Land cover (LC) maps generated by the classification of remote-sensing (RS) data allow for monitoring Earth processes and the dynamics of objects and phenomena. For accurate LC variability quantification in environmental monitoring, maps need to be spatiotemporally consistent, continually updated, and indicate permanent changes. However, producing frequent and spatiotemporally consistent LC maps is challenging because it involves balancing the need for temporal consistency with the risk of missing real changes. In this work, we propose a scalable and semiautomatic method for generating annual LC maps with labels that are consistently applied from one year to the next. It uses a Transformer deep-learning (DL) model as a classifier, which is trained on satellite time series (TS) of images using high performance computing (HPC). The trained model can generate stable maps by shifting the prediction window along the temporal direction. The effectiveness of the proposed approach is tested qualitatively and quantitatively on a multiannual Sentinel-2 dataset acquired over a three-year period in a study area located in the southern Italian Alps.**

***Index Terms*— Deep learning (DL), high-performance computing, remote sensing (RS), spatiotemporally consistent land cover (LC) maps, supervised classification, time series (TS).**

I. Introduction

CLASSIFICATION maps are vital for analyzing patterns on the Earth's surface across many research areas. Using open remote-sensing (RS) data, such as Sentinel-2 and Landsat-8 images, we can avoid the high costs associated with field surveys. These data enable more frequent global mapping of land cover (LC) classes, which is crucial for the timely identification of changes from extreme weather events and natural disasters as well as tracking urban construction trends. Brown et al. [1] used Google Earth Engine to retrieve Sentinel-2 data and train a convolutional neural network (CNN) model to generate frequent LC maps. The European Space Agency (ESA) has also recently released a global LC product based on Sentinel-2 and Sentinel-1 data[1] with a spatial resolution of 10 m. Although these approaches can produce accurate LC products, they do not avoid the risk of generating spatiotemporally inconsistent classification maps that show unrealistic year-to-year LC changes.

To guarantee spatiotemporal consistency across temporal updates, LC maps need to be generated with algorithms that are robust to noise (i.e., false alterations do not hinder real changes) and capable of generalizing on a temporal level. Furthermore, classification schemes must use consistent setups over time to ensure that LC classifications are semantically interoperable and can be compared. A method was proposed in [2] for mapping global LC types using multiple years time series (TS) of MODIS data. The method involves generating a yearly map using data from the preceding and subsequent years. Abercrombie and Friedl [3] proposed using a hidden Markov model (HMM) as a postprocessing step on a TS of LC maps to differentiate real LC change from spurious changes caused by classification errors. Recent works have focused on improving the temporal consistency of multiyear classification maps while maintaining sensitivity to LC changes [4], [5].

While these methods have improved the temporal consistency of classification maps, they may also risk obscuring inter- and intraannual LC changes, particularly when analyzing long TS of data. In addition, these methods are typically computationally demanding. This letter presents a scalable, semiautomatic approach using a Transformer deep-learning (DL) model to produce spatiotemporally consistent LC maps by continually ingesting satellite data. To this end, the proposed approach decouples the stable component of the LC map from the detection of changes, leveraging the ability of the self-attention Transformer model to classify long TS of RS data accurately. Finally, a user-guided analysis is requested to validate the LC mapping result obtained from the semantic viewpoint. The main contribution of this work is the proposal of a method that can use high-performance computing (HPC) systems to produce high-resolution LC maps that are: 1) regularly updated (e.g., annually); 2) able to identify permanent changes accurately; and 3) spatiotemporally consistent to

Manuscript received 27 August 2023; revised 7 October 2023; accepted 26 October 2023. Date of publication 1 November 2023; date of current version 13 November 2023. This work was supported in part by the Center of Excellence (CoE) Research on AI- and Simulation-Based Engineering at Exascale (RAISE) through European Union (EU) Horizon 2020 Research and Innovation Framework Program H2020-INFRAEDI-2019-1 under Grant 951733 and in part by the EUROCC2 Project through European High-Performance Computing Joint Undertaking (JU) and EU/European Economic Area (EEA) States under Grant 101101903. *(Corresponding author: Gabriele Cavallaro.)*

Rocco Sedona, Morris Riedel, and Gabriele Cavallaro are with the Jülich Supercomputing Center (JSC), 52428 Jülich, Germany, and also with the Department of Computer Science, University of Iceland, 107 Reykjavik, Iceland (e-mail: r.sedona@fz-juelich.de; morris@hi.is; g.cavallaro@fz-juelich.de).

