

Intelligent decision making on-board satellites



Vít Ružička
Hertford College
University of Oxford

DPhil Thesis
Primary supervisor: Andrew Markham
Secondary supervisor: Niki Trigoni

Hilary 2025

Acknowledgements

Here, I would like to thank several groups of people who were instrumental for the completion of this thesis and have positively influenced my life during the years spent at the Oxford University.

Firstly, I would like to thank my supervisor Andrew Markham, who has helped me during these long four years of research - helped and pushed me to try to answer the right and important questions. I also want to thank my co-supervisor, Niki Trigoni, with whom we met as a lab group during our first year of study.

I would also like to thank professors and teachers who inspired me in the first place to go on this academic path - Franz Franchetti, Eunsu Kang, Konrad Schindler and Eva Podzimeková.

Further, I want to thank the entire Frontier Development Lab (FDL) community, which first accepted me as a researcher and excited me about conducting Space research, and later allowed me in as a team lead. I consider many of these my colleagues, but more importantly also as my friends. They allowed me to tackle interesting problems, and to realise and appreciate the depths of other application domains. Particularly my thanks belong to my FDL 2021 teammates and mentors Gonzalo Mateo-Garcia, Anna Vaughan, Daniele De Martini, James Fulton, Valentina Salvatelli, Valentina Zantedeschi and Chris Bridges. And my FDL 2022 team with Beichen Zhang, Huiqi Wang, Amani Alabri, Karol Bot, Cole McCall and Dale Hamilton. My big thanks also belongs to James Parr, who is the heart and brains behind the FDL community and motivates all of us, and to Cormac Purcell who has helped me with several of my projects conducted with FDL. *Ad astra per algorithmos!*

I would also like to thank the team inside the ESA's PhiLab for welcoming me in their lab in Frascati for couple of months, namely to Nicolas Longép  , and to the friends I made there.

And, because not everything is about work, I would also like to thank all my friends and relatives for their support. I want to thank my fellow schoolmates from Oxford University and from the Hertford College, you all made the experience of studying there much more pleasant.

My thanks and best wishes for his own research and life belong to Guopeng Chen who has been studying alongside me for the entire duration of my studies. Also, to Matias Mattamala with whom I could discuss about PhD life.

To the people from the Banbury house, a small family made in the midst of the Corona chaos - and from the same year also to Çağla Usta and Katie Slaughter (take care folks!).

To the Czech community at Oxford and to Peter Kessler for tirelessly organising the Magdalen Monday Movies society (allowing me to decongest my brain from the usual problems with a good film). I would also like to thank the News Offices at the University of Oxford and at the Computer Science department.

To my friends in Czech Republic, who repeatedly agreed to meet up at random times when I (suddenly / finally) show up.

To the members of our lab in the Cyberphysical Systems group - Sangyun Shin, Madhu Vankadari, Jialu Wang, Jiaying Zhong, Kai Lu - with whom we shared similar experiences and difficulties of being doctoral students, and with whom we also shared tips on how to survive.

Finally, I would like to thank people who are the most important to me. I want to thank my partner Qian Feng for being there for me, including during hard times and for walking alongside me despite these (from her I know the advice to “*take it seriously easy*”). Of course, I want to thank my family, my parents Alice and Jan Růžičkovi, without whom I would never be where I am and who I am. Your lives inspired me and shaped me. You keep making me search for more meaning in what I do. You gave me my love for the Arts and Sciences, the interest in the world, because of you I keep exploring and try to make my own experience about everything before judging it. I will also forever remember the advice you gave me on my first trip abroad into the unknown: *Be the inferior of no man, or of any men be superior. (Nad nikoho se nepovyšuj, před nikým se neponižuj.)* a very livable quote from William Saroyan.

I also thank my little brother Kajetán Růžička (who is not so little anymore), for being my motivation for searching for yet another cool example of space technology or programming example that I can show him to inspire him about sciences, studying, knowledge-making and creativity.

Because of all of you, this makes sense!

Abstract

In this thesis we explore opportunities that are given to us with Remote Sensing data collected by satellites and their processing using Machine Learning. Satellites provide unprecedented quantity of data, repeated and uniform observations of Earth and other planetary surfaces and they allow for detection of many events of interest. Some of these events are closely tied with one of the key challenges of this generation, Climate Change.

However, the current systems on most devices in communication constrained environments (such as satellites) do not rely on on-board processing of the observed data and operate only in passive modes. They need to be manually scheduled, and the data needs to be manually processed by experts on the ground. Furthermore, the quantity of the observed data is often larger than the communication capabilities and the downlinked data is limited. As a result, events of potential interest are often lost or detected late. The uplink constraints are even more severe, making this an interesting asymmetrical scenario that could be addressed with model adaptation on-board.

The central topic of this thesis is the question of how to enable smart on-board processing of data to allow for autonomous decision making. We present three works where we explore this challenge.

In our first project, we successfully deploy a change detection machine learning model on-board of a real satellite and we test it in space. Our encoder-decoder model enables data compression into latent vectors, multi-temporal comparisons between observed images and allows training on-board of the satellite as an experimental proxy for adaptation.

In our second project, we focus on processing data from a variety of sensors for the task of automated methane leak detection. We build benchmark datasets of hyperspectral data and explore classically used methods which we improve with trained models.

Finally, in our last project, we design novel machine learning architectures aiming at deployment of automated general detectors on-board of hyperspectral satellites. We show that imaging spectroscopy data requires specific adaptations of machine learning architectures to prevent informational and computational bottlenecks. We demonstrate the efficacy of our models on two example tasks, further improving methane detection and exploring mineral identification from space. We measure the inference speeds of our models on low-compute hardware which was previously tested in harsh space environments.

Contents

List of Figures	v
List of Tables	ix
List of Abbreviations	xi
1 Introduction	1
1.1 Overview	1
1.2 Challenges and Motivations	3
1.3 Research Questions	6
1.4 Research Contributions and Overview	8
1.5 Outputs	11
1.5.1 Core publications	11
1.5.2 Media releases	13
1.5.3 Other co-authored publications	13
1.5.4 Datasets and public resources	15
2 Background Literature	16
2.1 Overview	16
2.2 Primer on Remote Sensing	16
2.2.1 Instruments	17
2.2.2 Levels of processing	20
2.2.3 Data portals, datasets	20
2.3 Primer on Machine Learning	22
2.3.1 Task descriptions	23
2.3.2 Training	24
2.3.3 Model architectures	26
2.4 Primer on Methane Detection	27
2.4.1 Motivation	27
2.4.2 Methane visibility and sensor considerations	30
2.4.3 Methane detection pipeline	31
2.4.4 Classical methods	33

2.4.5	Deep learning approaches	34
2.5	Machine Learning for Remote Sensing	36
2.5.1	Example models used with Remote Sensing data	37
2.5.2	Model adaptations for Remote Sensing data	38
2.6	Systems in compute constrained environments	39
2.6.1	Autonomy on-board	39
2.7	Efficient machine learning	43
2.7.1	Machine learning techniques	44
2.8	Definitions of efficiency	45
2.9	Summary	47
3	Change Detection on Satellites	49
3.1	Introduction	49
3.2	Background	51
3.2.1	Prior methods	51
3.2.2	Challenges and opportunities	54
3.3	Method	54
3.4	Data	55
3.4.1	Instrument	55
3.4.2	Datasets	55
3.4.3	Pre-processing	57
3.5	Models	57
3.5.1	Task formalisation	57
3.5.2	Baselines	58
3.5.3	RaVAEn models	60
3.5.4	Model adaptations through training on-board	61
3.6	Experimental Setup	62
3.6.1	Devices	63
3.7	Results	63
3.7.1	Change detection on disaster events	63
3.7.2	Inference times	65
3.7.3	Latent space and compression	67
3.7.4	Training on-board	68
3.8	Discussion	69
3.9	Summary	70

4	Methane Detection	73
4.1	Introduction	73
4.2	Background	76
4.2.1	Prior methods	76
4.2.2	Challenges and opportunities	77
4.3	Method	77
4.4	Data	78
4.4.1	Hyperspectral dataset STARCOP	78
4.4.2	Multispectral views of the data	81
4.4.3	Data from EMIT sensor	81
4.5	Models	82
4.5.1	Feature extraction from multispectral data	83
4.5.2	Feature extraction from hyperspectral data	84
4.5.3	Machine learning models	86
4.6	Experimental Setup	87
4.7	Results	89
4.7.1	Performance on multispectral data	89
4.7.2	Performance on hyperspectral data	91
4.7.3	Zero-shot generalisation on EMIT	92
4.8	Discussion	93
4.9	Summary	94
5	Universal Detection from Satellites	97
5.1	Introduction	97
5.2	Background	100
5.2.1	Domain	100
5.2.2	Datasets	102
5.2.2.1	Methane leak datasets	103
5.2.2.2	Mineral datasets	104
5.2.3	Machine learning models	104
5.2.4	On-board processing	105
5.2.5	Challenges and opportunities	106
5.3	Method	106
5.4	Data	107
5.4.1	Mineral maps	107
5.4.2	Methane leaks	109

5.4.2.1	Synthetic events	110
5.4.2.2	Real events	111
5.4.3	Datasets	112
5.5	Models	114
5.5.1	Baseline approaches	115
5.5.2	Deep machine learning	116
5.5.3	Current limitations	117
5.5.4	Model adaptations	119
5.6	Experimental Setup	120
5.6.1	Tasks	120
5.6.2	Hyperparameters	121
5.6.3	Devices	123
5.7	Results	124
5.7.1	Methane detection on EMIT	124
5.7.2	Methane detection on AVIRIS-NG	130
5.7.3	Mineral identification	130
5.7.4	Inference times	131
5.8	Discussion	133
5.8.1	Methane detection	134
5.8.2	Mineral identification	136
5.8.3	On-board deployment	136
5.9	Summary	138
6	Conclusions	143
6.1	Discussion	143
6.1.1	Change detection on-board (Q1)	144
6.1.2	Methane detection in hyperspectral data (Q2)	144
6.1.3	Universal detectors (Q3)	145
6.2	Future Work	146
6.2.1	Deploying machine learning models	149
	Bibliography	153

List of Figures

1.1	High-level illustration of on-board processing with machine learning models for increased autonomy.	1
1.2	Example task with a change detection model.	4
1.3	Current state of methane detection depends on manual interpretation of the downlinked data by trained experts.	5
1.4	Universal detection satellite equipped with general sensing capabilities.	6
2.1	Simplified illustration of light passing through the Earth’s atmosphere.	17
2.2	Visualisation of the electromagnetic spectrum.	18
2.3	Simplified abstraction of machine learning model training and inference schema.	22
2.4	Illustration of the estimated temperature by the end of the century depending on different reductions of greenhouse gasses.	28
2.5	Illustration of the presence of methane signal in comparison with the bands available in commonly used satellites.	30
2.6	Outline of hypothetical methane detection pipeline showing the individual steps and their formalisation as machine learning tasks.	32
3.1	Illustration of the change detection task on-board of satellite.	50
3.2	Photos of the D-Orbit satellites.	51
3.3	Illustration of the traditional processing pipelines and the difference that on-board processing could enable.	52
3.4	Illustration of the proposed RaVAEn system.	55
3.5	Locations used for training and validation images.	56
3.6	Diagram of the RaVAEn system detailing the training and inference steps.	59
3.7	Timing of the model for tasks of change detection and few-shot learning on-board of satellites.	62
3.8	Comparison of changes detected using the baseline and the <i>large</i> VAE model on examples of flooding and fire disaster events.	64

3.9	Timed model inferences, using the RaVAEn encoder network with different devices available available on the satellite.	66
3.10	Detailed view into the individual batches used in the RaVAEn encoder, showing delays in the CPU regime.	67
3.11	UMAP visualisation of encoded tiles from a flooded scene.	67
3.12	Training a classification model on top of the encoder network.	68
3.13	Predictions of the on-board trained model and visualisation of samples from the training dataset.	69
3.14	Example data containing two types of changes.	71
4.1	Predictions of the STARCOP models on example plume event from the AVIRIS-NG data.	74
4.2	Illustration of the automated methane detection using machine learning models and the unification of processing pipelines used with different sensors.	75
4.3	Examples of the AVIRIS-NG data captures.	78
4.4	Showing the limitations of the available annotation in the original labels.	79
4.5	Hyperspectral dataset made from the AVIRIS-NG sensor.	80
4.6	Visualisation of band ratios used to highlight methane plume.	84
4.7	The proposed HyperSTARCOP and MultiSTARCOP machine learning models.	86
4.8	Showing the qualitative results of our models evaluated on strong and weak plume samples and on confounders.	90
4.9	Results on example plume and confounder samples from the EMIT dataset.	92
5.1	Illustration of unifying processing pipelines for different downstream tasks with general machine learning models.	98
5.2	Aggregation of mineral components into mineral classes.	108
5.3	Illustration of the new bi-temporal dataset with near-global coverage.	111
5.4	Locations of tiles used to create the OxHyper datasets of EMIT data.	113
5.5	Comparison of the Spectral Angle Mapper (SAM) and Tetracorder methods over one example scene.	116
5.6	Illustration of the limitations of previous approaches and our proposed HyperspectralViTs, adapted Transformer based architectures.	118
5.7	Illustration of the Hyper SegFormer model for semantic segmentation of methane plumes in Hyperspectral data.	119

5.8	Qualitative results on the OxHyperSyntheticCH4 dataset of synthetic methane leak events.	125
5.9	Quantitative results on the OxHyperRealCH4 dataset.	128
5.10	Qualitative results on the OxHyperRealCH4 dataset of real methane leak events.	129
5.11	Qualitative results comparing our new models on the benchmark STAR-COP dataset.	131
5.12	Qualitative results comparing our new models on the mineral dataset OxHyperMinerals showing our proposed Hyper SegFormer model variants.	133
5.13	Detailed timings of different methods for processing a full granule of EMIT data.	134
5.14	Illustration of the methane detection system deployed either on-board of the satellite, or with an on-ground element and the potential gained speed-ups in detections in contrast with the current traditional pipelines.	139
6.1	Visualisation of the uncertainty estimation method used with an ensemble of trained models.	147
6.2	Proposal of pre-training Foundation models on general tasks such as fine-grained mineral identification and later using them for other specific task.	148
6.3	United Nations PlumeViewer application using the HyperSTARCOP model.	150

List of Tables

1.1	Released datasets with links to publicly accessible data.	15
2.1	Example current and future multispectral and hyperspectral sensors.	19
2.2	Example remote sensing datasets.	21
2.3	Types of example machine learning tasks with a typical remote sensing data sample on the input.	23
2.4	Selected machine learning models used to detect methane in multispectral data.	35
2.5	Selected machine learning models used to detect methane in hyperspectral data.	35
2.6	Example missions in Space with autonomy on-board of limited compute hardware.	40
2.7	Commercial off-the-shelf devices with hardware specifications.	41
3.1	The RaVÆn dataset statistics.	56
3.2	Differences in the architecture for different proposed model sizes.	60
3.3	AUPRC for baseline and VAE methods with time window $k = 1$	64
3.4	AUPRC for the best performing metrics with and without extended history.	65
3.5	AUPRC for models with different latent sizes.	65
3.6	AUPRC and timings for different sizes of model.	66
4.1	Distribution of the tiles stratified by emission rate into train and test sets.	80
4.2	Results of the MultiSTARCOP models on the test set.	91
4.3	Results of the HyperSTARCOP models on the test set.	91
4.4	Results of the hyperspectral models on EMIT.	93
5.1	Overview of the OxHyper datasets created in this chapter.	112
5.2	Quantitative results on the OxHyperSyntheticCH4 dataset.	126
5.3	Ablation study of the HyperSegFormer architectures on the OxHyper-SyntheticCH4 dataset.	127

5.4	Ablation study of the HyperEfficientViT architectures on the OxHyperSyntheticCH4 dataset.	127
5.5	Quantitative results on the benchmark STARCOP dataset.	130
5.6	Quantitative results on the mineral dataset OxHyperMinerals.	132
5.7	Timing of models on different devices.	132

List of Abbreviations

AI		Artificial intelligence
AUPRC		Area under the precision recall curve
AVIRIS-NG		Airborne Visible InfraRed Imaging Spectrometer - Next Generation
BCE		Binary cross-entropy
BOA		Bottom of atmosphere
C&DH		Command and data handling
CHIME		Copernicus Hyperspectral Imaging Mission for the Environment
CNN		Convolutional neural network
COTS		Commercial off-the-shelf
CV		Computer vision
CVA		Change vector analysis
DAG		Directed acyclic graph
EMIT		Earth Surface Mineral Dust Source Investigation
ESA		European Space Agency
FCNN		Fully convolutional neural network
FPGA		Field programmable gate array
GAN		Generative adversarial network
GHG		Greenhouse gas
GPU		Graphics processing unit
HPC		High performance computing
HSI		Hyperspectral instrument
IID		Independent and identically distributed

IME		Integrated mass enhancement
ION SCV		InOrbit Now Satellite Carrier Vehicle
IoT		Internet of things
IoU		Intersection over union
IPCC		Intergovernmental Panel on Climate Change
IR		Infrared
ISS		International Space Station
LES		Large eddy simulation
LSTM		Long short term memory
MAE		Mean absolute error
MF		Matched filter
ML		Machine learning
MLR		Multiple linear regression
MSE		Mean squared error
MSI		Multispectral instrument
NASA		National Aeronautics and Space Administration
NDVI		Normalized difference vegetation index
NDWI		Normalized difference water index
NIR		Near infrared
PCA		Principal component analysis
RF		Random Forest
RGB		Red green blue
RNN		Recurrent neural network
RS		Remote sensing

S/M/L WIR	Short/Middle/Long wavelength infrared
SRF	Spectral response function
SVM	Support vector machine
SWaP	Size, weight and power
TOA	Top of atmosphere
UAV	Unmanned aerial vehicle
VAE	Variational autoencoder
VO	Visual odometry
VPU	Vision processing unit
VSWIR	Visible to short wavelength infrared

Introduction

1.1 Overview

This thesis focuses on processing Remote Sensing data with Machine Learning models in tasks connected to areas such as humanitarian aid and climate change related processes. More concretely it focuses on challenges and opportunities arising from the deployment of Machine Learning models on-board of devices to increase the autonomy in communication constrained environments, such as satellites, or robots on other planetary bodies. Examples of autonomous behaviour enabled by on-board machine learning models is illustrated on Figure 1.1. The proposed models also aim to address and leverage sensor-specific properties of the data, most importantly the high spectral dimensionality of the captured images - in extreme scenarios working with hyperspectral sensors.

Aiming at on-board deployment of the used machine learning models requires considerations for the efficiency of the used models, as the compute of these devices is

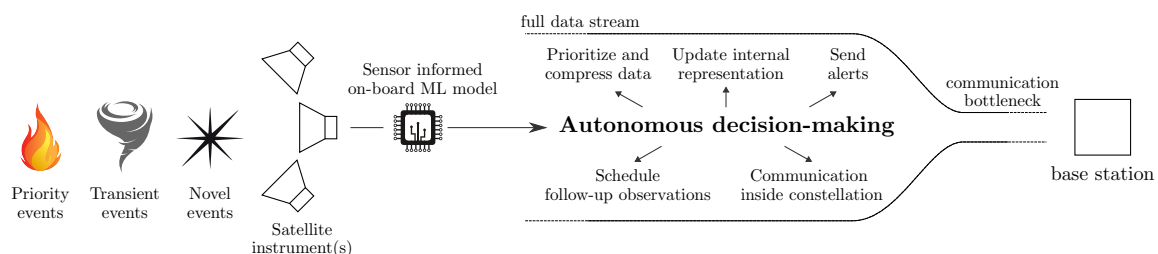


Figure 1.1: High-level illustration of on-board processing with machine learning models for increased autonomy on satellites.

typically quite low [Meoni *et al.*, 2024]. It is therefore important to test these models on reasonably representative compute environments that serve as proxies for the deployed devices. On-board deployment however enables several crucial capabilities of autonomous behaviour. An early detection of an event can trigger an alert that can save lives (for example detecting wildfires in remote locations before they spread, or fast delineation of flooded areas). Such alert can also use priority channels, as it wouldn't need to send large quantities of data. Also, in cases of satellite and other deep space missions, the uplink is even more limited than the downlink. It is therefore of interest to also explore adaptability of the pre-deployed models to newly observed data.

This thesis focuses mostly on the domain of remote sensing data, as it allows for homogeneous observations of the whole world and gives vision into areas that are otherwise hard to access by humans. It however requires adaptation of machine learning models typically used in computer vision for a new domain.

We further focus on one type of remote sensing data, that is hyperspectral data, which has much larger footprint and the requirements for downlinked data is also larger. Typical classical methods also process these large data volumes slowly, which is potentially a limiting factor for on-board deployment. Additionally, there aren't many large, annotated datasets of high-resolution hyperspectral data, which serve as benchmarks for training machine learning models. Hyperspectral data is needed to detect events that typically don't appear in spectral ranges used in multispectral sensors, or in cases where these events get collated with other types of background features.

One such example is the detection of the presence of methane gas in captured images [Sánchez-García *et al.*, 2022]. Super-emitter methane leak events can be detected even with multispectral sensors, however smaller methane leaks and accurate delineation of plumes, which is needed to quantify the size of the leak, can only be detected in hyperspectral data. Additionally, the range of types of events that can be detected and also differentiated in hyperspectral data is much larger than in cases of other sensors - taking a maximalist approach, one could conceive a general satellite which would be capable of detecting all types of materials inside the observed data and which could decompose these detections into individual chemical components directly from Space using general machine learning models. Such capabilities would be of interest in many applied scenarios. Examples listed by [Cawse-Nicholson *et al.*, 2021; Rast *et al.*, 2021] include detection of target trace gases, precise types and health of plants from space or the detections of surface mineral types and their

grain sizes. Furthermore, NASA’s Decadal survey [NASEM, 2019] states that for example for identification and quantification of chemical components in plant canopies, imaging spectrometers (hyperspectral sensors) are the only suitable technology.

In combination, exploring machine learning models adapted to specifically work with hyperspectral data which would also be deployable on-board of satellites, would allow for quite significant reduction of the downlinked and stored data. Autonomy inside a constellation of satellites with different sensing capabilities would also allow for improved detections by autonomously scheduling follow-up measurements of specialised satellites after a tip-off signal from an initial weak detection made by a general satellite.

1.2 Challenges and Motivations

Here we outline a list of challenges encountered with the current state of technology of satellites and their sensors and the current state-of-the-art machine learning models. We explore recent innovations and present motivation for the research conducted in this thesis.

Outside of specialised cases and recent demonstrator missions, satellites observe the Earth using traditional non-reactive processing pipelines [Furano *et al.*, 2020]. This leads to slow reaction times relying on on-the-ground data processing even in critical cases related to humanitarian assistance, disaster response and high priority targets connected to Climate Change. On-the-ground processing pipelines can take up to few days for detection after the potential event of interest was observed. Additionally, many sensors operate in an on-demand mode and capture data only after manual scheduling of acquisitions, which prevents the discovery of new events at unexpected locations. With systems running on-board of the satellites, the initial detection could lead to intelligent decision-making and could trigger an array of potential follow-up actions. These detections could trigger alert signals within minutes after the observation leading to a more responsive and reactive system. This could enable event discovery and automated follow-up observation scheduling inside constellations of satellites. By intelligently selecting the salient data of interest (anomalies, changes, detected events) we could lower the requirements of communication channels by downlinking only the relevant data - this is illustrated on Figure 1.2. This would be also relevant for deep space missions with even more restricted communication channels, where imaging spectroscopy data processing on-board could enable autonomous targeting of places of interest and selection of salient data for downlinking.



Figure 1.2: Example task with a change detection model. Having the change map predicted on-board of a satellite could be used for several triggered actions - for example sending a fast alert signal or downlinking only the relevant parts of the image (or even just the predicted mask).

The computation power of traditional satellites is highly constrained, as the their so-called payload subsystem doesn’t do any advanced data processing [Maini & Agrawal, 2014]. For example the Landsat 8 satellite uses a RAD750 CPU (with only up to 200 MHz) for it’s Command and Data Handling (C&DH) Subsystem [Berger *et al.*, 2001; Nelson *et al.*, 2012] - this is less powerful than a nowadays mobile phone. Only several demonstrator missions by NASA [Thompson *et al.*, 2012; Swope *et al.*, 2022], ESA [Giuffrida *et al.*, 2021; Marin *et al.*, 2021] and private sector [Růžicka *et al.*, 2023a; Mateo-Garcia *et al.*, 2023] test specific hardware accelerators in the harsh space environment and outline what will be possible in the future.

Remote sensing data enables unique monitoring capabilities of areas around the world with specialised sensors that provide observations at wavelengths outside of the visible spectrum typical for computer vision data. While this data enables a wide array of tasks, it also requires very specific classical algorithms that usually need domain knowledge which gets encoded into rule-based decision systems. The detections made by these classical methods can be quite noisy and may require manual intervention. We illustrate this on Figure 1.3 on the task of methane detection. Given large datacubes of data with high spectral dimension, more advanced methods also have slow processing speeds and may require auxiliary and heavily processed data. These factors prevent easy deployment on-board of autonomous devices, where automated detections are needed. We can leverage machine learning models to either clean and post-process these classical products (to differentiate between confounders and real events) or employ deep learning models to avoid the use of classical methods altogether. As hyperspectral data is not commonly used in Computer Vision research, the proposed machine learning architectures may require specific adaptations to this domain. Designing models to comply with the constraints of limited compute devices,

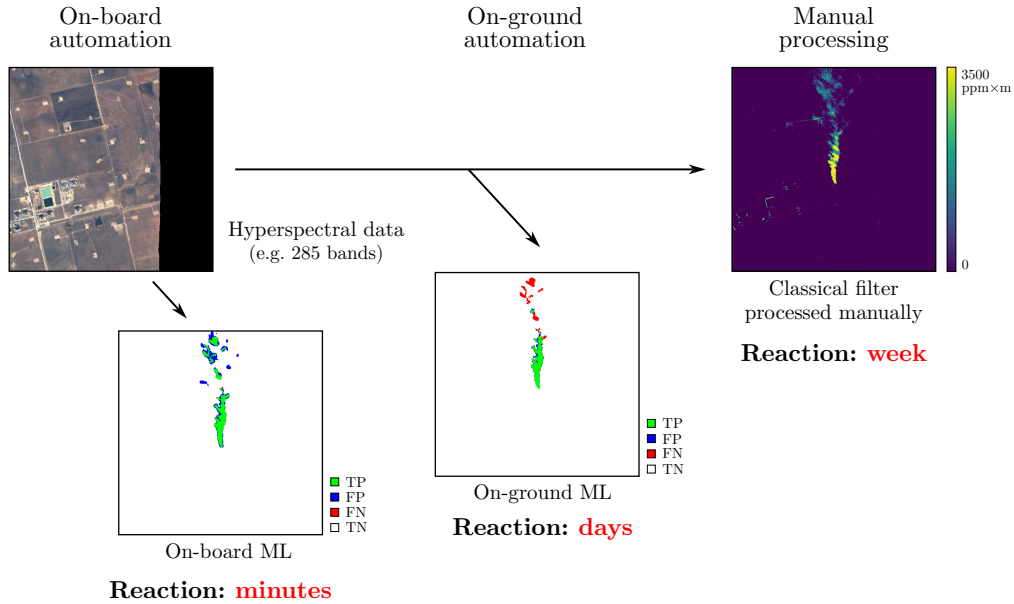


Figure 1.3: Current state of methane detection depends on manual interpretation of the downlinked data by trained experts (shown in the first row). Machine learning models processing of the data on-ground could aid their search and models running on-board of satellites could enable detections within just few minutes after the capture of the data.

while efficiently accessing all the information contained in the data is a challenge we address in this thesis.

Many traditional satellites provide very specific sensing capabilities as they aren't targeted for general use. Machine learning models applied on data from these satellites heavily depend on the specifications of the used sensors. These were designed with very specific target observables, which influences the choice of the observed wavelengths. Correspondingly, the created datasets of captures from different remote sensing sensors often focus only on a few specific locations and events of interest. It is difficult to evaluate and compare proposed machine learning models on this diverse data. Some of the used sensors however observe at wide spectral ranges targeting many downstream applications at once. Only recently, missions like EMIT [Green, 2022] started providing near-global hyperspectral coverage of the Earth with publicly accessible data archives.

On Figure 1.4 we envision general retrainable machine learning architectures that can be tuned to different target chemical component signatures. These would enable universal detection satellites, that would host general models easily extended to new detection tasks of interest. Satellites equipped with hyperspectral imaging spectroscopy sensors are particularly suited for this as they support a large number

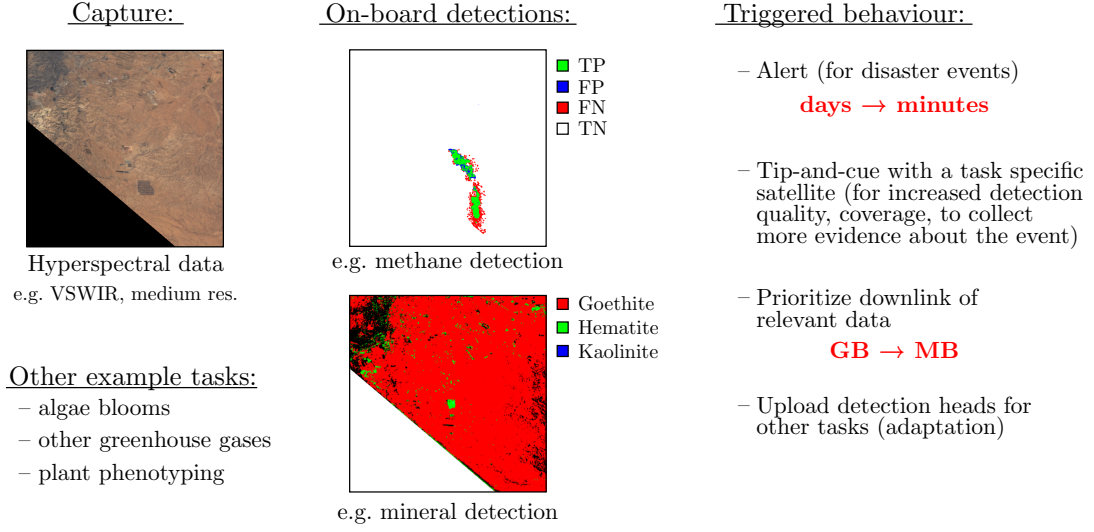


Figure 1.4: Universal detection satellite equipped with general sensing capabilities (for example VSWIR sensor with medium resolution) could host machine learning models that detect many types of events at once - here we show a prediction of a methane leak (actually present at this location) and a mineral map of three selected mineral classes (later explored in this thesis).

of observable features. The recently created public data archives [Green, 2022] and machine learning research in unsupervised learning and Foundation models [Yang *et al.*, 2023] make this a prime area of research.

1.3 Research Questions

Motivated by the outlined challenges, this thesis aims to address the central question of enabling smart data processing on-board of satellites to allow for intelligent decision-making. We split this large topic into several more easily approachable research questions that we cover in each of the chapters of this thesis.

Q1 - How can we perform change detection on-board of a satellite?

Change can be treated as an universal signal for the chance of collecting new information. This holds true for remote observations of our Earth, where change could correspond to regions of human activity including Climate Change relevant events, or for observations on other planetary bodies, where they could indicate new impact craters or areas of recent geologic activity. Can we leverage unsupervised change detection methods to detect potentially any event of interest, while being robust to noise and small variations of on-board low-level data? When comparing a current state of

the captured scene, we also need to have information about its past states. This could either be stored from previous revisits at costs of large memory requirements or communicated inside a network of satellites in a constellation. Can we design models that use low dimensional latent vectors instead of full data captures to reach lower storage and communication requirements? How can we design small models to allow for fast inference speeds to match the data capture rates? Proposed research methods need to be tested on devices that serve as realistic proxies for satellites or be deployed on real missions.

Q2 - How can we detect a particular faint signal (e.g. methane) on-board satellites with different sensors?

In cases when the detected phenomena is very weak and can be observed only with specific instruments (such as imaging spectrometers also known as hyperspectral imagers) we need methods that augment the signal of the event over the background data. These further need to be adjusted depending on the sensor used to collect the data. As an example, we explore the detection of methane, a greenhouse gas transparent in the visible spectrum, which is overlaid on top of variable surface scenes. Classical methods formulate the problem as anomaly detection and subtract estimations of the background in search for a modelled chemical signature of the gas. Given imperfect information and observation noise, they produce many false positive detections and trigger on other confounder materials. In most cases, they require manual interventions of domain experts, which prevents their automation and the potential deployment on-board of autonomous systems. Can machine learning models automate detections while being robust to confounder events?

Q3 - How can we design a universal detector that could be trained to detect any chemical signal of interest without modification?

Data from hyperspectral sensors supports unprecedented range of potential applications, however it may require specialised processing pipelines. Models explored in computer vision research do not consider data with high spectral dimension and as such their use with remote sensing data can lead to loss of information. Can we envision novel machine learning architectures that respect the high spectral dimensionality of the data, while complying with low compute restrictions of typical deployed hardware? How can these models support multiple tasks of interest, in the extreme the potential detection of any target chemical component present in the captured scene? How can we use these models as a strong foundation for adapting to

any other downstream task of interest? Furthermore, can we entirely replace classical methods with Deep Learning systems and end-to-end machine learning models trained on low levels of processing data without any other auxiliary information? Can we simulate synthetic datasets and events and later use these for detection on real-world data? Can we prepare models before the relevant mission has even launched? Could these enable a future of autonomous universal detection satellites?

1.4 Research Contributions and Overview

Here we briefly summarise the content of each chapter, link them to the outlined research questions and list our contributions.

Chapter 2 - Background Literature

Chapter 2 provides a general background which is then referred to in the later chapters. We address some of the **domain** specific research used for the applications of interest alongside with the typically used data sources. We also provide a primer on some of the machine learning **methods** we use in this thesis. Finally, we delve into the **systems** specific knowledge that will allow us to design machine learning models tuned for specific hardware platforms. Additionally, each of the following chapters focuses on self-contained project, and if we use a specific technique or data source, we go into more depth in that chapter’s background literature section.

Chapter 3 - Change Detection on Satellites

Broadly speaking, in Chapter 3 we explore the usage of small unsupervised change detection models for deployment on satellites. We explore simple machine learning architectures, focusing on small memory imprints and fast processing speeds. We remain application agnostic, by framing this task as an unsupervised change detection problem in a time-series of geographically collocated data. Our system is evaluated on a dataset of disaster events from data captured by the Sentinel-2 satellite. The main focus of this chapter is to propose a **change detection system on-board a satellite (Q1)**.

Our contributions in this chapter are:

- We train an unsupervised change detection system called **RaVAEn** based on the VAE architecture on a large dataset of Sentinel-2 images.

- The system is shown to generalise to samples of floods, wildfires, landslides and hurricanes. We demonstrate that our latent-space comparison method outperforms image-space baselines on a newly created dataset of disaster events by on average 18% in the AUPRC score (by 27% in hurricane and by 6% in fire scenarios).
- We measure model inference speeds on several proxy devices and deploy the model on-board of a real satellite flown by the industrial partner D-Orbit – specifically on the ION-SCV 004 satellite which is part of the *Dashing through the Stars* mission. There we conduct the world’s first experiments in training small neural network classifiers on-board of the satellite.
- On-board of the ION SCV 004 satellite we demonstrate change detection in about 0.11 seconds per a full captured scene (area of 4.8 km x 4.8 km). When comparing inference speeds between devices, we encounter almost 16.5x times speedup when running our models on the Myriad X chip on the ION SCV satellite in contrast to the earlier runs on a CPU of the Xilinx Pynq board. Training of an entire machine learning model takes only about 1.55s seconds including the encoding time.

Chapter 4 - Methane Detection

In Chapter 4, we explore **faint signal detection on-board of satellites with hyperspectral and multispectral sensors (Q2)**, in the form of semantic segmentation of methane leak plumes. This specific application is of unique interest to us, given its timely nature - methane is the second most important contributor to global warming through radiative forcing. Yet it is also the most actionable target for reduction given its relatively short atmospheric lifetime. Many new satellite missions are being launched to observe methane emissions and as such the data deluge will be enormous and will require automation.

Our contributions in this chapter are:

- We propose the **STARCOP** models based on the U-Net architecture and train them on data from different sensors and compare them against classical baselines. We see reductions in false positive detections and significant improvements in scores for methane leak detection.

- Our hyperspectral model improves the F1 score over a widely used matched filter baseline by 21.51% for strong events and by 8.68% for weak events while reducing the false positive rate by over 42.12%. Meanwhile our multispectral model improves the F1 score of classical band ratios baselines from the very low 7.44 F1 score for strong events up to 31.89.
- Furthermore, we demonstrate generalisation capabilities of our model trained on the AVIRIS-NG data when evaluating it on data from the EMIT sensor, the model improves the F1 score of the matched filter baseline by 40.28% on a small dataset of manually found events.
- With multispectral and hyperspectral model variants we propose a tip-and-cue system for satellite constellations with non-heterogeneous sensing capabilities.
- We curate a new dataset, improve existing labels and release the data publicly to aid research in this domain. Instead of having to download and process almost 6TB of raw data, we package the dataset into machine learning ready formats at vastly reduced sizes (60 or 633 GB depending on the needed subset of spectral bands).

Chapter 5 - Universal Detection from Satellites

In Chapter 5 our focus broadens from a single application of interest (of methane leak detection) into a generalised detection scenario. We focus on the **design of universal detection architectures (Q3)** working with hyperspectral data, which we demonstrate on two example downstream tasks of methane detection and mineral identification. To do so, we adapt existing model machine learning models to better process hyperspectral data in an end-to-end manner. We identify informational bottlenecks present in the default versions of the models (if they were used with hyperspectral data), and the limitations of relying on classical features.

Our contributions in this chapter are:

- We propose **HyperspectralViT architectures**, adaptations of machine learning models based on the Transformer architecture using a modular design of three explored changes. We explore these changes in ablation studies with two popular and recent semantic segmentation models.
- We create three large new datasets from the near-global EMIT data and we release these publicly under the shared name **OxHyper datasets**. First dataset

contains simulated methane leak events, second one is a smaller dataset of real methane leak events. The third dataset contains novel generated labels for the task of mineral identification.

- On the demonstrated tasks we show further improvements of our adapted end-to-end models over classical machine learning approaches. In methane detection we reach the new state-of-the-art results improving scores on a synthetically created dataset by 27.87% in F1 (and by 43.85% in IoU) over previous model (proposed in the previous chapter) and by 97.26% in F1 (and by 155% in IoU) over the matched filter baseline.
- We further demonstrate that pre-training machine learning models on a synthetically created dataset has positive impacts for detection of real events, which is highly relevant for newly deployed sensors. Fine-tuning our models pre-trained on the synthetic dataset improves the F1 score by 6.9% over the scenario of only training from the real data from scratch (in both cases using our best proposed model variant).
- We measure the significantly reduced processing time across several devices which serve as proxies for the hardware available on satellites (and other communication constrained devices). Using our newly proposed models, we estimate that to process an entire archive of data collected by the used satellite in one day we'd need 2.5 hours (with 30 sec per single capture) on a proxy of the space tested environment on the ION-SCV 004 satellite and only 3.2 minutes (with 0.64 sec per single capture) if we used high-power GPU on the Jetson AGX Xavier board. We would gain about 317x times speed-up over models depending on the MF baseline (with 203.3 sec per single capture).

Finally in Chapter 6, we conclude the thesis and answer the research questions outlined here in the introduction. While we dedicate each of these chapters to one central question, some topics span across the entire thesis.

1.5 Outputs

1.5.1 Core publications

This thesis compiles the work presented in several past publications. For the first authored publications we also detail the contributions made in each paper. In most cases, we release the code and the pre-trained model weights.

Publications directly relevant to the chapters in this thesis are:

1. **Růžička, V.**, Vaughan, A., De Martini, D., Fulton, J., Salvatelli, V., Bridges, C., Mateo-García, G., Zantedeschi, V. (2022, October). RaVÆn: unsupervised change detection of extreme events using ML on-board satellites. *Scientific reports*, 12(1), 16939. Růžička *et al.* [2022]
Codebase: <https://github.com/spaceml-org/RaVAEn>.
In this paper, V.R. was responsible as the first author for writing, reviewing, conducting experiments and their evaluation and publishing code and results. He also shares the conception of the experiments, coding and measuring the model on real-world hardware.
2. **Růžička, V.**, Mateo-García, G., Bridges, C., Brunskill, C., Purcell, C., Longépé, N., Markham, A. (2023, July). Fast model inference and training on-board of satellites. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium* (pp. 2002-2005). IEEE. Růžička *et al.* [2023a]
Codebase: <https://github.com/previtus/RaVAEn-unibap-dorbit>.
In this paper, V.R. was responsible for conceptualisation, methodology, experiments, code, writing and reviewing.
3. **Růžička, V.**, Mateo-García, G., Gómez-Chova, L., Vaughan, A., Guanter, L., Markham, A. (2023, November). Semantic segmentation of methane plumes with hyperspectral machine learning models. *Scientific Reports*, 13(1), 19999. Růžička *et al.* [2023b]
Codebase: <https://github.com/spaceml-org/STARCOP>.
In this paper, V.R. was responsible as the first and corresponding author, he was proposing the machine learning models, conducting and evaluating the experiments and publishing the code repository, results and writing the paper. He shares the conception of the experiments and coding, and collection and processing of the dataset.
4. Upcoming publication from the work presented in Chapter 5. **Růžička, V.**, Markham, A. (2024, October). HyperspectralViTs: General Hyperspectral Models for On-board Remote Sensing. Růžička & Markham [2024]
Pre-print publicly at: https://arxiv.org/abs/2410.17248.
In this paper, V.R. was responsible for conceptualisation, methodology, experiments, code implementation and model training, writing and reviewing.

1.5.2 Media releases

The following **media releases** and public talks cover the presented work in the core publications:

- *AI on satellites for autonomous detection of disaster events*. Newsletter of the Oxford University, Computer Science Department. **Inspired Research**, Winter 2022, Issue 21, page 20, October 2022. [link](#)
- *Researchers successfully train a machine learning model in outer space for the first time*. **Oxford University News**. July 2023. [link](#)
- *World breakthrough in onboard AI model training presented by Φ -lab at IGARSS*, **ESA Φ -lab News**. August 2023. [link](#)
- *AI that automatically detects methane plumes from space could be a powerful tool in combating climate change*. **Oxford University News**. November 2023. [link](#)
- *Na stopě metanu - Umělá inteligence proti klimatické změně. (On the trail of Methane - Artificial Intelligence for Climate Change tasks)*. **WIRED CZ**. November 2023. [link](#)
- *Potřebuji vidět hlubší smysl své práce. (I need to see a deeper meaning behind my work)*. **ČVUT FEL News CZ**. December 2023. [link](#)
- *How AI helps us track methane - from space* at the **TED AI in Vienna**. October 2024. Video is available at: https://www.ted.com/talks/vit_ruzicka_how_ai_helps_us_track_methane_from_space.

1.5.3 Other co-authored publications

Additional co-authored publications are:

1. Zhang, B., Wang, H., Alabri, A., Bot, K., McCall, C., Hamilton, D., **Růžička, V.** (2022, November). Unsupervised wildfire change detection based on contrastive learning. *AI + HADR Workshop*, NeurIPS 2022. Zhang *et al.* [2022] Codebase: <https://github.com/spaceml-org/FireCLR-Wildfires>.

This paper explores unsupervised change detection in Sentinel-2 images while specializing on tasks related to wildfire detection - this includes the detection of active fires, ash types and presence of fuel loads relevant for wildfires. During

this project, I served as a machine learning team lead. I was responsible for the choice of SimCLR and helped with VAE baselines explored during development. Exploring contrastive learning was an alternative technique to the VAE used in Chapter 3, although we ended up using smaller and more local datasets connected with selected wildfires.

2. Parr, J., Acciarini, G., Bridges, C., Mateo-García, G., Portalés, E., Purcell, C., **Růžicka, V.**, Spalding, A., Veitch-Michaelis, J. (2024, July). Live Twinning: A Vision of ML Enabled Assets in LEO for Rapid Response to Natural Catastrophes. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium* (pp. 685-688). IEEE. Parr *et al.* [2024]

This paper covers a selection of projects developed with the Frontier Development Lab focusing on topics of autonomy and AI on-board of satellites. These projects include works mentioned in Chapters 3 and 4 and papers Růžicka *et al.* [2022, 2023a,b]. I was responsible for providing descriptions about the projects I was involved in. This paper forms a bridge between the research presented in this thesis and several other projects conducted in the Frontier Development Lab internships (such as AI models aimed for the planned ESA VIGIL mission to the fifth Lagrangian point (L5) to provide a side view of coronal mass ejection events from the Sun).

3. Vaughan, A., Mateo-García, G., Gómez-Chova, L., **Růžicka, V.**, Guanter, L., Irakulis-Loitxate, I. (2024, May). CH4Net: a deep learning model for monitoring methane super-emitters with Sentinel-2 imagery. *Atmospheric Measurement Techniques*, 17(9), 2583-2593. Vaughan *et al.* [2024b]

This paper focuses on detection of methane leaks in the images from the multi-spectral sensor Sentinel-2. During my research on the work presented in Chapter 4 and in the paper Růžicka *et al.* [2023b], we conducted a wide research of prior methods used in this field (for hyperspectral and multispectral data) - this includes discussing band ratio methods, potential sources of data and limitations of connected sensors. It also includes working together on developing the motivation for these projects and their presentation to the public.

4. Aybar, C., Mateo-García, G., Acciarini, G., **Růžicka, V.**, Meoni, G., Longépé, N., Gómez-Chova, L. (2024, October). Onboard Cloud Detection and Atmospheric Correction With Efficient Deep Learning Models. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. IEEE. Aybar *et al.* [2022]

Codebase: <https://github.com/spaceml-org/DTACSNet>.

This paper effectively explores translation of Sentinel-2 data between different processing levels (L1C to L2A, including cloud detection) using small and fast machine learning models - as a simulation of classical, typically slower methods used on-ground. My participation was to advise on topics related to AI on-board of satellites, as the aim was to run these models on the edge to allow in space domain to domain translation. Experience with hardware proxies for real satellites is also leveraged throughout Chapters 3 and 5.

1.5.4 Datasets and public resources

Throughout our research, we also created many new datasets, which we publicly release to aid other researchers in the connected fields. These are listed in Table 1.1. Codebases, pre-trained model weights and details about data processing tools are also provided in the repositories for each project.

Table 1.1: Released datasets with links to publicly accessible data.

Dataset name	Sensor	Variant	Size	Links
RaVAEn	Sentinel-2, L1C	disasters evaluation dataset	11.73 GB	Google Drive
STARCOP	AVIRIS-NG, L1B	MF, RGB, WV3 simulated 125b. (1573-1699, 2004-2480)	60.7 GB 633 GB	Zenodo.7863343 Train 1, 2, 3, Test
	EMIT, L1B	small generalisation dataset	341 MB	Google Drive
OxHyper	EMIT, L1B			
-SyntheticCH4		RGB, MF	6 GB	HuggingFace repo.
		86 bands (RGB, 1573-2478nm)	114 GB	HuggingFace repo.
		multi-temporal version	228 GB	HuggingFace repo.
-RealCH4		86 bands (RGB, 1573-2478nm)	47 GB	HuggingFace repo.
-Minerals		285 bands (381-2492nm)	372 GB	HF Train, TestVal

2

Background Literature

2.1 Overview

In this chapter we explore the background literature for the domain of Remote Sensing in Section 2.2 and the methodology of Machine Learning in Section 2.3. We specifically focus on the application of Methane detection in Section 2.4. Further examples of Machine Learning methods applied for Remote Sensing tasks are listed in Section 2.5. Finally, in Section 2.6 we detail systems and hardware used for missions deployed in Space.

2.2 Primer on Remote Sensing

Applications explored in our work use data from the domain of Remote Sensing. Here, we will provide high-level description of this discipline and mention examples of types of tasks, types of data and practices relevant to the applications tackled in this thesis. For more information we refer the reader to the text collection of [Njoku, 2014]. We differentiate the data used in the discipline of Remote Sensing namely in contrast with the domain of Computer Vision which has experienced widespread attention by the Machine Learning community. Machine Learning models used on Remote Sensing data face several unique challenges and motivate creation of new adapted architectures. Also, while models trained on Computer Vision data are in daily use in the real-world (for example on mobile phones), Remote Sensing data has not yet experience similar wide adoption of Machine Learning. In this section, we explore some of the reasons why Remote Sensing data is unique.

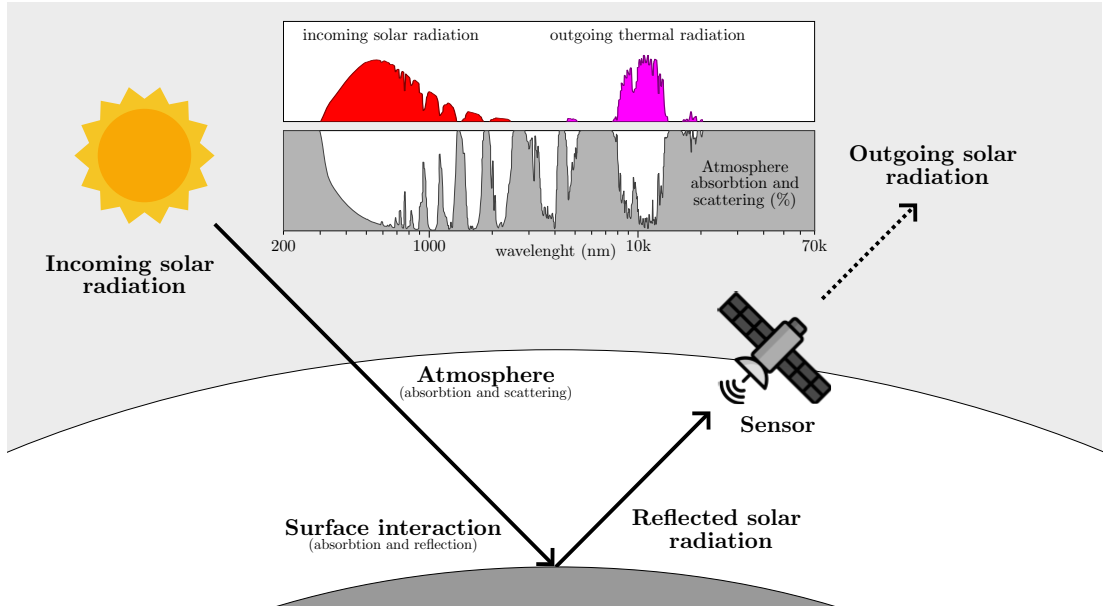


Figure 2.1: Simplified illustration of light passing through the Earth’s atmosphere, bouncing by the surface and returning to the satellite’s sensors. Different constituents of atmosphere absorb incoming light at different wavelengths - this is why we are able to detect them in the reflected at-sensor radiation.

2.2.1 Instruments

Remote Sensing data comes from a wide range of instruments deployed on various types of devices. We will present several examples and create few simplified categorisations. We however note that this is not meant as a complete and exhaustive description of the entire discipline.

We can differentiate between the types of deployment of the used sensors. Firstly, data can be collected aerially, with airplanes flown over the locations of interest. These will create mostly local data captures (although at times they can cover areas of whole countries or sample scenes across different countries) that usually aim to monitor a targeted event. Although efforts are taken to standardise data acquisitions, some parameters of data could differ between the captures - this could be due to the different atmospheric conditions, time of day or height of the flight. One such example is the AVIRIS-NG sensor, which has been flown over areas in the USA [Thorpe *et al.*, 2017], India [Bhattacharya *et al.*, 2019] and Europe [Kuhlmann *et al.*, 2023] with more planned captures. Data collected by unmanned aerial vehicles (UAVs) or drones have similar properties, but their operation can be easier and less costly [Dooley *et al.*, 2024]. Secondly, data can be captured from satellites orbiting the Earth. These often have global coverage of the entire planet and because they are using

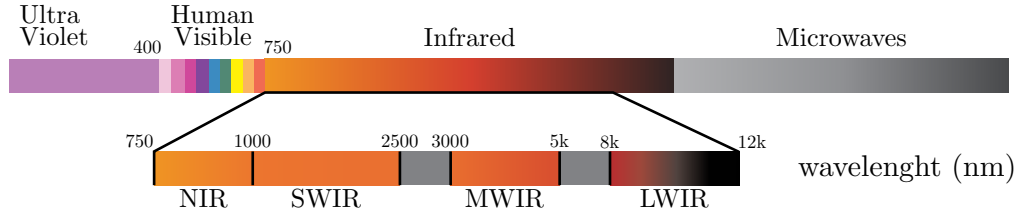


Figure 2.2: Visualisation of the electromagnetic spectrum.

the same processing pipelines, the collected data tends to be more uniform. While some missions capture data in only on-demand scenarios, other record everything and effectively create archives of all observed data. One such example is the Sentinel-2 satellite, which observes areas around the whole world with a revisit time of about 5 days (using two near identical units). The data captures also tend to be synchronised with a time of day on the ground. Thirdly, sensors can be deployed on other specific platforms, such as for example the International Space Station (ISS), which hosts for example the EMIT sensor. The properties of these are somewhat similar to a satellite, however the position of the ISS itself is less predictable and can be changed. As such, the time of the day between captures can also differ.

Figure 2.1 shows an example of the light spectrum that can be captured with different instruments and the effect our atmosphere has on different wavelengths. Instruments deployed in space typically observe the reflected solar radiation after it has passed through the atmosphere on the way from the Sun, which causes many complex interactions and requires specific data processing. The visible part of the spectrum covers the red, green and blue (RGB) bands similar to those typically used in Computer Vision. Figure 2.2 shows the division of the infrared portion of spectrum into near infrared (NIR), short-wave (SWIR), middle-wave (MWIR) and long-wave (LWIR) infrared (which includes thermal imaging ranges). The exact ranges differ depending on the definitions. The visible and short-wave infrared (VSWIR) is also sometimes combined. When describing which parts of the spectrum a satellite covers, we encounter classification into multispectral (MSI) and hyperspectral (HSI) instruments [Hagen & Kudenov, 2013; Cawse-Nicholson *et al.*, 2021]. MSI satellites typically have fewer bands, and each band covers a relatively wide area (for example Sentinel-2 has 13 bands and its “Red” band covers 641 to 689nm). Meanwhile HSI satellites collect many more bands with narrower spectral resolutions (for example the AVIRIS-NG sensor has 285 bands and the band near to “Red” covers only about 639 to 645nm).

Table 2.1: Example current and future multispectral and hyperspectral sensors. Used shortcuts: S. range/res. for spectral range and resolutions (both in *nm*), sat. for satellite, aer. for aerial capture (from a plane), pub. for public, priv. for private and acq. denotes that manual acquisition needs to be scheduled. Finally, GSD refers to ground sample distance (also known as spatial resolution).

Name	Type	Bands	S. range	S. res.	Year	GSD	Revisit
MSI							
Landsat 8 OLI	sat., pub.	9	433-2294	20-80	2013	15-30m	16d
Landsat 9 OLI	sat., pub.	9	433-2294	20-80	2021	15-30m	16d
WorldView-3	sat., priv.	16	400-2365	30-180	2014	.31-3.7m	1d
Sentinel-5P	sat., pub.	7	270-2385	.225-.65	2017	3.5×7km ²	1d
Sentinel-2A	sat., pub.	13	443-2190	15-180	2015	10-60m	10d
Sentinel-2B	sat., pub.	13	443-2190	15-180	2017	10-60m	10d
Sentinel-2C					2024		
HSI							
AVIRIS-C	aer., pub.	224	380-2510	10	1989	4-20m	N/A
AVIRIS-NG	aer., pub.	425	380-2510	5	2013	3-10m	N/A
AVIRIS-3	aer., pub.	284	390-2493	7.4	2023	0.5-13m	N/A
GHGSat-D	sat., priv.	N/A	1630-1675	~0.3	2016	50m	14d
GHGSat-CX	sat., priv.	N/A	1630-1675	~0.1	2020	30m	14d
PRISMA	sat., acq.	240	400-2505	11-13	2019	30m	29d
EnMap	sat., acq.	246	420-2450	6.5	2022	30m	27d
EMIT	ISS, pub.	285	381-2493	7.5	2022	60m	N/A
Tanager-1	sat., priv.	>420	400-2500	5	2024	30m	weekly
MethaneSAT	aer., pub.	N/A	1598-1683	0.25	~2025	100x400m	7-14d
SBG-VSWIR	sat., pub.	220	380-2500	10	~2032	30-35m	<16d
CHIME	sat., pub.	>200	400-2500	<10	~2029	30m	~12.5d

Hyperspectral sensors typically provide a finer view of the imaged phenomena, however at the cost of increased data volume and processing requirements. Multispectral sensors are more common, and the archives of their data is also more easily accessible. While the family of the multispectral Landsat satellites operates globally since 1972 (and for comparison Sentinel-2 from 2015), most of the early hyperspectral sensors were deployed locally on aerial missions with AVIRIS first used in 1986. Satellite deployed sensors by the private actors such as EnMap or PRISMA followed later, but their data captures are only on-demand. Only recently missions like the ISS deployed sensor EMIT provide near-global view of the Earth with hyperspectral instruments. We note that the spectral range of HSI isn't defined, so some sensors operate only until about 1000nm, while other capture wavelengths up to 2500nm - this is relevant for detection of events visible only in shorter wavelengths (for example methane).

We list some of the relevant satellites and sensors in Table 2.1. We note that the Sentinel missions operate in tandem halving the revisit time from 10 to 5 days (similarly as the Landsat missions halving their revisit time from 16 to 8 days). Missions such as the planned MethaneSAT operate in very narrow windows monitoring carbon dioxide and methane separately [Miller *et al.*, 2024], the captured data will likely directly create methane concentration products. The aerial mission AVIRIS-3 has been recently used to monitor methane and CO₂ plumes in the preliminary report of [Thorpe *et al.*, 2024]. We omit radar based sensors (such as synthetic aperture radar or LiDAR) and thermal imaging sensors, as these are beyond the scope of this thesis. Similar lists are available in the recent survey by [Tiemann *et al.*, 2024] focusing on instruments observing methane.

2.2.2 Levels of processing

Data product levels refer to the pipelines used to process the data from the Level 0, raw unprocessed data available on-board of the sensor, all the way to levels presenting heavily analysed and interpreted data [Njoku, 2014]. The exact definitions of these aren't standardised across different missions, satellites and agencies, although some general similarities apply. Level 0 data is on-sensor raw capture, with noise and sensor artifacts due to dark signal, erroneous pixels and other effects. It also typically lacks bands coregistration, as explored in [Llaveria *et al.*, 2023]. Level 0 data is usually not easily available, only recently some works collected and released small, annotated datasets [Meoni *et al.*, 2023; Del Prete *et al.*, 2023]. Level 1 data typically contains georeferenced Top-of-Atmosphere (TOA) reflectance data - however for example, the exact method used for orthorectification often differs between sensors. Level 2 data typically contains surface reflectances, referred to as Bottom-of-Atmosphere (BOA), and as such is atmospherically corrected. It may also contain some early scene classifications and cloud masks (as these may be needed for more exact atmospheric correction). We note that while these are usually processed on-the-ground from the downlinked satellite data, some recent works considered models to simulate this process on-board [Aybar *et al.*, 2024].

2.2.3 Data portals, datasets

In this section, we describe several data portals relevant for our work. First, we note how important it is for data to be provided with an easy and free access for its widespread adoption in the community. Data collected by a satellite can typically be

Table 2.2: Example remote sensing datasets. We note that we informally approximate the dataset size as Small (with only few satellite captures), Medium or Large (with many thousands of captures). S1 and S2 refers to the Sentinel satellites.

Sensor	Name	Coverage	Size	Labels	Models
S2	WorldFloods	global	L	Water extent	U-Net
	BigEarthNet	global	L	Land cover	CNN
	Dynamic World	global	L	Land cover	FCNN
	MajorTOM	global	L	N/A	N/A
S1+S2	CloudSEN12	global	L	Cloud masks	U-Net
AVIRIS	Single scenes (Indian Pines, ...)	local	S	Materials	N/A
	JPL methane leaks	local	M	Methane leaks	N/A
	MethaneMapper	local	M	Methane leaks	DETR
	STARCOP (Ch. 4)	local	M	Methane leaks	U-Net
EMIT	Presented in Ch. 5:				
	OxHyperSyntheticCH4	global	M	Methane leaks	SF, EV
	OxHyperRealCH4	global	M	Methane leaks	same as ^
	OxHyperMinerals	global	M	Minerals	same as ^

used for a wide variety of tasks, which might not be known at the date of capture. New methodology, or machine learning models developed in the future can be used retrospectively and this archive can be studied as was for example the case when detecting methane in older captures of data [Irakulis-Loitxate *et al.*, 2022].

To name a few examples, data from the Copernicus Sentinel satellites is relatively easily accessible on several portals, for online viewing on the Copernicus Browser¹ and for data download on the Google Earth Engine portal² (in multiple levels of processing including TOA and BOA). Similarly, data from the NASA’s hyperspectral sensor EMIT is continually being uploaded on the Land Processes Distributed Active Archive Center³ (LP DAAC) on daily basis. In contrast, data from other hyperspectral satellites such as EnMap or PRISMA requires the users to manually schedule acquisitions over areas of interest. Although it is possible to access the archive of past acquisitions, there are typically limits to the number of scenes that can be downloaded and locations that haven’t been scheduled manually will simply be missing from the archives. Licences for sharing data processed from these satellites are also more complex - namely one cannot share the original data, which potentially prohibits creation of large benchmark datasets hosted on third party websites.

¹<https://browser.dataspace.copernicus.eu/>

²<https://code.earthengine.google.com/>

³<https://search.earthdata.nasa.gov/search>

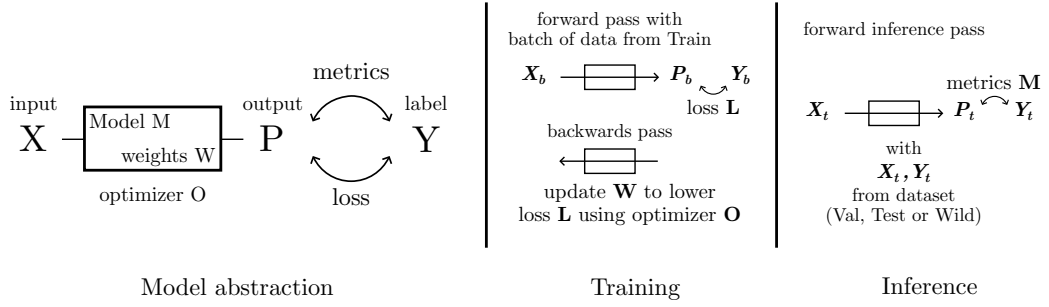


Figure 2.3: Simplified abstraction of machine learning model training and inference schema.

Table 2.2 shows a non-exhaustive list of publicly accessible large remote sensing datasets (including the recent large dataset of Sentinel-2 images called MajorTOM [Francis & Czerkawski, 2024]) and example used machine learning models, SF stands for SegFormer [Xie *et al.*, 2021], EV for EfficientViT [Cai *et al.*, 2023] and DETR [Carion *et al.*, 2020].

2.3 Primer on Machine Learning

In this thesis we heavily rely on Machine Learning models which we apply to several concrete applications (such as tasks in the domain of Remote Sensing) given specifically defined constraints (for example low compute environments). It is out of the scope of the thesis to provide exhaustive research into all existing Machine Learning techniques and models. Instead, we chose to provide a very general and high-level description of trends used in Machine Learning. For an introduction into this field, we refer the reader to the classical textbooks such as [Goodfellow *et al.*, 2016] and we also suggest following the references for more details about each of the mentioned techniques or architectures.

An abstraction of a machine learning model is shown on Figure 2.3, where the model itself is described generally as M with some trainable parameters W (weights). The typical modern model is composed of a sequence of layers, which apply specific learned data transformations using the weights of each of the layers. When constructing this model, we typically plug existing building blocks proposed in prior research, following the type and dimensionality of the data and the type of the task and labels.

Table 2.3: Types of example machine learning tasks with a typical remote sensing data sample on the input. In some cases the exact used output shape depends on the formulation of the problem and could differ between papers. For example, bounding boxes can be defined in multiple ways.

Name	Input	Output
Classification	$[b, w, h] \in \mathbb{R}^{B,W,H}$	$c \in \text{Classes}$
Regression	$\wedge$	$r \in \mathbb{R}$
Domain to domain transfer	$\wedge$	$[w,h] \in \mathbb{R}$
Semantic segmentation	$\wedge$	$[w,h] \in \text{Classes}$
Object detection	$\wedge$	b. boxes with classes
Instance segmentation	$\wedge$	b. boxes with instances
Panoptic segmentation	$\wedge$	b. boxes with both
Feature learning	$\wedge$	depends on the definition
Change detection	$[t, b, w, h] \in \mathbb{R}^{T,B,W,H}$	$[w,h]$ in Classes
Anomaly detection	$N * [b, w, h] \in \mathbb{R}^{B,W,H}$	depends on the definition
Sequence classification	$[t, b, w, h] \in \mathbb{R}^{T,B,W,H}$	$c \in \text{Classes}$
Sequence regression	$\wedge$	$r \in \mathbb{R}$
Sequence modelling	$\wedge$	$[b, w, h] \in \mathbb{R}^{B,W,H}$ for $t + 1$

2.3.1 Task descriptions

Table 2.3 provides a non-exhaustive list of few selected tasks and how these inform the shapes of the input data and the used labels. Most of applications explored in this thesis use Remote Sensing image-like data with many bands, from this we frame the typical input composed of images potentially obtained at multiple time snapshots, with multiple spectral bands and with spatial coordinates given by the width and height of the image (typically tiled into into smaller square tiles). In several examples explored in this thesis, we define the task at hand as semantic segmentation, this means that the spatial dimensionality of the data is usually maintained even in the label (although potentially at lower resolutions).

In addition, we typically differentiate between three types of learning, depending on the type of labels we have at our disposal. In the so-called fully supervised learning scenario, each sample has a label. A simple illustration can be the early work of [Krizhevsky *et al.*, 2012], where a model was trained on image classification task and where all labels were known. In the so-called weakly or semi-supervised learning scenario, we have labels only for a subset of the data. In extreme cases known as few-shot or one-shot learning, only few or just one sample contains a label. In these cases, we typically leverage some existing pre-trained models (ideally trained on a similar task) and for example fine-tune them on our much smaller labelled datasets.

Clustering methods are typically used in few-shot learning or one-shot learning cases. In the unsupervised learning scenario, we don't have any labels at all, in these cases we either try to generate some pseudo-ground truth labels (and convert the task into supervised learning), or we learn some auxiliary tasks. With generative machine learning models, these could be for example data reconstruction from a bottleneck feature [Kingma & Welling, 2014] or other tasks that require the model to somehow understand the data such as masked image reconstruction [He *et al.*, 2022]. With contrastive learning methods, we generate lists of positive and negative samples and instead learn distance functions (for example using the triplet loss, where positives are encoded close to each other while negative samples are pushed away from each other). Models obtained in unsupervised learning can also be used for semi-supervised tasks.

Finally, in the so-called zero-shot learning, we use models pre-trained on other data for predictions on our own data and we hope for generalisation. Definitions are not entirely consistent, however this term could also be used to describe a scenario, where we are using models originally trained on a clean and well-behaved benchmark datasets with data from in-the-wild, real-word deployment of similar sensors. As an extreme example of this, we can also consider the so-called Foundation models, which are typically used in a zero-shot manner, and which expect generalisation ability on newly shown data [Yang *et al.*, 2023]. Some definitions of these also add requirements that the used pre-training datasets have to be very large and that the size of the model itself needs to be beyond some number of parameters. We however note, these definitions are quite arbitrary, and that new papers sometimes tend to define their own new classifications. We refer the reader to overview papers of [Shrestha & Mahmood, 2019; Tuia *et al.*, 2024] for more categorisations.

2.3.2 Training

The illustration on Figure 2.3 also includes an abstraction of the training process, for more details we refer to [Goodfellow *et al.*, 2016]. The model learns a function that translates data on the input into predictions at the output side of the model. We typically calculate loss as a distance between the predictions (in some cases after applying a threshold) and the ground truth labels. Without labels, pseudo-ground truth or distances between positive and negative samples can be used instead. The loss function depends on the type of predicted task. As an example, a typical regression loss is the L1 loss also known as mean absolute error (MAE) and the L2 loss also known as mean square loss (MSE). One of the commonly used classification losses is

for example the binary cross-entropy (BCE) loss. A survey of losses can be found in [Ciampiconi *et al.*, 2023].

In the training process we select the so-called optimizer, which is an algorithm that defines how we change the model weights given the observed loss. The goal of the optimizer is to update weights in a way that will decrease the overall loss. In doing so, we are moving through the high-dimensional space of all possible weights to find areas with low loss, without getting stuck in areas of local minimum. Commonly used optimizers include the stochastic gradient descent (SGD), in this thesis we use Adam [Kingma & Ba, 2014] or AdamW [Loshchilov & Hutter, 2017].

Metrics are also calculated like losses from the distance between the predictions and labels, they should however be informed by domain knowledge and give us an indicator of how well is the model doing. During binary classification, we typically quantify samples into types of errors such as True or False Positives or Negatives. From these we can compute metrics such as $Precision = TP / (TP + FP)$, $Recall = TP / (TP + FN)$, the F1 score $F1 = (2 * Precision * Recall) / (Precision + Recall)$ and the Intersection over Union score $IoU = TP / (TP + FP + FN)$. We use multiple metrics in cases when dealing with datasets with imbalanced distribution of classes (for example when we have only few samples of interest and many samples from the background class). Model predictions are typically continuous, and we have to convert them to binary classifications by selecting a threshold. Other metrics don't need threshold and give us a measure of alignment between continuous predictions and binary labels - one example is the area under the precision recall curve (AUPRC) metric.

Finally, dataset splits also influence how we train our models. We illustrate this on an example of fully labelled dataset $D = (X, Y)$ where X are inputs and Y are associated labels. In most benchmarks, these splits are provided to us, but when we are creating new datasets, we need to split them ourselves. Samples are typically split into *Train* and *Test* subsets, each following the independent and identically distributed (IID) assumption. Sometimes k-fold cross-validation is used instead, where we create k different splits of different subsets of the entire dataset D . In other cases, we create an additional *Val* subset. This enables us to tune hyperparameters on this set aside set, or to select the best performing model. To prevent model overfitting and false result numbers, each subset needs to be non-overlapping with the other subsets. We can also consider creating a dataset of samples from the wild, usually from a real-world application. We should ideally consider the creation of these already when training our models and when testing their generalisation abilities.

2.3.3 Model architectures

Machine learning architectures can also be described as a list of layers, or of individual building blocks used to process the input data into predictions. These create directed acyclic graphs (DAGs) that uniquely describe the passage through a network. Most of layers will be reused across different machine learning architectures at times regardless of the application. Only in some cases the domain of the data calls for a new machine learning architecture and design.

Here, we briefly describe some of the commonly used layers in machine learning models, for the exact specification we however refer the reader to the overview of [Shrestha & Mahmood, 2019]. The simplest layer is the so-called dense or fully connected layer, which connects each input with each output using multiplication by learned weights and addition of learned bias using a projection through (often non-linear) activation function. Another typical layer type is the so-called convolutional layer, which applies several learned convolution kernels over the whole layer’s input data. This layer maintains the spatial resolution and typically uses fewer parameters than fully connected layer. Few non-parametric layers include down-sampling layers, which reduce the resolution using a pooling function (for example an average of a given neighbourhood window). Another is simply an identity operation, which can be used across different levels of the model - for example the residual connections of [He *et al.*, 2016] or the skip connections of [Ronneberger *et al.*, 2015]. The famous AlexNet [Krizhevsky *et al.*, 2012] model which is a predecessor of many modern convolutional neural networks, can be described using the listed layers. The addition of typically learned upscaling or unpooling layers then allows a construction of the so called encoder-decoder models that reconstruct the shape of the data from high level features (for example [Ronneberger *et al.*, 2015]).

So far, we omitted layers used for modelling sequential data, which form the basis of the recurrent neural networks (RNN) or the later proposed long short-term memory (LSTM) [Hochreiter, 1997] layers. One of the latest additions into the family of machine learning building blocks is the Transformer block, initially used to model sequential data in large language learning models [Vaswani *et al.*, 2017] and later adapted for image data in [Dosovitskiy, 2020]. These add attention mechanisms that allow the layer to model connections in data globally across the input resolution. Models using these are typically better at modelling long-range interactions inside images than those relying only on convolutions [Tuli *et al.*, 2021], however this is at the cost of larger computational demands (some specialised architectures however try to address this [Cai *et al.*, 2023]) and the need for larger datasets.

We will create a high-level generalisation of the types of machine learning models given the flow of the data shapes passing through the layers of the model. This is naturally governed by the shape of the data and labels. When we predict a single value for an entire image, the models will be progressively decreasing the resolution of the data ending up in networks similar to an encoder layout. Models like AlexNet [Krizhevsky *et al.*, 2012], ResNet [He *et al.*, 2016], VGG [Simonyan & Zisserman, 2014], ViT [Dosovitskiy, 2020] and ConvNeXt [Liu *et al.*, 2022] can all be described like this. When we instead predict data of the same resolution, we usually use the architecture which can be described by the encoder-decoder layout. In cases of auto-encoders and similar models [Kingma & Welling, 2014; He *et al.*, 2022], networks are kept separate. In case of architectures such as U-Net [Ronneberger *et al.*, 2015], SegFormer [Xie *et al.*, 2021], EfficientViT [Cai *et al.*, 2023] and others, we use identity connections between different layers processing the data at the same resolutions, which effectively allows for separation of the scale of the learned features. Naturally, there are more complex architectures, but even these typically contain blocks which could be described with this simple differentiation. We will use this description throughout the chapters in this thesis.

2.4 Primer on Methane Detection

2.4.1 Motivation

The background of our motivation for why we'd like to detect methane stems from messages presented in the Intergovernmental Panel on Climate Change (IPCC) report from 2023 [Lee *et al.*, 2023]. The single most important parameter for the future development of Climate on planet Earth currently is the amount of greenhouse gasses (GHGs) contained in the atmosphere. This is because of their influence on the phenomena called radiative forcing. Effectively there's an energy balance between the energy coming from the Sun and the energy released (radiated) back into Space [Ramanamswamy *et al.*, 1991]. Radiative forcing can be positive or negative depending on if the amount of energy that enter Earth's atmosphere differs from the energy leaving it causing increase in temperature (positive radiative forcing) or decrease (negative radiative forcing). Typically, this is due to a mixture of different physical parameters, current effects lead to temperature increases.

The individual constituents are quite complex, even the influence of GHGs is at least two-fold. GHG molecules absorb and reflect heat that would otherwise escape into space. Larger quantities of GHGs in the atmosphere also changes the altitude at

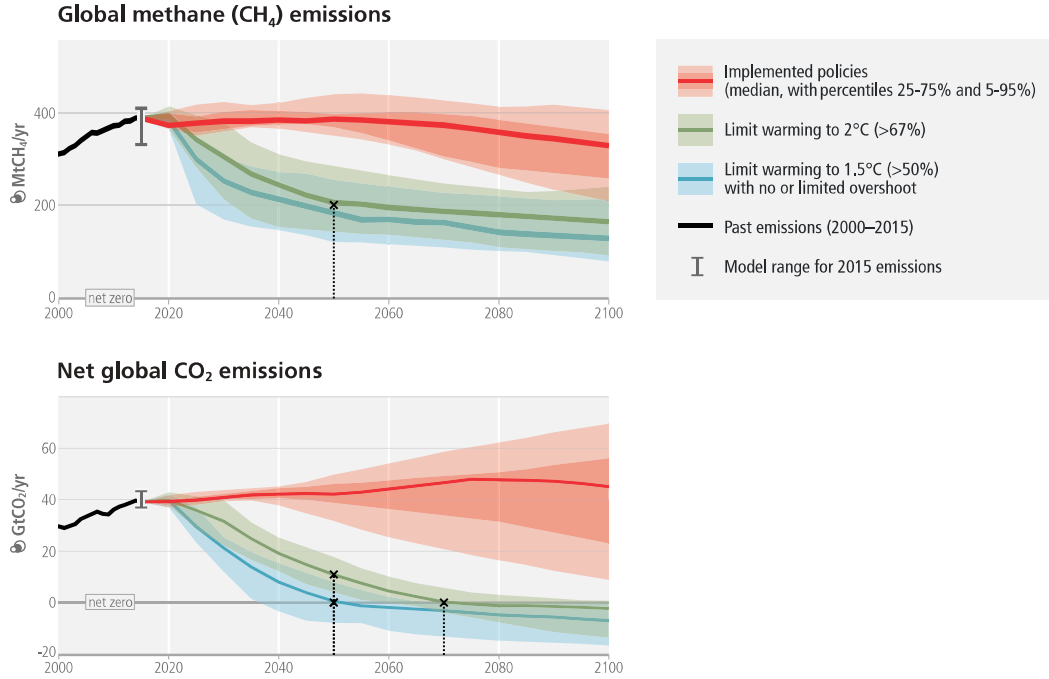


Figure 2.4: Illustration of the estimated temperature by the end of the century depending on different reductions of greenhouse gasses, split to show carbon dioxide and methane separately. This graph has been reproduced from the IPCC 2023 report [Lee *et al.*, 2023]. We additionally mark mentioned reduction targets for both methane and net zero in carbon dioxide.

which radiation escapes into space - at higher altitudes the emission occurs at lower temperature, which means that the atmosphere retains more energy (this is called a change in the effective radiative altitude). Other constituents can have both positive and negative forcing effects, such as for example aerosols. Calculation of radiative forcing is therefore complex and is beyond the scope of this thesis.

Before the industrial era, the incoming and outgoing radiation was close to balance, which is why the year 1750 is often used as the baseline for comparison, however the years between 1850-1900 are also used. Various modelled scenarios will lead to different outcomes given our behaviour in the next few decades. As is shown on Figure 2.4, there is a temporal delay, and almost regardless of our current actions we will likely not avoid a future which has $+1.5^{\circ}\text{C}$ in comparison with the pre-industrial era. However, our actions now will heavily influence the expected outcome by the year 2100 - we can either limit global warming to $+1.5^{\circ}\text{C}$ or go beyond $+4^{\circ}\text{C}$. If we continue with current rate of GHG emissions, we are likely to reach $+2.7^{\circ}\text{C}$ by the end of the century (this is known as scenario “*SSP2-4.5*” in [Lee *et al.*, 2023], where

SPP x - y stands for “Shared Socioeconomic Pathway”, x denotes the pathway number and y for the level of obtained radiative forcing by 2100). If we reach net zero state, where we do not emit more GHGs than what is removed, only by the year 2070, we might face future of just under $+2^{\circ}\text{C}$ by 2100 (scenario “SSP1-2.6”). Finally, if we are able to reach net zero by 2050, it is estimated that we will have just under $+1.5^{\circ}\text{C}$ by 2100 (scenario “SSP 1-1.9”). These numbers refer to all GHGs, on Figure 2.4 we see separation into CO_2 and methane.

In terms of the impact these temperature changes would have, every 0.1°C matters. There are vulnerable environments both in the animal kingdom in land-based and oceanic ecosystems (for example warm water corals) and for us as human society. These impacts are expected to be the worst for vulnerable populations. Impacts on the irreversible losses of biodiversity, costs stemming from food insecurity and losses in agricultural yields and costs connected to estimated medical treatments due to pollution are vast and all-encompassing.

While CO_2 is the largest contributor to global warming, methane closely follows as the second one. As is detailed in the United Nation’s Methane Assessment Report [Kuylenstierna *et al.*, 2021], methane emissions come from a mixture of sources, some natural and some anthropogenic. A significant source of pollution comes from leaks in the oil and gas industry and the used infrastructure (pipelines and storage tanks). These are however relatively easy to fix, once the information about the breakage is known. For this, an early warning system is needed. Satellites provide a unique source of potentially global information - and in fact detection made by Remote Sensing instruments already lead to fast fixes of the detected leaks as early as in 2016 during flights of the AVIRIS-NG sensor as reported in [Duren *et al.*, 2019]. Published research of [Irakulis-Loitxate *et al.*, 2022] which used archival data and detected some of the largest methane leaks in the history has led to new policies being made and countries joining the Global Methane Pledge. Signed countries aim is to reduce their methane emissions by 30% by the year 2030 in comparison with the 2020 levels. For context, in Figure 2.4 it is shown that global methane emissions need to be reduced by 34% of 2019 emission levels by the year 2030 to limit global warming to $+1.5^{\circ}\text{C}$. This reduction will need to continue to about 44% by 2040. By year 2050 this would mean a reduction from 400 Mt released per year to about 200 Mt per year. Net zero in CO_2 will likely be reached before other GHGs, due to the long effect of some hard to abate residual GHG emissions.