Claudia Paris is with the Department of Natural Resources, Faculty of Geoinformation Science and Earth Observation, University of Twente, 7514 AE Enschede, The Netherlands (e-mail: c.paris@utwente.nl).

Jan Ebert is with the Jülich Supercomputing Center (JSC), 52428 Jülich, Germany (e-mail: ja.ebert@fz-juelich.de).

Digital Object Identifier 10.1109/LGRS.2023.3329428

[1]ESA WorldCover 2021: https://worldcover2021.esa.int/.

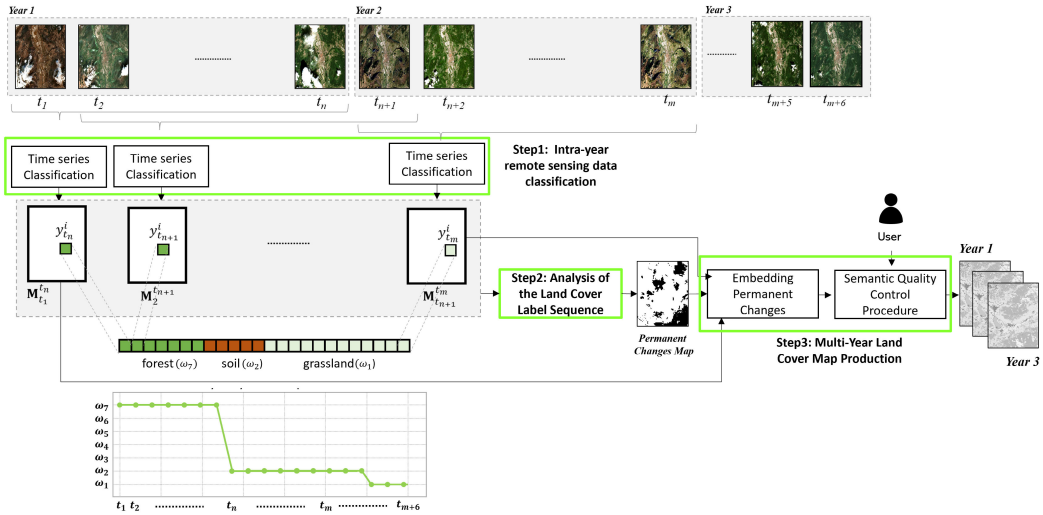


Fig. 1. Proposed workflow that classifies the intrayear RS data stream to generate spatiotemporally consistent LC maps. The three main steps of the method are highlighted in green. In the final step, the user is requested to validate the automatically detected changes only from the semantic viewpoint.

facilitate the monitoring of ongoing environmental processes (e.g., desertification, urbanization, and deforestation).

II. Consistent Multiyear Maps Production

Fig. 1 shows the workflow of the proposed method (PM), which is based on three main steps: 1) intrayear RS data stream classification; 2) analysis of the LC label sequence; and 3) the production of multiyear LC maps. To generate spatiotemporal coherent thematic products, the PM produces an up-to-date LC map every time new RS data is acquired. This condition allows us to generate an intrayear sequence of LC labels that can be used to: 1) reduce spurious year-to-year changes; 2) strengthen the consistency of the annual maps; and 3) continually improve the accuracy of previously generated maps via a backpropagation strategy. To ensure high-quality mapping, the user is involved in the final step to revise the obtained classification results only from the semantic viewpoint, that is, approve/discard the presence of a change and assign the new LC label.

A. Problem Formulation and Notation

Annual LC maps are typically generated by classifying TS of RS images instead of individual RS data, as it yields more accurate results. Although this approach is effective, it has two main limitations. First, the classifier assigns each pixel the LC label that best fits most of the RS images in the TS. This leads to accurate results for stable LC components and land surface seasonality (e.g., crop phenology), while it may discard permanent changes visible in a small portion of the TS (e.g., occurred at the end of the year). Second, the spatiotemporal consistency of annual LC maps generated by separately classifying different TS of images is affected by classification errors that do not indicate real changes.