Reduction of methane emissions needs to be one of the priorities, however other efforts to reduce other GHGs in the atmosphere also need to be taken. Methods

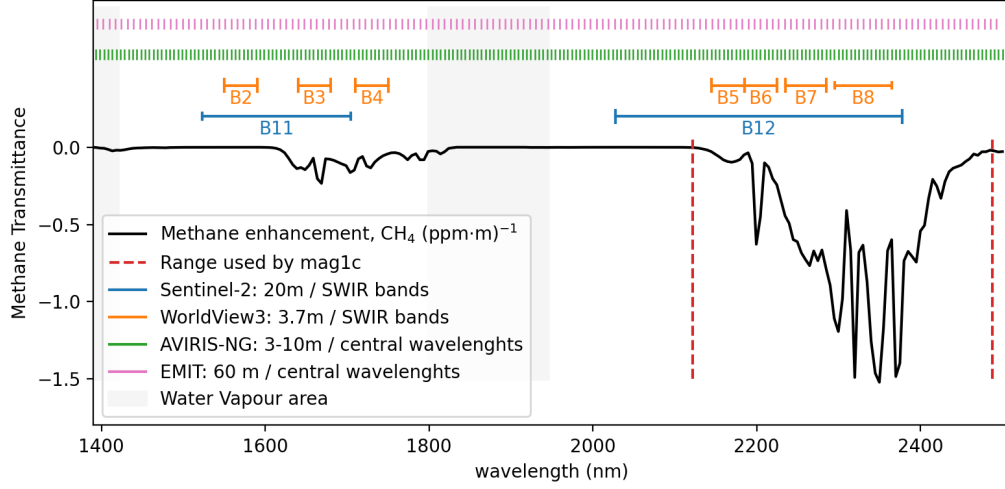


Figure 2.5: Illustration of the presence of methane signal (shown through the methane transmittance) in comparison with the bands available in commonly used satellites. For clarity the hyperspectral sensors of AVIRIS-NG and EMIT, show the central wavelengths, while for the multispectral instruments of Sentinel-2 and the WorldView-3, we show the whole band ranges in the short-wave infrared (SWIR) region.

developed to detect one gas could be used for detection of others [Thorpe *et al.*, 2023]. New efforts are being undertaken also with launches of new satellite missions (in the private sector and also space agencies around the world) and with policy work taken by governments and non-governmental actors. Imaging spectrometers used to detect methane were for example highlighted as one of the important observables for future NASA’s missions in the Decadal Survey for 2017-2027 [NASEM, 2019]. This complex challenge needs to be tackled by experts from multiple disciplines and people from around the world.

2.4.2 Methane visibility and sensor considerations

Methane is a greenhouse gas (GHG), which is transparent to human eye, it is however visible at several other wavelengths outside of the optical range using specialised instruments. How well can these instruments detect methane depends on their spectral range and resolution in the relevant areas. Figure 2.5 illustrates this by showing the methane gas signature (estimated in [Rothman *et al.*, 2013] and for visualisation sampled at points typical for the selected sensor) in comparison with spectral ranges observed by selected satellites. Multispectral instruments tend to have fewer bands overlapping with areas where methane is visible, and coarser spectral resolution, meaning that often other confounding materials will be visible as well. In

contrast, hyperspectral instruments usually provide a much more refined view of methane simply by having bands with narrow widths in areas with largest visibility of methane. This is namely the peaks between 1600-1800nm and 2100-2500nm. Details about other satellites relevant for detection of methane are in the Section 2.2.1 and in Table 2.1. While some hyperspectral sensors specialised for methane detection follow the strategy of sampling images at very specific wavelengths (for example the GHGSat or the more recent MethaneSAT with spectral range zoomed into approximately 1600-1680nm), often with very narrow bandwidths, other general-use satellites sample at full VSWIR ranges and remain usable for other types of tasks (for example the AVIRIS, EnMAP, PRISMA and the newer EMIT and Tanager-1 sensors).

In cases of poor intersection between the satellites spectral range and methane signal, multi-temporal captures can be used to boost visibility of methane (assuming that there are only few changes on the ground of the scene) - sensors with high temporal cadence can somewhat counter low visibility due to spectral range (this is the case of for example Sentinel-2 or Landsat-8 satellites) [Varon *et al.*, 2021].

We also differentiate between the types of methane mapping instruments depending on the ground resolution of the sensor. While for example the Sentinel-5P sensor with ground resolution of at best 3.5-7 km² can be used for global mapping and modelling of global trends, sensors like AVIRIS-NG with 3-10m or EMIT with 60m provide enough details for local mapping sometimes referred to as point source estimate. With these sensors, each individual methane plume can be detected and outlined. Finally, some sensors aim in between these two extremes, to be used for area monitoring such as for example the upcoming MethaneSAT satellite with ground resolution of 100x400m.

Multi-sensor systems could be used to detect areas with suspected elevated methane concentrations using global mapping instruments, zoom in with captures from area mapping instruments and finally detect individual plumes with high resolution data of local mapping instruments. This setup is sometimes used manually when searching for real-world methane leaks as illustrated by [Pandey *et al.*, 2023; Schuit *et al.*, 2023].

2.4.3 Methane detection pipeline

We can broadly differentiate between several tasks of interest in the methane detection pipeline as illustrated on Figure 2.6. We will mostly focus on tasks connected to high resolution instruments, where individual methane plumes can be detected and outlined. Given an input datacube (of multispectral or hyperspectral data), we can first consider the task of classification into categories of “plume” and “no plume” for

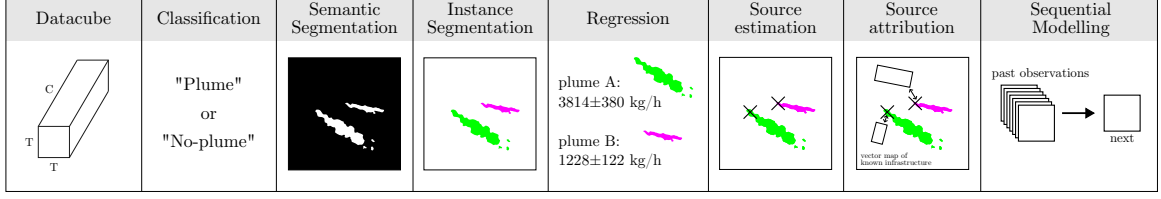


Figure 2.6: Outline of hypothetical methane detection pipeline showing the individual steps and their formalisation as machine learning tasks. Shown plume shapes and their quantification are just examples.

the whole considered tile. A more refined approach could also distinguish between plume sizes (for example by dividing emissions into subclasses such as “weak” and “strong” plumes - similarly as one could differentiate between different types of cloud cover [Aybar *et al.*, 2022]). Both of these could be approached for example with CNN type models and would be similar to image classification tasks (for example [Krizhevsky *et al.*, 2012] and others).

The most common task is however to detect pixels belonging to the methane plume leak, which can be formulated as semantic segmentation (for example in Chapter 4 and [Růžicka *et al.*, 2023b]). Potential output of the semantic segmentation model can be converted into a per-tile classification using thresholds on the number of active pixels. Following trends from Computer Vision, models for instance or panoptic segmentation [Kirillov *et al.*, 2019] could be used to differentiate multiple plumes detected on a single input tile. We note that while these are not too common in real datasets, some samples with multi-plume complexes appear. For each instantiated plume, source point estimation could be used to detect its origin and automated source attribution to a specific a priori known oil and gas infrastructure could be estimated [Guisiano *et al.*, 2023].

Finally, for each individually detected plume, an emission quantification could be used to estimate the size of the methane leak in kg/h. We note that quantification could also be ran across the entire tile. Model presented for example in [Joyce *et al.*, 2022] uses a series of separately trained machine learning models on a subset of these tasks. They first consider the task of semantic segmentation, then predict per-pixel concentrations and finally a per-tile classification and regression. We note that methods for multi-task learning employed in Computer Vision research [Kokkinos, 2017] could be used to boost performance of each individual task.

Lastly, we can consider a separate task of sequential modelling when dealing with data of high temporal cadence, but usually low ground resolution to provide forecasting capabilities previously explored in climate modelling research [Eyring *et al.*,

2024; Rasp *et al.*, 2018] with models such as [Lam *et al.*, 2022] or the recent [Schmude *et al.*, 2024].

2.4.4 Classical methods

As classical methods we denote those that do not leverage machine learning in order to detect methane plume. We note that many machine learning approaches leverage the outputs of these classical methods and use them as classically computed input features and only few use all bands of the input data in a deep learning manner (a detail which is explored in Chapter 5).

Methods used with multispectral data typically leverage band ratios between bands that are in and out of the expected methane signal regions. These methods could be used with a variety of sensors, we will use the example of Sentinel-2 bands. As shown on Figure 2.5 the band B12 is in an area with a strong overlap with methane signal, while B11 has lower overlap. We could therefore use the difference between these two bands (computed as a ratio, optionally with a matching parameter to map the typical values of the two bands onto each other) as a potential measure of methane signal. This exact setup has been named the multi-band single-pass retrieval in [Varon *et al.*, 2021]. It further introduces the multi-pass retrievals, which use bands from an earlier data capture by a satellite. Another type of band ratios has been proposed by the work of [Sánchez-García *et al.*, 2022], where multiple bands outside of the methane signal range are used to simulate a single band inside the overlap with multiple linear regression (MLR). Finally, the simulated and truly observed data is then compared using a ratio. We provide more details about these multispectral classical methods in Section 4.5.1, where we implemented both ratio products and explore their suitability as baselines - in most cases they are quite noisy and require manual intervention or further processing by a machine learning model.

Methods using hyperspectral data include two main families of methods. First family of methods is based on the matched filter approach, which processes data in a pixel-by-pixel manner and compares each pixel with an estimation of the scene background using a covariance metric. The isolated anomaly signal is computed for all spectral bands and then compared against the known or estimated signature of methane. The basic matched filter product tends to have false positive detections due to confounder materials with similar spectral signatures as methane. Iterative variants of the matched filter are proposed as the *mag1c* method in [Foote *et al.*, 2020]. More practical details about this method are in Section 4.5.2, where we implemented it as one of the baselines for comparisons with machine learning models.

Additionally, wider ranges of used hyperspectral bands are used in the matched filter variant proposed by [Roger *et al.*, 2023].

Second family of methods could be considered as physics based, as they leverage radiative transfer models to simulate changes in radiation due to several absorbers in the atmosphere - the simulated at sensor radiances are then compared with the observed data. The so-called differential optical absorption spectroscopy (DOAS) algorithm was developed by [Platt, 1994; Platt & Stutz, 2008], and later improved as the weighting function modified variant (WFM-DOAS) by [Buchwitz *et al.*, 2000] and as iterative maximum a posteriori variant (IMAP-DOAS) by [Frankenberg *et al.*, 2005]. These have then been used with the data from AVIRIS sensor in [Thorpe *et al.*, 2014] and AVIRIS-NG in [Thorpe *et al.*, 2017].

For plume quantification, the most commonly used approach is the integrated mass enhancement (IME) method which is in detail described by [Varon *et al.*, 2018] and used by prior works such as [Frankenberg *et al.*, 2016]. More recent works such as [Chulakadabba *et al.*, 2023] propose further adaptations to the method. This IME method uses wind information alongside with plume length estimated from manually created detection. As the wind data is often not available when using satellite imagery, the the U₁₀ NASA GEOS-FP reanalysis product of [Molod *et al.*, 2012] is often used, as was also done in [Varon *et al.*, 2021]. Alternatively, when only coarse information is needed, the ERA5 product [Hersbach *et al.*, 2020] can be used as in the case of [Vanselow *et al.*, 2024], which used Sentinel-5P (TROPOMI) data. Many authors consider these wind products to hold a large degree of uncertainty [Tiemann *et al.*, 2024], which propagates into the plume quantification predictions.

2.4.5 Deep learning approaches

Table 2.4 lists several machine learning models used with multispectral satellite data. Given the lack of large annotated datasets, most works rely on simulated data using real plume shapes from other datasets, simulation tools such as the Large Eddy Simulation (LES) or simpler, but faster simulation methods such as the recent work of [Rouet-Leduc & Hulbert, 2024]. Few works also leverage multi-temporal data such as [Vaughan *et al.*, 2024a; Rouet-Leduc & Hulbert, 2024].

Table 2.5 lists selected machine learning models used to process hyperspectral data. We note that most approaches leverage some kind of methane enhancement product, either in the shape of a matched filter, or another output of a classical method. Given the lack of annotated data due to large amounts of manual labour required and few publicly available data sources (as indicated by Table 2.1), only

Table 2.4: Selected machine learning models used to detect methane in multispectral data. “P.” refers to whether the data has been made publicly available (Y=Yes / N=No / R=Request). “C.” refers to coverage, we differentiate between broad categories of local (L) and global (G).

Paper, year	Sensor	ML model used	Data simulated	P.	C.
[Kumar, 2022]	S2	Transformer	yes, with LES	N	L
[Zortea, 2023]	S2	ResNet	yes, with HYSPLIT	N	L
[Radman, 2023]	S2	CNN regression	yes, with LES	R	L
Ch.4	WV-3	U-Net	yes, from real HSI data	Y	L
[Vaughan, 2024b]	S2	U-Net	no	Y	L
[Vaughan, 2024a]	S2	U-Net	real & simulated	Y	G
[Rouet-L., 2024]	S2	U-Net	yes, Gaussian plumes	N	G

Table 2.5: Selected machine learning models used to detect methane in hyperspectral data. We note that most models use either a matched filter or similar methane enhancement products as the input - this is marked in the table in column “MF” (Y=Yes / N=No / S=Similar). “P.” refers to whether the data has been made publicly available (Y=Yes / N=No). “C.” refers to coverage, we differentiate between broad categories of local (L) and global (G). SF refers to SegFormer and EV EfficientViT.

Paper, year	Sensor	Model	Data simulated	P.	MF	C.
[G.A., 2022]	PRISMA	U-Net	yes, from S2	N	Y	L
[J.S., 2022]	AVIRIS-NG	CNN regress.	yes, with LES	N	Y	L
[Joyce, 2022]	PRISMA	U-Net, ResNet	yes, with LES	N	N	L
[Kumar, 2023]	AVIRIS-NG	~DETR	no	Y	Y	L
Ch.4	AVIRIS-NG, EMIT	U-Net	no	Y	Y	L
[Schuit, 2023]	TROPOMI	CNN classifier	no	Y	S	G
[Jahan, 2024]	AVIRIS-NG	VGG regress.	yes, with LES	N	Y	L
[Bruno, 2024]	GHGSat-C1	U-Net	yes, with LES	Y	S	L
Ch.5	EMIT	SF., EV.	real & simulated	Y	N	G

some works use data globally sampled from around the world. Only more recent works tend to release their datasets publicly.

Finally, we note that some machine learning models were used specifically for methane plume quantification, these include [Jongaramrungruang *et al.*, 2022; Joyce *et al.*, 2022; Radman *et al.*, 2023; Bruno *et al.*, 2024; Jahan *et al.*, 2024]. These typically frame the problem as a regression task and approach it with a CNN-like model. As reliable plume quantification data is limited (and controlled releases are rare and hard to organise at scale [Thorpe *et al.*, 2016; Sherwin *et al.*, 2023]), these works usually rely on simulated data products (where the plume quantification is also the input provided to the simulator). As is highlighted by the overview of [Tiemann *et al.*, 2024], most of the used models don't directly encode wind information (particularly due to the large uncertainties present in this information). We note that this would further add a dependency on a product which might be difficult to obtain or estimate while on-board of a satellite.

More details about the current state of the art in machine learning models used for methane detection is also provided in Section 5.2. We also refer the reader to the overview work of [Tiemann *et al.*, 2024] which explores machine learning models used for the task of methane detection.

2.5 Machine Learning for Remote Sensing

In this section we cover some of the high level topics concerning using Machine Learning models with Remote Sensing data. For more details we refer the readers to several overviews of using Deep Learning techniques for applications in Remote Sensing in [Ball *et al.*, 2017], specifically classification tasks in [Maxwell *et al.*, 2018] and an overview of Machine Learning applied to tasks related to Climate Change in [Rolnick *et al.*, 2022]. The recent overview of [Tuia *et al.*, 2024] further provides information about some active areas of research at this intersection.

Remote sensing data differs from the data used in typical computer vision research, most notably by having a non-standard number of bands. Furthermore, each sensor has its own specific ground resolution (how many meters on the ground are captured by one pixel) and also different spectral response for each band. Therefore, while most computer vision models generalise across different sensors, in remote sensing we often encounter different models tailored for each satellite.

2.5.1 Example models used with Remote Sensing data

As in other disciplines, classical methods are being replaced by shallow or deep machine learning based approaches, in some cases post-processing the different classical features, in other by building end-to-end deep learning architectures which derive their own feature representations. Given the heavy spatial nature of the data (neighbouring pixels typically matter a lot in images from a satellite), we will focus on the task of semantic segmentation, which uses spatial data on both input and output sides of the model. This means that each pixel is assigned a single label (discrete, or continuous value), which is informed by the nearby pixels. Furthermore, data of nearby pixels gets processed together, usually tiled into spatial and spectral cubes of data.

One of the commonly applied machine learning model for remote sensing data is the U-Net architecture [Ronneberger *et al.*, 2015], which has encoder and decoder networks connected by skip-connections alongside different encoding levels. Each level considers data at progressively decreasing resolution, these eventually get reduced to create a bottleneck feature, and then reconstructed into the original spatial resolution. The architecture for the encoder network is usually reused from various classification models, retaining only the convolutional, input resolution independent section of the model.

More recently, the Transformer architecture [Vaswani *et al.*, 2017] has been widely adapted as an innovation over the previous convolutional neural network (CNN) based architectures. The original Transformer architecture was later adapted to the task of semantic segmentation by several models [Zheng *et al.*, 2021; Wang *et al.*, 2021; Xie *et al.*, 2021; Cai *et al.*, 2023]. For example, the SegFormer architecture [Xie *et al.*, 2021] uses Transformer blocks inside a similar layout of encoder-decoder networks with similar connections across different resolutions. Interestingly, the input resolution of the SegFormer model doesn't match its output resolution. Instead, in one of the first layers of the model, the resolution is downscaled by the factor of 4. This is done consciously to prevent the model from having large memory requirements, but the impacts on the results are not fully addressed. While this could negatively impact the detection capabilities of tiny objects (for example smaller than the 4x4 pixel region), this is perhaps not fully reflected in the used benchmark datasets.

Current state-of-the-art models for semantic segmentation include the relatively lightweight SegFormer [Xie *et al.*, 2021], fast EfficientViT [Cai *et al.*, 2023], multi-task OneFormer [Jain *et al.*, 2023], SegNext [Guo *et al.*, 2022] and others - more details including results on Computer Vision benchmark datasets are provided in

the overview of [Li *et al.*, 2024]. We note that for our work, which is interested in deploying the trained machine learning models on-board of satellites and low compute devices, we will often use smaller encoder versions of these models and prioritise faster inference speeds over few percentage increases on benchmark datasets.

Despite the intended spectral dimensionality, which is typically 1 or 3 bands (grayscale or RGB), these architectures can be used with remote sensing data after some small adaptations in the code. These will typically consist of adjusting the desired number of features after the initial convolutional layers. We note that two things can happen here - either we maintain the same number of channels as we were given by the original architecture, or we try to increase them according to the increase of the spectral bands on the input. In the extreme scenarios we either force a too strict bottleneck, which will lose useful data, or we create a too large volume of features which will cause memory problems further down the line in the architecture. Naturally, we are interested in more domain inspired alterations to the machine learning architectures.

2.5.2 Model adaptations for Remote Sensing data

Research in machine learning models applied to the area of multispectral or hyperspectral data proposes some adaptations to the general architectures. Some works, including [Firat *et al.*, 2022] for classification of hyperspectral images, first reduce the dimensionality of the data by running a PCA for each pixel. This approach however loses a lot of information in the preprocessing step and as such would not be applicable for tasks with hard to detect events. Other works, such as [Jongaramrungruang *et al.*, 2022] use classical domain specific methods to extract a single band of information (in this case also a methane enhancement product), which is later processed by a regular convolutional neural network.

The work of [Niu *et al.*, 2022] proposes a hyperspectral variant to the TransUNet architecture [Chen *et al.*, 2021] when working with hyperspectral data from a UAV drone. To address the high spectral dimensionality of the data, it uses the Squeeze-and-Excitation layers [Hu *et al.*, 2018], that allow the model to re-weight each band on the input of each layer. The reported work however uses a relatively small dataset with only two data captures (2332x959 px and 864x1618 px) and a random checkerboard split between the train and test datasets, which means that very similar samples end up in both subsets.

The model proposed by [Sun *et al.*, 2019] uses an array of 1D convolutional layers with kernel size of 1x1, which first processes the data only in its spectral dimension. The paper however tests this model only on the tiny Pavia University dataset.

The SpectralFormer model proposed by [Hong *et al.*, 2021] focuses on the same task using the datasets of Pavia, Indian Pines and Houston - we however note that these remain very small.

The work of [Kumar *et al.*, 2023] uses a custom model adapted from the DETR [Carion *et al.*, 2020] architecture. It ingests hyperspectral data with multiple encoders as: RGB, computed methane enhancement product and 100 selected bands in the range where methane is visible. Each of these is projected by a ResNet model into a feature representation. On the output it predicts both the bounding boxes and the masks of the detected events. The efficiency of the used methods is however not addressed in this work.

Works such as [Cong *et al.*, 2022] adapt the masked autoencoder (MAE) architecture to remote sensing data by masking individual bands and individual samples in a time-series. They later use the model for downstream tasks as a self-supervised feature extractor.

The work of [Fuchs & Demir, 2023] presents a large dataset of hyperspectral images from the EnMAP sensor, and explores different variants of autoencoder models (including adaptations with 1D convolutions) for the task of image compression.

2.6 Systems in compute constrained environments

We can very broadly differentiate between different types of compute constrained environments. While on-the-ground edge devices [Murshed *et al.*, 2021] such as camera traps are stationary and typically use Computer Vision sensors and processing pipelines, devices on satellites or in deep space exploration missions may either be orbiting a planet with a certain revisit time or may be driving on one. These have unique sensor types, either specialised for an application of interest, or general remote sensing ones which can support a number of potential applications. Furthermore, hardware used in space needs to be radiation resistant or hardened [Furano *et al.*, 2020; Kothari *et al.*, 2020; Parr *et al.*, 2024].

2.6.1 Autonomy on-board

In this section, we list several selected deep space and Earth orbiting missions and provide details about the hardware used on these platforms. Due to harsh Space

Table 2.6: Example missions in Space with autonomy on-board of limited compute hardware. Details filled when publicly known. References were shortened for legibility as well as the number of cores available in the CPUs (for example 2c means 2-cores).

Mission	Year	Methods or tasks	Parameters	RAM
<u>Rovers</u>				
Opportunity	2004	AEGIS, VO [2012]	RAD6000@20MHz	128MB
Curiosity	2012	VO, hazards, ... [2020]	RAD750@200MHz	256MB
Perserverance	2021	Navigation, ... [2023]	RAD750@200MHz, Xilinx V5QV	256MB
Ingenuity	2021	Navigation, ... [2022]	Qualcomm Snapdragon 801	
<u>Satellites</u>				
EO-1	2000	SVM, minerals [2006; 2012]	Mongoose V@12MHz	256MB
IPEX CubeSat	2015	RF [2015a]	Atmel AT91SAM9@210MHz	
OPS-SAT	2020	ML demos [2024]	2c Cortex-A9@800MHz, Cyclone V	1GB
PhiSat1	2020	CNN, clouds [2021; 2020]	Details N/A, Myriad 2	
PhiSat2	2024	ML demos [2021]	Details N/A, Myriad 2	
ION SCV-003	2021	Floods mapping [2023]	4c 64-bit CPU, Myriad X	2GB
ION SCV-004	2022	Training [2023a], [2022]	4c GX-412HC@1.2GHz, Myriad X	2GB
Europa Clipper	2030	SVM & RF [2019]	Estimated: RAD750@200MHz	
<u>Space station</u>				
ISS	2022	Image processing [2022]	Snapdragon 855, Adreno 640	

environment, many missions reuse the same radiation hardened devices and bespoke highly reliable solutions. As is noted by [Furano *et al.*, 2020], some commercial off-the-shelf (COTS) devices have also been recently tested for Space applications with vastly higher demonstrated compute capabilities.

Table 2.6 shows the list of devices and missions in space demonstrating autonomy on-board, some of these devices can also be considered as COTS. Meanwhile Table 2.7 lists several COTS devices typically used outside of space environments for on-board processing and autonomy. We note that some of these have been proposed for deployment in space only for mission non-critical applications in [Lofqvist & Cano, 2020; Guertin, 2022]. We also highlight that several of the so-called COTS devices were also used in real-world missions in space [Furano *et al.*, 2020], namely the Intel Movidius Myriad chips which are used in a number of missions (ION SCV satellites by D-Orbit, Phi-Sat satellites by ESA) or the Qualcomm Snapdragon accelerators typically used in mobile devices, which were recently tested in the Ingenuity helicopter [Tzanetos *et al.*, 2022] on Mars and for various experiments on the International Space Station (ISS) [Swope *et al.*, 2022]. This list is not exhaustive, for other devices used in non-space environments for Internet of Things (IoT) and other on the edge applications we refer the reader to [Murshed *et al.*, 2021].

Table 2.7: Commercial off-the-shelf devices with hardware specifications. Locally used abbreviations include: RasPi for Raspberry Pi., Jet. for NVIDIA Jetson and TC for Tensor Cores. Number of CPU cores was also shortened (for example 2c means 2-cores).

Device	Year	CPU	VPU/GPU	RAM
Xilinx Pynq Z2	2011	2c Cortex-A9 @ 650MHz	FPGA	512MB
RasPi 3B+	2018	4c Cortex-A53 @1.2-1.4GHz	VideoCoreIV @250MHz	1GB
RasPi 4B	2019	4c Cortex-A72 @1.5GHz	VideoCoreIV @600MHz	8GB
RasPi 5	2023	4c Cortex-A76 @2.4GHz	VideoCoreVII @800MHz	8GB
Jet. AGX Xavier	2018	8c Carmel @2.265 GHz	GV10B GPU (16GB)	16GB
Jet. Nano	2019	4c Cortex-A57 @1.47 GHz	Maxwell GPU	4GB
Jet. Orin Nano	2023	6c Cortex-A78AE @1.5 GHz	Ampere GPU, 16 TC	4-8GB
Jet. AGX Orin	2023	12c Cortex-A78AE @2.2 GHz	Ampere GPU, 64 TC	32-64GB

Early example of [Castano *et al.*, 2006] demonstrates a deployed SVM classifier working on-board of the Earth Observer 1 (EO-1) spacecraft as part of the so-called NASA’s Autonomous Sciencecraft Experiment (ASE). The processing runs on a payload processor equipped with radiation-hardened Mongoose V CPU (12 MHz and 256 MB RAM) and the Hyperion imaging instrument. It has shown that a learnt SVM model can outperform other more manually tuned algorithms and that it can run on near-raw, only partially processed data (so-called level 0.5 which skipped some processing steps used when converting raw data of level 0 into processed data of level 1 [Barry, 2001]). Later in [Thompson *et al.*, 2012] mineral detection was performed on the same hardware using a pipeline of classical algorithms for superpixel endmember detection.

In comparison, a follow-up work conducted by [Thompson *et al.*, 2015a] on the smaller Intelligent Payload Experiment (IPEX) CubeSat spacecraft carrying a cheaper radiation non-hardened Atmel AT91SAM9 CPU (210 MHz), used a trained random forest classifier to differentiate between clear, cloudy, planetary-limb or outer-space pixels to allow for filtering of captured data samples (for example selecting pictures with a threshold on how many clear pixels it contains).

The work of [Estlin *et al.*, 2012] describes the hardware available on board of the Mars Exploration Rover (MER) mission Opportunity equipped with “The Autonomous Exploration for Gathering Increased Science” (AEGIS) targeting system which aids with rock selection from images captured by MER Panoramic camera for follow-up ChemCam spectrometer measurements. The Opportunity rover hosted a radiation resistant RAD6000 flight processor (with 20 MHz and 128 MB RAM) with

requirements for the AEGIS system to use at maximum 4 MB RAM not to interfere with its other systems. The reported processing typically took less than 15 minutes (while on a modern personal computer of that time it would take less than 1 second). It uses the Rockster edge detection algorithm to find rocks and a series of user specified attributes to prioritise the final target. Meanwhile the newer Mars rovers Curiosity (active since 2012) and Perseverance (active since 2021) are equipped with the RAD750 CPU (with max clock rate of 200 MHz). Perseverance is furthermore equipped with another set of CPUs and the Xilinx V5QV FPGA for computer vision tasks for navigation [Verma *et al.*, 2023] - and is accompanied with the Mars Helicopter Ingenuity [Tzanetos *et al.*, 2022] which uses Qualcomm Snapdragon 801 on-board (which is technically more powerful than the processor of the Rover itself). Usage of FPGA's as technology for data processing in space missions was further explored in [Guerrieri *et al.*, 2018].

The planned on-board detection of scientifically important events for the Europa Clipper mission [Wagstaff *et al.*, 2019] considers the same radiation robust RAD750 CPU (with max clock rate of 200 MHz) and plans to use SVMs and Random Forest classifiers instead of deep learning neural networks given the hardware restrictions.

More recently, deep learning methods have been considered by the water segmentation model deployed on board of the Phi-Sat 1 experimental spacecraft, which is capable of running a fully convolutional neural network (FCNN) designed with only 0.26M parameters on board of the Intel Movidius Myriad2 visual processing unit (VPU) while processing a 12 MP image in less than 1 minute [Mateo-Garcia *et al.*, 2021]. It is shown to outperform two tested baseline methods based on the normalised difference water index (NDWI) thresholding and a per-pixel linear model. Machine learning models were also proposed for the demonstrator mission of the follow-up Phi-Sat 2 satellite [Marin *et al.*, 2021].

Finally, our own work from Chapters 3 and 5 has been deployed on limited hardware machines to measure inference speeds of the proposed machine learning models. Both use the so-called Unibap's SpaceCloud Framework (SCFW) software environment [Flordal *et al.*, 2021], which works similarly as a Docker environment allowing relatively easy deployment of python trained models. The software environment is furthermore run on the flatbed devices using the SpaceCloud iX5-100 radiation tolerant environment. This gives us a unique chance to test models on a realistic hardware, which has been flown in space. Similar hardware was also used for running machine learning models on previous demonstrator missions in [Mateo-Garcia *et al.*, 2023].

Platforms with higher compute power such as SpaceCloud iX10-100 were also considered in the recent work of [Gómez & Meoni, 2024] for on-board image compression using machine learning models.

2.7 Efficient machine learning

Crucial when considering autonomous models on-board of deployed devices are the limitations stemming from the computational power available, acceptable energy consumption over the lifetime of the mission or the cyclic nature of availability of renewable energy sources (for example solar energy on some deployed devices). These are even more highlighted in the case of deep learning models, which may often contain unique features not available otherwise, or may outright outperform traditional, non-deep learning methods, however, have increased requirements for hardware specifications.

Quite exciting is the research in the EdgeML area, which explores how we can minimise the existing deep learning models and make them available for on-the-edge devices - the more extreme version of this research is TinyML, which focuses on ultra-low-power devices such as Internet of Things (IoT) devices. In our work we use the definitions for EdgeML and TinyML as given by the overview work of [Lin *et al.*, 2023] - further details about the definitions of efficiency are explored in Section 2.8.

When considering which methods to use to optimise the used ML model, we should also consider the available hardware specific to each of the missions, which can range from very compute-limited devices (for example space radiation resistant flight processors) to more powerful devices having their own specialised compute units to perform demanding operations (for example the GPU or specialised Tensor core processing units on mobile devices). We note that in most cases relevant to our work, we do not need to aim at IoT devices as the target platform and that hardware on-board of typical satellite is more powerful [Furano *et al.*, 2020]. As such, we would be able to deploy models designed for Mobile Platforms such as MobileNets [Sandler *et al.*, 2018], ShuffleNets [Ma *et al.*, 2018], but also the more recent EfficientViTs [Cai *et al.*, 2023] on the relevant hardware.

We understand machine learning model efficiency mainly in terms of reducing inference speed while fitting into constraints defined by the hardware limitations (such as memory) and the device’s power budget.

2.7.1 Machine learning techniques

The overview of [Verhelst & Murmann, 2020; Menghani, 2021] and the feature piece by [Lin *et al.*, 2023] highlights many options for making deep neural networks more efficient and allows them to be used on-board of low-compute and low-energy devices.

As listed by [Lin *et al.*, 2023], we can broadly differentiate between software (e.g. software implementation of a more efficient computation of convolution operations), hardware (e.g. inference engine built with a certain hardware capability in mind) and systems solutions (e.g. frameworks such as TensorRT or OpenVino not depending on Python and allowing for efficient inference independent on the used library).

Over the past years, model architectures became more and more *efficient by design*, motivated by allowing larger models capable of learning from bigger datasets. This was the case with designing more efficient reused building blocks in Inception net [Szegedy *et al.*, 2015], Xception net [Chollet, 2017], or the introduction of residual connections in ResNet [He *et al.*, 2016]. These advances allow for more efficient model architectures, but also for better distribution of which concepts are learnt at which part of the neural network, as can be exemplified by using residual (skip) connections in U-Net models [Ronneberger *et al.*, 2015]. We can propose alterations to some of these original machine learning architectures to make them more efficient according to the task we deal with. The simplest example of this would be the reduction of the number of convolutional channels, or layers of reused blocks in some of these models.

Additionally, *model quantization*, reduces the precision point of the used operations and the saved weights [Han *et al.*, 2015a; Rastegari *et al.*, 2016]. This can lead to smaller model impact not only on storage space, but also on energy required to run these networks [Moons *et al.*, 2017]. More recent research by [Xiao *et al.*, 2023] focuses on the extreme end of quantization of large language models (LLMs), which can have many billions of parameters and where the prior used methods weren't capable of maintaining accuracy.

Model pruning is another efficiency technique which removes nodes in the network with low impact on the overall model's prediction [Han *et al.*, 2015b; He *et al.*, 2017]. The resulting pruned networks however need adapted inference engine to make use of their sparsity [Han *et al.*, 2016].

Other approach is to consider a larger trained model as a teacher and train a smaller student network which *distills the knowledge* in a more efficient manner [Hinton *et al.*, 2015].

Alternative approaches try to speed up matrix-matrix multiplications in convolution with custom implementations [Zhang *et al.*, 2018] or by applying tensor decomposition [Lebedev *et al.*, 2014].

The proper functioning of all mentioned techniques is highly interconnected and might need to be explored at the same time, which is where the concept of AutoML, automatic exploration of the search space of possible model architectures [Liu *et al.*, 2018], can be useful also for searching for efficient and easily deployable models on devices with limited compute capabilities [He *et al.*, 2018].

As highlighted by [Lin *et al.*, 2023], EdgeML and TinyML doesn't only concern model inference, but also model training. We touch upon a similar topic in Chapter 3 when training small models on-board of satellites leveraging few-shot learning paradigms, where only few samples are needed to train a model (this depends on having a pre-trained model with qualitative embedding space). One common method is to rely on features of pre-trained models such as [Sharif Razavian *et al.*, 2014] and to fine-tune few layers for the task at hand on the edge device. Other works such as [Chen *et al.*, 2016] propose more efficient backpropagation algorithm implementation at the cost of increased training time. Recent works such as [Wang *et al.*, 2022] propose memory efficient training for mobile devices.

Finally, embedded ML may benefit from custom made processing units such as Nvidia's Tensor Cores [Markidis *et al.*, 2018], or more relevant for TinyML, even more customised processors [Moons *et al.*, 2019].

2.8 Definitions of efficiency

What do we mean when we talk about "efficiency"? In the works presented in this thesis, we usually have a relatively concrete idea of which model or which algorithm we want to run on which platform. Our needs and targets are quite specific and binary - can this model run, or not? However, as we listed in Section 2.6, there are many different devices in use, each with their unique constraints.

The work of [Furano *et al.*, 2020] provides a very useful overview of typically considered parameters in satellite and deep space missions and also mentions several parameters unique for the deployment of machine learning models. Traditionally, we consider the size, weight and power (SWaP) of the system. Size and weight directly influences the cost of launching satellites and the available power budget influences the devices that can be used on the system. All recent overviews [Furano *et al.*, 2020; Swope *et al.*, 2022; Kothari *et al.*, 2020] agree on the fact that we are entering a new

era of space hardware, where commercial off-the-shelf (COTS) hardware achieve much better characteristics in terms of SWaP than traditional satellite missions relying on bespoke and custom made hardware. For that comparison [Furano *et al.*, 2020] suggests a calculation based on the following:

$$\text{performance [Mpixel/s]} / \text{weight [kg]} / \text{power [W]} / \text{cost [\$]} \quad (2.1)$$

A small satellite mission will have a much better score than a traditional satellite. We however note that the “performance” parameter is at best approximative - an example of 600 MHz Myriad 2 chip is given as capable of processing at the speed of 600 Mpixel/s. However, this number naturally depends on the used algorithm and other system conditions (such as speed of input / output). Nonetheless, we see that measuring inference time needed for processing a given image size is an informative metric - this allows us to consider “efficiency” in similar terms.

Parameters of these systems need to be considered together as they are highly interdependent. While the usual tradeoffs between accuracy and speed of the used algorithm applies, in some cases, allowing potentially costly on-board processing can actually save the costs of the overall system, if a useful decision can be taken. For example in the case of the Perseverance rover [Verma *et al.*, 2023], an OnBoard Planner (OBP) system can adjust the daily planned tasks (such as where exactly to drive, managing pre-heating of required instruments and others) on-board and is estimated to save the overall energy cost and effectively allow longer rides. There are costs associated to running this system that are lower than these estimated savings. Another example could be data compression on-board, as mentioned by [Kothari *et al.*, 2020], power costs linked to data transmission can be lowered by using a deep learning model for lossy data compression. In these two examples, if we care about power costs, running extra computations can actually be more *efficient*. The considered algorithm classes naturally need to follow the hardware available on each mission - for example the tree search algorithm planned for the Perseverance rover is adjusted to not allow backtracking to keep the computation time linear [Verma *et al.*, 2023].

In addition to the usually considered parameters such as SWaP, machine learning models deployment need to pay extra attention to the available memory budget. Remote sensing images processed on satellites are typically tiled into a non-overlapping grid and in some cases it is possible to process a larger number of these tiles at once by processing them in a batch. Therefore, for the deployment, parameters such as tile size and inference time batch size may have to be tweaked to fit into the specifications

of the target device. As a general rule of thumb, selecting the largest batch size that still fits the device tends to be the fastest (up until some number - this is given by details and limitations of the hardware). However, care needs to be taken when the same chip is used by other systems on the satellite. In those cases, there may be unexpected execution delays. We have also experienced these in our work on [Růžicka *et al.*, 2023a], where executing models on a set aside device has led to more reliable worst execution time estimates (namely see Figure 3.10). As another example, Mars rovers prior to Perseverance used to have to stop to re-compute visual odometry information after each passed meter, while Perseverance is able to effectively think while driving by having another set of the same processing hardware (RAD750 CPU).

The efficiency of machine learning architectures can also be seen in the view of the number of samples required for training. For example, in the case of new satellite missions, finetuning existing models will be preferred over training large foundation (and usually data hungry) models.

As is also highlighted by [Furano *et al.*, 2020], machine learning models are best used for specific tasks inside fault tolerant systems. For example predicting a specific class mask for newly observed images, rather than full control of the entire system. This is due to the non-deterministic nature of data driven models.

In this work we understand **efficiency as the measured inference speed on devices that form realistic proxies to the final deployed environment**. This should be usable as a means of comparison between different variants of models tested on the same device, but also to make estimations of the “performance” metric used by [Furano *et al.*, 2020]. This allows us to build our models as “efficient enough” to fit under other system parameters - such as the available memory budget or power costs. Additionally, we deploy our models on several types of devices (from low-powered to medium-powered with specialised processing units). In most cases we propose machine learning architectures that are as small as possible (usually in terms of the number of parameters) - in the spirit of being efficient by architectural design. Namely, in Chapters 3 and 5 we test different variants of models (using different encoder sizes and numbers of layers) to explore the tradeoffs between accuracy and inference speeds.

2.9 Summary

The work presented in this thesis combines three main disciplines - research in methods, as explored in Section 2.3, domain specific knowledge explored in Sections 2.2

and 2.4 and knowledge of limited compute hardware and deployed systems as explored in 2.6.

Our works aims to innovate mainly in methodology using the particular restrictions and unique configurations presented by the tasks such as change and weak signal detection on-board of satellites.

Change Detection on Satellites

3.1 Introduction

The main focus of this chapter is the deployment of an **anomalous change detection system on-board of a satellite (Q1)** to promote intelligent decision-making on-board and to enable several downstream tasks. We frame our problem as unsupervised change detection in a time series of images from the satellite’s sensor. In this case, we use low processing level data from the Sentinel-2 satellite to train our models.

The context of our work is the global presence of satellites - they observe Earth in unified manner globally. The uniformity of data capture and the harmonised products allow for observation over a selected location of interest at multiple time steps. Sentinel-2 has on average about 5 days revisit time. In this setting, we propose a system which enables detection of changes on-board of satellite, while being agnostic to small natural fluctuations in between data captures due to noise and natural growth. Previously tasks such as cloud detection for selection of clean samples for downlinking was tackled with simple machine learning models deployed on-board in [Giuffrida *et al.*, 2020]. More recently, models detecting particular disaster event, such as floods, were also deployed in [Mateo-Garcia *et al.*, 2021].

Instead, we focus on general anomalous event detection, which would enable the system to detect any large change observed in the data. We illustrate the question we are trying to answer on Figure 3.1. We also explore on-board deployment of our system, which will enable us to react to detected events of time sensitive nature. Additionally, we aim to address the question of adaptability of the deployed model

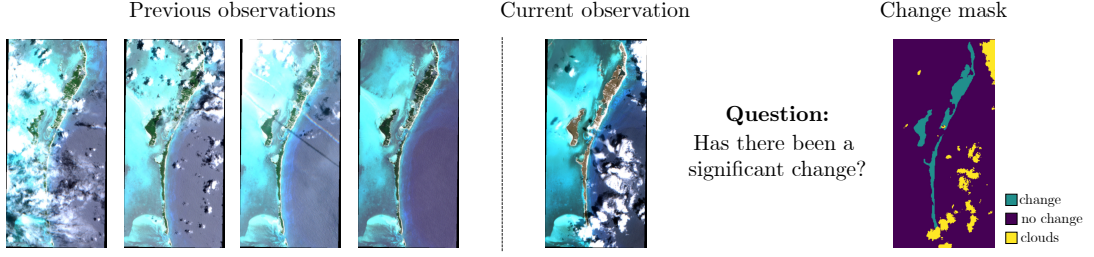
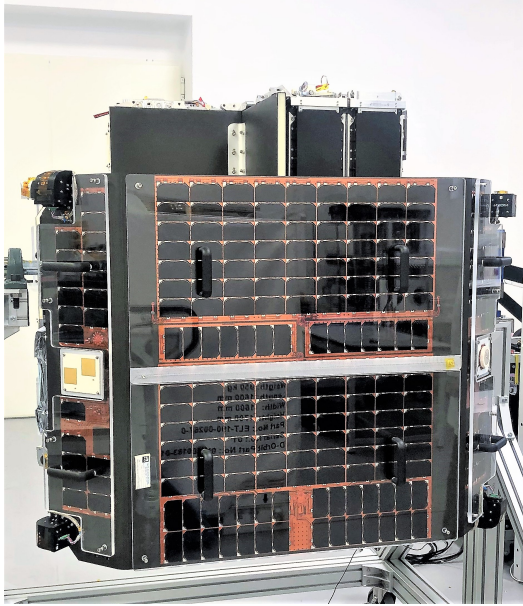


Figure 3.1: Illustration of the change detection task on-board of satellite. We show a sequence of Sentinel-2 images, the most up-to-date one is after a hurricane event. On the right we show the ground truth annotation.

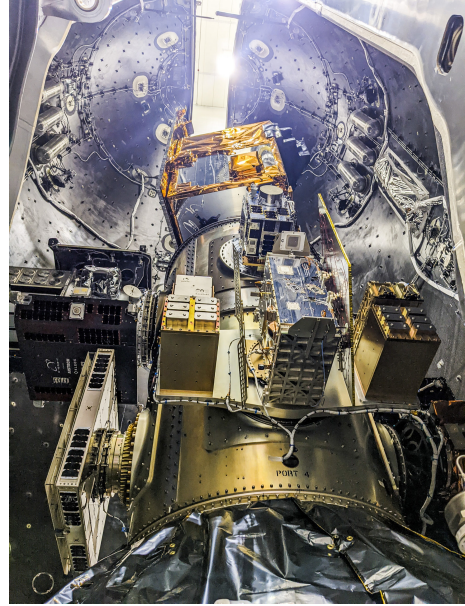
in scenarios where we for example encounter sensor degradation, or supervision from other satellites in a constellation.

To summarize, in this chapter we propose a machine learning model based on the variational autoencoder (VAE) [Kingma & Welling, 2014] architecture, which we train on a dataset of low level of processing Sentinel-2 data (L1C). We evaluate this model on the task of unsupervised change detection on a dataset of disaster events with samples from several distinct event classes (such as wildfires, floods, hurricanes and landslides). We test the proposed model on low compute hardware and we also deploy it on-board of a real ION-SCV 004 satellite as part of the *Dashing through the Stars* mission by the aerospace company D-Orbit. Figure 3.2 shows the ION-SCV 004 satellite in a clean environment and a typical attachment style onto the transponder rocket. On this satellite, we demonstrate the world’s first training of a neural network architecture on-board of satellite’s limited hardware to explore model adaptation to newly received information.

This research is significant, as it explores real-world limitations of our machine learning system, and as it enables opportunities for intelligent decision-making on-board. Detected anomalies can trigger a fast alerting signal which could be sent via a priority channel, and they can also enable prioritization by downlinking salient data of interest. As systematically illustrated on Figure 3.3, the traditional processing pipelines allow detections of events within days, while our proposed method could enable detections within minutes of the data capture. Latent representation of the data also reduces storage requirements on-board of the satellite and can be used for compression when exchanging information in a constellation of satellites.



(a) ION-SCV 004 satellite



(b) SpaceX Falcon 9 Transporter-4

Figure 3.2: Photos of the D-Orbit satellites: (a) shows the ION-SCV 004 satellite in a clean room and (b) shows the SpaceX Falcon 9 Transporter-4 rocket with a similar ION-SCV 005 satellite attached (on the left of the image). The SpaceX rocket itself operates as a carrier through the smallsat rideshare program, while the D-Orbit’s ION Satellite Carrier Vehicle also eventually launches smaller cubesats when orbiting in space during the Phase 1 of the mission, while in Phase 2 it runs experiments on-board of the SCV. Photos by SpaceX and D-Orbit from public posts (a, b).

3.2 Background

We formalise our problem as unsupervised change detection in a time series of data. Depending on how we frame the problem, we could also use the notation of anomaly detection and use the relevant methods interchangeably. We provide an overview of the referred methods and task definitions in Chapter 2 and details about change and anomaly detection in this section.

3.2.1 Prior methods

Change and anomaly detection has been classically approached by comparing normalised samples in a set of data (one sample against potentially all others in anomaly detection, or previous sample with current one in change detection). We can broadly distinguish between classical methods and methods using either shallow or deep machine learning models. For more details, we refer the reader to overviews of [Radke *et al.*, 2005] and for classical approaches to [Lu *et al.*, 2004].

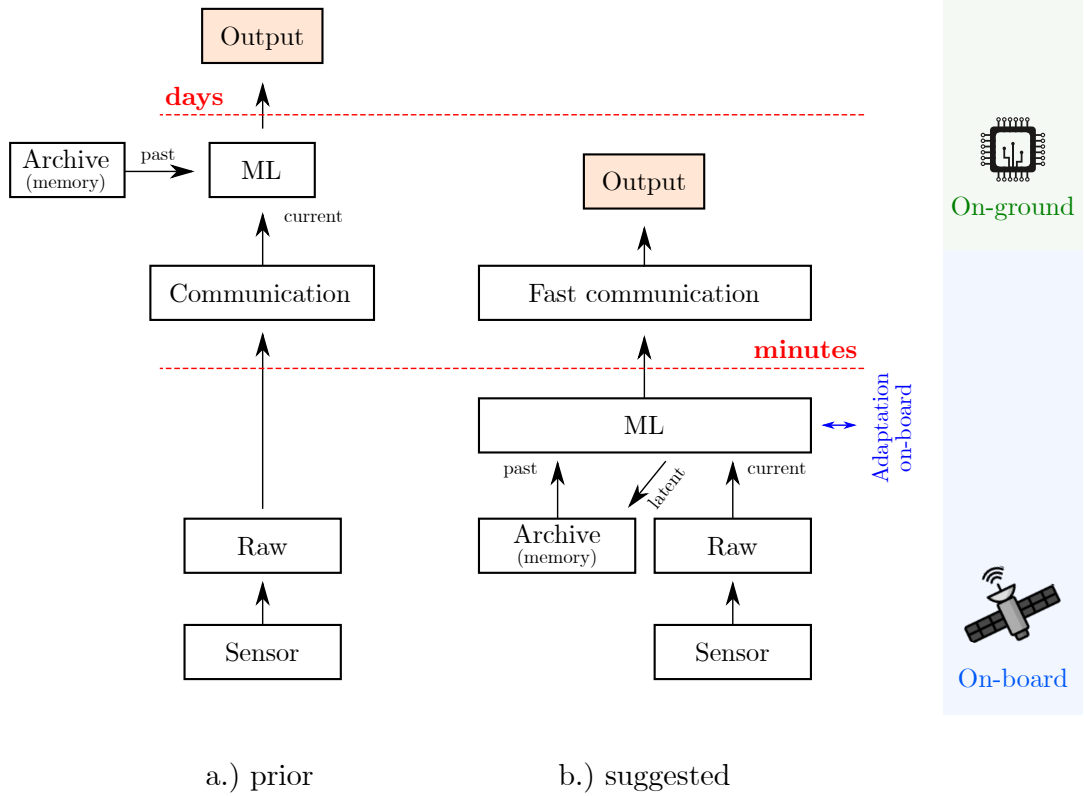


Figure 3.3: Illustration of the traditional processing pipelines and the difference that on-board processing could enable.

Change detection can most easily be considered as a measure of difference between two samples. With the same dimensionality of the data volumes, we are able to compute the difference image $D = I_{t_2} - I_{t_1}$ between all pixel values across all bands. When considering remote sensing data, we may instead use a classical index representation, for example the Normalized Difference Vegetation Index (NDVI) which is used to describe the density of the green vegetation on the captured area. It is calculated as $NDVI = \frac{NIR-RED}{NIR+RED}$ using the near infrared (NIR) and red (RED) bands of the image (for example with Sentinel 2 data this corresponds to “B8” and “B4”). The difference image would then be computed between two computed NDVI indices. Finally, the Change Vector Analysis (CVA) technique has been used with remote sensing images measuring angle magnitudes and directions of the subtracted images [Johnson & Kasischke, 1998].

Instead of using the raw data values, shallow machine learning methods for dimensionality reduction are often employed to remove noise in the data. We can for example use the Principal Component Analysis (PCA) [Dunteman, 1989] method for change detection by making a vector representation of the pixel values between the

paired images in sequence as (I_{t_2}, I_{t_1}) for all bands and over all coordinates. If these vectors were to be plotted on a 2D chart, the unchanged pixel values would have near identity relation, while the changed pixel values would deviate. Using this intuition, we can calculate the principal components (PC) of the data and consider the pixels corresponding to the first PC as those not containing any change. Meanwhile pixels corresponding to the second PC (orthogonal to the first principal axis) can be considered as containing changes, alongside with the following PCs. This approach has however been reported to require visual inspection by a domain experts to estimate which of the components truly correspond to the change [Radke *et al.*, 2005].

In [Volpi *et al.*, 2013] a Support Vector Machine (SVM) classifier is used for per-pixel change detection with high resolution data (ground resolution of 0.7m). Each pixel is described with local statistics calculated from a sliding window. These statistics include for example local mean and variance information, co-occurrence frequencies of pixel values and minimum and maximum differences within the pixels inside the window. Dimensionality of the tested extracted features ranges between 4 and 63, which we note as relatively low when compared with features of the later deep learning models. SVM models are further reported in [Nemmour & Chibani, 2006] to outperform a shallow neural network model and in [Bovolo *et al.*, 2010] to outperform a traditional CVA approach.

We note that we can consider the task at hand also as an instance of the anomaly detection task. For anomaly detection we use similar classical methods, including Reed-Xiaoli (RX) [Reed & Yu, 1990], PCA in [Shyu *et al.*, 2003] and SVM in [Banerjee *et al.*, 2006].

Machine learning provides many methods for performing feature extraction, that can be used in the framework of change or anomaly detection. In supervised change detection scenarios, where we have a dataset of labelled events, we can use Siamese models proposed by [Zagoruyko & Komodakis, 2015] and later applied on remote sensing data in the letter by [Zhan *et al.*, 2017]. The work of [Daudt *et al.*, 2018] adapts the popular U-Net model as Siamese U-Net which is later also used by the author of this thesis in [Růžička *et al.*, 2020].

In unsupervised change detection scenarios, we can obtain feature representations instead of the original data for conducting feature space comparisons. Bottleneck feature of a pre-trained model can be used to describe data - either using the encoders of classical computer vision deep learning models such as [Krizhevsky *et al.*, 2012], or models trained on sensor specific datasets [Sumbul *et al.*, 2019]. Unsupervised learning methods further include generative autoencoder models [Kingma & Welling,

2014; He *et al.*, 2022; Larsen *et al.*, 2016] or contrastive learning approaches such as [Chen *et al.*, 2020; Manas *et al.*, 2021].

On-board deployment of classical or machine learning models has been explored in [Giuffrida *et al.*, 2020; Du *et al.*, 2024] for cloud detection, in [Mateo-Garcia *et al.*, 2021] for water extent segmentation and while searching for other suitable tasks also in [Meoni *et al.*, 2024]. In such cases exact parameters of the used system govern the selection of the methods. We have to consider the power, compute and memory of the deployed device [Furano *et al.*, 2020]. As this thesis focuses on on-board deployment of machine learning models in multiple chapters, a more holistic analysis is provided in Section 2.6.

3.2.2 Challenges and opportunities

Specialised detection system trained on a single type of event [Mateo-Garcia *et al.*, 2021] cannot easily generalise to other tasks. We note that many unsupervised machine learning models can be leveraged here to enable generality of our solution.

Many existing change detection systems do not optimize the proposed models for on-board deployment on a device with restrictions in terms of compute and memory. While slow and complex models can also be deployed on satellites when the number of processed samples is very small [Serief *et al.*, 2023], in real-world scenarios it may be necessary to propose small and highly efficient models.

3.3 Method

To achieve unsupervised change detection deployable on-board of satellite, we propose a machine learning model based on the variational autoencoder (VAE) architecture. We train this generative model on the task of reconstruction of the input data after passing through a compressed feature bottleneck. This model is further described in Section 3.5. We use an unlabelled dataset of images from the Sentinel-2 satellite, from which we consider high resolution bands from the L1C level of data processing - we describe this in Section 3.4. We evaluate the trained model on a dataset of disaster events, containing samples from classes such as wildfires, floods, hurricane and landslide events. Each location is captured with a time series of images, last one with the change.

Finally, as illustrated on Figure 3.4, we also deploy the proposed model on-board of a real satellite and explore model adaptation through training on-board. We cover each of these steps in the following sections.

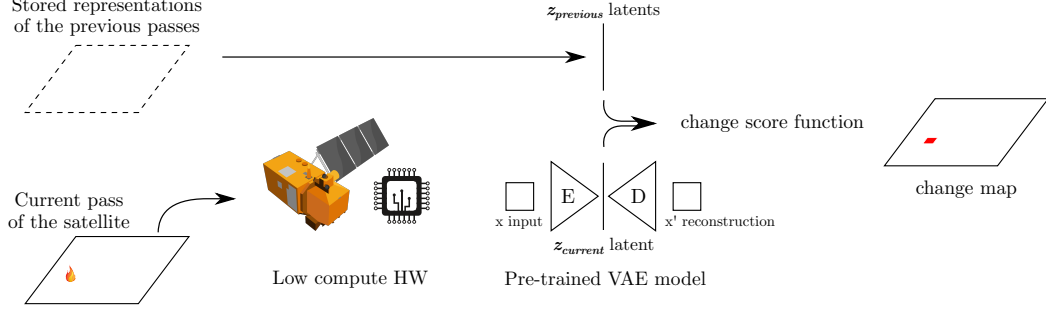


Figure 3.4: Illustration of the proposed RaVAEn system. A pre-trained VAE model is used to encode incoming data captures, the resulting latent representation provides better signal for anomalous change detection.

3.4 Data

3.4.1 Instrument

We have selected the Sentinel-2 satellite for this project, given its high temporal cadence of data (only 5 days revisit time on average), relatively high ground resolution (in most bands at 10m for one pixel) and the number of spectral bands that are relevant for a wide selection of tasks [Hoersch, 2013]. Furthermore, it provides global coverage of the Earth and an easy and free access to the data. Many other works in the domain of Remote Sensing also use Sentinel-2 data, which extends the relevance of our research into other domains.

To better mimic the environment on-board of a satellite, we selected low processing level of data denoted as L1C. This data product has been orthographically corrected and misaligned bands were adjusted, however it hasn’t been corrected atmospherically. We note that at the time of working on this project in [Růžicka *et al.*, 2022], there were no so-called L0 data products available. These would be the raw data captures directly from sensors on the satellite. Only recently some small subsets of L0 data were released in [Meoni *et al.*, 2023], however even these wouldn’t be sufficiently large and diverse for training our models.

3.4.2 Datasets

In total, we use three datasets made from Sentinel-2 L1C data. For training our proposed models, we leverage the globally sampled WorldFloods dataset (using its v1 release) from [Mateo-Garcia *et al.*, 2021]. It contains 233 scenes, each as a time series of on average five images per scene. We note that this dataset has a diverse

Table 3.1: The RaVÆn dataset statistics. Each location is captured in 4 time-steps before the event and once after the event. Positive rate denotes the ratio of changed to non-cloudy pixels in the last pair of images (the only ones that are annotated).

	Number of locations	Cumulative area (km ²)	Positive rate (%)
Landslides	5	108	10.48
Floods	4	1301	6.74
Hurricanes	5	1622	24.31
Fires	5	3485	53.79

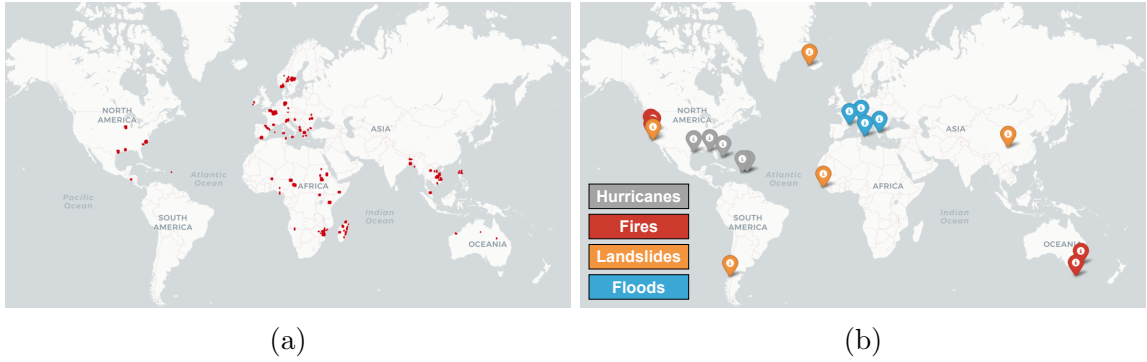


Figure 3.5: Locations used for training (a) and validation (b) images.

representation of samples from around the world, and usually captures images around cities and rivers. When tiled into 32x32 px tiles, this creates a training dataset of almost 7.6M samples.

For validation of our model, we use the RaVÆn dataset manually selected and annotated by the co-authors of our paper [Ružička *et al.*, 2022] - this dataset contains four disaster event types (wildfires, floods, hurricanes and landslides) as is detailed in Table 3.1. Each location contains a time series of five images, the last one containing change related to a disaster event. Each has a manual label which assigns each pixel into classes of no-change, change or clouds (clouds were marked for either the last or the second to last image). Four images in the time series before the event contain only natural changes in cloud cover, vegetation growth, variable viewing angle of the satellite and sensor noise. The images were sampled with a window of 180 days before the event allowing only samples with less than 20% cloud cover. Cloud cover masks were generated using the s2cloudless [Sentinel Hub, 2021]. Figure 3.5 shows the distribution of samples in these two datasets.

Finally, for experiments on-board, we also used a smaller dataset of Sentinel-2 images which has been pre-deployed on-board of the satellite before it’s launch. This

dataset covers only the region of Slovenia. Furthermore, we have selected a very small subset of this dataset (in total only 6 images each of 480x480 px) and labelled each 32x32 tile with a label of cloud or no-cloud. This dataset was later used to demonstrate few-shot learning on-board of the satellite.

In most of our experiments we used the 10 high resolution bands of Sentinel-2 at ground resolutions of 10m and 20m. One exception was the pre-loaded dataset on-board of the satellite, which contains only 4 bands (RGB+NIR).

3.4.3 Pre-processing

We normalise each separate band of the data using statistics calculated from the training dataset. We use the following formula:

$$\begin{aligned} x' &= \log(x) \\ x'' &= 2 * \frac{x' - \min(x')}{\max(x') - \min(x')} - 1 \end{aligned} \quad (3.1)$$

Here, $\min(x')$ and $\max(x')$ denotes values precomputed from the training dataset. This transforms the data into range between $-1; +1$, which is consistent with the steps used in other papers [Brown *et al.*, 2022].

Finally, we tile each dataset into small non-overlapping tiles of 32x32 pixels (this covers an area of 320x320 meters), while keeping the time series of the tiles at each location together.

3.5 Models

3.5.1 Task formalisation

We denote $x_t^{a,b} \in \mathbb{R}^{32 \times 32 \times 10}$ as a data sample at location a, b and at time t (a, b is a generically described coordinate of the tile). For tile $x_t^{a,b}$, we are detecting if a significant change occurred from the k -long history of samples at the same location $\{x_{t-k}^{a,b}, \dots, x_{t-1}^{a,b}\}$. We define a novelty score function $S(\mathbb{R}^{32 \times 32 \times 10}) \rightarrow \mathbb{R}$ as:

$$S(x_t^{a,b}) = \min_{i=1 \dots k} d(x_{t-i}^{a,b}, x_t^{a,b}) \quad (3.2)$$

where $d(\mathbb{R}^{32 \times 32 \times 10}, \mathbb{R}^{32 \times 32 \times 10}) \rightarrow \mathbb{R}$ is an arbitrary difference function between two tiles. In other words, using this formalisation, we define the novelty score given to the last image in each time series as the minimal distance between all previous samples and the last sample. If at least one previous sample looks similarly as the last image

(and correspondingly has low distance between the two samples), then we do not flag it as a significant change. This allows us to be more robust to small spurious changes and the presence of for example clouds. We assume that the potential anomalous event is present only in the last image - if there were anomalies in the previous images, they would have been detected by our system earlier.

3.5.2 Baselines

Firstly, we use simple distance functions in the pixel space of two images inside the novelty score function 3.2. Here, we can use different distance metrics, such as Euclidean or cosine distances between pixel values of the samples. We do so after applying the same pre-processing steps as with the other methods. We note that these baselines are quite simplistic, but they are also very fast to compute and as such they are suitable as candidates for easy on-board deployment.

Secondly, we use approaches based on the principal component analysis (PCA) method, as has been explored in prior research in [Dunteman, 1989; Radke *et al.*, 2005]. We can consider PCA as a dimensionality reduction method which is first fitted on a training dataset and then can be used to project new samples (for example from the test dataset) into a lower-dimensional space. These samples can later be reconstructed and compared with the original data (also by selecting the number of kept principal components). Using PCA can follow either change detection (CD) or anomaly detection (AD) paradigms. In CD, we consider samples in a sequence (image before and after event), while in AD we consider most samples as normal with a minority of anomalous events (in most cases these anomalies are not included in the training datasets).

We use PCA in three modes: (i) PCA for CD, where we fit PCA on all tiles from a particular location and then project images before and after into low-dimensional representation (in this case keeping 50 principal components). (ii) PCA for CD using difference vectors (values of tile before minus tile after the event), we then check the reconstruction loss of these difference vectors using only 1 principal component. Intuitively, anomalous data (assumed to belong to the change) will not be reconstructed well when using such a low number of principal components. (iii) PCA for AD, where we fit the PCA on all tiles extracted from images before the event (depending on the selected memory size) and check the reconstruction error (again keeping only 1 principal component) of tiles from the tiles of image after the event.

Finally, we chose to extract smaller tiles from the Sentinel-2 images when using PCA. Namely, we extract tiles of 8x8 px without overlap and keep the same number

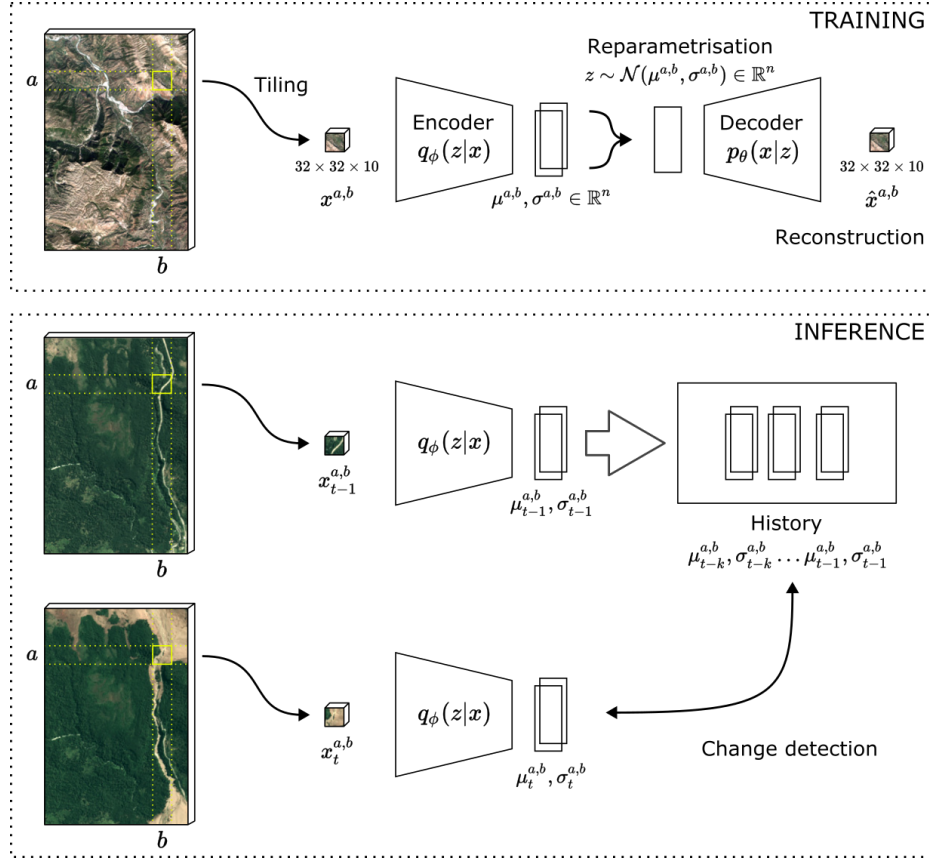


Figure 3.6: Diagram of the RaVAEn system detailing the training and inference steps. Tiles $x^{a,b}$ of dimension $32 \times 32 \times 10$ from the original Sentinel-2 multiband L1C data from the training dataset are fed to a VAE model. Here, a and b correspond to the location of the tile. The VAE is trained in an unsupervised fashion as its encoder learns to compress the tile in an Gaussian embedding representation $\mu^{a,b}$ and $\sigma^{a,b}$ and the decoder to reconstruct them from there. At inference, only the trained encoder is needed as we compress evaluation dataset tiles $x_t^{a,b}$ into their embeddings $\mu_t^{a,b}$ and $\sigma_t^{a,b}$ which can be compared against an history of k embeddings extracted at the same location to assess whether the tile has changed significantly and prioritise for downlink.

of high resolution bands (10) as for the other methods. This means that a single sample in the created dataset has $8 \times 8 \times 10$ values. We note that as in the case of other methods, we use the same data normalisation. All of these different ways of computing distances between tiles from images before and after the event use the same novelty score function of 3.2.

3.5.3 RaVAEn models

We propose a model based on the variational auto-encoder (VAE) architecture which learns a feature embedding space for input tiles $x^{a,b}$. Auto-encoder models consist of two separate networks, the encoder q and decoder p as shown on Figure 3.6. These generative models learn a data representation space by projecting input samples using the encoder network into a lower dimensional space called bottleneck or latent space z . The decoder network then tries to reconstruct the encoded feature back into the original data domain. VAEs impose additional restrictions on the distribution of the learned latent space [Kingma & Welling, 2014] - in particular they impose a diagonal Gaussian distribution for the embedding, which is parametrised by mean and standard deviation of each dimension $\mathcal{N}^{a,b} = \mathcal{N}(\mu^{a,b}, \sigma^{a,b})$. Sampling from this distribution we obtain the latent variable z , but practically speaking, we can use the mean vector $\mu^{a,b}$ for feature representation.

The model is trained using data reconstruction L2 loss between the input tile $x^{a,b}$ and the reconstruction $\hat{x}^{a,b}$, and the Kullback-Leibler (KL) divergence - as is custom in the literature [Kingma & Welling, 2014].

During model inference, we compute the novelty score for each tile from a larger captured scene. Namely, for samples $x_{t-1}^{a,b}$ and $x_t^{a,b}$, we compute the latent representations and define the distance functions as one of the following: the Euclidean and cosine distance between the means $\mu_{t-1}^{a,b}$ and $\mu_t^{a,b}$, or the KL divergence between the Gaussian latents $\mathcal{N}_{t-1}^{a,b}$ and $\mathcal{N}_t^{a,b}$. Recall that we do so for each of the previous images in the k -long history window. This means that for change detection, we only need to retain the encoder network, which reduces the compute requirements of using the model. Furthermore, instead of storing the full previous images, we only need to store the selected latent representation.

Table 3.2: Differences in the architecture for different proposed model sizes (keeping latent size fixed on 128). Note that during inference, we only need the encoder network of the VAE model. We also only need to process the newly acquired image to obtain their latent representation, while the latent vectors of the previous image can be loaded.

	Total par. (M)	Encoder par. (M)	Extra depth	Hidden channels	Latent size
Small model	0.443	0.285	0	16, 32, 64	128
Medium model	0.979	0.617	0	32, 64, 128	128
Large model	1.463	1.007	2	32, 64, 128	128

To optimise the model architecture for low compute environments, we parametrize the number of layers used in the encoder and decoder network into configurations (small, medium, large) presented in Table 3.2. Each of the models has 3 downsampling blocks (2D convolutions with kernel size 3, stride 2, zero padding of 1) which halves the spatial resolution keeping the number of channels according to the “hidden channels” parameter. The extra depth adds sequence of extra 2D convolutional layers after each block with skip connections, building a residual block. Each convolution uses the leaky ReLU activation and is followed by batch normalisation. Later, fully connected layer is used to model the means and log variance, for which the target latent size of the bottleneck is set to 128 (we also explore different values for this hyperparameter). The decoder uses same layers as the encoder, but in reverse order, with nearest neighbour upsampling to avoid checkerboard artifacts [Odena *et al.*, 2016; Wang *et al.*, 2020]. Last layer reconstructs the input data range with a linear activation function.

3.5.4 Model adaptations through training on-board

What we would like to achieve here is some form of model adaptation after it has already been deployed and uplinked on the communication constrained device. Our research is also reinforced by the overview of [Lin *et al.*, 2023], which highlights the need for model training on EdgeML and TinyML devices - we focus on this task specifically in the context of satellites (as an example of communication and compute constrained device). This may be needed when the nature of the input data drifts over time, for example due to sensor degradation. Another example may be from practical considerations of using the change detection system of RaVAEn proposed in Section 3.5.3. In some cases, we would like to reject selected detections made by the system, perhaps by building a blacklist and a whitelist of events. If we want to detect events beyond the anomalous change scenario, this would lend itself more as a task of tile classification - one such example is cloud detection.

More concretely, we want to classify the tile $x^{a,b}$ into either cloudy or non-cloudy classes. Perhaps if enough tiles are detected as containing clouds in a single image, the entire data capture can be rejected. This classifier could also be used to mask out clouds directly on-board.

We can do this by further leveraging the trained encoder model beyond the task of change detection as is shown on Figure 3.7. We assume that the bottleneck feature contains discriminative description of the original input tile. As such, we can consider the encoder network as a learned dimensionality reduction function. Using the latent

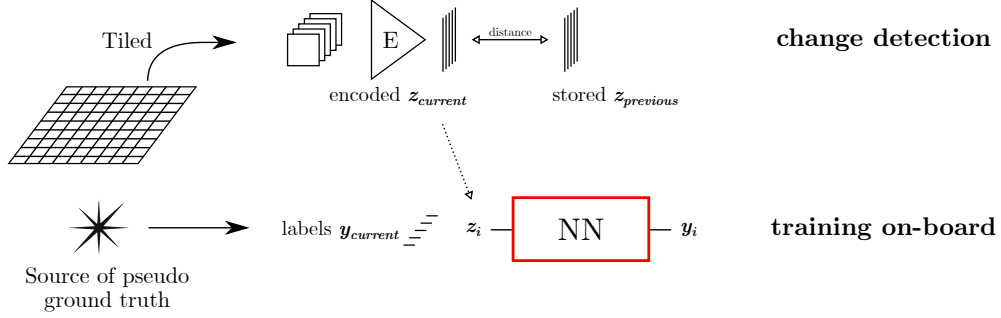


Figure 3.7: Timing of the model for tasks of change detection and few-shot learning on-board of satellites.

space of our pre-trained model, we can train new and extremely small classification models in a few-shot learning manner on-board of the device.

We note that for this, we will need a source of (pseudo) ground truth, labels for each of the observed tiles. We consider this challenge to be not trivial, but in our demonstration, we assume to have at least a small, labelled dataset of tiles, as has been also described in Section 3.4.2. We may be given these labels from another, more reliable instrument in a constellation of satellites, or by flying over a calibration zone.

Importantly, we want to achieve this task on-board of the satellite. This means using a frozen encoder model to extract latent representation from all $x^{a,b}$ tiles in the labelled dataset obtaining latents $\mu^{a,b}$. Using the known labels $y^{a,b}$, we want to train a small classification model.

3.6 Experimental Setup

For training of the VAE model we use the Adam optimizer [Kingma & Ba, 2014], learning rate of 0.001 and loss composed of data reconstruction and KL divergence terms [Kingma & Welling, 2014]. We use the batch size of 256 tiles (of 32x32 px and 10 bands) and train for a single epoch (we note that the large size of the dataset makes this enough - experiments with longer training didn't lead to better performance). We use the PyTorch and PyTorch Lightning libraries for training and our own custom code for evaluation. We deploy the models using ONNX format with PyTorch and OpenVino libraries. We train the models in mixed precision mode (allowing 16 bit floats).

3.6.1 Devices

We initially train our RaVAEn model using a n1-standard-16 instance on Google Cloud Platform (GCP) with two NVIDIA Tesla V100 GPUs. We then test the model inference speed on the Xilinx Pynq FPGA board using its 650 MHz ARM Cortex-A9 CPU and 512 MB RAM (the model is not encoded on the FPGA’s programmable array). This device serves as a proxy for a satellite environment, motivated by similar compute environments used by [Rapuano *et al.*, 2021; Gretok & George, 2021].

We also deploy the model on a real-world satellite ION-SCV 004 provided by the D-Orbit and the Unibap company. This satellite holds the 4-core 64-bit AMD GX-412HC SOS CPU running at 1.2GHz, Intel Movidius Myriad X VPU and 2GB of RAM.

We use the insight from measuring inference speed on the Xilinx board to inform our selection of model variants for the real satellite. Then we use the Myriad X VPU for fast model inference, and the CPU of the satellite for training a small classification model. This is due to the limitation that libraries running models on the Myriad chip do not allow for changes in the model weights. Further details are provided in the result section discussing the time measurements 3.7.2.

3.7 Results

3.7.1 Change detection on disaster events

We present qualitative comparison of results in Figure 3.8. There, we show the scene just before and just after the event of interest (flooding and wildfire in this case), note that the model was also given two more previous views (denoted as using the history window of $k=3$). Clouds detected in either of the last two images are masked in the predictions.

We compare the predicted change maps against the ground truth labels (these are available only in the evaluation set) using the AUPRC metric. We remark that this metric allows comparison without having to choose a threshold after which we would classify each tile as containing a change. The continuous values associated per each tile would allow us to for example sort the tiles and downlink only the most anomalous ones given a communication budget.

Table 3.3 compares our proposed model against the baselines testing different distance functions. Table 3.4 further explores the effect of adding a history of window of up to 3 previous samples and compares different baseline approaches previously

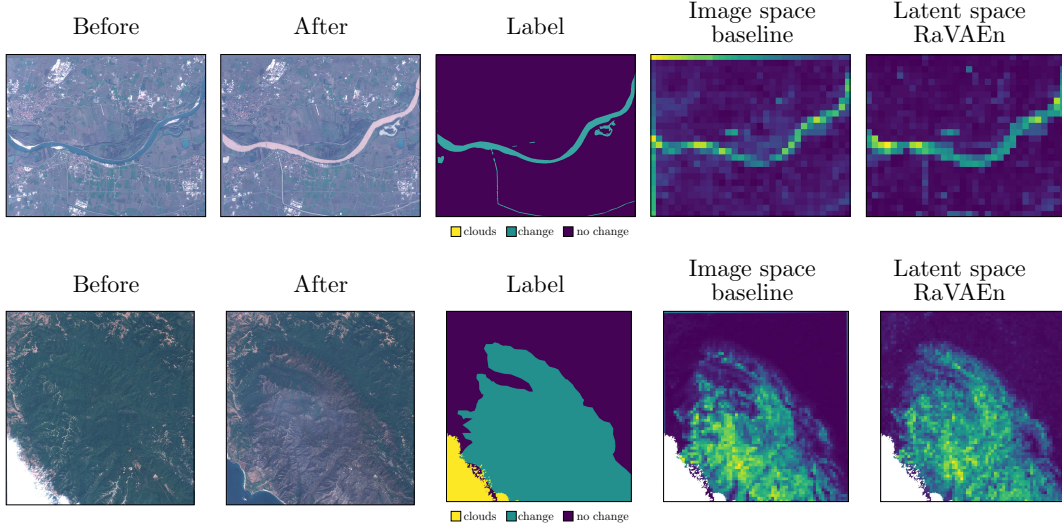


Figure 3.8: Comparison of changes detected using the baseline and the *large* VAE model (with latent size 128) on examples of flooding and fire disaster events. Both methods used a history of $k = 3$ frames. The image space baseline (using cosine distance) prediction seems to more closely copy the details present in the image, making it susceptible to small, noisy variations between the two images.

Table 3.3: AUPRC for baseline and VAE methods (here with the latent size of 128) with time window $k = 1$ (averaged over 5 training runs).