Let us focus on the first target year γ_1, where a training set representing the k LC classes present in the scene $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$ is used to train a classification model C^{γ_1}. Let $\mathbf{TS}_{t_1}^{t_n} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ be the TS of n RS images of γ_1, where $\mathbf{X}_1 \in \mathbb{R}^{r \times q \times b}$ is the RS image acquired at time t_1 having $r \times q$ pixels and b spectral channels. By applying C^{γ_1} to the TS of images acquired from t_1 to t_n, we generate the LC map $\mathbf{M}_{t_1}^{t_n}$. Let $\mathbf{TS}_{t_{n+1}}^{t_m} = (\mathbf{X}_{n+1}, \mathbf{X}_{n+2}, \ldots, \mathbf{X}_m)$ be the TS of RS images acquired for the subsequent year γ_2, and $\mathbf{M}_{t_{n+1}}^{t_m}$ be the corresponding annual map. Let $\mathbf{x}^i$ be the i pixel of the considered TS, which is associated with the annual LC labels $y_{t_n}^i$ and $y_{t_m}^i$ visible in $\mathbf{M}_{t_1}^{t_n}$ and $\mathbf{M}_{t_{n+1}}^{t_m}$, respectively, where $y_{t_n}^i, y_{t_m}^i \in \Omega$. For simplicity, we focus here on two years. However, the PM can be easily scaled to multiyear data.

B. Intrayear RS Data Stream Classification

To strengthen the consistency of the annual LC maps, the intrayear RS data stream is classified using the same model C^{γ_1} by shifting the prediction window along the temporal direction. For instance, at the beginning of γ_2, the updated version of $\mathbf{M}_{t_1}^{t_n}$ (i.e., $\mathbf{M}_{t_2}^{t_{n+1}}$) is generated by classifying the TS of images $\mathbf{TS}_{t_2}^{t_{n+1}} = (\mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_{n+1})$, which includes the first image of γ_2 and excludes the first image of γ_1, that is, $\mathbf{X}_{n+1}$ and $\mathbf{X}_1$, respectively. In this letter, we use a Transformer deep learning (DL) model as a classifier, which has rapidly challenged recurrent networks in tasks using sequences of data [6]. Transformers can be scaled efficiently because they are based on the concept of attention, which is inherently nonrecurrent, allowing sequence parallelization during training. Moreover, Transformers can model both short- and long-term dependencies since each attention head attends to information from different positions in the sequence. Multiple heads of attention are used to obtain different representations of this attention information. This leads to stable classification results obtained even under varying cloud cover conditions, thus enabling the

classification of the continuous stream of RS data without needing any data preparation step. At the end of γ_2, the method generates a TS of LC maps ($\mathbf{M}_{t_1}^{t_n}, \mathbf{M}_{t_2}^{t_{n+1}}, \ldots, \mathbf{M}_{t_{n+1}}^{t_m}$) obtained by classifying the TS ($\mathbf{TS}_{t_1}^{t_n}, \mathbf{TS}_{t_2}^{t_{n+1}}, \ldots, \mathbf{TS}_{t_{n+1}}^{t_m}$). At the end of this step, for each pixel $\mathbf{x}^i$, we have an LC label sequence ($y_{t_n}^i, y_{n+1}^i, \ldots, y_{t_m}^i$) (where n and m are the number of RS images acquired in the year γ_1 and γ_2, respectively) that allows us to model the temporal trajectory of each pixel, independent of neighboring pixels (see Fig. 1).

C. Analysis of the LC Label Sequence

Once the label sequences are completed for at least two years, we can start the production of the spatiotemporal consistent annual LC maps. LC maps can be divided into three main components: 1) stable component ϕ_{un}; 2) seasonal component ϕ_s (e.g., snow and crop phenology); and 3) permanent changes ϕ_c (e.g., deforestation and urbanization). In this work, we focus our attention on annual maps that aim to represent stable components and interannual permanent changes, neglecting the seasonality of the land surface. For instance, for the "Snow" LC class, only the permanent snowfields are represented in an annual thematic product, while the snow patches visible only in the winter period are typically neglected.

To automatically discriminate the stable LC component from the land surface seasonality and the permanent changes, we perform an analysis of the LC label sequence generated in the previous step, that is, ($\mathbf{M}_{t_1}^{t_n}, \mathbf{M}_{t_2}^{t_{n+1}}, \ldots, \mathbf{M}_{t_{n+1}}^{t_m}$). First, the labels are mapped from categorical into numerical data. Then, we apply a 1-D median filter $\in \mathbb{R}^{1\times p}$ with length equal to p to the LC labels sequence pixel, that is, ($y_{t_n}^i, y_{n+1}^i, \ldots, y_{t_m}^i$), where $i \in [1, r \times q]$, to reduce noisy classification results. The binary difference between the obtained annual maps $\mathbf{M}_{t_1}^{t_n}$ and $\mathbf{M}_{t_{n+1}}^{t_m}$ is computed (i.e., the standard method used in the literature to detect changes between maps). This binary output provides all the changes that occurred on the ground, that is, [ϕ_c, ϕ_s]. To remove noisy changes and determine when they occurred, the binary intrayear LC label sequence is convoluted with a step function to retrieve the maximum. The date of permanent changes can be automatically computed by shifting its index backward along the temporal axis of half of the inference window. Finally, we disentangle the permanent changes ϕ_c from the seasonal ones ϕ_s. In the considered implementation of the method, we assume that a change can be identified as permanent if: 1) it is visible in the TS of LC maps for at least half of the inference window (stable change); 2) the LC sequence must not return to its initial LC more than once since this kind of patterns typically belongs to seasonal changes ϕ_s; and 3) changes that occurred at the beginning of the TS of LC are discarded to avoid unreliable changes.

D. Multiyear LC Maps Production

In the final step, the method generates the spatiotemporally consistent TS of LC maps by embedding the permanent changes identified in the previous step. Changes having an area lower than a certain threshold (ath) are discarded. By shifting the prediction window along the temporal direction, the method can capture permanent changes that occur at any time of the year. The changes are correctly embedded in the annual LC maps according to the date estimated in the previous step. The estimated change date is injected into the yearly LC output to: 1) increase the accuracy of previously generated LC maps and 2) avoid incorrectly identifying changes in the subsequent years. The new yearly map is produced by adding the detected permanent changes onto the initial map. The integration of the automatically detected permanent changes is finally confirmed by the user, who can visually check their presence in the RS data and assign the new LC label. We would like to remark that the user is involved only in checking the change from the semantic viewpoint (i.e., the presence of the change and the new LC label). The PM automatically identifies the identification of the changes and their geometrical extent in an unsupervised way. The proposed quality control procedure carried out by the user aims to increase the reliability of the final thematic products.

III. Dataset Description and Experimental Setup

The considered study area is located in the northeast part of Italy, where a major deforestation event occurred in 2018 due to storm Vaia, which caused significant forest damage. To generate the TS of annual LC maps, we considered the Sentinel-2 data (Tile TPS32) able to acquire dense TS of images at high spatial resolution (10 m). The considered TS comprises 45 images acquired from January 2018 to December 2020, that is, 15 yearly acquisitions equally distributed over the three years. The length of the median filter $p = 5$, while the area threshold ath is equal to 100 according to the minimum mapping unit of the changes that we aim to detect. Only the images having cloud coverage lower than 40% were automatically downloaded through the Sentinelsat API. The available training set representative of 2018 was reporting nine LC classes, namely, "Artificial land," "Grass," "Crops," "Bareland," "Broadleaves," "Conifers," "Shrub," "Water," and "Snow." The 2018 training dataset is made up of 10 000 points per class (a total of 80 000 samples for the training set), located in an area spatially disjoint from the predicted one used in the experiments.