Detection method	Dataset			
	Landslides	Floods	Hurricanes	Fires
Cosine baseline	62.9	37.8	51.3	81.8
Euclidean baseline	26.7	32.6	35.1	77.0
Cosine embedding	59.9 ± 1.2	44.8 ± 1.1	67.6 ± 1.4	83.3 ± 0.8
Euclidean embedding	26.6 ± 0.4	45.0 ± 0.7	47.8 ± 1.9	80.0 ± 1.1
KL-Divergence	25.8 ± 2.2	24.7 ± 1.8	30.1 ± 3.5	73.1 ± 1.6

Table 3.4: AUPRC for the best performing metrics from Table 3.3 with and without extended history k (averaged over 5 training runs). The VAE model uses latent size of 128 and the default large configuration.

Detection method	k	Dataset			
		Landslides	Floods	Hurricanes	Fires
Cosine baseline	1	62.9	37.8	51.3	81.8
	3	62.2	37.8	57.0	86.5
PCA (i) (enc. dist.)	1	29.7	42.8	34.0	76.0
	3	33.7	49.3	41.1	87.0
PCA (ii) (diff. vec.)	1	31.7	64.0	52.7	65.4
	3	69.3	67.3	56.4	70.8
PCA (iii) (reconstr.)	1	20.7	61.1	54.0	74.5
	3	27.2	54.3	60.2	75.5
VAE, cos. embedding	1	59.9 $\pm$ 1.2	44.8 $\pm$ 1.1	67.6 $\pm$ 1.4	83.3 $\pm$ 0.8
	3	75.9 $\pm$ 2.4	44.3 $\pm$ 0.9	72.6 $\pm$ 1.1	91.3 $\pm$ 0.8

Table 3.5: AUPRC for models with different latent sizes (averaged over 5 runs). The AUPRC results are for the cosine similarity of the embedding with a history of 3 frames. Models use parameters for the default large model, but with variable latent sizes. We also show the total number of parameters of each model.

Latent size	Dataset				Params. total (M)
	Landslides	Floods	Hurricanes	Fires	
512	76.1 $\pm$ 1.0	43.8 $\pm$ 1.0	71.4 $\pm$ 0.7	90.9 $\pm$ 0.4	3.823
256	75.5 $\pm$ 2.1	44.0 $\pm$ 0.7	72.7 $\pm$ 0.7	91.1 $\pm$ 0.4	2.249
128	75.9 $\pm$ 2.4	44.3 $\pm$ 0.9	72.6 $\pm$ 1.1	91.3 $\pm$ 0.1	1.463
96	72.3 $\pm$ 0.5	41.9 $\pm$ 1.0	68.7 $\pm$ 3.4	90.5 $\pm$ 0.4	1.266
64	69.9 $\pm$ 1.5	39.2 $\pm$ 1.7	72.6 $\pm$ 2.2	90.3 $\pm$ 0.6	1.069

described in Section 3.5.2 against our proposed VAE based method. Table 3.5 explores different model configurations, namely varying the latent vector size hyperparameter.

3.7.2 Inference times

The model was tested on several platforms, the CPU of the Xilinx Pynq board and the Myriad X VPU of the ION-SCV 004 satellite. Table 3.6 provides an insight into the performance of three model variants, it also shows the runtime needed to encode a single 32x32 px tile on the CPU of Xilinx Pynq. We report average inference time from running on a larger scene of 574x509 px (approximately 5x5 km^2 at Sentinel-2 10m resolution) which was cropped into multiple of 32 and tiled as 255 tiles (we

Table 3.6: AUPRC and timings for different sizes of model (using latent size of 128; averaged over 5 runs). The AUPRC results are for the cosine similarity of the embedding with a history of 3 frames. Runtime is reported for an average single tile of 32x32 px on-board of the Xilinx Pynq board.

Config.	Dataset				Runtime (ms)
	Landslides	Floods	Hurricanes	Fires	
Small	74.8 $\pm$ 1.4	44.5 $\pm$ 1.4	74.8 $\pm$ 0.2	90.7 $\pm$ 0.2	8.08
Medium	75.8 $\pm$ 0.7	42.8 $\pm$ 0.4	73.8 $\pm$ 1.8	91.2 $\pm$ 0.3	19.06
Large	75.9 $\pm$ 2.4	44.3 $\pm$ 0.9	72.6 $\pm$ 1.1	91.3 $\pm$ 0.1	54.83

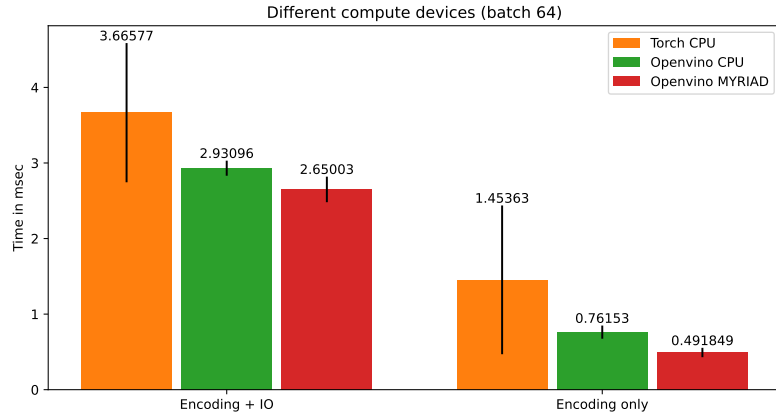


Figure 3.9: Timed model inferences, using the RaVAEn encoder network with different devices available on the satellite. We show the encoding time for a single tile of 32x32px, we measure these as averages from the total of 225 tiles while processing an entire 4.8x4.8 km^2 scene.

report average inference time for a single tile). During these measurements we report that at least 67% of the total RAM (512MB) was left available. We note that this measurement includes any background processes that ran alongside our code on the clean Pynq environment.

Figure 3.9 shows inference speed measures on the ION-SCV satellite. There we ran our model on two different processing units, the CPU and the Myriad X VPU, using two libraries (PyTorch and OpenVino). We used a dataset that was preloaded on the satellite prior to its launch, which contained Sentinel-2 images with only 4 bands (RGB+NIR). We note that the only difference between the models are in the first layer, which either projects 10 or 4 bands of data, the rest of the model architecture remains the same. As such, the influence on the inference speed is minimal.

Figure 3.10 shows measurements of individual batches used with the CPU and the VPU of the device - we observed random slowdowns when using the CPU of the device

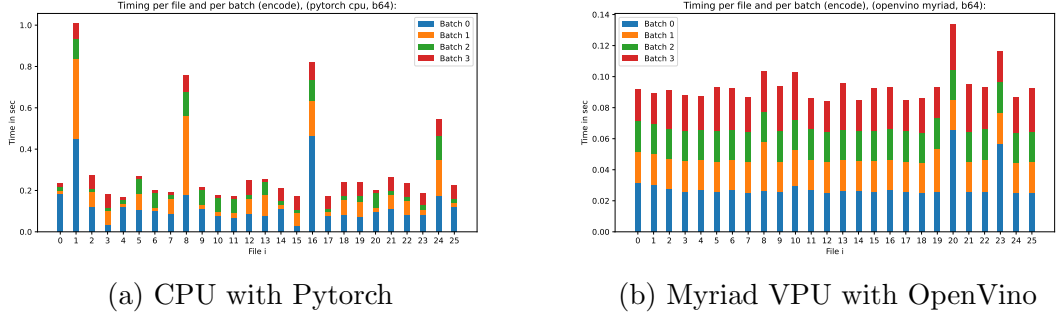


Figure 3.10: Detailed view into the individual batches used in the RaVAEn encoder, showing delays in the CPU regime. Batch size of 64 splits the 225 tiles into unequal batches (namely the last one has only 33 tiles). We show evaluation over 26 loaded scenes of the same resolution.

particularly with the PyTorch library, this is likely due to background processes on the satellite.

3.7.3 Latent space and compression

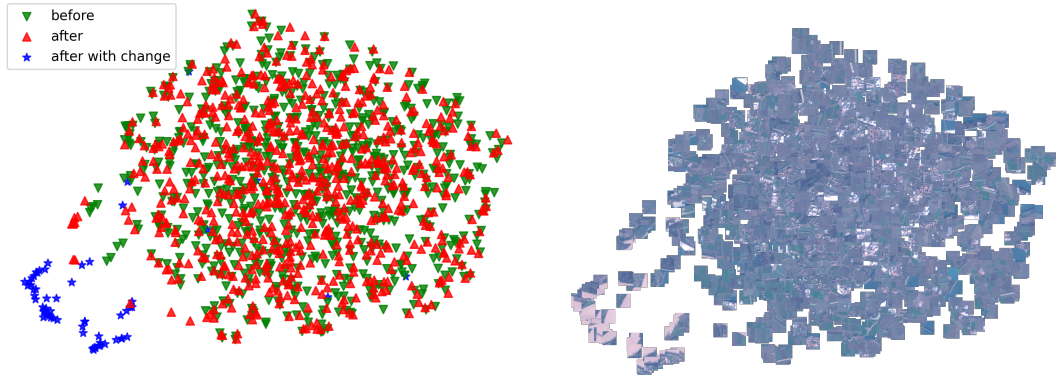


Figure 3.11: UMAP visualisation of encoded tiles from a flooded scene presented on Figure 3.8. Tiles from the image before the event are marked as green, while the tiles from after the event are shown in red. Tiles corresponding to the flooded tiles, marked with blue, can be seen clustered together in contrast to the rest of the data from this scene.

In Figure 3.11, we present the UMAP visualisation [McInnes *et al.*, 2018] of the latent space learned by the RaVAEn model. In this visualisation, we show embedded tiles from both before and after scenes, we also highlight the tiles which were labelled as containing change (flooded water in this case).

We note that the proposed system also provides a lossy compression from the original Sentinel-2 data. One tile of 32x32 pixels with 10 bands is originally stored with

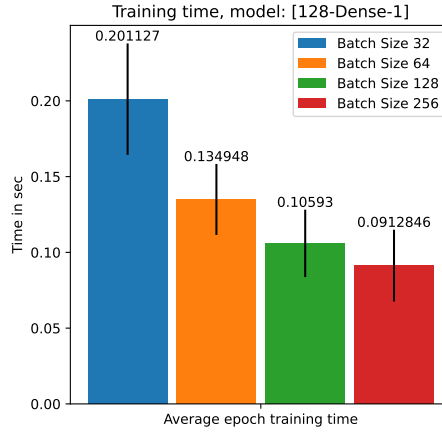


Figure 3.12: Training a classification model on top of the encoder network, timing while changing the used batch size. The full training dataset is composed of 1305 tiles.

12bit as per the Sentinel-2 radiometric resolution specification. Using the embedding with vector size of 128, we store the encoded mean vectors at 16bit float precision. This approach provides the users with 60x reduction of the necessary storage space for each of the saved sample. Further compression of the latents would be also feasible [Ballé *et al.*, 2016].

Existing satellite practices of lossless compression such as CCSDS-137 [CCSDS, 2019] achieve compression ratio of 6.5, and JPEG-LS [Schelkens *et al.*, 2009] ratio of 2.5. Our method is in contrast to these lossy, but the original tile can be reconstructed using the decoder network on the ground (again with a loss of quality). In fact, further research using auto-encoder based models for compression was shown in [Bégaint *et al.*, 2020]. For change detection applications or for communication across satellites in constellations, having just the latent vector is sufficient.

3.7.4 Training on-board

Figure 3.12 reports training time needed for a single epoch for a cloud classification model. This is an example of few-shot learning, as we first encode the dataset using the encoder networks with Myriad X VPU. Given the dataset size of 1305 tiles, this encoding takes further 0.64s sec. We observe that the classification models converges after 10 epochs. Prediction examples are shown on Figure 3.13.

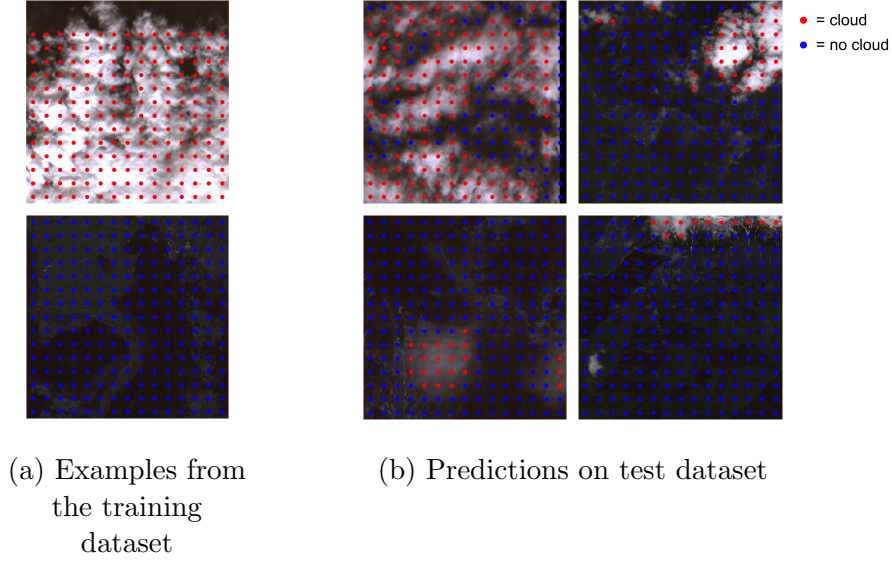


Figure 3.13: Predictions of the on-board trained model and visualisation of samples from the training dataset. Training dataset was created using scenes where almost all tiles contain one class type (tiles classified as cloud or no-cloud class).

3.8 Discussion

Results show that it is possible to detect disaster events with unsupervised change detection models on-board of the satellite. To interpret the results, we compare the performance of the proposed RaVAEn model with the pixel space comparison baselines. On Figure 3.8, we see the qualitative difference, where our model is able to detect smaller true changes (such as the small, flooded water channel on the bottom of the image), while being less sensitive to small noisy variations between the two images. Tables 3.3, 3.4 confirm this quantitatively, we see that our model outperforms the baseline on most tasks and that the inclusion of history window of $k=3$ further benefits our model. We see that multi-temporal information is helpful for unsupervised change detection.

Additional baselines depending on the PCA method tend to have lower performance than other approaches (except for the case of floods dataset), we also observed high sensitivity to the selection of the used hyperparameters (such as the number of included principal components and the way how we compare tiles before and after the event).

In Table 3.5, we explore different latent sizes used with the Large variant of the VAE model. It seems like there are only small gains in using values beyond 128 (all models above the latent size of 128 have their average results within each others

standard deviation window given by the five repeated trainings). Also considering the number of parameters of the created models, we prefer smaller models. We further observe that the reduction of latent size below 128 reduces the performance of the large model. In Table 3.6 we show that we are able to retain the model’s performance within $\pm 3\%$ while reducing the model size (by changing other parameters than the latent size) and the consequential runtime by 85%.

We use this insight to directly deploy only the “small” model variant on the ION-SCV satellite. We achieve the best inference speed when using the Myriad X VPU with the OpenVino library. This can be used for much faster change detection on-board of the satellite, or for other downstream tasks such as training on-board. Computing change detection for a single tile on the Myriad X takes just under 0.5 msec, while an entire scene of 4.8km^2 (with 225 tiles) takes about 0.11 seconds. We note that there is almost 16.5x times speedup when running the models on the Myriad X chip on the ION SCV satellite in contrast to the CPU of the Xilinx Pynq board, in both cases we tried to optimize the parameters for fast running on the specific hardware (such as selecting batch size).

We demonstrate that an entire training run would take only 0.64s to encode the dataset with Myriad X and OpenVino, and 0.91s to fully train a classification model using the satellite’s CPU and PyTorch. This means that in only about 1.55s in total, we were able to adapt a small classification model for a completely novel task - either to extend the capabilities of the change detection system (by for example blacklisting tiles with clouds), or to adapt to potentially changing quality of the input data (due to for example sensor degradation).

We show that our change detection system generalizes well to a large variety of disaster events, and that it can be deployed in many types of low compute devices.

3.9 Summary

In summary, we have proposed, tested and deployed a system that enables general anomalous change detection on-board of satellites. Doing so, it enables intelligent decision-making, such as sending alert signal, prioritization of anomalous samples for downlinking (potentially in their compressed form) and also on-board few-shot training. The RaVAEn system has been deployed in space and ran on-board a real satellite. It also demonstrated the world’s first training of machine learning model on-board using a combination of devices (Myriad X VPU and the device’s CPU) and libraries.

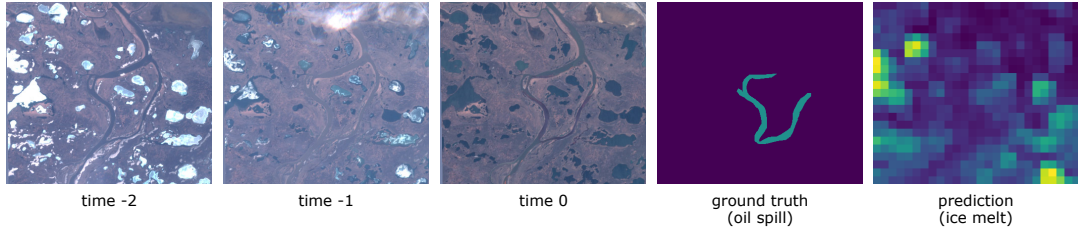


Figure 3.14: Example data containing two types of changes. While the official dataset annotation marks an oil spill, the prediction of our model detects the natural ice melting instead.

Our research has identified several limitations which can be addressed in future research. Firstly, hardware available on-board of low compute devices such as satellites vastly differs given the mission requirements. As such, even more efficient hardware needs to be considered, for example with deployment of the model directly on the programmable field of FPGAs, or by leveraging GPUs (these are however not as robust to space radiation and as such are not widely used).

Secondly, more modern machine learning models can be leveraged for the task of unsupervised change detection. These include improved variants of auto-encoder models such as MAE [He *et al.*, 2022], VQ-VAE [Van Den Oord *et al.*, 2017] or domain specific designs such as ScaleMAE [Reed *et al.*, 2023]. Methods for contrastive learning such as SimCLR [Chen *et al.*, 2020] or models specifically proposed in the domain of Remote Sensing such as SeCo [Manas *et al.*, 2021] can also be used.

In fact, our original paper [Růžička *et al.*, 2022] already inspired several follow-up works exploring these directions. The conference paper of [Zhou *et al.*, 2024] deploys the RaVAEn model on a FPGA chip and encounters further speedups over our results on the Myriad X chip. The Master thesis of [Herec, 2024] adds finetuning round inspired by contrastive learning to our pre-trained RaVAEn model and further improves its performance. Finally, in our own work [Zhang *et al.*, 2022], we propose a SimCLR inspired training of an unsupervised change detection model on the specific domain of wildfire detection, in a scenario, where we lacked large, labelled datasets.

In Figure 3.14 we show an interesting encountered sample, where our proposed model detects only one of the two occurred changes (ice melting leaving a stronger change signature than oil spill), this motives a potential future direction of research in semi-supervised learning to create whitelisting and blacklisting filters.

Work on wildfire detection on-board of satellites was also implemented in the private sector by the work of [Gottfriedsen *et al.*, 2023] - we note that the details about the model are however not published.

To aid research in this area, we have released all our data, code and the weights of the trained machine learning models publicly^{1,2}.

Finally, given recent focus on general Foundation models in the domain of Computer Vision [Yang *et al.*, 2023] and applied to Remote Sensing tasks [Lu *et al.*, 2024], we see our extremely small RaVAEn model as a potential path forward in contrast to the very large and demanding Foundation models. We achieve general behaviour and can use the model across a wide range of downstream tasks, despite aiming at small and practically deployable model sizes.

¹<https://github.com/spaceml-org/RaVAEn>

²<https://github.com/previtus/RaVAEn-unibap-dorbit>

4

Methane Detection

4.1 Introduction

In this chapter we explore the research question of **detecting a faint signal on-board of a satellite using different sensors (Q2)**, in particular we focus on a single concrete application of methane leak detection using classical and machine learning methods. We pay attention to the specific requirements this task dictates in terms of the types of used sensors and consequentially the types of data we process. We frame this problem as semantic segmentation and train machine learning models with hyperspectral and simulated multispectral data on datasets we create for this challenge.

Detection of methane leaks in the atmosphere is an important, yet challenging task. After carbon dioxide (CO_2), methane (CH_4) is the second largest contributor to global warming through its effects on radiative forcing [Kuylenstierna *et al.*, 2021]. Furthermore, methane has a much shorter atmospheric lifetime than CO_2 , so any reduction of unwanted emissions can have tangible and fast effect. In fact this task has been called the low hanging fruit of climate action. Methane itself is a greenhouse gas which is transparent in the visible spectrum - as such, specialised instruments are needed to detect it. Different sensors and satellites provide observations with different sensitivities to signals of methane presence in the scene, as well as vastly different ground resolutions. While satellites such as the Sentinel-5P (TROPOMI) can detect methane at ground resolution of only $7 \times 3.5 \text{ km}$ per pixel, sensors such as NASA's EMIT observe methane at resolutions of 60 m per pixel. Current aerial missions provide even finer sampling of up to $3\text{-}10 \text{ m}$ per pixel such as the NASA's AVIRIS-NG mission. Broadly speaking, we can differentiate between low ground resolution

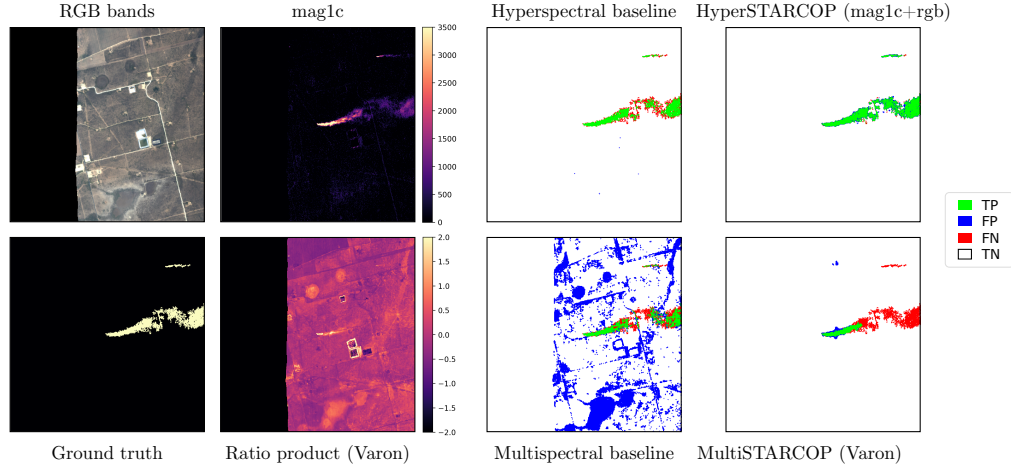


Figure 4.1: Predictions of the STARCOP models on example plume event from the AVIRIS-NG data. On the left, we show some of the inputs used by our models and by the baselines and, on the right, we show the comparison of the predictions with the ground truth labels. It can be seen, that the major issue with the existing baselines is the high false positive ratio.

instruments which can be used to provide global picture about methane distribution over the whole Earth, and mid to high ground resolution instruments which can be used for point source detection of individual methane leak plumes. The later allows for attribution of methane leaks to individual facilities and for more precise follow-up quantification. Furthermore, the number of spectral bands and their wavelengths heavily influence the detection capabilities of these instruments.

There are several limitations of the currently used approaches for the task of methane detection. Firstly, the currently used classical methods (these create what we call methane enhancement products) tend to be noisy and produce many false positives. This is due to the presence of confounder materials which appear to trigger these classical methods in similar ways as methane. As such, these products typically have to be monitored manually. However, with the increased number of satellites capable of seeing methane, manual detection will no longer be feasible and methods for automation will be needed. Secondly, there aren't many large, qualitatively labelled datasets and research in this field often relies on private data and the reported models are often kept proprietary. Finally, data driven exploration of detection limits using different instruments doesn't exist globally - detections are reported anecdotally, and existing methods have large amount of uncertainty associated with their detections.

Concretely, in this chapter we propose a machine learning models to automate the detection of methane leaks in hyperspectral and simulated multispectral data. We

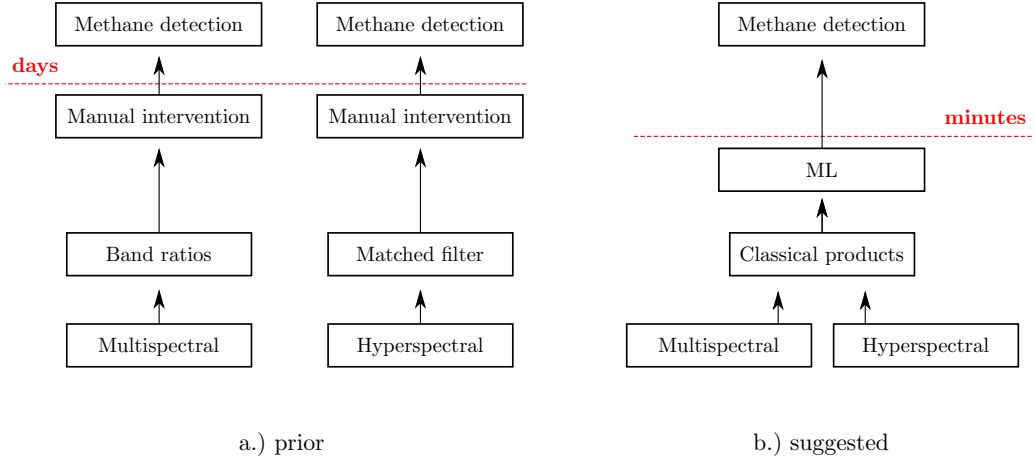


Figure 4.2: Illustration of the automated methane detection using machine learning models and the unification of processing pipelines used with different sensors.

improve upon classically used methods by significantly reducing their false positive rate. We study existing data portals, collect a dataset of methane leak events and manually improve initial coarse labels. Finally, we release all our trained models and created datasets publicly in a machine learning ready format. We present detections in different modalities of data as a demonstration of a tip-and-cue system, and we explore generalisation of our models across different hyperspectral sensors. Figure 4.1 shows an example scene with a strong methane leak event and illustrates the comparison of the detection limits between models using multispectral and hyperspectral data.

We consider this research significant as it could allow for automation of methane leak detections in real-world data. Figure 4.2 illustrates the potential difference that automated methane leak detection enables in terms of the reduced time and manual labour needed and the simplification of existing methodologies. Detections made by our system could be verified and used by experts (reducing their labour by filtering our vast quantities of empty data) which could then lead to fixing of previously unattended leaks. For example, manual detection made in the work of [Duren *et al.*, 2019] have led to fixing methane leaks in only few days after the detection - automated detections could make this even faster and could allow larger scale. Our work aims to directly address the needs outlined by several governmental agencies and independent organisations, for example in the Global Methane Assessment report made by United Nations in 2021 [Kuylenstierna *et al.*, 2021], the NASA’s decadal survey for 2017-2027 [NASEM, 2019] and finally also the IPCC report of 2023 [Lee *et al.*, 2023]. While our work contributes to only a small part of the problems outlined there, it remains to be of high importance.

4.2 Background

4.2.1 Prior methods

Methods and data products used for detection of methane leaks in satellite data are more closely explored in Section 2.4. Here we outline the most relevant prior methods used in this field which are used throughout this chapter.

Methane is a greenhouse gas, which is weakly visible in wavelengths between 1600-1800nm and strongly in the region between 2100-2500nm. Given the low overlap of the methane signal with most of the spectral bands typically used in most multispectral satellites, augmentation methods such as the so-called band ratios need to be used to enhance visibility. A single spectral band with known methane visibility is divided by a nearby spectral band without such visibility. It is assumed that nearby bands will have similar view of the background information (therefore filtering out this information in the ratio product), while the methane signal is present in only one of them (therefore remaining in the product). The work of [Varon *et al.*, 2021] proposes two methods for computing band ratios - either with a single or multiple temporal captures of the same scene. Other methods such as [Sánchez-García *et al.*, 2022] use multiple bands outside of the methane visibility to estimate background of the band with visible methane signal. The subsequent ratios should better differentiate between true methane signature and other confounder materials.

Hyperspectral data, which is also referred to as imaging spectroscopy data, gives better view of methane signal than data from multispectral sensors. This is given by their improved overlap with the methane signal. The so-called matched filter uses the knowledge of methane signature over individual spectral bands (measured in laboratory by [Rothman *et al.*, 2013]) and searches for it in the hyperspectral datacubes. By going one-by-one for each pixel and subtracting estimation of the background information, we can compare correlation with this known target signature. Improved versions of matched filter such as `mag1c` [Foote *et al.*, 2020] are often used to iteratively reduce the false positive detections in these products.

However, these classical methods contain quite a lot of noise and need to be analysed manually by experts. This slows down any potential detection pipeline used for observing methane in large data captures (for example in daily observations produced by satellites).

4.2.2 Challenges and opportunities

The first challenge faced in this domain is the lack of large, annotated datasets, that could be used to train models. This has been observed by many overviews [Signoroni *et al.*, 2019; Paoletti *et al.*, 2019; Gewali *et al.*, 2018] and the opinion piece given by [Thompson & Brodrick, 2021]. This extends to machine learning research applied to processing hyperspectral data, which uses only very small datasets such as Indian Pines [Baumgardner *et al.*, 2015] (and other similar datasets with only few scenes). To make matters worse, the similarity between the used data splits in these datasets leads to very severe model overfitting. Consequentially the proposed methods cannot really generalise well enough to be used globally. This makes it difficult for comparison between methods proposed in this field. Many works also use private datasets, and they do not release the code of their published research. At the publication date of [Růžička *et al.*, 2023b], there were no usable large datasets in this domain prior to our work, only raw data that needed processing, or tiny datasets. One of the contributions of this chapter is the release of a large, high-quality dataset which can be used to benchmark models in this area. While there are some recent datasets released in this area, ours remains to be the best one to use (due to the quality of the fixed manual labels).

The second challenge is in the high false positive ratio and the noise the classical methods produce. This prevents real implementation of these methods beyond pipelines that depend on manual detection by experts. This in turn slows down any detections and cannot really scale with the upcoming increase of satellites used for detecting methane. In our work we show that machine learning models provide pathways of reducing the false positive ratio and that they could be used for automated detection of methane leaks.

4.3 Method

In this chapter, we propose machine learning models based on the U-Net architecture for the task of methane detection, which we formalise as semantic segmentation. We create a new dataset of hyperspectral images with improved manual annotations, and we also create simulated multispectral views of this data. On this dataset we compare machine learning approaches against several classical baselines.

In Section 4.4 we describe the creation of the used datasets and the simulation of multispectral views of source hyperspectral data. In Section 4.5 we present the used models and classical baselines for both data modalities. And finally, Section 4.6

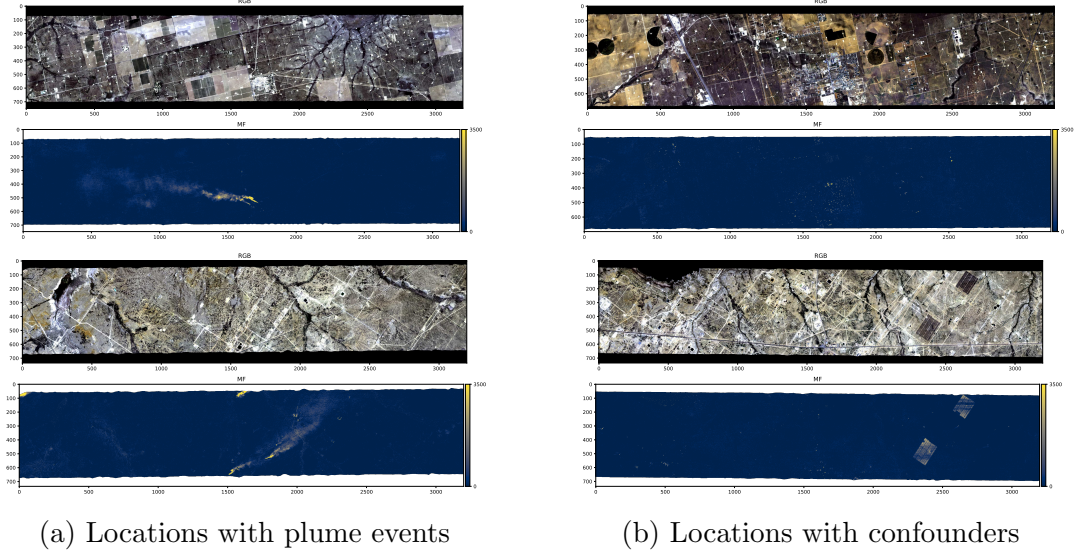


Figure 4.3: Examples of the AVIRIS-NG data captures. We have zoomed-in onto two samples of clear plume events and two cases with confounders (man-made infrastructures and solar panels). We note that the flight paths were rotated for easier visualisation. We show the methane enhancement product created with the mag1c matched filter [Foote *et al.*, 2020].

presents the used training hyperparameters and our proposed matched filter weighted loss function.

4.4 Data

4.4.1 Hyperspectral dataset STARCOP

To train any machine learning models, we need to collect a large a diverse dataset of samples with enough positive events (in our case methane plumes) present in the data. For long time, this kind of data was missing and most of research in methane detection discussed only few manually detected events. More recently, the data from the aerially flown AVIRIS-NG sensor created a large enough archive in order to train machine learning models. It has been repeatedly deployed across regions with known methane emissions, including test releases of methane to calibrate the used instruments.

Figure 4.3 shows example events from the original AVIRIS-NG data alongside the computed matched filter product. Note that the data was collected during an aerial campaign and that the flight path is visible (as the collected strip of data follows the flight).

We use the data from the 2019 campaign conducted by [Cusworth *et al.*, 2021],

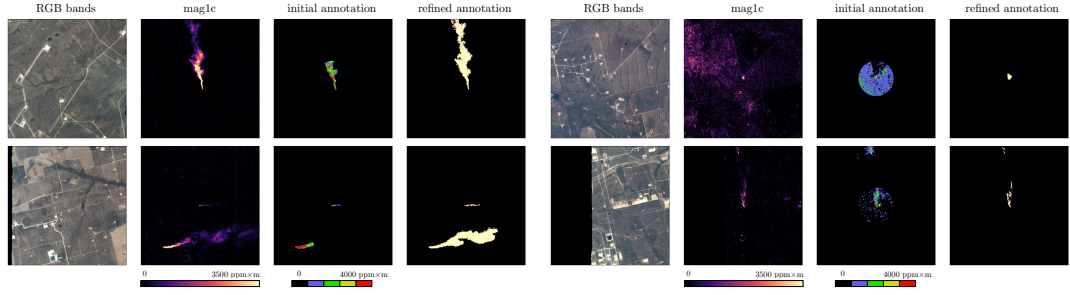


Figure 4.4: Showing the limitations of the available annotation in the original labels released by [Cusworth *et al.*, 2021], with a circular mask cut out around the center of each plume. We manually refine and extend these plumes using the mag1c product as guidance.

which contains up to 2069 individual methane plumes in 564 images retrieved during 31 different days. These initial labels also contain an estimated quantification for each plume (emission rate in kg/h). Out of the two instruments used during this campaign, the AVIRIS-NG and the GAO sensor, only the AVIRIS-NG data was made publicly available. As such we use only the public subset of the data with total of 1878 plume events observed in 300 separate granules from flights - total size of these in the L1B level of processing takes about 6TB to download. Here, we will describe the steps taken to create the STARCOP dataset used in this chapter and in our paper [Růžička *et al.*, 2023b].

Firstly, we explored the initial manual annotation made for this data by [Cusworth *et al.*, 2021] - we observed several limitations which are illustrated on Figure 4.4. Each annotation is cropped by a circular mask with a diameter of 151 pixels. As we wanted to extract larger tiles from the original data, and as some methane plumes were larger than these initial labels, we had to extend and improve these labels.

Secondly, we downloaded all hyperspectral granules of the source data (approximately 6TB) and computed the matched filter product mag1c [Foote *et al.*, 2020] for each separate one. We used this product to semi-automatically extend the original labels using thresholding and connected areas of active pixels. We have manually inspected all plume events and refined the labels (we manually redrawn the labels in the graphical interface of the IRIS tool [Mrziglod & Francis, 2019]). For each event we considered 512x512 px tile (the potential plume being in the center of the tile - for this reason it is important to use methods such as cropping of smaller tiles and shifting to break this regularity). In order to balance the created dataset, we sampled the same number of no-plume tiles as the number of plume tiles. For negative sample selection, one half was selected randomly, while the second half was chosen

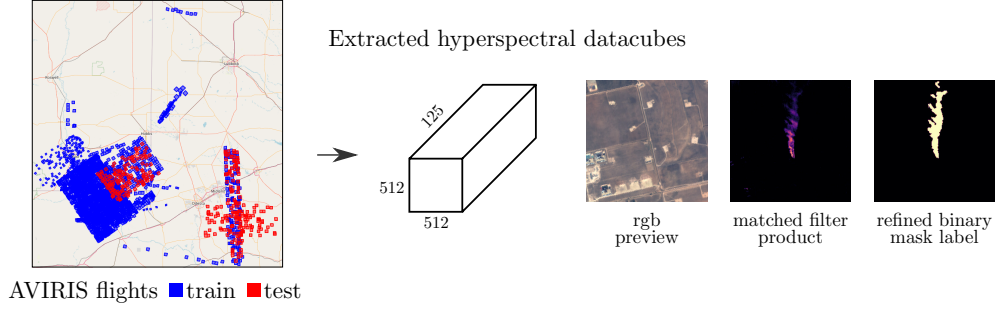


Figure 4.5: Hyperspectral dataset made from the AVIRIS-NG sensor, we primarily store the RGB bands, the matched filter product and the binary label, we also select 125 bands to save separately for future models. We note that additional views of the data may also be made from the original hyperspectral datacubes.

Subset	Flights	Days	Number of tiles by emission rate (kg/h)					
			0	10-200	200-500	500-1000	1000-3000	3000+
■ train	256	19	1713	352	708	379	247	26
■ test	44	3	176	13	56	40	47	10

Table 4.1: Distribution of the tiles stratified by emission rate into train and test sets.

using mag1c product as an indicator of locations with high amount of confounders (locations where the matched filter output was high, but where there were no plumes in the checked labels).

Finally, for creation of the train and test splits, we manually selected acquisitions from three days at the end of the campaign (18th, 21st and 25th October) into the test set. These samples do not have temporal overlap with the rest of the data which is then kept in the train dataset.

Particular care was taken to avoid data overlaps between these subsets. In this project, we chose to use temporally non-overlapping source images. The extracted larger tiles (512x512 px) therefore also don’t introduce any informational overlaps between the created subsets.

Figure 4.5 and Table 4.1 shows the spatial locations of train and test tiles and the statistics of the plume leak events stratified by emission rate. As can be seen, efforts were made to maintain diverse representation in both subsets. Later for evaluation, we divide these tiles into two broad categories: “strong” plumes with more than 1000 plume pixels in the label and the rest of the events as “weak” plumes (this does roughly correlate to the size of the plume, another potential division would be using the kg/h estimation).

In total the training dataset contains 1712 tiles with plumes and 1713 tiles without plumes. The test dataset has 166 plume tiles and 176 tiles without plumes. The final dataset size, containing all pre-computed products is about 60GB. We keep these as 512x512 tiles centred around methane leak events, and further sub-tile them during training using the tile size of 128 px with 64 px overlap. This creates a training dataset of 167825 images, each with all the required intermediate feature products.

4.4.2 Multispectral views of the data

To explore the performance of machine learning models on different modalities of data, we simulate multispectral views from the source hyperspectral datacubes. We note that we do not simulate instrument specific noise profiles and as such this data works only as a proxy of the real data capture. However, this allows us to reuse the same high quality and manually annotated labels with products from other instruments and sensors.

We convolve the available hyperspectral bands with the spectral response function (SRF) of two selected sensors - from the WorldView-3 and the Sentinel-2 satellites. We further convert the radiance values of top-of-atmosphere (TOA) reflectance using the acquisition date, centre location coordinates and the solar irradiance. For WorldView-3, we obtain the solar irradiance of each band by convolving the SRF of the sensor with the Thuillier solar spectrum [Thuillier *et al.*, 2003]. We also repeat the process with Sentinel-2 sensor as the target, but we couldn't observe even the strongest events in the classically computed methane enhancement products. We note that here we were limited by having only a single temporal view of the data and that in practice, multitemporal data could be used with better performance [Varon *et al.*, 2021; Vaughan *et al.*, 2024b]. Namely, for WorldView-3 satellite we simulate bands from SWIR1 as B1 (with central wavelength of 1210nm) to SWIR8 as B8 (central wavelength of 2330nm).