We used a Transformer with five encoding layers, each with two attention heads. We set the number of expected features in the encoder layers $d_{\text{model}} = 128$. No positional embedding was used. We carried out the training of the model for 100 epochs. The Adam optimizer [7] was used with a cyclic learning rate schedule with its initial value set to 0.003. The code is based upon a PyTorch implementation of the Transformer [8]. The experiments were run on the JURECA-dc[2] HPC at the Jülich Supercomputing Center (JSC), in particular, on the partition where each node is equipped with four Nvidia A100 graphics processing units (GPUs). The Transformer was trained from scratch using a training dataset representing the LC present in the scene, associated with CORINE LC labels, for the year 2018 (i.e., on the sequence of 15 acquisitions), and it was used to generate the map of the first target year, that is, $\mathbf{M}_{t_1}^{t_n}$ and the intrayear LC maps. We used PyTorch DistributedDataParallel to scale the training of the Transformer on multiple GPUs, speeding it up from ∼500 s on one GPU to ∼45 s on 16 GPUs. That model is then used for inference, shifting the prediction

[2]JURECA-dc: https://apps.fz-juelich.de/jsc/hps/jureca/index.html.

TABLE I

INTERYEAR AGREEMENT OF LC MAPS OF 2018–2020 EVALUATED AT THE CLASS LEVEL BY COMPARING THE PM WITH THE BASELINES: 1) RF AND 2) TRANSFORMER

		RF			Transformer			PM		
	Class	2018-2019	2019-2020	2018-2020	2018-2019	2019-2020	2018-2020	2018-2019	2019-2020	2018-2020
	Artificial land	85.4	83.1	80.3	79.3	88.0	84.4	100	100	100
	Grass	87.5	79.6	79.6	84.7	52.7	62.6	100	100	100
	Crop	88.0	85.1	81.2	79.3	77.8	74.2	99.9	99.9	99.9
	Bareland	70.5	73.4	58.3	72.7	46.4	46.4	99.8	98.8	98.7
	Broadleaves	88.1	90.5	82.8	79.6	88.1	87.3	99.9	99.8	99.7
	Conifers	87.0	86.9	79.0	88.3	83.0	84.6	97.5	95.5	93.1
	Shrub	74.3	81.8	66.7	45.0	62.8	64.6	99.9	99.9	99.9
	Water	97.2	97.7	96.4	93.4	78.1	80.1	99.9	99.9	99.9
	Overall	86.6	86.7	79.9	80.4	80.3	81.2	99.1	98.5	97.7

TABLE II

PRECISION, RECALL, F1 SCORE, AND INTERSECTION OVER UNION (IOU) FOR THE RF, TRANSFORMER, AND PM AGAINST THE REFERENCE MAP REPRESENTING THE LC CHANGES

	precision	recall	F1 score	IoU
RF	0.20	0.67	0.30	0.18
Transformer	0.13	0.75	0.22	0.13
PM	0.72	0.59	0.65	0.48

window of length equal to 15 acquisitions along the temporal direction from the end of 2018 to the end of 2020. Due to the availability of 30 Sentinel-2 images acquired in 2019 and 2020, 30 intraannual maps have been generated. We performed the inference for each pixel of the TPS32 tile (∼120 000 000 pixels) to generate an output label for each element of the input sequence. Reference data created by expert annotators were used to validate the results.[3]

IV. EXPERIMENTS RESULTS

The multiyear annual maps generated by the PM are compared with those generated by separately classifying the annual TS of Sentinel-2 images with the: 1) Transformer and 2) random forest (RF). While the comparison with the Transformer allows us to emphasize the added value of the PM, RF is typically used as the baseline method due to its robustness to noise. Fig. 2 shows a qualitative comparison of the agreement between the LC maps produced by the baseline methods and the PM. As expected, the baseline methods' interyear agreement appears noisier than that produced by the PM, which can correctly detect major changes while keeping the stable LC component invariant between years. These qualitative results are confirmed by the quantitative ones reported in Table I. As expected for the "Bareland" and "Broadleaves" classes, all the methods show an agreement lower than 90% due to the impact of the permanent deforestation event that occurred in November 2018, which also had consequences in subsequent years. However, although no changes occurred for most of the LC classes, the baseline methods show lower interyear agreement than the one achieved by the PM. For instance, the "Artificial land" class presents an agreement lower than 90% for the baselines, while the PM correctly keeps this class in the stable LC component. Similarly, the