4.4.3 Data from EMIT sensor

To explore generalisation abilities of the models proposed in this chapter, we also collected few samples from another hyperspectral instrument by NASA, the EMIT sensor. We compute the same matched filter products from both sensors. We provide more details about the sensors in Section 2.2.1.

Despite having similar instrument design, the AVIRIS-NG and the EMIT data differs on several important fronts. Firstly, the AVIRIS-NG sensor is deployed aurally,

and is flown over locations with known or expected emissions. In contrast, the EMIT sensor is deployed on the International Space Station (ISS) and as such the ground resolution of the two instruments differs significantly. While the AVIRIS-NG sensor has about 3-10m per pixel, the EMIT sensor has only 60m per pixel. The influence of atmospheric effects is also more prominent in the EMIT data. Additionally, AVIRIS-NG has in total 425 spectral bands at 5nm spectral resolution, while EMIT has only 285 bands at 7.4nm spectral resolution. Finally, AVIRIS-NG is mostly local (relatively speaking, with the currently considered data spanning the Four Corners area of the USA), while the EMIT sensor is one of the first hyperspectral instrument to provide a near global coverage of the Earth without relying on manual scheduling. Furthermore, while the EMIT mission started observing only the arid regions of the planet, it has been extended to capture images globally. The ISS is however not as predictable as a satellite and it also observes locations at different time of day conditions.

During the work on this chapter’s paper [Růžička *et al.*, 2023b], only limited number of EMIT captures were available with flagged methane leak events. We have manually selected 11 full scene granules with known methane leak plumes and 9 scenes in urban areas (which are known producers of confounder signals) without any plumes. We have manually labelled these events. For these scenes, we use the L1B level of data processing without applying orthorectification.

4.5 Models

As we discussed in the previous section, we prepared datasets of methane leak events as hyperspectral and simulated multispectral views of the data. This allows us to compare the performance of machine learning models proposed to work with either modality, and to also compare them against the classical methods. Here we first describe the feature extraction methods which convert the raw satellite data into what is often referred to as methane enhancement products. We proposed baselines which threshold these products, or machine learning models, which instead use combinations of these products as inputs. In this way, we can consider the models proposed in these sections as examples of classical machine learning. There, a classical feature extraction process is followed with a machine learning model, in contrast to deep machine learning which would learn these features from the data in an end-to-end manner.

4.5.1 Feature extraction from multispectral data

Given the weaker overlap of many multispectral instruments with the regions of spectrum where methane is visible (this is described in more detail in section 2.4), we typically have only few bands that can “see” methane. Furthermore, these bands are typically spectrally quite wide, and as such they will contain activations from other materials. Isolating these from the real methane plumes is not a trivial task. Therefore, we typically compare two bands, one within the methane absorption (referred to as signal band S) and the other one outside to serve as a background reference (we will refer to it as background band B). Comparing these bands we get the so-called band ratio products.

In this chapter we will implement two band ratio products from the recent literature. First one, proposed by [Varon *et al.*, 2021] we refer to as Varon ratio, and second one proposed by [Sánchez-García *et al.*, 2022] we refer to as Sanchez ratio. In practice, while these methods were published with specific multispectral instruments, they can be used across different sensors.

The work of [Varon *et al.*, 2021] explores mono-temporal and multi-temporal methane enhancement products, we use the so-called “multi-band-single-pass” (MBSP) variant. Given a signal band S and a background band B we define:

$$VaronRatio(S, B) = (c * S - B) / B \quad (4.1)$$

where the parameter c is used to scale one band into the range of the other band. It can be obtained using least square fit, or as a simplified formula $c = \text{sum}(B') / \text{sum}(S')$, where S' and B' are signal and background bands with removed outlier values (using data between lower quartile of 5th percentile and higher quartile of 95th percentile). We use the notation of symbol $A \leftrightarrow B$ as Varon ratio between signal band A and background band B . We could use different bands as signal (for example B7 or B8 of the simulated WorldView-3 data) and as background (for example bands B5 or B6 of the WorldView-3 data). As the default Varon ratio we use $B7 \leftrightarrow B5$.

The work of [Sánchez-García *et al.*, 2022] instead uses multiple linear regression (MLR) to first estimate the background information in the signal band S from a combination of several background bands B_i . The combined estimated band S_{MLR} is then compared with the signal band S using:

$$SanchezRatio(S) = VaronRatio(S, S_{MLR}) \quad (4.2)$$

where we note that the MLR estimation is calculated separately for each tile. In the SWIR region, we consider either the band B7 or B8 as the signal bands and

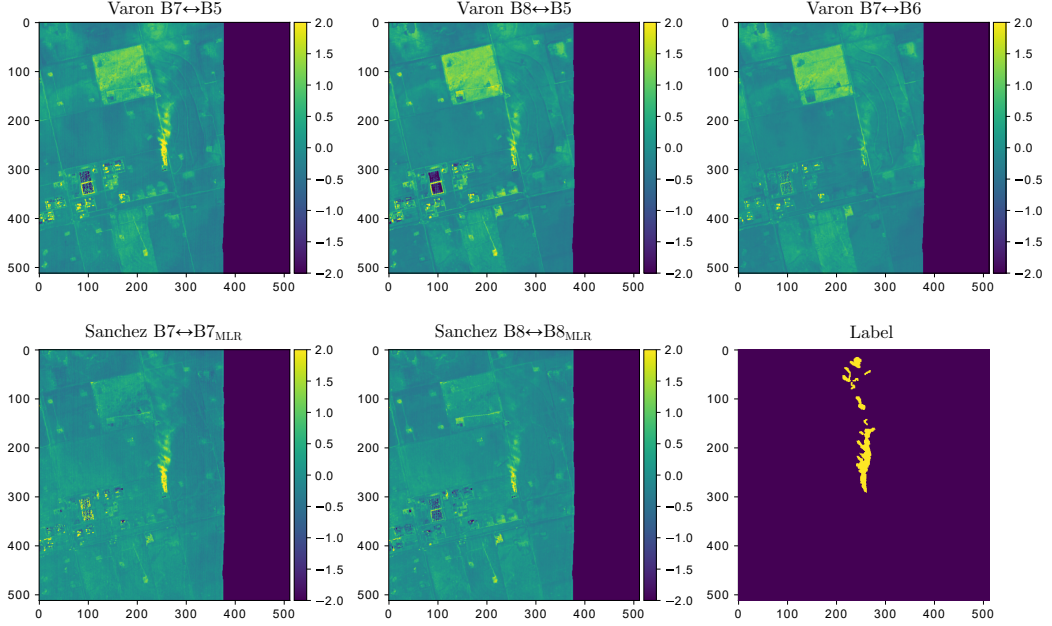


Figure 4.6: Visualisation of band ratios used to highlight methane plume. We remark that confounder materials in the background of the image seem to trigger the band ratio products as well. We show normalised and clamped products (between -2 and $+2$) which are later used as inputs to machine learning models.

combination of B1-2 and B4-6 as the background bands when computing the MLR. We use the notation of $A \leftrightarrow A_{MLR}$ as Sanchez ratio between signal band A and the MLR estimation of this band from other background bands.

We use these calculated band ratios as inputs to the machine learning models, and also as baseline classical methods. For the multispectral baseline we primarily use the Sanchez ratio between B8 and the $B8_{MLR}$, which we threshold using the experimentally found value of 0.05. The binary output is then post-processed with the opening morphological operation (to reduce the speckle noise or the salt and pepper effect of the isolated active pixels). We note that we tested all of the potential ratio products and their different configurations with different threshold values, but they all remain very noisy regardless of the selected variant. We show an example of the computed band ratio products in Figure 4.6 for a strong methane plume event.

4.5.2 Feature extraction from hyperspectral data

Hyperspectral data provides more information about the methane presence than limited multispectral views (this is in more details described in Section 2.4). As such, we can use more complex methods than simple band ratios. Given the known shape of

the methane signal across different spectral wavelengths modelled from the HITRAN 2012 database [Rothman *et al.*, 2013], we can search for this signature directly inside the hyperspectral data.

Matched filter (MF) is a method which for each hyperspectral pixel measures the similarity between the pixel value minus the estimated background value against the known methane signature. In missions such as AVIRIS-NG we consider the average value of the corresponding column of pixels as the background for each single pixel. Other pixel neighbourhoods could also be considered, but the column sampling is usually preferred due to the push-broom scanning of the hyperspectral instrument. Note, that we ideally try to select the same pixels of the push-broom sensor, this can be done before orthorectification, or by using the backwards pointing geometric lookup information.

Matched filter can be described with the following formula:

$$MF = \frac{(L - \mu)^T \Sigma^{-1} t}{t^T \Sigma^{-1} t} \quad (4.3)$$

where L refers to the vector of values for each processed pixel (1 hyperspectral pixel with the dimensionality given the number of selected hyperspectral bands), μ is the background estimation (typically computed as an average of pixels in a square neighborhood or a column of pixels from the same sensor element) and finally t is the target signature (in this case of methane).

On its own MF products are also quite noisy and sensitive to the presence of confounder materials, which is why iterative matched filter approaches such as for example mag1c [Foote *et al.*, 2020] try to improve the method by adding sparsity regularization and albedo correction to the target matched spectrum and estimated background. Although mag1c significantly reduces false positives, the computed methane enhancement product still contains high amount of false detections. As illustrated on Figure 4.3, we have observed this specifically in urban areas (rooftops), water bodies and human made infrastructures (such as solar panels or roads).

We will use the computed mag1c MF as an input to our trained hyperspectral machine learning models. We also use it as a hyperspectral baseline method, after applying a threshold of $500 \text{ ppm} \times \text{m}$ and an opening morphological operation (again to remove speckle noise). We further note that a variant of the MF products is usually shown to the manual annotators in pipelines currently employed by the industry and research labs. We are therefore trying to explore if machine learning models can improve these classical MF approaches.

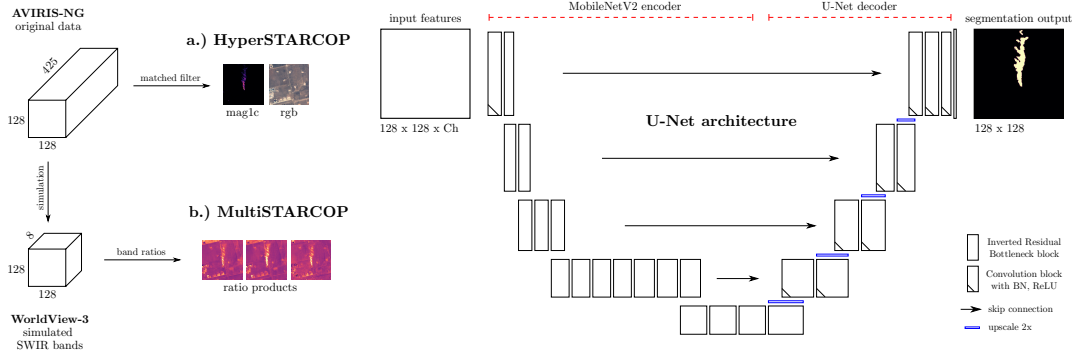


Figure 4.7: The proposed HyperSTARCOP and MultiSTARCOP machine learning models based on the U-Net architecture with MobileNetV2 as its encoder network.

4.5.3 Machine learning models

We propose two machine learning models illustrated on Figure 4.7 working on the two modalities of data (multispectral and hyperspectral). These models share the same proposed architecture and most of the used training hyperparameters. Both use a combination of the classical methane enhancement products, but they differ in the type of the used source data and the number of used input bands. We chose the U-Net architecture [Ronneberger *et al.*, 2015] with the MobileNet-v2 encoder [Sandler *et al.*, 2018] given its widespread usage in research in remote sensing applications.

The multispectral variant, MultiSTARCOP, uses several configurations of the Varon and Sanchez band ratios computed from the simulated WorldView-3 data. These products typically contain quite noisy view of the methane plume mixed with the background information. Large pixel values in these products could mean both the presence of methane, but also the presence of confounder materials (due to the width of multispectral bands, these can be many). Clearly, a single threshold values isn't enough to separate these two classes. In this case, our proposed model has to learn to differentiate between methane and the background using the shape and the strength of the signal.

The hyperspectral variant, HyperSTARCOP, uses the matched filter as one of its inputs, in addition to few selected bands (to provide a better view of the background information of the scene). With this model, we aim to improve the detections of the matched filter product. We assume that the model can learn, which of the pixels outlined by mag1c are true methane plumes, and which ones are just confounders. As MF methods operate on pixel-by-pixel basis, we believe that our model could improve differentiation by considering the spatial context in the tile and the shape of

the plume. We test two variants of the model, one uses only the mag1c MF product, while the other adds the RGB bands from the AVIRIS-NG data (namely the bands with central wavelengths at 640nm, 550nm and 460nm). The assumption is, that if a human expert can use the RGB information to detect a potential confounder material (such as a roof of a house), so should be able our trained model.

As a machine learning baseline, we also use a simple 5-layer convolutional neural network (CNN) for the hyperspectral scenario. This model serves as an exploration of if models simpler than the U-Net architecture could be leveraged (we are interested in this, namely due to their decreased number of parameters and the correspondingly faster training and inference times). Namely, this model has 4 convolutional layers with kernel size of 3 (with the number of channels increased from the input dimension of 4 to 64 and eventually to 128) each followed by a ReLU activation layer. Finally, the model uses a single convolutional layer with kernel size of 1 to match the number of the output bands (in this case 1).

In general, we use several simple rules to convert the semantic segmentation output of any of these listed models into classification per a whole considered tile. We note that more advanced methods (including training classification models) could be used. We threshold the output of the model by the value of 0.5 to obtain a binary output map. If the sum of the active pixels is larger than 10, we consider this tile to contain a plume. We will use this conversion when evaluating the performance of our model on per tile classification scores.

4.6 Experimental Setup

Both variants of the models (multispectral and hyperspectral) are trained with similar settings, however there are few differences we experimentally confirmed as providing better performance. While the HyperSTARCOP model is trained from scratch (randomly initialised weights), we initialise the encoder network of the MultiSTARCOP with weights pre-trained on the ImageNet dataset.

For all training runs, we use the Adam optimizer with learning rate of 0.001. We use a scheduler which halves the learning rate on plateau (after 4 epochs of patience). In total we train for 15 epochs. We use a batch size of 32 tiles, each with 128x128px (sampled with 64px overlap) and a variable number of bands (from one up to four, depending on the used inputs).

Despite selecting the same number of tiles with and without plumes, we are dealing with unbalanced data at least in terms of the number of pixels for each of these classes

(methane plumes tend to be small in the context of the entire tile). For both models we assign each extracted tile in the dataset a weight given the inverse of the fraction of plume pixels in said tile. Then we use the `WeightedRandomSampler` provided by the PyTorch library, which effectively balances the distribution of the data. When training the MultiSTARCOP model, we also use the weighted binary cross-entropy loss with plume weight of 15.

For the HyperSTARCOP model, we instead consider a novel training loss for the context of methane plume detection with hyperspectral data. We compute the binary cross-entropy loss for each tile, but before computing the average over all pixels in one tile, we weight the loss values by a prepared product derived from the matched filter. This approach is similar to the steps taken in the U-Net paper [Ronneberger *et al.*, 2015], where regions at the boundaries between two segmented objects were weighted higher as to improve the delineation of the detections. In our case, we use the matched filter product, which we scale between 0.1 and 1 (first dividing it by 400 and then clipping it into this range) as the weight for each pixel. This means that areas with high values in the matched filter, will contribute higher to the computed loss of the model. This concerns both areas of methane plumes, but also areas of confounder materials (where we show the model through the label that they shouldn't be detected despite having large values in the MF product). The 5-layer CNN baseline model uses the same settings.

Finally, for training these models we use the `n1-highmem-8` instance on the Google Cloud Platform with one NVIDIA Tesla V100 GPU and 52GB of RAM. A typical training run takes about 6-8 hours, depending on the number of inputs of the models (and also the I/O speed of the used data storage).

In all our trained models we normalise the data band by band using min-max normalisation with statistics calculated from the training dataset. We compute these normalisation parameters for each of the methane enhancement products including band ratios and matched filters.

When using our trained hyperspectral models with data from the EMIT sensor, we are able to compute the same matched filter products and to select the closest bands to the ones that were used from AVIRIS-NG. We re-normalise these bands using the statistics from the original training dataset to move the data into ranges expected by the models.

4.7 Results

In this section we present the results of our proposed multispectral and hyperspectral machine learning models compared with the classical baselines used for both modalities of the data.

For semantic segmentation results we report the F1 scores of all pixel values in each tile, dividing the dataset into two broad categories - strong and weak emissions. In the codebase of Ružička *et al.* [2023b] we differentiate these by the number of plume pixels in the label, if they are beyond 1000. We note that we also considered splitting by the quantification of 1000 kg/h - this can be also done, but then the splits won't match our reported numbers. We also report the area under the precision-recall curve (AUPRC) score, which is independent of the choice of the threshold (by default for F1 scores set to 0.5). Classification per each tile is converted by using the simple rule of having more than 10 active pixels (after thresholding by 0.5). We note that the used thresholds could be tweaked to create a more sensitive system (balancing recall and precision).

Figure 4.1 shows prediction of our models and the classical baselines over a single selected large plume event from the evaluation dataset. We note that the multispectral baseline has the most false positive pixels, as it is difficult to differentiate between high band ratio values caused by methane plume or by the presence of confounder material. The MultiSTARCOP model does better in filtering out false positives in the scene, however it is able to reliably detect only the start of this strong plume (where the methane concentrations are the highest). In contrast, the hyperspectral data allows both the baseline and our trained HyperSTARCOP model to improve the semantic segmentation of the methane plume.

Figure 4.8 shows more qualitative examples from three types of events in our dataset - strong plume events, weak plume events and no-plume tiles.

4.7.1 Performance on multispectral data

We have trained machine learning models on different methane enhancement products computed from the simulated WorldView-3 data (namely the bands in the SWIR region). In Table 4.2 we present the AUPRC and F1 scores of the semantic segmentations and the FPR of the converted classifications of the trained models compared against classical baselines. As a baseline we show the Sanchez band ratio ($B8 \leftrightarrow B8_{MLR}$), we however note that all baseline methods had similarly low scores. We show that these methods cannot be easily used for automating detections.

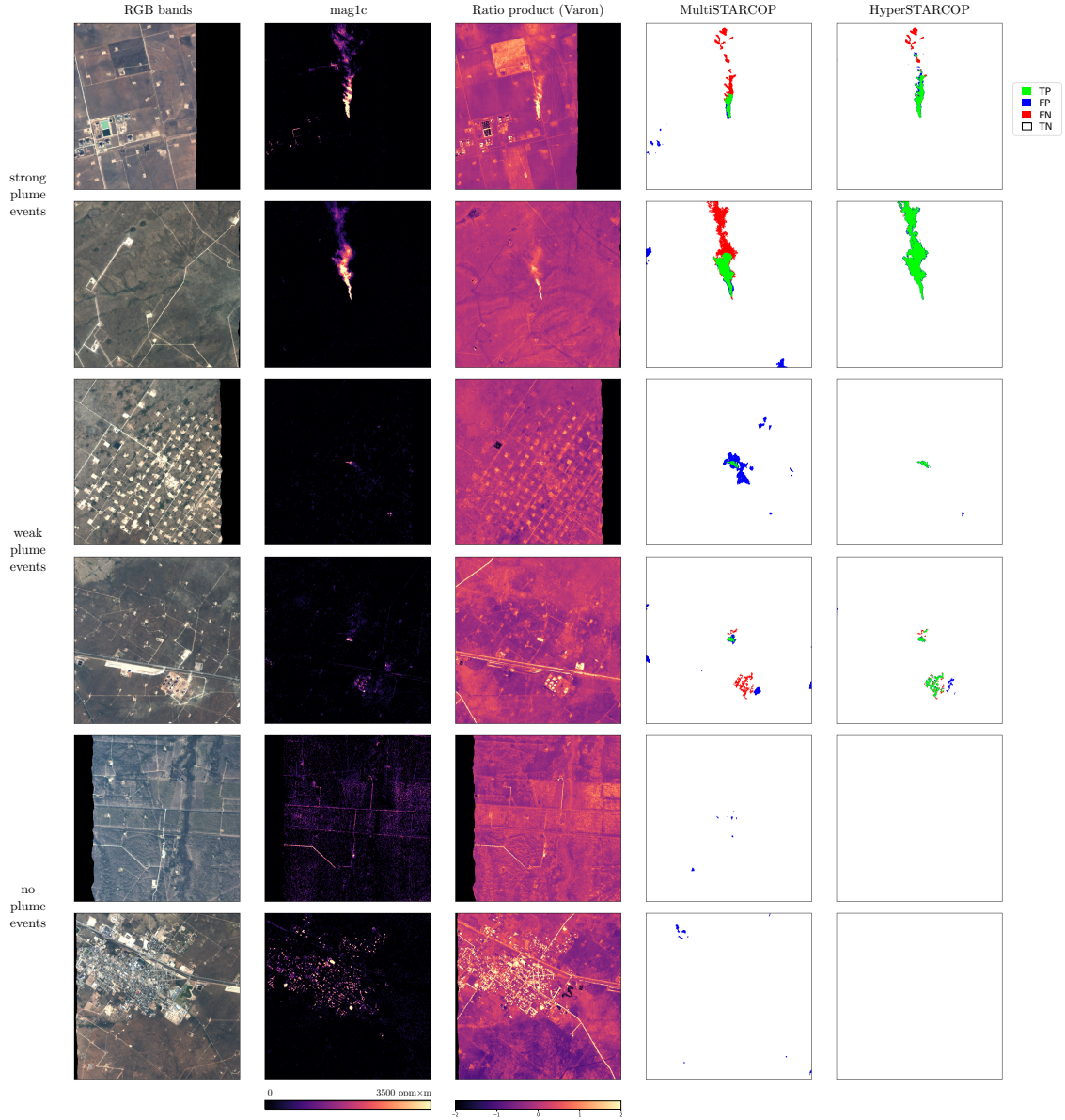


Figure 4.8: Showing the qualitative results of our models evaluated on strong plume samples (in the first two rows), on weak plume samples (in the second two rows) and finally on samples that are known as confounders (in the last two rows). Left to right columns in each sample shows: the RGB bands of the AVIRIS-NG data, the normalised mag1c product computed from the hyperspectral data, the normalised Varon ratios between bands B7 and B5 from the multispectral view of the data, prediction of the MultiSTARCOP model and the HyperSTARCOP model in comparison with the ground truth label. In the first three columns we show the areas outside of the sensor swath (no-data areas) with black background.

Table 4.2: Results of the MultiSTARCOP models on the test set, we show results of our proposed model in comparison with the existing baseline. We show the average results of 5 training runs of our models.

	F1 (strong) $\uparrow$	F1 (weak) $\uparrow$	FPR by tile $\downarrow$	AUPRC $\uparrow$
Ratio baseline	7.44	0.5	100.0	N/A
Our (Varon)	30.72 ± 2.87	10.35 ± 1.52	87.89 ± 4.67	11.92 ± 1.35
Our (Sanchez)	26.59 ± 3.13	9.32 ± 1.05	94.4 ± 1.30	9.96 ± 1.43
Our (Varon+Sanchez)	31.89 ± 2.44	11.04 ± 0.75	90.51 ± 4.23	13.04 ± 1.96

Models trained with a combination of Varon and Sanchez ratios obtain the best F1 and AUPRC scores, while the model using only Varon ratios achieved the best (lowest) false positive rate by tile. In all cases the performance drops rapidly when considering weak methane plume events.

4.7.2 Performance on hyperspectral data

Table 4.3: Results of the HyperSTARCOP models on the test set, we show results of our proposed model in comparison with the existing baseline. We show the average of training 5 runs of our models. F1 score is reported also for subsets of strong and weak plumes. FPR is reported for tiles after converting to classifications. MF here refers to the mag1c matched filter product.

	AUPRC $\uparrow$	F1 $\uparrow$	(strong)	(weak)	Prec. $\uparrow$	Rec. $\uparrow$	IoU $\uparrow$	FPR $\downarrow$
MF thr. baseline	N/A	40.14	67.50	39.95	30.57	58.42	25.11	75.43
5-layer CNN	42.89	40.23	73.06	34.03	29.17	69.22	25.24	59.20
	± 1.2	± 3.6	± 2.5	± 3.1	± 6.2	± 9.0	± 2.9	± 3.8
Ours, MF only	49.41	49.23	74.15	47.57	37.25	74.37	32.72	52.11
	± 5.5	± 3.8	± 6.1	± 4.2	± 4.8	± 9.0	± 3.4	± 11.0
Ours, MF+RGB	51.99	50.26	81.96	43.42	37.53	78.39	33.63	43.66
	± 2.8	± 3.8	± 3.7	± 5.7	± 5.8	± 6.6	± 3.4	± 7.4

We have trained machine learning models using the hyperspectral methane enhancement product computed with matched filter from the original AVIRIS-NG data. We also add a variant with extra RGB bands alongside this MF product, as means of accessing the background information in the scene. Both trained models are compared against the thresholded matched filter product which we consider as the classical hyperspectral baseline. Table 4.3 shows the comparison between these variants - in this case we report more detailed metrics to allow for easier comparison with later developed models from Chapter 5. When comparing the proposed models against

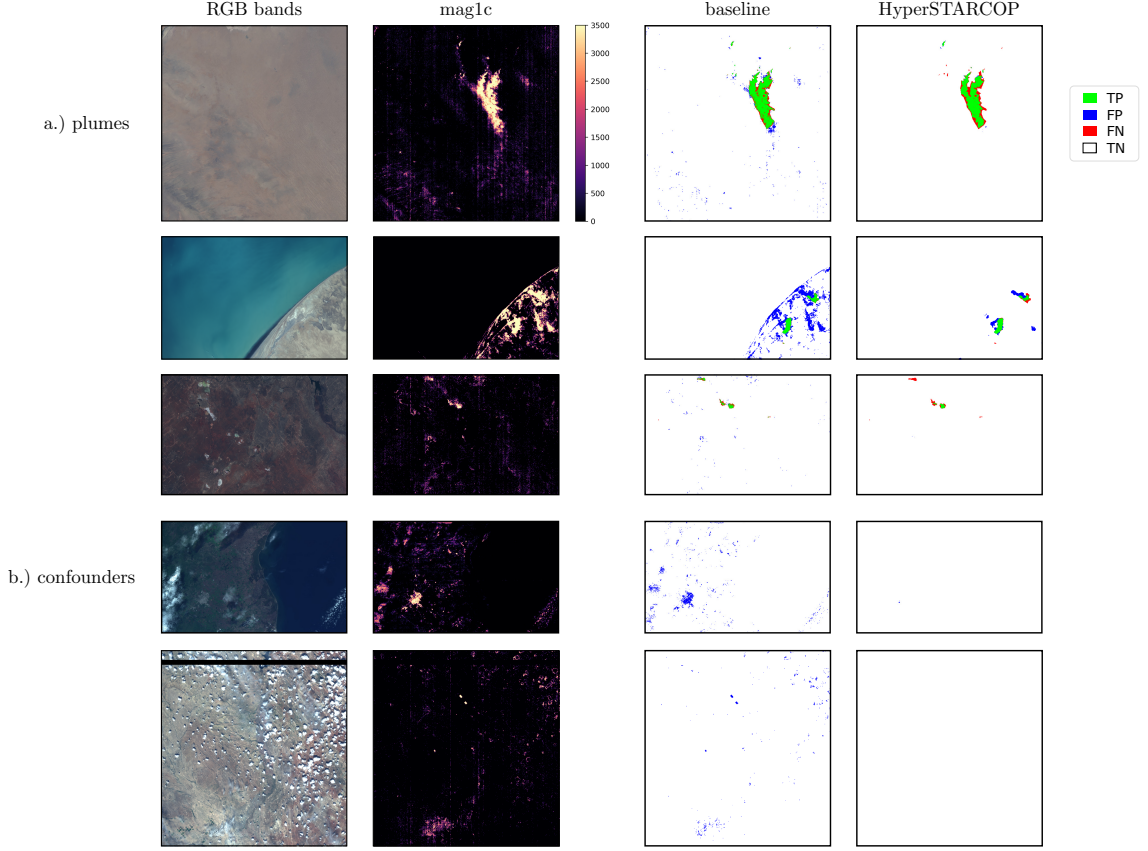


Figure 4.9: Results on example plume and confounder samples from the EMIT dataset, showing the baseline method and our HyperSTARCOP model using MF+RGB as inputs.

the classical and simple machine learning model baselines, we observe better scores in all metrics. Interestingly, our model with “MF + RGB” outperforms the “MF only” variant on strong plume events in F1 score, while receiving lower F1 score for weak events. Overall, the proposed models provide better semantic segmentation of the methane plumes, while also being less sensitive to noise and confounder materials.

4.7.3 Zero-shot generalisation on EMIT

We use the hyperspectral models trained on dataset from the AVIRIS-NG sensor in a zero-shot manner with data from the EMIT sensor. Despite the differences in the sensor configurations, ground and spectral resolution, we see that the model can be used to detect methane plumes. Figure 4.9 shows the predictions made on tiles with and without plume events. We see that the HyperSTARCOP model refines the detections made by the matched filter baseline (namely in reducing the number of blue

false positive pixels). Importantly, two desired properties were transferred despite using the model with data from another instrument, the capability to outline methane plumes and the robustness to rejecting the false positive detections of confounder materials (for example in urban environments).

Table 4.4: Results of the hyperspectral models on EMIT. We show the average of training 5 runs of our models. MF again refers to the mag1c product.

	AUPRC $\uparrow$	F1 $\uparrow$	Rec. $\uparrow$	Prec. $\uparrow$
Baseline on EMIT	N/A	35.25	65.8	24.08
Ours, MF only	46.61 ± 4.61	42.54 ± 5.22	31.09 ± 5.43	68.93 ± 4.72
Ours, MF+RGB	53.29 ± 8.08	49.45 ± 10.79	38.78 ± 12.48	73.17 ± 7.40

Table 4.4 compares the performance of the matched filter baseline against our two trained model variants. We see increase in most metrics when using the machine learning models trained with RGB and the MF product as inputs in contrast with the MF only variant. We note that the standard deviation of these results is higher than when using the AVIRIS-NG data, this is however consistent with the fact that the models has to generalise to previously unseen data and has to cope with associated spectral and spatial biases learned for the other instrument.

4.8 Discussion

The presented results confirm that machine learning models can reduce the number of false positives and that they outperform classical baselines. This holds true for the simulated multispectral scenario, where the high noise in band ratios prohibits their usage for automated plume detection. The proposed MultiSTARCOP model improves the detection of plumes in multispectral data, however the scores remain relatively low. We estimate that, as was confirmed by other works [Varon *et al.*, 2021], having access to multiple timesteps would further improve this performance.

In the hyperspectral scenario, we see that the HyperSTARCOP model could be used to automate methane leak detections in satellite data. Our “MF + RGB” model outperforms the MF baseline by 21.51% in F1 score for strong events and by 8.68% in F1 score for weak events. It also reduces the false positive rate by over 42.12% in contrast to the baseline. We note that the improved semantic segmentations of the detected methane plumes can also be used in other following tasks, such as plume quantification. There, the MF product is accumulated over the whole area of the

plume, so having a binary mask predicted from our model could reduce the effect of nearby confounders.

When evaluating our model pre-trained on AVIRIS-NG data for generalisation across other hyperspectral instruments, we see improved results in a small evaluation dataset of events in EMIT. There, we improve the F1 score by up to 40.28% over the baseline. Furthermore, we see the “MF + RGB” version outperforming the “MF only” version. We highlight that our work [Růžička *et al.*, 2023b] was the first one to use machine learning models to make predictions on data from the then new sensor EMIT for the task of methane leak detection.

Our multimodal dataset creation allows for direct comparisons between multispectral and hyperspectral instruments and models. Detection presented for example in Figure 4.1 illustrates the potential use of our proposed models for a tip-and-cue scenario on-board of satellites. Even a weak detection can trigger a tip-off signal which could autonomously schedule follow-up observations by other hyperspectral instruments. While the shape of the plume may change with the passage of time, it would provide another evidence and would allow a more accurate estimation of the plume emission size.

4.9 Summary

In summary, we answer the explored research questions connected to the specific application of methane leak detection showing that automation of these detections would be possible. With more labelled datasets from upcoming satellites, our proposed models will become even more reliable.

In the context of the task, many future instruments will be used for detecting methane leaks. We explore this multimodal environment by training models both on the multispectral and hyperspectral modalities of the data. Furthermore, we demonstrate zero-shot generalisation capabilities between two instruments.

Importantly, we contribute to the research in this domain by publicly releasing all data of our datasets, the code and the trained model weights¹. Our is one of the first large enough datasets that could be considered as a benchmark dataset for this domain, furthermore it includes higher quality labels given the manual inspection and refinement described in this chapter.

In our work, we encounter several limitations, which in turn inspire directions for future research. We note that all of the models presented in this chapter still rely on

¹<https://github.com/spaceml-org/STARCOP>

classically computed products (band ratios or the matched filter products). If we lost some information by computing these products, the models have no way of recovering it from the original raw data in individual bands. The chosen iterative version of the matched filter also remains to be prohibitively slow for potential on-board deployment of the proposed models on satellites. In follow-up research presented in the next chapter, we therefore address this by considering all hyperspectral bands with end-to-end machine learning models. We also note the relative simplicity of the used U-Net architecture, which we address by exploring more modern semantic segmentation architectures.

In future work, more global datasets should be considered. While the used AVIRIS-NG campaign provides large amount of samples, they remain to be concentrated in relatively local area of Four Corners in the USA. Additional flown campaigns improve this coverage, but we would like to explore truly global datasets with samples from geographically diverse zones. Thankfully the current missions such as EMIT and future missions such as SBG will provide just that. Finally, methane detection includes many additional tasks beside semantic segmentation of the plumes. Additional outputs could be considered when constructing new datasets and when training new machine learning models.

Universal Detection from Satellites

5.1 Introduction

The central problem addressed in this chapter is the **design of architectures that would act as universal detectors capable of training to detect any potential target chemical signals (Q3)** in data from hyperspectral instruments (also known as imaging spectrometers). These instruments capture images at high spectral resolutions allowing a wide range of applications for detecting phenomena observable in this data. Each specific application introduces domain specific solutions for detecting the events of interest. These are usually manually crafted sets of rules, which tend to be brittle to real-world conditions and noise in the data. They may also introduce new dependencies on data from external sources. Such an example was explored in the previous chapter on the task of methane detection, where we saw that machine learning models present a pathway for improving existing classical methods such as matched filters. Another example would be the broadly defined task of mineral identification in hyperspectral images, sometimes described as spectral end-member disentanglement using a library of spectral signatures and a set of cascading hand-crafted rules used to identify each hyperspectral pixel in the image. These and other examples can benefit from deep learning approaches, where salient features are learned directly from the data (rather than relying on classically derived products). By removing the dependency on classically computed products, we also gain in computational efficiency in several tested tasks. We aim to deploy the proposed general machine learning architectures on-board of hyperspectral satellites.

These aspects are illustrated on Figure 5.1, where we compare prior specialised approaches with unified processing pipeline using general machine learning models.

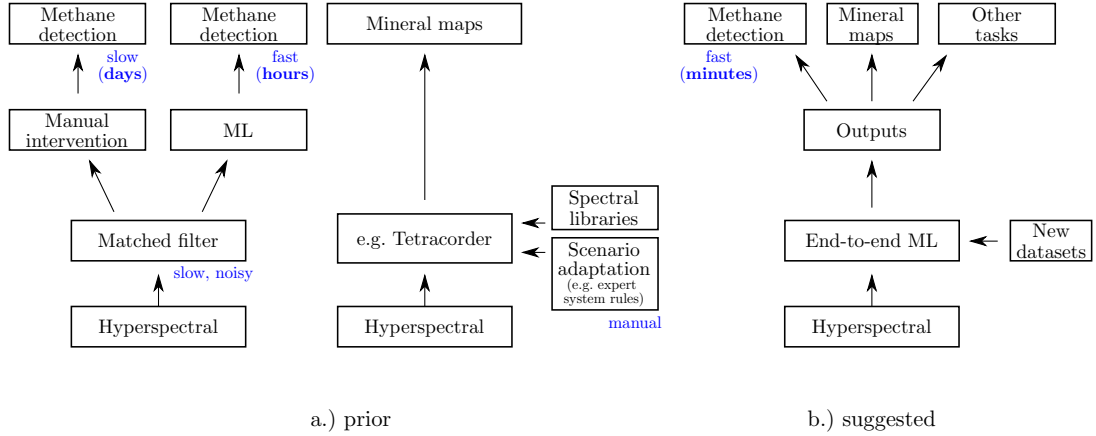


Figure 5.1: Illustration of unifying processing pipelines for different downstream tasks with general machine learning models. On the example of methane detection we also illustrate the potential speed-up of event detections.

Hyperspectral imaging is recently undergoing a revolution in terms of novel sensing instruments being deployed on satellites and other deep space exploration missions. The NASA’s EMIT mission, an early example of such an instrument, already provides near global observations of the entire Earth at relatively high ground resolution (1 pixel being 60x60m area) and high spectral resolution (spectral sampling of 7.4nm). Data from hyperspectral instruments can be used for a wide range of tasks, beyond just the example of methane leak detection explored in the previous chapter. This is in fact the motivation behind NASA’s upcoming Surface Biology and Geology (SBG) mission [Cawse-Nicholson *et al.*, 2021], which aims to provide data collection for tasks relevant to climate change, biodiversity monitoring, carbon storage and for differentiation between minerals, chemical pollutants, plant health monitoring and others. The aim is to capture data at improved ground resolutions (30x30m) and with vastly improved revisit time (proposed 16-days instead of the unpredictable schedule of ISS). We are familiar with such global and high cadency coverage of Earth by other multispectral satellites such as Landsat 8 and Sentinel-2. These have enabled detection of many events of interest, and also provide a true archive of the state of the planet over the years of their missions. As such, they have been heavily used in the Remote Sensing literature [Santos & Gonçalves, 2014; ED Chaves *et al.*, 2020]. More indepth description of current and upcoming imaging spectroscopy missions is available in Section 2.2.1.

When training machine learning models on remote sensing data, we often encounter simple reuse of of-the-shelf architectures originally proposed in the domain of

computer vision. One such example would be the adaptation of the U-Net architecture we presented in Chapter 4 for the task of methane detection. We however note that these models are not inherently suitable for data with high spectral dimensions due to reasons explored in this chapter and that reusing default versions of these models can lead to a loss of performance.

To explore and propose novel machine learning architectures or their adaptations to new types of tasks, we primarily need large and qualitatively annotated datasets. These are still mostly missing, which is why we have to create new dataset and sometimes we need to opt to simulation of synthetic events in the data. We note that the findings we make on these created datasets generalise beyond the simulated datasets into real world data and to other tasks.

Furthermore, a number of tasks related to the SBG mission could benefit from on-board automation. Namely, the detection of disaster events, hazardous materials or as previously explored gas emissions. On-board detection could trigger an early warning signal allowing fast on-the-ground reaction to the event. Automated scheduling of follow-up observations would also ultimately bolster the evidence of these events and maximise the collected science. This is often an underexplored direction of research, which dictates additional constraints to the proposed machine learning architectures.

Concretely, in this chapter, we propose adaptations to several modern machine learning architectures used for semantic segmentation. Concretely, in this chapter we introduce a family of machine learning models adapted to work with hyperspectral data called **HyperspectralViTs**. Their naming follows the rule that these are **Hyperspectral** adaptations of modern Visual Transfer (**ViT**) models. Two instances of these are the **HyperSegFormer** adapted from [Xie *et al.*, 2021] and **HyperEfficientViT** adapted from [Cai *et al.*, 2023]. We improve the processing of hyperspectral data using deep learning end-to-end models. Key insight is that end-to-end deep learning models are capable of extracting more valuable information from all provided bands of hyperspectral data - in contrast models depending on classical features are limited by their quality.

We explore two specific detection tasks with data from the EMIT sensor, for which we create three new labelled **OxHyper** datasets. Firstly, using the insight from the previous chapter, we create **OxHyperSyntheticCH4**, a global synthetic dataset of methane leaks events in EMIT data using shapes of real events from the AVIRIS-NG data for simulation. We also evaluate the trained models on **OxHyperRealCH4**, a new, smaller dataset of real methane leak events captured by the EMIT sensor. Secondly, we create **OxHyperMinerals** a dataset of mineral classes in the EMIT data

using a robust classical method as a source of pseudo-ground truth labels. Finally, we explore the deployment of the proposed machine learning models on-board of low compute environments including a proxy of a real-world satellite.

Our research has implications for many real-world tasks connected with the explored hyperspectral (imaging spectrometer) data. These span across novel Remote Sensing missions observing the Earth, such as NASA’s SBG [Cawse-Nicholson *et al.*, 2021], NASA’s Carbon-I, the ESA’s CHIME [Rast *et al.*, 2021] and others from private companies such as the MethaneSAT Miller *et al.* [2024] or Planet’s Tanager-1 [Barentsen *et al.*, 2024]. They are however also relevant to missions to other planets equipped with imaging spectrometers including the Moon [Ehlmann *et al.*, 2022] with the Lunar Trailblazer mission, the Europa Clipper mission to Jupiter’s moon [Poulet *et al.*, 2024] with the MISE (Mapping Imaging Spectrometer for Europa) sensor Blaney *et al.* [2024] and others. On-board autonomy would enable maximisation of collected science [Wagstaff *et al.*, 2019; Kerner *et al.*, 2020] by intelligent decision making and salient sample selection for downlinking.

Finally, the improvements to the existing machine learning models have concrete implications to the two explored tasks of interest, methane leak detection and mineral identification from space. Namely, on the task of methane detection we reach a new state-of-the-art performance. The work presented in this chapter has been made public in our pre-print paper Ružička & Markham [2024].

5.2 Background

In this section we outline methods and data sources relevant for our work in this chapter. More details are included in the Chapter 2.

5.2.1 Domain

Imaging spectroscopy (or hyperspectral) data allows for detection and modelling of a wide range of tasks. As an overview, we explore two specific missions: NASA’s SBG and ESA’s CHIME instruments.

The planned NASA’s Surface Biology and Geology (SBG) instrument [Cawse-Nicholson *et al.*, 2021] is based on the identified need for global high-resolution instrument with observation of wavelengths ranging from visible, shortwave infrared (VSWIR; 380-250 nm) to midwave (MWIR; 3-5 μm) and thermal infrared (TIR; 8-12 μm). Namely the VSWIR range overlaps to what we encountered with other instruments such as AVIRIS-NG and EMIT. Due to its wide spectral range, it is relevant

for a very large number of tasks. To name a few identified by the NASA’s Decadal Survey [NASEM, 2019]: terrestrial and aquatic ecology, atmospheric science, geology, hydrology (water use and quality), biology (vegetation traits and health), agriculture, snow and ice, disaster events and changing land use. The SBG Algorithm Working group has explored algorithms used for a subset of these tasks in [Cawse-Nicholson *et al.*, 2021]. Majority of these consider classical domain specific methods and machine learning models remain a minority. Nonetheless their exploration highlights that machine learning models will likely be needed in the light of the large quantity of data collected every day and the required speed of processing.

The Copernicus Hyperspectral Imaging Mission for the Environment (CHIME) instrument [Rast *et al.*, 2021] from ESA follows similar motivation and observes spectral wavelengths between 400-2500 nm. Relevant domains include: agriculture, food security, forestry, raw materials and soils, biodiversity monitoring and detection of environmental degradation both inland and in coastal regions.

In fact, both instruments have similar observation requirements (in the VSWIR range), provide similar products and are planned for launch at similar time frames. NASA-ESA cooperation has been highlighted by both [Cawse-Nicholson *et al.*, 2021] and [Rast *et al.*, 2021] papers. Harmonisation of the CHIME and the SBG-VSWIR data could reduce the average revisit time for each location from the estimated 16 days to about 8 days, similarly as has been previously done with harmonisation of Landsat 8 and Sentinel-2 imagery in [Claverie *et al.*, 2018]. Domain-to-domain generative machine learning models based for example on the CycleGAN [Zhu *et al.*, 2017] architecture could be used to translate between these modalities.

For this chapter, two types of tasks are especially relevant: methane plume detection belonging to atmospheric sciences and mineral identification which can be considered as spectral unmixing in the field of geology. Methane leak detection has been more closely explored in the previous chapter. Most methods depend on manual detection made by experts after processing the raw hyperspectral data into so-called methane enhancement products. A family of these products are based on matched filter (MF) approaches and their improvements [Foote *et al.*, 2020]. Mineral identification in hyperspectral data can be formalised as spectral unmixing, that is identifying which individual endmembers from a spectral library of chemical and mineral constituents create the observed signature. One classical approach is the Tetracoder program [Clark *et al.*, 2003] which uses a cascade of manually derived rules to compare observed spectral feature with a reference spectral library of minerals. It isolates discriminative portions of spectra known for each mineral using continuum removal

techniques [Clark, 1999] and then computes degree of fit metrics for each item in the library. It has been described as a simulation of the process a trained mineralogist would do when looking at the hyperspectral data.

Despite the differences between the domains of the two tasks there are several similarities in the used methodologies (matched filter and Tetracoder). MF tries to detect a single weak gas signal (here methane) layered over an estimation of changing scene background. In contrast Tetracoder focuses on the surface materials, this time comparing the observed signature with a library of candidate minerals and their mixtures. Both methods operate only on pixel-by-pixel basis. While MF models the background from a surrounding neighbourhood, the spatial shape of the potential methane plume isn't considered. Likewise, Tetracoder doesn't consider the spatial pattern of the detected minerals. The compute environment needed to run some of these classical methods requires a lot of effort to setup. The iterative versions of the MF can also be prohibitively slow to run on low compute devices. While Tetracoder was used on data from very different sources - Earth, Moon, Mars and Saturn - it needs a considerable amount of manual tuning of the used rules and addition of scenario specific decision trees and items in the spectral library. Similarly, the MF based techniques require manual input by trained experts to detect the actual leaked methane plumes.

What connects the classical methods used across different disciplines is the reliance on manual tuning and brittleness to in-the-wild and potentially noisy data. Deep learning approaches could outperform these classical baselines, by being able to automatically derive robust discriminative features from raw data in an end-to-end learning scenario. The natural bottleneck for training deep learning models is the access to large, labelled datasets.

5.2.2 Datasets

Datasets with large number of hyperspectral images and reliable source of labels are rare and usually there is a lack of diversity in terms of the coverage (many earlier sensors were aerially deployed only over specific location around the world). To make matters worse, many research papers work with very small datasets composed from just a few data captures such as for example: Indian Pines, University of Pavia, Houston or Salinas. As noted by the overview of [Signoroni *et al.*, 2019], this data is further split into a very similar and very local training and testing datasets, which leads to vast model overfitting and unrealistic results that couldn't really generalise to any new locations.

One example of a recent and large dataset is presented by the work of [Fuchs & Demir, 2023], which provides hyperspectral images from the EnMAP sensor. The focus of the paper is however on the task of image compression, which doesn’t require labelled data. Very recent examples are presented in the works on Foundation models in Wang *et al.* [2024] presenting dataset of hyperspectral images collected from the sensor on Hyperion EO-1, and Braham *et al.* [2024] with data from the EnMAP sensor. These datasets however don’t have labels for methane leak events.

5.2.2.1 Methane leak datasets

Several works in the domain of methane detection started using aerial data from the AVIRIS-NG sensor collected over several flown campaigns partially labelled by [Cusworth *et al.*, 2021]. The same source was used to create datasets for machine learning models in [Kumar *et al.*, 2023] and work presented in the previous chapter [Růžička *et al.*, 2023b] - in the latter case, we also further manually checked and improved the quality of labels.

More recently, data from the near-global sensor EMIT became available, and while the quantity of the captures scenes is very large (in February 2025 over 131k granules), there aren’t any established benchmark datasets. Some initial annotation of methane leaks is gradually released as the “EMITL2BCH4PLM.001” product on the NASA’s data portal [Green *et al.*, 2023]. Furthermore this mission is planned to continue collecting data at least through the year 2026. We note that we collect some of this data for use in this chapter.

Earlier works in the domain of methane detection resorted to simulation of methane leak events into clean hyperspectral scenes. In [Jongaramrungruang *et al.*, 2022] the Large Eddy Simulation (LES) [Matheou & Chung, 2014] was used to create realistic plume outlines which are mixed with background noise of specific AVIRIS-NG scenes. The workshop paper of [Groshenry *et al.*, 2022] instead uses the shapes of methane leaks detected in the Sentinel-2 data and mixes them with background noise of scenes from the PRISMA instrument. Finally, the work of [Joyce *et al.*, 2022] simulates several plume configurations and mixes them into 36 selected background scenes from the PRISMA instrument. In contrast to the previous two works, they simulate the methane leak events directly into the hyperspectral datacubes instead of simply adding them as a combined matched filter product. We however note that none of these works have released the data used for training their models or their code.

5.2.2.2 Mineral datasets

Datasets for global mineral identification across the entire globe are also missing. Works such as [Wan *et al.*, 2021] evaluate their methods on just two airborne hyperspectral captures. Works such as [Clark *et al.*, 2003] compare their methods against several well studied scenes such as the Cuprite or Alunite Hill scenes - these have been sparsely in-situ sampled to provide verification data. Note that large scale differences exist when comparing in-situ measurements and space based instruments (where one pixel can cover over 60x60m area).

Only recently, the data from the EMIT instrument were processed with the adjusted Tetracorder code to produce very initial, near-global maps of minerals. These are available as the “EMITL2BMIN.001” product in the NASA’s data portal [Green, 2023]. We note that we can use these initial maps as a sort of pseudo-ground truth for training machine learning models.

5.2.3 Machine learning models

The de facto standard machine learning model used for the task of semantic segmentation is the U-Net architecture [Ronneberger *et al.*, 2015]. With its encoder-decoder networks connected by skip connections which effectively allow the models to separate the scale of the learned features, it is often used even on relatively small datasets. It’s Siamese adaptation has been used in an active learning loop for change detection and has proved to learn relevant features even in very small initial datasets [Růžička *et al.*, 2020].

In this chapter we will explore more modern Transformer based models used for semantic segmentation such as SegFormer [Xie *et al.*, 2021] and EfficientViT [Cai *et al.*, 2023]. Transformer based architectures have recently shown promise when working with very large datasets, where they outperform the older convolutional models. They can also be abstracted into the encoder-decoder paradigm, where in the encoder network they use transformer blocks and, in the decoder, a MLP block. Furthermore, due to higher computational and memory requirements, they employ few efficiency tricks, such as strict spatial resolution reduction throughout the model. As such, the basic version of these models operates with output at 4x reduced resolution in contrast to the input.

While there are adaptations of existing architectures specifically in the domain of hyperspectral data, these models remain to be tested only on very small datasets. Overview of [Shirmard *et al.*, 2022] lists typically used methods, but only a subset

of these uses machine learning models and only a fraction adapts these models for hyperspectral data. The work of [Sun *et al.*, 2019] experiments with adding 1D convolutions into simple models to separate processing of spectral and spatial data from scenes such as University of Pavia, Indian Pines and Salinas. Meanwhile the work of [Yu *et al.*, 2020] adds 3D convolutions to learn spatial as well as spectral relationships on data from the same scenes and adds the Kennedy Space Center scene. The work of [Zhao *et al.*, 2020] uses a LSTM based model on one region in Nevada with data from AVIRIS-C (Classic). We note that in all of these works, the used scenes are very small, and in fact even just two samples from our previously proposed datasets would cover larger area. We underline the need for large, diverse datasets to properly evaluate any proposed new machine learning architectures.

5.2.4 On-board processing

While on-board processing on multispectral satellites have been demonstrated with few recent works [Mateo-Garcia *et al.*, 2021], including our own work presented in Chapter 3 [Ružička *et al.*, 2022], similar processing of hyperspectral data on-board remains limited.

As early as during the demonstrator mission of Earth Observer One (EO-1) spacecraft, six specific spectral channels and a thresholding algorithm was used for on-board cloud detection (other tasks were also explored such as flood mapping, change detection and generalised feature detection of sand dune scenes) [Chien *et al.*, 2005]. Anomalous spectral discovery on data from the EO-1 was later explored in [Thompson *et al.*, 2012] to produce scene summary maps and to generate a list of salient features which could be used for prioritization of scenes for downlink. In the work of [Thompson *et al.*, 2015b] a non-iterative matched filter was used for real-time visualisation of data during an airborne campaign. It was shown that having a direct visualisation of methane plumes from the overflowed scenes has increased the number of captured events of interest. We note, that these visualisations however needed a human interpretation during the flight, which prevents easy automation of the process. Finally, the recent instrument EMIT used an on-board cloud detection system (threshold detection in five specific spectral channels) to filter which scenes should be sent to the ground [Pham *et al.*, 2023].

5.2.5 Challenges and opportunities

First and foremost, there aren't many machine learning ready datasets in the domain of hyperspectral data that would be large, diverse and with high quality of labels. We are however able to use the datasets prepared in the previous chapter with new archives of data from the recently deployed near-global sensor EMIT to create datasets with synthetic and real event labels. Furthermore, the outputs of classical methods for mineral identification were released alongside the raw data captures. These can be used as pseudo-ground truth labels for our models. With these, we are able to create large datasets for two types of tasks of interest: methane plume detection and mineral identification.

Many existing classical methods are prohibitively slow for potential on-board deployment and simple approaches such as selected bands thresholding and default versions of the matched filter method produce many false positive detections. Trained machine learning models running on-board of communication constrained devices could enable a new paradigm shift. Many tasks relevant to the SBG mission could benefit from on-board detection and intelligent decision-making in the form of alert signals, prioritized downlink or automated follow-up observation scheduling. Similar instruments will be hosted on mission to other planetary bodies, where the communication bottleneck is even more constrained and the need to maximize the amount of scientifically relevant data is more prominent.

5.3 Method

In this chapter we approach two classical tasks with newly adapted deep learning architectures. In order to do so, we create three new datasets using data from recently launched sensor EMIT. When needed, we simulate synthetic events, we also use real events and in one case resort to generation of pseudo-ground truth labels using prior classical methods.

The first approached task continues our work from the previous chapter on methane leak detection in high resolution hyperspectral data. This task focuses on the detection of faint gas signals in the atmosphere against variable surface backgrounds. We create a new dataset of synthetic events in the EMIT data using our previous STARCOP dataset from the AVIRIS-NG data as a source of methane plume shapes. Additionally, we collect a smaller dataset of real methane leak events and also evaluate against previous datasets made in Chapter 4.

The second task is mineral identification, effectively a spectral unmixing from observed hyperspectral data. Here we typically search for particular minerals on the surface of the observed scene. Due to lack of available global datasets, we use a prior classical method as a source of pseudo-ground truth labels. We aggregate several separate components into larger mineral classes and select subset of three frequently present minerals as a multi-hot label of classes: “Goethite”, “Hematite” and “Kaolinite”.

Both tasks can be formulated as semantic segmentation using machine learning models. The models learn a translation from the observed spatial data into selected categorical labels predicted for all pixels in the image, the output prediction having the same spatial resolution as the input data.

We explore classical models as baselines and compare them with what could be called classical machine learning models. These use prior feature extraction methods to project the rich hyperspectral data into more easily interpretable products, such as matched filters, used by experts in the field. We instead propose deep machine learning models that learn their own features directly from the data in an end-to-end learning manner.

Section 5.4 describes the creation of the needed datasets, 5.5 presents the used baselines and proposed models and finally 5.6 discusses the used training setup and hyperparameters.

5.4 Data

In this chapter we use data from two hyperspectral instruments. The aerial AVIRIS-NG sensor previously used in Chapter 4 and the more recent sensor EMIT. These sensors are more closely described in Section 2.2.1.

5.4.1 Mineral maps

Given the focus of the EMIT mission on mapping surface mineral dust sources, the data is being released alongside annotation of mineral labels. The exact steps taken to produce these labels are described in the theoretical basis document [Brodrick *et al.*, 2023]. The Tetracorder method of [Clark *et al.*, 2003] was adapted for this new source of data and its outputs are released in the NASA’s data portal as the “EMITL2BMIN.001” product for each captured tile [Green, 2023].

The Tetracorder program defines its own categorisation of minerals in the so-called Expert System Groups and Cases (this is however separate from geologic groups).

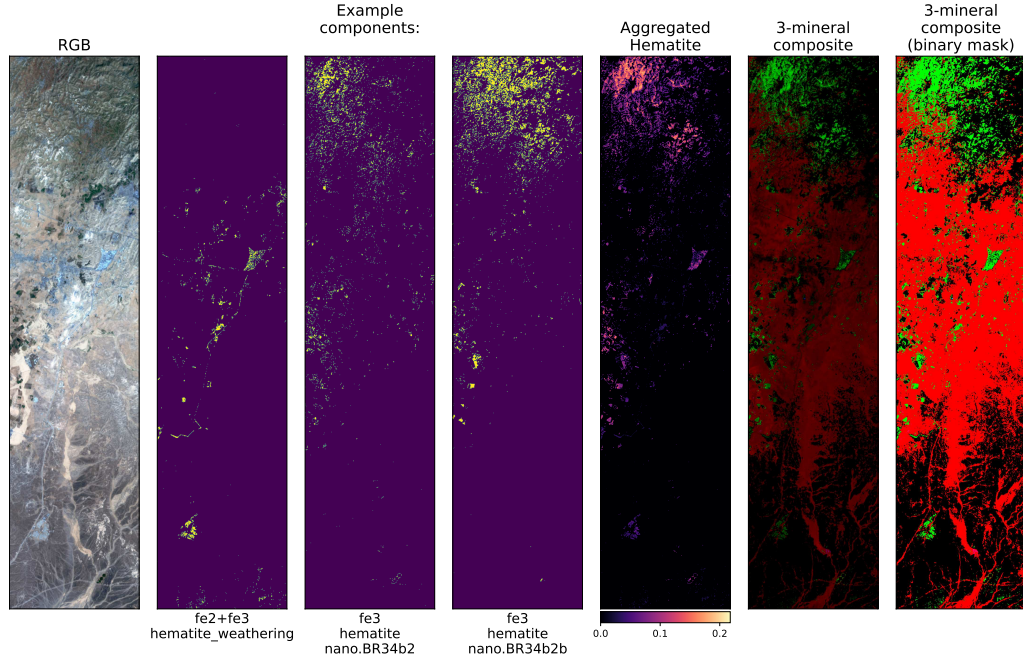


Figure 5.2: Aggregation of mineral components into mineral classes. From left to right we show a RGB visualisation of the scene, binary mask of three common components for Hematite, the aggregated Hematite product (using in total 14 components) and visualisations of a 3-minerals composites. In this example we map Goethite, Hematite and Kaolinite as RGB channels. We note that for training models, we convert the values of individual aggregated minerals into binary maps (as shown in the last column).

Mineral detection groups 1 and 2 are defined in [Brodrick *et al.*, 2023] depending on the region where they are observable. Group 1 consists of materials with electronic absorptions in the visible and 1-micron region. Group 2 instead consists of observables with narrow absorptions in the 2-2.5 micron region (such as clays, carbonates and sulphates). The spectral range of EMIT (380-2500 nm), covers both ranges. These two groups contain 87 and 294 individual constituent classes (in total 381 items in the USGS spectral libraries). Each pixel is assigned an identification of a constituent in Group 1 and in Group 2, both could also be kept without classification. This makes the corresponding labels multi-hot (one pixel can have multiple classes).

Only an earlier version v0.2 of the theoretical basis document [Clark *et al.*, 2020] includes the detailed description of how these individual components are to be aggregated together into what will be called the “EMIT10” product. At the moment of writing and using this data, this product doesn’t exist yet, however, we were able to reproduce the described steps. We aggregate the 381 constituents into 10 mineral

classes, formally defined as $M = \{\text{Calcite, Chlorite, Dolomite, Goethite, Gypsum, Hematite, Illite, Kaolinite, Montmorillonite, and Vermiculite}\}$.

For the 381 constituents predicted by Tetracorder, we know fractional abundances of how much they appear in a given mineral class $m \in M$. Following [Clark *et al.*, 2020], we aggregate the “EMITL2BMIN.001” product [Green, 2023] into binary labels of each of the mineral classes as is illustrated on the Figure 5.2. We note that these labels are only initial and that we omit the information of the estimated quantity of each mineral (abundance) and the uncertainty around these estimates - these will be included in the future EMIT10 product, for simplification we are using only binary maps (the difference is illustrated in the last two columns). We also highlight that these products are computed from the Tetracorder outputs and despite the in-situ validation of this methods at sites throughout the USA [Clark *et al.*, 2003; Swayze *et al.*, 2014] there may remain some incorrect predictions. Given these factors, we consider the generated labels only as pseudo-ground truth labels for training our models. We however note, that given any updated version of the Tetracorder software, or the full EMIT10 product which will be released in the future, or any potential manual refinement of these labels, our models can be easily retrained.

Finally, we collect the aggregated mineral maps as labels and select a subset of three mineral classes for full training: “Goethite”, “Hematite” and “Kaolinite”. These were well represented in the datasets we were creating. Each pixel is assigned a multi-hot categorical label, as there can be an overlap between the individual classes due to shared components that created these three class aggregates. One possible model output is also a prediction that none of these minerals is present, this was highlighted as an important feature of the Tetracorder system in [Clark *et al.*, 2003]. The choice of these selected classes was also motivated by [Connelly *et al.*, 2021], which states that “Hematite” is one of the most important mineral classes for its effect on radiative forcing of dust, as it has approximately an order of magnitude stronger effect than other minerals.

5.4.2 Methane leaks

We have previously created a large dataset of methane leak events in Chapter 4, however it is only located in the Four Corners region in the USA. In this chapter, we instead chose to use the data from the near-global sensor EMIT and sample more diverse datasets. When conducting this research, the number of known real methane leak events present in this data was relatively low, which is why we chose to create another dataset with purely synthetic events in addition to one with real events.

This also allows us to train models on synthetic data and then to test how much they generalise to real events (optionally fine-tuning them if needed).

5.4.2.1 Synthetic events

We can simulate synthetic methane leak events in the clean hyperspectral datacubes, given the knowledge of the expected signal of methane and plume concentration data from real events. Unlike previous study of [Jongaramrungruang *et al.*, 2022], which used plume shapes simulated using the expansive Large Eddy Simulation (LES) [Matheou & Chung, 2014], we use real methane leak events from the similar hyperspectral sensor AVIRIS-NG, to inform the simulation. Namely, we use the STARCOP dataset created in the previous chapter and in [Růžička *et al.*, 2023b], given its improved and manually refined labels and high diversity of plume sizes.

Mathematically, we can describe the relation between a datacube with and without methane leak event using the Beer-Lambert absorption law [Foote *et al.*, 2020] as:

$$L_{simulated}(\alpha, s) = L_{clean} * e^{-\alpha * s} \quad (5.1)$$

where L refers to at sensor radiances in the hyperspectral datacubes (clean and resulting simulated one), α to the concentration of methane at given location and s the methane absorption at given wavelength. We note that the quantity α is usually the one we are trying to estimate with matched filter approaches [Foote *et al.*, 2020]. As such, we can use existing cleaned matched filter products as sources for this quantity when simulating synthetic events. For s we query the values of the methane signature at the wavelength of each band. Similar notation can be derived from the works of [Sánchez-García *et al.*, 2022; Gorroño *et al.*, 2022].

We then recompute the matched filter product from the simulated datacubes, this product now contains both the simulated event and the background profile of the scene (sometimes with confounder noise). We then manually explore these products to make sure that the data points can be used.

Furthermore, we were interested in building a multi-temporal dataset, in order to enable future research to test whether having access to multiple images of the same location is beneficial for the model. The EMIT sensor is in operation since mid-2022 and has data available since the August of 2022. These observations contain many cases of intersecting data captures. We uniformly sample these locations to obtain balanced representation around the world. We avoid selecting locations with known or suspected real methane leak events.

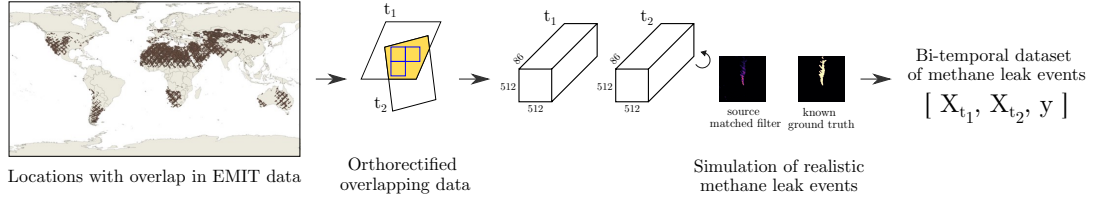


Figure 5.3: Illustration of the new bi-temporal dataset with near-global coverage.

We illustrate the steps used to make this dataset in Figure 5.3. First, we find areas with a large enough overlap between captures (filtering cloudy and invalid data), and we tile these overlapped regions as 512x512 px hyperspectral tiles. We note, that the size of the original EMIT data captures is very large (typical granule of 1280x1242px has around 1.9 GB), which is why the extraction of smaller tiles (512x512 px) around the events of interest is needed. We keep the temporal distance between two passes under 30 days. Each location is represented by a pair of hyperspectral datacubes, we select only the second one for simulation of a randomly selected plume from the original STARCOP dataset using eq. 5.1. We maintain the splits between train and test source data so that the same plume shape doesn’t appear in the newly constructed train and test subsets, as that could lead to overfitting. We further split the original test labels into validation and test subsets in the created EMIT dataset (again, not repeating the same plume shapes into the two splits). This will allow us to better evaluate the models.

We use the L1B level of data processing of the EMIT data. We orthorectify each sample and tag them with geo-location, this is needed in order to compute the intersection between the data. We note that despite the initial orthorectification, the pairs are not pixel-perfectly matched. This is consistent with the low processing level of data on-board the satellite. As an optional step, we suggest the use of the GeFolki algorithm [Brigot *et al.*, 2016] to align pixels of the earlier sample with pixels of the later ones. We note that running this algorithm further slows down the proposed system. We note that we currently don’t leverage the multi-temporal data of this dataset in our experiments and that it is published and prepared for future research.

5.4.2.2 Real events

In addition to these efforts, we also collected a dataset of real methane leak events using products released as “EMITL2BCH4PLM.001” in NASA’s data portal as [Green *et al.*, 2023], which can be used as initial annotation. We searched for events between August 2022 until October 2023, which resulted in 862 source plume events located

Table 5.1: Overview of datasets created in this chapter. Each large tile has 512x512 px, when training we further tile these into 64x64 px tiles with 32 px overlap - the size of the created training dataset is also listed. 2T refers to bi-temporal data. We note that the STARCOP dataset from Chapter 4 used different tile sizes (128x128px with overlap of 64px).

Datasets	Bands	Spectral range (nm)	Train	T. tiled	Val	Test
OxHyper-SyntheticCH4	2T, 86	RGB, 1573-1699, 2004-2478	796	174341	198	200
RealCH4	86	RGB, 1573-1699, 2004-2478	279	54805	91	98
Minerals	285	381-2492	796	174341	198	200
STARCOP	125	RGB, 1573-1699, 2004-2480	3425	N/A	N/A	342

in the total of 510 raw files (total of 1.1TB of data). We computed the matched filter product for all raw files and selected a dataset of real plume events in a total of 468 large tiles of 512x512px. This was manually curated, namely we selected only one tile for each cluster of nearby plume events (to prevent repetition of the same scene), we also excluded entire tiles where we found missing annotation of some weaker plumes (this could be later improved by manual annotation). Finally, we also excluded few very weak and very diffuse events, and samples where we encountered computational instability when computing the matched filter.

For these tiles, we used similar manual process as when we created the dataset from AVIRIS-NG data in the previous chapter. Namely, we applied an experimentally found threshold of 200 to the released matched filter product in [Green *et al.*, 2023] and processed the resulting binary maps with the opening morphological operation to reduce the speckle noise in the labels. We note that this set of events can be expanded in a multiple of ways. Primarily a more source events can be targeted, as they get detected in the future (and from the so far not included months in 2023-2024 and beyond). The missing plume annotation can also be manually fixed in the tiles which we excluded.

5.4.3 Datasets

In total, in this chapter, we create three new datasets, which we release under the shared name of “OxHyper”, summarized in Table 5.1. These datasets are made from EMIT data, we also compare them with the previously created STARCOP dataset from Chapter 4. Their global coverage is shown in Figure 5.4. While the newly created datasets are smaller than our previous dataset, they are sampled globally following

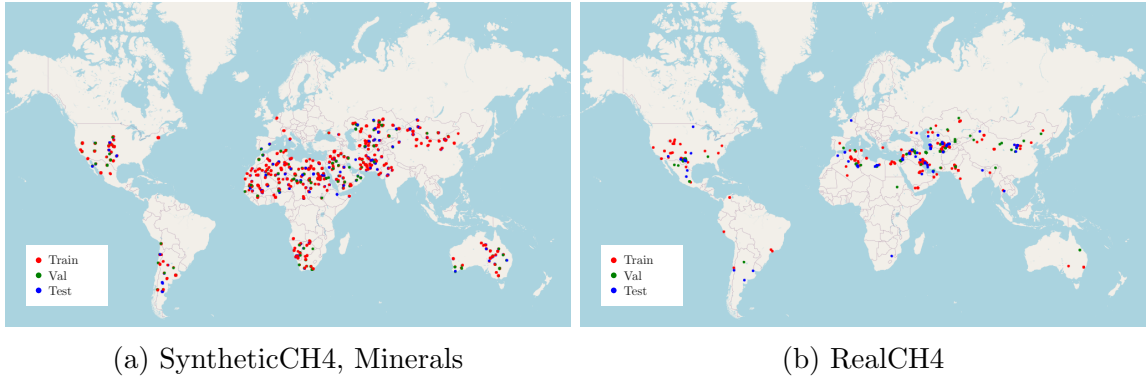


Figure 5.4: Locations of tiles used to create the OxHyper datasets of EMIT data. We note that the same source tiles (including the same dataset splits) are used for the minerals (where they contain more spectral bands) and the synthetic methane datasets (where the methane leak events are added into the clean datacubes).

the availability of the EMIT data (and as such provide much improved diversity of background scenes).

A particular care was taken when considering overlap between dataset subsets: When constructing the Train, Validation, and Test subsets, we made sure that these are made from not overlapping source data. Given the larger number of source images available for the creation of these datasets, we chose to create these splits using geographically and temporally non-overlapping regions (in contrast to the datasets created in Chapter 4, where we instead used only temporally non-overlapping data over the same geographic area). As also discussed by Signoroni *et al.* [2019]; Thompson & Brodrick [2021], some prior works that use hyperspectral datasets with machine learning models fail to split the data between the training and evaluation subsets. This may occur for example in cases of extracting tiles (sometimes called patches) from original data and using these to train models that predict values only for the central pixel. Although these works separate data according to different central pixels, some of the area caught by the neighborhood of the tile may end up being included in multiple subsets (allowing the models to overfit using this extra information). In this work, we check for overlaps between the full extracted tiles to avoid this problem.

Firstly, we created the “OxHyperSyntheticCH4” dataset, a bi-temporal hyperspectral dataset of synthetically simulated methane leak events using the labels of the STARCOP dataset with newly downloaded near-global hyperspectral datacubes from the EMIT sensor. In total we keep 796 large tiles (of 512x512 px) in the training dataset, 198 in the validation dataset and 200 in the test dataset. Each set is made from completely non-overlapping sources of data (clean EMIT datacubes and labels

from the STARCOP dataset from AVIRIS-NG data). In each subset, we uniformly sample the available plume sizes, so that we have a similar distribution of event sizes. Additionally, half of these tiles contain simulated events. We note that these large tiles (of 512x512 px) are further tiled when training the models. When using the tile size of 64 pixels with an overlap of 32 pixels, we finally get a training dataset of 174341 samples. The total size of this data is 228 GB.

Secondly, we create the “OxHyperRealCH4” dataset, a new mono-temporal hyperspectral dataset of real methane leak events. In total we have 279 large tiles in the training dataset, 91 in the validation dataset and 98 in the test dataset. These have reliable ground truth labels, which we calculated from the original labels provided on the EMIT data portal and consequentially manually checked. We note that this dataset is smaller than the other ones and training a machine learning model from scratch could in theory lead to overfitting and poor generalisation, however it can also be used for model fine-tuning and to test generalisation ability of models trained on other datasets. The total size of this data is 47 GB.

Thirdly, we create “OxHyperMinerals”, a completely new dataset for mineral identification, with labels of 3 selected mineral classes and source annotation for all 381 constituents (separated into two groups). In total we have 796 large tiles in the training dataset, 198 in the validation and finally 200 in the test dataset. These large tiles exactly match the tiles used in the synthetic methane events dataset, however they do not contain the simulated events and contain more bands. The total size of this data is 372 GB.

All large tiles have the spatial resolution of 60m and have 512x512 px. We further tile these using tile size of 64 px with an overlap of 32 pixels. The two methane datasets contain 86 spectral bands made of the RGB bands and ranges between 1573-1699nm and 2004-2478nm. In contrast, the mineral dataset contains all 285 spectral bands in between spectral range of 381-2492nm.

5.5 Models

In this section we define the used classical and machine learning baselines, list several limitations encountered when using machine learning models in the domain of hyperspectral data processing and finally we propose adaptations of several modern machine learning architectures to overcome these limitations. While baseline methods for the two tested tasks differ, we will demonstrate that our proposed machine

learning architectures can be used on in theory any task with similar input-output modalities.

5.5.1 Baseline approaches

In this chapter, we tackle two distinct tasks, detection of methane leaks and mineral identification, in both cases using hyperspectral data.

Methane detection, as has been extensively explored in Chapter 4, is usually approached with matched filter (MF) methods, one of these is the iterative MF called `mag1c` [Foote *et al.*, 2020]. Simply put, MF methods process selected bands in the original data and compare background contrasted data against a known signature of, in our case, methane. They produce a single band methane enhancement product (with values in $\text{ppm} \times \text{m}$, parts-per-million-meter), which is typically used by experts to manually detect methane plumes. As a baseline, we threshold the output of the `mag1c` filter with the experimentally found value of 500 $\text{ppm} \times \text{m}$ and apply opening morphological operator to reduce speckle noise.

In Chapter 4, we proposed classical machine learning models based on the U-Net architecture, that instead use the `mag1c` product as one of its inputs, alongside the RGB bands. We denote this model as “STARCOP MF+RGB”. It can be considered as a classical machine learning model baseline.

Mineral detection in remote sensing data has been approached with a wide range of classical methods which are described in the overview of [Shirmard *et al.*, 2022], and only a subset of these can be used with hyperspectral data. We explore two selected methods, the Tetracorder system [Clark *et al.*, 2003] and the classical approach known as spectral angle mapper (SAM) [Yuhas *et al.*, 1992]. We note here that the pseudo-ground truth labels that we want to train our machine learning models with, were created with the Tetracorder system. As such, we can’t really use this widely applied method as one of the baselines. When exploring the potentially simpler approach of SAM, we noted that there are large differences in the predictions for a sample scene. We note that there is a whole range of potential other distance metrics that could be used between the currently observed data and samples in the spectral library as is explored in [Deborah *et al.*, 2015].

As shown in Figure 5.5, the two methods don’t agree on the predictions. Therefore, instead of using the SAM method, we limit our comparisons only between different variants of the machine learning models. The potential usage of Tetracorder would unfairly obtain perfect score (given the pseudo-ground truth labels) and other methods such as SAM would get unfairly low scores (given the observed mismatch between

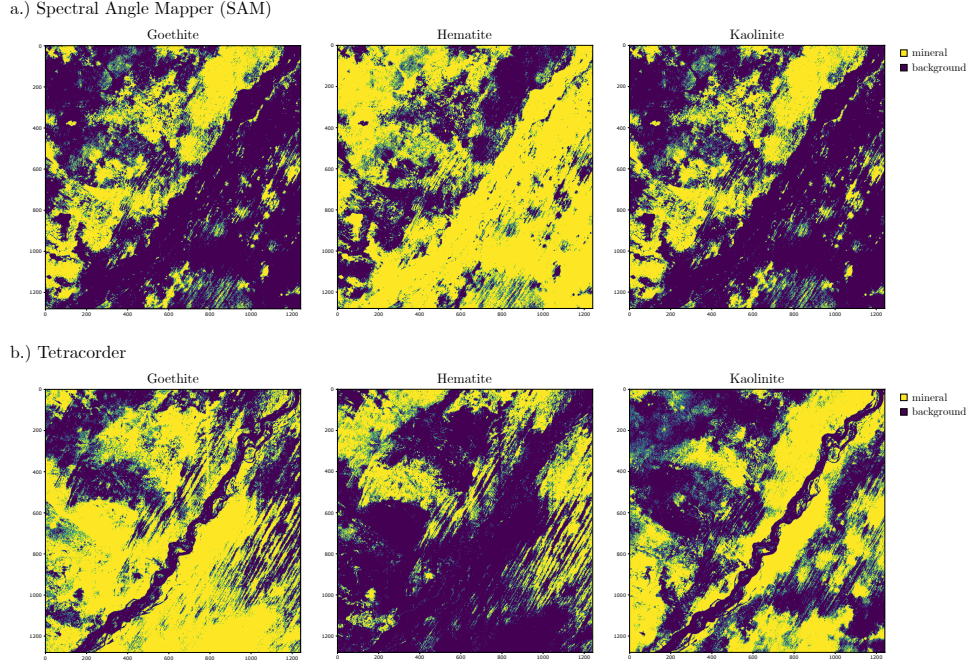


Figure 5.5: Comparison of the Spectral Angle Mapper (SAM) and Tetracorder methods over one example scene. We show the binary mask aggregated for three selected mineral classes. Note that SAM predicted the same values for Goethite and Kaolinite, as there is an overlap between components used for these two classes (and in this scene there wasn't anything else that would differentiate the labels).

the two method outputs). Until there is a manually verified dataset with validated ground truth labels, we will only compare how well are the proposed machine learning architectures capable of learning this pseudo-ground truth. Only a very small amount of validation data is known from the in-situ measurements made on calibration scenes such as Cuprite [Clark *et al.*, 2003].

5.5.2 Deep machine learning

In contrast to the used baselines and classical machine learning models that operate on the data produced from these baselines, we instead propose to use deep learning architectures on the full hyperspectral data. Similar paradigm has been embraced in the Computer Vision (CV) community by training deep learning architectures in an end-to-end learning manner ever since training models on the ImageNet dataset in [Krizhevsky *et al.*, 2012] instead of depending on prior classical features such as for example [Viola & Jones, 2001]. One critical difference is, that the typical Remote Sensing (RS) data contains more spectral bands than what is common in CV (usually just RGB). Data from Hyperspectral instruments typically have much higher spectral

dimensionality than other sources in both CV and RS.

In this chapter, we will compare several modern machine learning models based on the Transformer architecture with previously used classical machine learning models such as U-Nets. These architectures include the SegFormer [Xie *et al.*, 2021] and the EfficientViT [Cai *et al.*, 2023] models. Both can be described as encoder-decoder models, where the encoder network uses Transformer blocks to project the input data into high dimensional representations in sequentially reduced spatial resolution. The decoder block is then connected by skip-connections with different spatial encodings or the data. In the SegFormer model, a MLP block is used in the decoder network. The EfficientViT architecture has been designed to reduce computational and memory requirements of the other Transformer based models. In this chapter we will use default versions of these two models as deep learning baselines.

5.5.3 Current limitations

Existing classical baselines in methane leak detection suffer from noise and high false positive rate, also due to confounder materials. These show up on the computed MF products and confuse potential detectors. These baselines also omit spatial information about the shape of the detected plume or mineral patch, as they operate only on pixel-by-pixel basis. Dependency on classical products may introduce loss of information. If the MF product misses a real plume event, the model has no way of recovering it from the original data.

In addition, baseline methods such as the mag1c filter or the Tetracorder system introduce limitations due to slow computation and dependencies on other libraries. We describe this as **computational bottleneck**, which would prevent the deployment of these methods on-board of satellites and other systems. We measure the computation speed of methane detection baselines in several test devices. Machine learning models could overcome this computational bottleneck by providing faster inference speed on general and dedicated hardware.