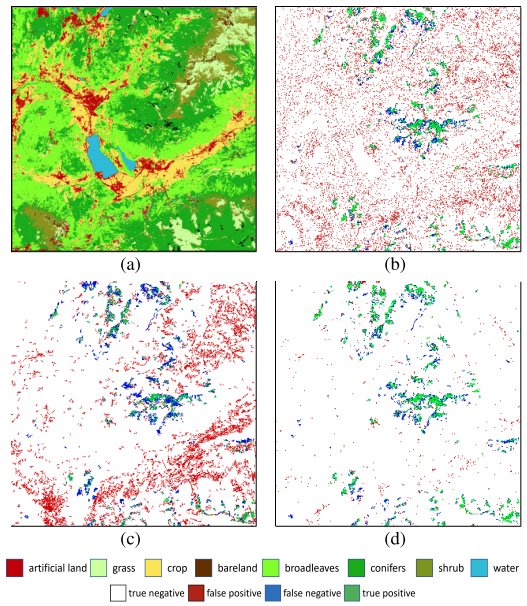


Fig. 2. Comparison of the multiyear (2018–2020) LC maps agreement (a) 2018 LC map $\mathbf{M}_{t_1}^{t_n}$ produced by the Transformer, (b) RF change map, (c) Transformer change map, and (d) PM change map. The true positive (green), true negative (white), false positive (red), and false negative (blue) are computed considering the reference change map.

"Shrub" class is frequently confused with "Grass," leading to LC variation in the annual maps that do not correspond to real changes. Table II reports quantitative results of the detected change for the three methods considering the reference change map. The PM shows significantly better scores for the detected changes than results obtained by the two baseline methods because of the sharp reduction of false-positive changes. Fig. 3 shows a small portion of the study area. The Sentinel-2 images acquired in 2018–2020 clearly show deforestation in late 2018 (i.e., November 2018). Differently from the PM, neither the RF nor the baseline Transformer can detect it in the 2018 annual maps. This is a common problem when generating annual LC maps. The changes that occurred at the end of the year are not identified by the classifier. Note that the LC changes that occurred at the end of 2018 are correctly identified by

[3]Vaia storm (Italy): https://www.politicheagricole.it/flex/cm/pages/ServeBLOB.php/L/IT/IDPagina/18158.

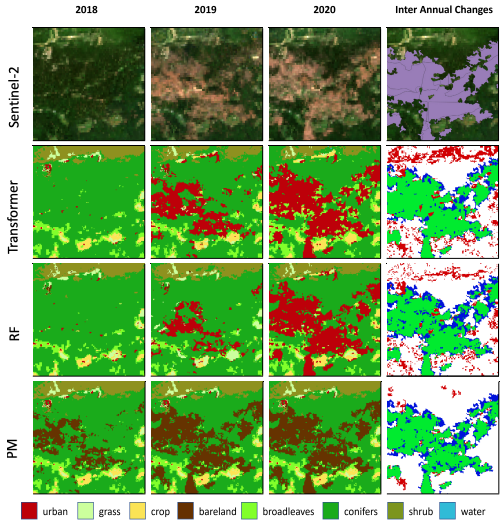


Fig. 3. Examples of multiyear maps and changes obtained for a small portion of the considered study area for 2018–2020 using the: 1) Transformer, 2) RF, and 3) PM. The Sentinel-2 images acquired in 2018–2020 are reported with the changes reference map.

the PM when moving the prediction window into 2019 (i.e., when the change is present in at least eight images out of the 15 of the considered prediction window). However, the date of the change can be correctly backpropagated in the 2018 annual map when the user validates the presence of the change. Moreover, from the results obtained, one can see the spatiotemporal consistency of the LC maps generated by the PM, in which classification errors present at the pixel level are neglected. Furthermore, because of the visual check of the user, the correct LC label can be assigned to the changed areas, that is, "Bareland," thus ensuring high-quality mapping. In contrast, both the baseline methods misclassify the deforestation areas with "Artificial land" pixels because of the similarity of the urban and bare rocks spectral signatures.

V. Conclusion

We proposed a method for updating LC maps and detecting permanent changes while maintaining spatiotemporal consistency in the resulting products.[4] In the future, we plan to investigate how to extract additional information from the Transformer model, such as visualizing the attention weights, to provide users with more detailed insights into the temporal dynamics of the LC products. Although we did not apply any manual correction to the produced maps in this work, active learning methods could be investigated to continually provide feedback to the model and produce more accurate LC maps. Dynamically adjusting the set of LC classes is vital for enhancing model robustness and reliability by addressing unpredictability in real operative scenarios, ensuring consistent performance and adaptability to variations on the ground. Moreover, we plan to update the initial training data to maintain reliable LC products and, in future developments, test the method on datasets from various areas to address different classification challenges.

[4]GitHub repository: https://gitlab.jsc.fz-juelich.de/sdlrs/land cover-classification-framework/-/tree/rocco-DL-module.

Acknowledgment

The authors would like to thank the computing time granted by the Jülich Aachen Research Alliance (JARA) Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA at the JSC. They would also like to thank the local unit of Helmholtz AI at the JSC for the support. They also thank the Autonomous Province of Trento and the Veneto Region for reference data of the storm Vaia and Daniele Marinelli for his invaluable suggestions.

References

[1] C. F. Brown et al., "Dynamic world, near real-time global 10 m land use land cover mapping," *Sci. Data*, vol. 9, no. 1, Jun. 2022, Art. no. 251.
[2] J. Wang, Y. Zhao, C. Li, L. Yu, D. Liu, and P. Gong, "Mapping global land cover in 2001 and 2010 with spatial–temporal consistency at 250 m resolution," *ISPRS J. Photogramm. Remote Sens.*, vol. 103, pp. 38–47, May 2015.
[3] S. P. Abercrombie and M. A. Friedl, "Improving the consistency of multitemporal land cover maps using a hidden Markov model," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 703–713, Feb. 2016.
[4] J. F. Brown et al., "Lessons learned implementing an operational continuous United States national land change monitoring capability: The land change monitoring, assessment, and projection (LCMAP) approach," *Remote Sens. Environ.*, vol. 238, Mar. 2020, Art. no. 111356, doi: 10.1016/j.rse.2019.111356.
[5] R. Sedona, C. Paris, L. Tian, M. Riedel, and G. Cavallaro, "An automatic approach for the production of a time series of consistent land-cover maps based on long-short term memory," in *Proc. IEEE IGARSS*, Jul. 2022, pp. 203–206.
[6] A. Vaswani et al., "Attention is all you need," 2017, *arXiv:1706.03762*.
[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[8] M. Rußwurm, C. Pelletier, M. Zollner, S. Lefèvre, and M. Körner, "Breizhcrops: A time series dataset for crop type mapping," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B2-2020, pp. 1545–1551, Aug. 2020.

B. Code Repositories

Adhering to the principles of FAIR (Findability, Accessibility, Interoperability, and Reusability) [105], the author has contributed to the following open source repositories:

- `https://gitlab.com/rocco.sedona/mdpi-paper-bigearth`, utilizing a ResNet50 on a version of BigEarthNet [95] converted to HDF5 for large scale training on HPC systems.

- `https://gitlab.com/rocco.sedona/igarss2020_paper`, which closely follows the aforementioned work but includes also the LARS optimizer [111].

- `https://gitlab.jsc.fz-juelich.de/sedona3/igarss2021_sat6`, with the code for training a ResNet50 on the SAT4 and SAT6 datasets [7] with the LAMB optimizer [112].

- `https://gitlab.jsc.fz-juelich.de/sedona3/mgan` collects the Tensorflow code utilized for the harmonization of S2 and S2 data using a multispectral adaptation of pix2pix.

- The branch `https://gitlab.jsc.fz-juelich.de/sdlrs/land-cover-classification-framework/-/tree/rocco-DL-module` (currently private, to be opened once the review of Paper VI is finished) collects the code used by the author to train LSTM and Transformer models on TS of data acquired by S2 for LC classification.

- The author contributed with the code for pre-processing BigEarthNet and training DL models using data parallelism to the repository located at `https://gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/switching_bs`.