However, even deep machine learning model may encounter difficulties when processing hyperspectral data. This is due to the large spectral dimension of the data. Machine learning architectures such as U-Net, SegFormer, EfficientViT and others, were designed to work well with typical Computer Vision data, usually with only 3 bands (RGB). These models suffer from what we describe as **informational bottlenecks**, when used end-to-end with hyperspectral data. We illustrate the limitations on Figure 5.6.

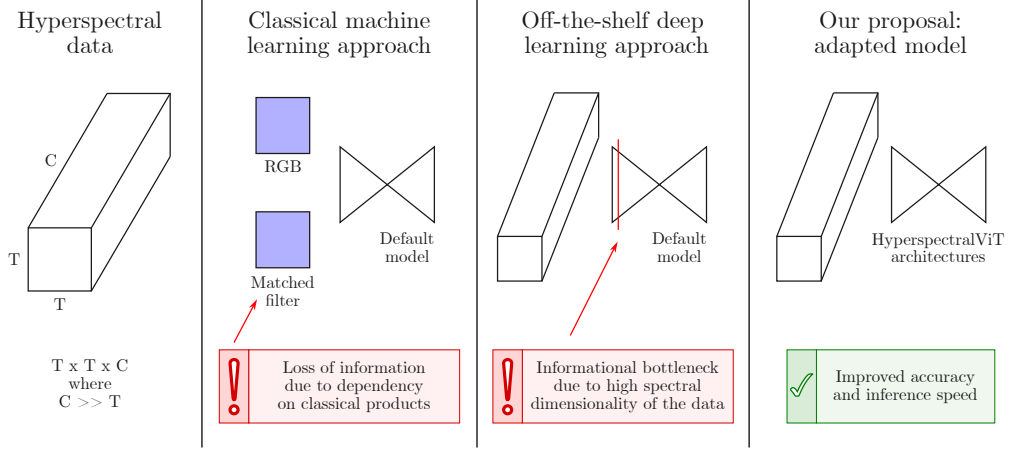


Figure 5.6: Illustration of the limitations of previous approaches and our proposed HyperspectralViTs, adapted Transformer based architectures.

As an example, if we select 86 bands of data which are relevant for our task of methane detection, the first layers in the vanilla versions of the used models reduces the data volume by -95.34% in the case of either the SegFormer, EfficientViT or U-Net (instead of increasing this volume by $+33\%$ as is common with RGB data). We claim that this **spectral bottleneck** forces the model to get rid of most of the potentially valuable information right at the entry point to the model, which means that the later layers cannot effectively extract high level compressed information, as this extraction has to occur in the first convolutional layer. In the case of SegFormer, the first layer is found in the first “Overlap Patch Embeddings” block and it is a 2D convolution with kernel size of (7,7), stride of 4 and padding of 3.

Furthermore, we have identified what we call a **spatial bottleneck** as other limitation encountered in the typical modern machine learning architectures such as the SegFormer, EfficientViT or any other architecture made from the same basic Transformer design. These models reduce the spatial resolution of the input data by the factor of 4, and use non-parametric bilinear interpolation to scale the predicted output to the original resolution. In the design of Transformer based models, we need what is referred to in [Li *et al.*, 2024] as fine-grained object segmentation, while maintaining fast model inference speeds. In contrast, the U-Net architecture reduces the spatial dimension of the data only by the factor of 2, and then uses a learned Upscaling layer at the end of the model to reconstruct the original resolution.

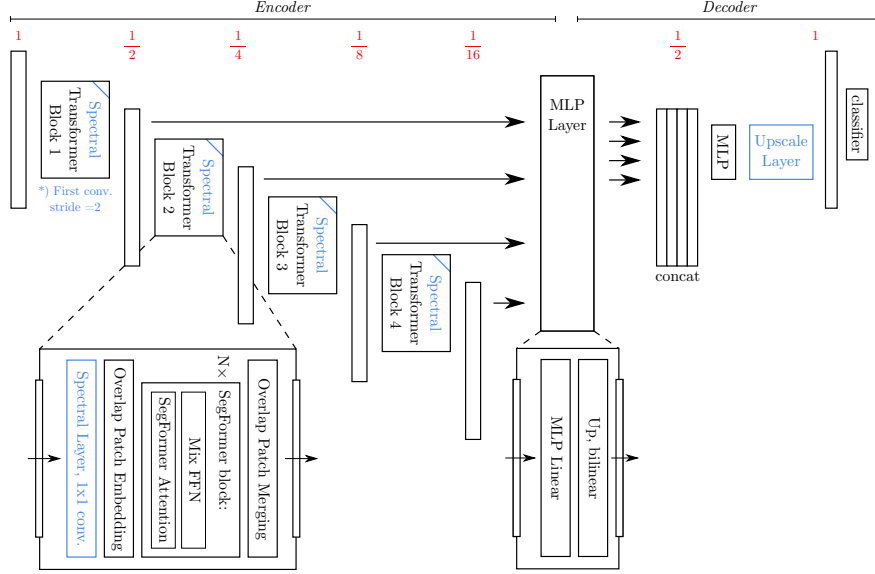


Figure 5.7: Illustration of the Hyper SegFormer model for semantic segmentation of methane plumes in Hyperspectral data. We highlight in blue the three proposed modular adjustments: 1.) Spectral layers (denoted as “1x1 Conv”) in the Transformer blocks, 2.) Upscaling layer in the decoder network (denoted as “Upscale layer”) and finally 3.) adjusting the stride of the first Transformer block to 2 (denoted as “Stride”). In red we show an example of the progressive decimation of the resolution throughout the model (using all three adjustments).

5.5.4 Model adaptations

We propose three modular changes to any Transformer based architecture for semantic segmentation to allow for better handling of data with high spectral dimension (such as data from hyperspectral satellites) as illustrated in Figure 5.7. We note that these changes are modular, and we test them in the SegFormer and the EfficientViT architectures, but they could be added to other models as well.

Firstly, to aid with the spectral dimensionality of the data, we propose an addition of a special spectral layer into all Transformer blocks throughout the encoder networks. We add a Convolutional layer with kernel size of (1,1), stride of 1 and padding of 0, keeping the output of each layer with the same spectral dimension as its input. This allows for learning operations that occur merely on the spectral data, on its own these allude to pixel-by-pixel operations of classical methods. While in theory the original convolutional layers with larger kernel sizes could learn similar operations, by restricting the operation to just the 1x1 convolutions, we obtain much sharper outputs. Effectively, this also allows the model to weight different input bands with a mechanism similar to attention, thus selecting only the relevant information

for the task of interest. While this is a simple addition to the model, we note that it is quite effective, and it doesn’t significantly increase the computational requirements of the model.

Secondly, we address the reduction of the spatial resolution of the data. We implement a similar Upscaling layer to the one used in the U-Net architecture at the end of the SegFormer model, just before its classification head. This block is composed out of two repeated 2D convolutional layers with kernel size of (3,3), padding and stride of 1, each followed by a ReLu activation layer and Batch normalisation layer.

As an additional solution for the recovery of the lost resolution, we also explore the change to the hyperparameters of the original SegFormer and EfficientViT architectures, namely changing the stride of the first convolutional layer from 4 to 2. This change effectively increases the spatial size of the data propagated through the network by the factor of 2. This doesn’t change the number of parameters of the resulting model, but as the effect of this change influences the whole model, the impact on the inference speed is larger than with the Upscaling layer.

These three proposed changes to the Transformer based architecture are modular and can be evaluated in isolation or used all together - the resolution of the output product will either be $\frac{1}{4}$, $\frac{1}{2}$, or 1 times the resolution of the input. We name the resulting adapted architectures as the HyperSegFormer and HyperEfficientViT variants - these are examples of the newly proposed **HyperspectralViT** architectures.

5.6 Experimental Setup

We separate our work into three types of experiments: a.) methane detection, b.) mineral detection and c.) on-board deployment. In all cases we use hyperspectral datasets mainly from the EMIT sensor, and we evaluate new adaptations of deep machine learning models against classical baselines and classical machine learning models. Our proposed models are task agnostic and as such we can estimate inference times on low compute hardware for any potential trained task.

5.6.1 Tasks

Firstly, we approach the task of methane detection, where we compare the classical baseline of matched filter and classical machine learning models that still rely on this product against our new end-to-end machine learning architectures. We train our models on the “OxHyperSyntheticCH4” hyperspectral dataset of synthetic methane leaks events simulated into clean EMIT datacubes. In order to explore the usefulness

of these models in the real world, we also evaluate their zero-shot generalisation ability on the “OxHyperRealCH4” dataset of real methane leak events in the EMIT data. We show how well can these pre-trained models be finetuned on a small training dataset of real methane leak events and compare them with models trained from scratch on this small dataset. In doing so, we demonstrate a scenario, where we only have a few real events observed by a newly deployed sensor, yet we still want to use it for detecting new events. Can pre-training on synthetic data help?

We also train the newly proposed end-to-end architectures on the benchmark dataset previously created in Chapter 4 made from the AVIRIS-NG data. In all of these experiments, we compare the newly adapted SegFormer and EfficientViT architectures against their vanilla versions and against the U-Net based HyperSTARCOP “MF+RGB” model.

Secondly, we compare the proposed architectures on another task, the mineral identification. For this, we use the created minerals dataset “OxHyperMinerals” from EMIT data with generated multi-hot labels of three selected mineral classes. These were produced by the Tetracorder system. We compare different adaptations of the end-to-end architectures against their vanilla versions for both the SegFormer and the EfficientViT architectures.

Thirdly, we evaluate the inference speeds of the proposed architectures against the classical baselines on devices with limited compute power. We note that we can use these models for both tested tasks (methane detection and mineral identification) and any other potential tasks which we will get ground truth labels for. We describe the target platforms in Section 5.6.3.

We use the F1 score, precision, recall and IoU (intersection over union) metrics for the pixel predictions considering two classes (“plume” and “no plume” pixels) and also the threshold independent AUPRC (area under the precision recall curve) score when comparing the continuous model predictions with the binary labels. Additionally we also report the false positive rate (FPR) of the converted classifications per whole tiles. For minerals datasets we report the F1 score for each mineral class and also the support weighted aggregated scores of F1, precision and recall.

5.6.2 Hyperparameters

In all experiments, we try to keep standardised training parameters, however there are some model specific deviations with experimentally found best working values. In other cases, we use the default parameters reported in the relevant papers. We train our models on the task of semantic segmentation, in methane detection with two

exclusive classes (“plume” and “no plume”) and in mineral detection with a multi-hot label (which allows no detection or potentially multiple active classes of “Goethite”, “Hematite” and “Kaolinite”). In all cases, we use the multi-label binary cross-entropy loss.

When training methane detection models on the “OxHyperSyntheticCH4” dataset, we use the following settings for all models: training for 50 epochs, batch size of 16 tiles, each at 64 px size (with a 32 px overlap). Losses are not weighted; we only use the WeightedRandomSampler from the PyTorch library to balance the positive and negative sampled tiles (using the fraction of plume pixels as a guidance). When fine-tuning the models on the “OxHyperRealCH4” dataset, we train the models for further 5 and 10 epochs from the best reached checkpoint. When training models from scratch on this smaller dataset, we keep the number of epochs to 30 and 50 (to limit and explore the effects of overfitting). End-to-end models use 86 bands of the EMIT data (RGB and bands between 1573-2478 nm).

To compare the newly proposed models with the HyperSTARCOP “RGB+MF” model from Chapter 5, we also train them on the AVIRIS-NG STARCOP dataset. Here we train the models for 15 epochs using a batch size of 16 (instead of the original 32) and tile size of 128 px with an overlap of 64 px. Except for the batch size, these hyperparameters are kept the same. We also weight the model loss by the MF product, as was done with the original HyperSTARCOP models. End-to-end models use 60 bands of AVIRIS-NG (namely bands between 2104-2400 nm).

When training models on the “OxHyperMinerals” dataset on the task of mineral detection, we train all models for 20 epochs with tiles of 64 px with 32 px overlap. In this case, trained models use all 285 bands of the available EMIT data (381-2492 nm). The data loader is not balancing the samples and losses are not weighted.

Finally, on all datasets we use model specific settings for learning rate and we keep the same selected backbone. U-Net models use the MobileNet-v2 backbone in the encoder and the Adam optimizer with learning rate of 0.001. SegFormer and EfficientViT models use the AdamW optimizer with the learning rate of 0.00006. SegFormer uses the B0 backbone architecture and EfficientViT the B1 backbone architecture, both are trained from scratch. This difference is given from the different number of parameters used in these backbones - the default SegFormer with B0 encoder has 3.845M parameters while the EfficientViT with the B1 encoder has 4.811M parameters (in ablation study we also explore larger encoder variants).

The OxHyper datasets presented in this chapter have three data splits (train, validation and test), which enables us to use the checkpoint with the best loss on

the validation dataset for evaluation on the test dataset. When training models on the STACOP dataset, we instead use the last checkpoint (this matches the approach taken in the previous chapter and allows fair comparison).

In all our models, we use min-max normalisation independently calculated for each band using the statistics of data from the training dataset. When calculating these, we remove outlier values, keeping data between lower quantile of 5th percentile and higher quantile of 95th percentile for calculating the min/max values.

5.6.3 Devices

We train the models on a high-performance computing (HPC) cluster usually with the NVIDIA Tesla V100 GPUs, each with 16-32GB of GPU memory, and the Intel Platinum 8628 (Cascade Lake) CPU with 2.90 GHz and 48 cores per node and RAM of up to 384GB. The most important part of this is the GPU and the available GPU memory as that governs the allowed batch size of training data (consider that each tile is hyperspectral and contains many spectral bands).

After training the models on a HPC cluster, we measure the models inference times and the needed data loading IO operations times on several limited compute platforms. As a proxy of the computational environment available on satellites, we use the flatbed machines provided by the UK aerospace company D-Orbit and by Unibap. We have experience in using these from work described in Chapter 3. These mimic the hardware available on satellites like the ION-SCV 004 we used to run our previous machine learning models deployed in space. The flatbed machine has a quad-core AMD GX-412HC SoC CPU at 1.2GHz and 2GB of RAM (from which only about 1.34GB was left free with the system running). It is also equipped with the Intel Movidius Myriad X VPU previously tested in Chapter 3. We used the ONNX Runtime python library for model inference. The Myriad X processing unit is estimated to consume only 1W when processing a 1 TFLOP of compute [Giuffrida *et al.*, 2021]. Here we however note that we couldn't fit some of the tested models on the VPU. Where this was the case, we instead used the CPU of the device.

We have also selected two other devices typically used for Internet-of-Things (IoT) applications, the CPU powered Raspberry Pi boards and the GPU equipped NVIDIA Jetson AGX Xavier board. As the device with the lowest compute, we use the Raspberry Pi 3B+ (released in 2018) with 4-core ARM Cortex-A53 SoC CPU at 1.2 GHz and 1 GB of RAM. We also use the Raspberry Pi 4B (released in 2019) with ARM Cortex-A72 SoC CPU at 1.5GHz (up to 1.8GHz) and up to 8GB of RAM. As has been noted by NASA's guidelines on using Raspberry Pi's in space [Guertin, 2022],

they are not suggested for deployment in any mission critical tasks due to their lower robustness in extreme environments. They have been however used as a proxies for devices with similar compute power in [Gretok & George, 2021]. The estimated power consumption of Raspberry Pi devices is around 5W during a benchmark.

We then use the more powerful NVIDIA Jetson AGX Xavier board (released in 2018) equipped with an 8-core NVIDIA Carmel ARMv8.2 CPU at 2.265 GHz and a GV10B GPU based on the Volta architecture with 16GB memory (shared between CPU and GPU). While initial tests of the effects of radiation on NVIDIA’s Jetson devices was explored in [Slater *et al.*, 2020] and proposed for space applications in [Lofqvist & Cano, 2020], these works also state that a careful examination and future tests are still needed. As such, while we test our models on these other hardware platforms, we reiterate that the most realistic point of comparison remains to be the Unibap’s flatbed machine as it has been space tested. We note that on the Jetson board we use the JetPack version of 5.1.3 and we experimented with both ONNX Runtime python libraries and the Tensor RT inference engine (using the *trtexec* command), as was also used in the work of [Cai *et al.*, 2023]. The Jetson AGX Xavier power consumption estimate is up to 30W, and while this seems like a lot, recent studies consider prototypes of 1U CubeSats with estimated 100W powered by solar panels [Pedivellano *et al.*, 2023].

5.7 Results

In this section we present the results of our proposed machine learning models adapted for hyperspectral data compared with prior approaches and classical baselines.

5.7.1 Methane detection on EMIT

Firstly, we present the results of training our proposed models on the OxHyperSyntheticCH4 dataset of simulated methane leak events. Figure 5.8 shows prediction of our performing model variants on a selection of scenes from the test dataset - here we use the ConvUpStride adaptations of the SegFormer models. In Table 5.2, we compare adaptations of the SegFormer and EfficientViT architectures against their vanilla versions and against classical machine learning models proposed in previous chapter. We also show the results of the classical baseline, the matched filter product which we threshold. Training runs were repeated 5 times and average and standard deviation values are shown.

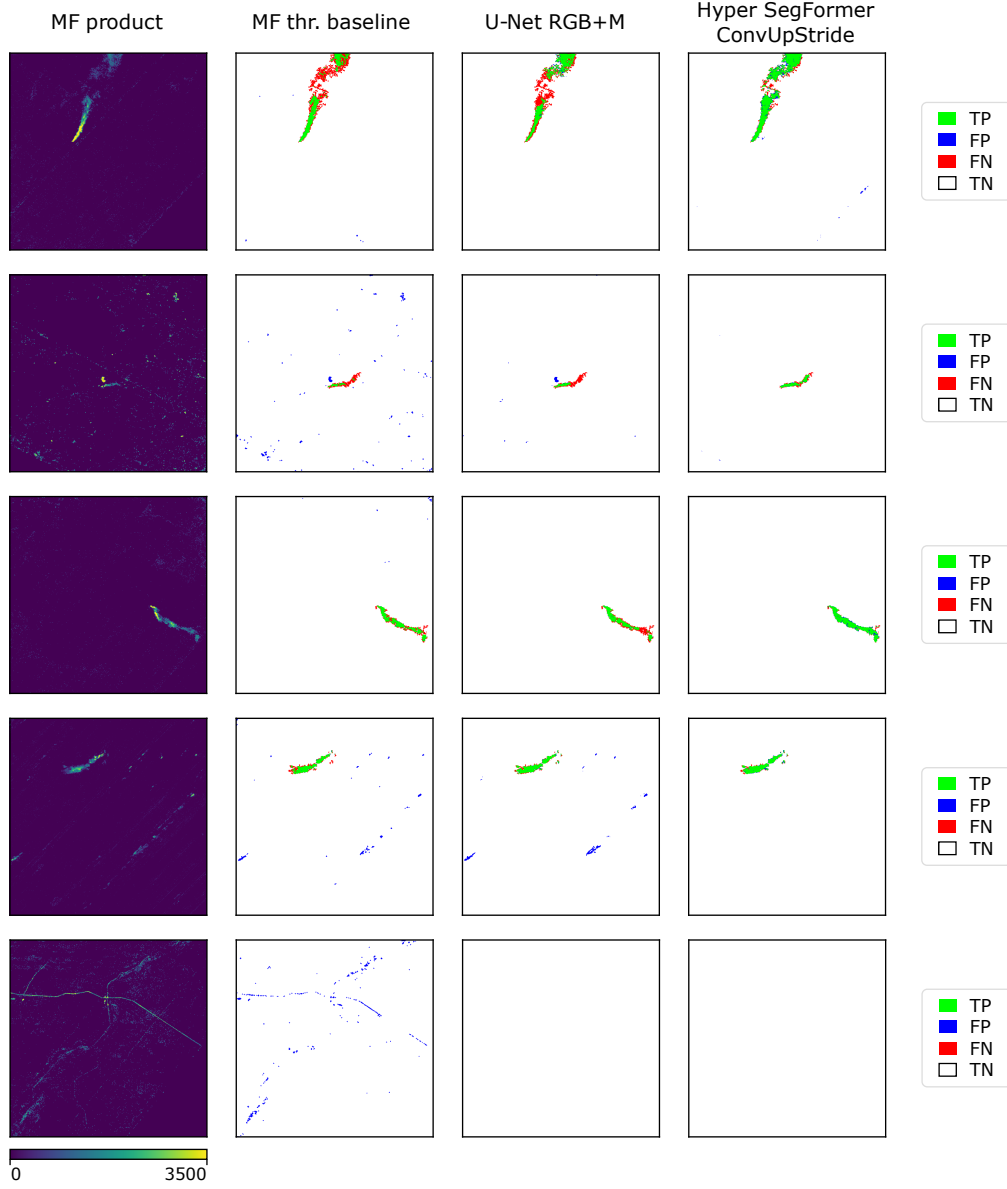


Figure 5.8: Qualitative results on the OxHyperSyntheticCH4 dataset of synthetic methane leak events. Showing the computed matched filter in the first column and comparing the performance of the matched filter baseline, our U-Net based model from Chapter 4 and from [Růžička *et al.*, 2023b] and finally our proposed Hyper SegFormer model (“ConvUpStride” variant). MF refers to the mag1c matched filter product.

Table 5.2: Quantitative results on the OxHyperSyntheticCH4 dataset. We show the average of training 5 runs of our models. SF refers to SegFormer and EV to EfficientViT. FPR is reported for tiles after converting to classifications. We compare the results against the RGB+MF HyperSTARCOP model from Chapter 4.

	AUPRC $\uparrow$	F1 $\uparrow$	Prec. $\uparrow$	Rec. $\uparrow$	IoU $\uparrow$	FPR $\downarrow$
MF thr. baseline	N/A	37.65	31.69	46.37	23.19	89.0
HyperSTARCOP	68.10	58.08	48.44	73.31	41.12	65.6
	± 4.7	± 5.3	± 6.7	± 2.8	± 5.2	± 6.7
SF base	67.76	60.26	53.14	70.70	43.15	27.2
	± 2.6	± 1.7	± 4.9	± 5.8	± 1.7	± 3.1
SF ConvUp	78.14	71.02	68.72	74.17	55.18	25.2
	± 3.0	± 3.4	± 7.4	± 3.2	± 4.2	± 7.2
SF ConvUpStride	82.24	74.27	74.15	74.88	59.15	18.4
	± 2.7	± 2.9	± 4.0	± 6.0	± 3.7	± 2.4
EV base	74.29	68.40	65.54	72.75	52.39	25.0
	± 7.3	± 6.9	± 10.3	± 6.0	± 7.8	± 5.0
EV ConvUp	77.86	70.89	68.31	74.06	55.00	40.2
	± 3.4	± 3.1	± 5.8	± 3.1	± 3.7	± 9.2
EV ConvUpStride	78.94	72.26	72.63	72.19	56.80	31.8
	± 5.3	± 4.9	± 6.3	± 5.5	± 6.0	± 5.9

Tables 5.3 and 5.4 detail ablation studies of the proposed model adaptations. We also show results of using larger encoder variants, but we note that these are likely too large for on-board deployment.

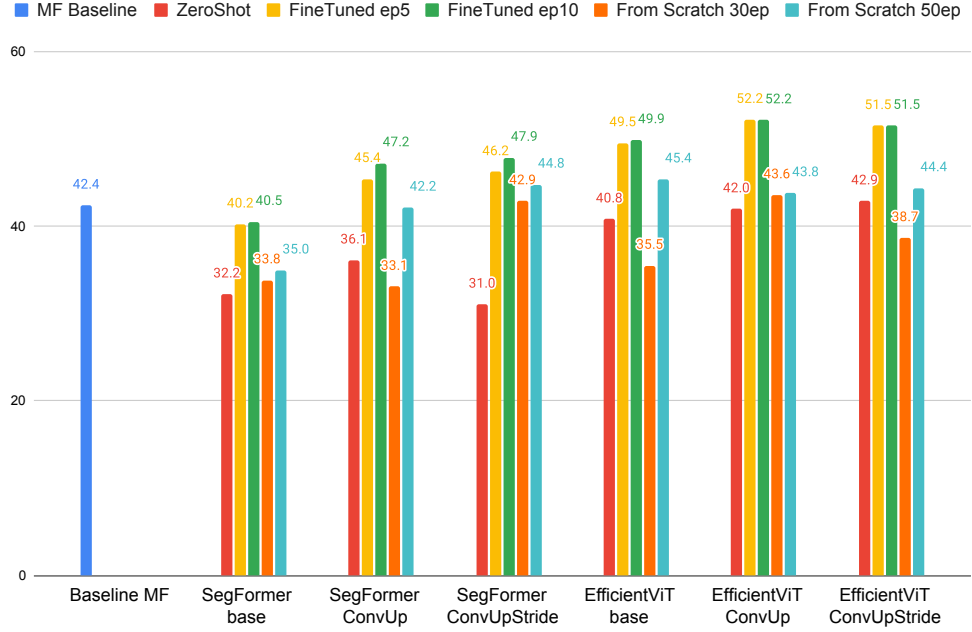
Secondly, in Figure 5.9 we evaluate our adapted SegFormer and EfficientViT models on the OxHyperRealCH4 dataset of real methane leak events captured by the EMIT sensor. Figure 5.10 shows the qualitative results on this dataset. Here, we use the models pre-trained on the synthetic dataset (which uses data of the same input/output shapes and uses the same data normalisation) and test their zero-shot generalisation on the dataset of real events. Furthermore, we fine-tune these models on the smaller dataset of real events. Finally, we also compare these results with training models from scratch only on the dataset of real events. We note that this experiment resembles a real-world scenario, where we train models on data from a newly deployed sensor, which hasn't yet seen enough of these rare events. We try to answer the question of if pre-training on a synthetically created dataset helps the final models detect events on a real but small dataset.

Table 5.3: Ablation study of the HyperSegFormer architectures on the OxHyperSyntheticCH4 dataset.

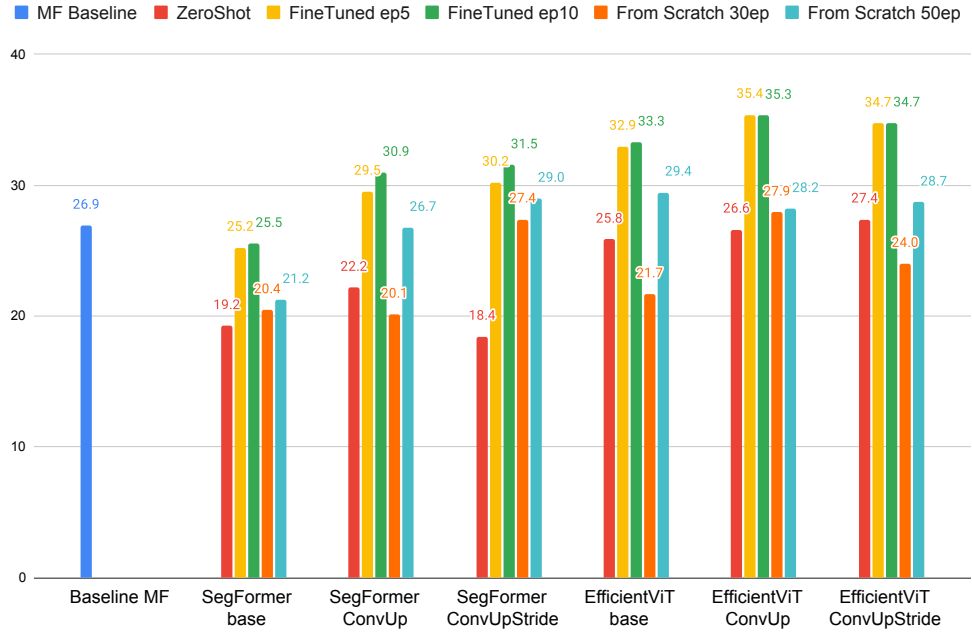
1x1 C.	S. 2	Up	Enc. var.	#Par. (M)	Out. res.	Score				
						AUPRC $\uparrow$	F1 $\uparrow$	Precision $\uparrow$	Recall $\uparrow$	IoU $\uparrow$
$\checkmark$	$\checkmark$	$\checkmark$	B0	3.85	$\frac{1}{4}$	67.76 $\pm$ 2.6	60.26 $\pm$ 1.7	53.14 $\pm$ 4.9	70.7 $\pm$ 5.8	43.15 $\pm$ 1.7
				3.88	$\frac{1}{4}$	75.94 $\pm$ 4.2	69.29 $\pm$ 3.8	68.27 $\pm$ 6.0	70.74 $\pm$ 4.2	53.15 $\pm$ 4.6
				3.85	$\frac{1}{2}$	71.94 $\pm$ 4.6	61.51 $\pm$ 2.7	52.4 $\pm$ 4.2	75.23 $\pm$5.6	44.47 $\pm$ 2.8
				4.29	$\frac{1}{2}$	73.22 $\pm$ 2.0	67.3 $\pm$ 2.0	67.41 $\pm$ 6.5	68.21 $\pm$ 5.5	50.75 $\pm$ 2.2
	$\checkmark$	$\checkmark$		4.33	$\frac{1}{2}$	78.14 $\pm$ 3.0	71.02 $\pm$ 3.4	68.72 $\pm$ 7.4	74.17 $\pm$ 3.2	55.18 $\pm$ 4.2
	$\checkmark$	$\checkmark$		3.88	$\frac{1}{2}$	82.17 $\pm$ 2.6	72.56 $\pm$ 3.9	66.8 $\pm$ 6.3	79.83 $\pm$ 1.6	57.09 $\pm$ 4.7
	$\checkmark$	$\checkmark$		4.29	1	71.62 $\pm$ 5.3	65.72 $\pm$ 6.1	63.5 $\pm$ 8.0	68.45 $\pm$ 4.1	49.23 $\pm$ 6.4
	$\checkmark$	$\checkmark$		4.33	1	82.24 $\pm$2.7	74.27 $\pm$2.9	74.15 $\pm$4.0	74.88 $\pm$ 6.0	59.15 $\pm$3.7
$\checkmark$	$\checkmark$	$\checkmark$	B1	13.94	$\frac{1}{4}$	76.93 $\pm$ 1.9	69.36 $\pm$ 1.4	66.46 $\pm$ 3.3	72.72 $\pm$ 2.8	53.11 $\pm$ 1.6
				14.51	$\frac{1}{2}$	77.00 $\pm$ 1.6	70.51 $\pm$ 1.2	68.14 $\pm$ 4.4	73.42 $\pm$ 3.6	54.47 $\pm$ 1.4
	$\checkmark$	$\checkmark$		14.51	1	84.25 $\pm$3.2	76.98 $\pm$3.3	75.94 $\pm$5.3	78.16 $\pm$1.7	62.67 $\pm$4.4

Table 5.4: Ablation study of the HyperEfficientViT architectures on the OxHyperSyntheticCH4 dataset.

1x1 C.	S. 2	Up	Enc. var.	#Par. (M)	Out. res.	Score				
						AUPRC $\uparrow$	F1 $\uparrow$	Precision $\uparrow$	Recall $\uparrow$	IoU $\uparrow$
$\checkmark$	$\checkmark$	$\checkmark$	B1	4.81	$\frac{1}{4}$	74.29 $\pm$ 7.3	68.4 $\pm$ 7.0	65.54 $\pm$ 10.3	72.75 $\pm$ 6.0	52.39 $\pm$ 7.8
				4.84	$\frac{1}{4}$	77.44 $\pm$ 1.8	71.05 $\pm$ 1.5	70.99 $\pm$ 4.8	71.43 $\pm$ 2.1	55.12 $\pm$ 1.8
				4.81	$\frac{1}{2}$	73.77 $\pm$ 2.9	67.54 $\pm$ 3.2	62.86 $\pm$ 5.7	73.39 $\pm$ 2.2	51.07 $\pm$ 3.7
				4.83	$\frac{1}{2}$	77.53 $\pm$ 2.5	70.26 $\pm$ 2.8	67.64 $\pm$ 5.5	73.53 $\pm$ 3.1	54.22 $\pm$ 3.3
	$\checkmark$	$\checkmark$		4.86	$\frac{1}{2}$	77.86 $\pm$ 3.4	70.89 $\pm$ 3.1	68.31 $\pm$ 5.8	74.06 $\pm$3.1	55.00 $\pm$ 3.7
	$\checkmark$	$\checkmark$		4.84	$\frac{1}{2}$	78.95 $\pm$2.6	71.86 $\pm$ 2.5	70.66 $\pm$ 6.2	73.55 $\pm$ 1.7	56.14 $\pm$ 3.1
	$\checkmark$	$\checkmark$		4.83	1	74.68 $\pm$ 5.2	67.63 $\pm$ 4.5	63.51 $\pm$ 7.9	73.00 $\pm$ 2.4	51.27 $\pm$ 5.2
	$\checkmark$	$\checkmark$		4.86	1	78.94 $\pm$5.3	72.26 $\pm$4.9	72.63 $\pm$6.3	72.19 $\pm$ 5.5	56.80 $\pm$6.0
$\checkmark$	$\checkmark$	$\checkmark$	B2	15.34	$\frac{1}{4}$	75.4 $\pm$ 6.6	68.96 $\pm$ 5.9	67.25 $\pm$ 11.2	72.14 $\pm$ 3.4	52.93 $\pm$ 6.6
				15.43	$\frac{1}{2}$	82.6 $\pm$2.8	75.24 $\pm$2.6	74.68 $\pm$6.1	76.21 $\pm$2.6	60.38 $\pm$3.4
	$\checkmark$	$\checkmark$		15.43	1	82.25 $\pm$ 1.9	74.29 $\pm$ 2.2	74.34 $\pm$ 6.7	74.77 $\pm$ 2.6	59.15 $\pm$ 2.8



(a) F1 score



(b) IoU score

Figure 5.9: Quantitative results on the OxHyperRealCH4 dataset. We compare the MF baseline against zero-shot generalisation, fine-tuning or training from scratch for both proposed models. Results are averaged from 5 training runs, and the values on top of individual bars are rounded for better clarity.

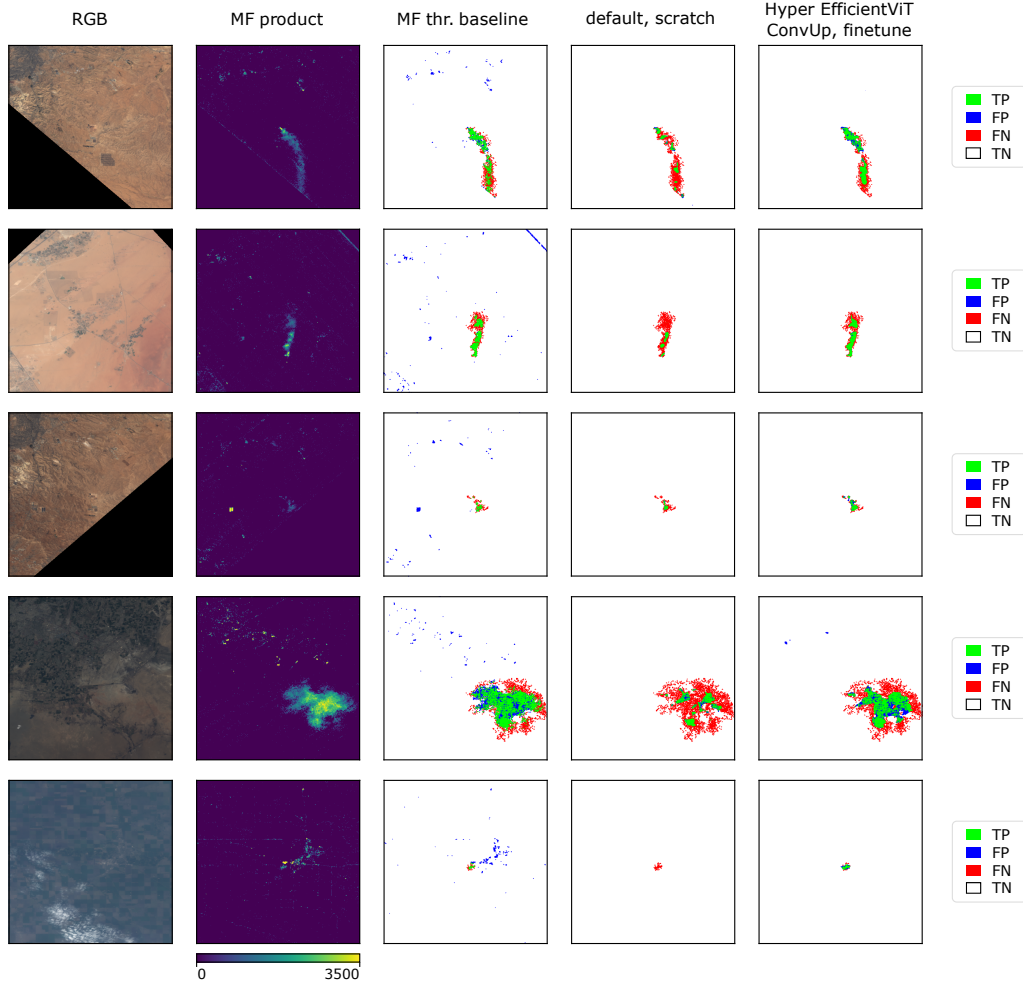


Figure 5.10: Qualitative results on the OxHyperRealCH4 dataset of real methane leak events. From left to right we show the predictions of the classical baselines, models trained from scratch and models pre-trained on the synthetic dataset (using the HyperEfficientViT model, ConvUp variant). We note that black parts are the no-data areas created upon orthorectification. MF refers to the magic matched filter product.

Table 5.5: Quantitative results on the benchmark STARCOP dataset. We show the average of training 5 runs of our models. SF refers to SegFormer and EV to EfficientViT architecture. F1 score is reported also for subsets of strong and weak plumes. FPR is reported for tiles after converting to classifications.

	AUPRC $\uparrow$	F1 $\uparrow$	(strong) $\uparrow$	(weak) $\uparrow$	Prec. $\uparrow$	Rec. $\uparrow$	IoU $\uparrow$	FPR $\downarrow$
MF baseline	N/A	40.14	67.50	39.95	30.57	58.42	25.11	75.43
U-Net RGB+MF	51.99	50.26	81.96	43.42	37.53	78.39	33.63	43.66
	± 2.8	± 3.8	± 3.7	± 5.7	± 5.8	± 6.6	± 3.4	± 7.4
SF base	48.84	37.88	68.20	17.91	25.71	75.58	23.52	32.80
	± 3.3	± 6.5	± 3.2	± 4.7	± 6.3	± 4.0	± 5.0	± 12.5
SF ConvUp	57.91	53.17	71.53	39.37	44.93	68.16	36.42	32.34
	± 5.6	± 6.5	± 4.0	± 8.7	± 10.8	± 5.9	± 6.0	± 20.2
SF ConvUpStride	60.98	56.75	73.16	45.65	50.87	65.73	39.69	19.89
	± 4.5	± 3.7	± 1.9	± 3.0	± 7.7	± 5.8	± 3.5	± 6.1
EV base	47.39	48.69	62.10	36.26	44.92	54.98	32.26	18.06
	± 5.0	± 4.1	± 6.0	± 5.0	± 7.3	± 8.1	± 3.6	± 6.6
EV ConvUp	56.10	53.47	66.84	40.60	49.58	59.84	36.58	15.54
	± 5.3	± 4.2	± 6.1	± 6.7	± 6.6	± 10.3	± 3.9	± 5.8
EV ConvUpStride	51.99	49.95	65.17	40.79	43.68	60.41	33.35	17.49
	± 4.1	± 3.7	± 2.5	± 3.1	± 7.6	± 6.8	± 3.2	± 3.8

5.7.2 Methane detection on AVIRIS-NG

In Figure 5.11 we evaluate our newly proposed end-to-end models on a benchmark STARCOP dataset of real methane leak events created in the previous chapter. It contains data from the aerial AVIRIS-NG sensor from the Four Corners area in USA. Table 5.5 compares the results of the classical machine learning models based on the U-Net architecture called HyperSTARCOP models with the newly proposed end-to-end models.

5.7.3 Mineral identification

In Table 5.6, we show the results of training the proposed machine learning architectures on the mineral dataset OxHyperMinerals. We show the individual F1 scores for each of the three mineral classes alongside the aggregated scores (which are support weighted by the number of pixels belonging to each class). In Figure 5.12 we also show examples of model predictions next to the pseudo-ground truth generated by the Tetracorder system. Note the gradual increase of output resolution depending on the used adaptations.

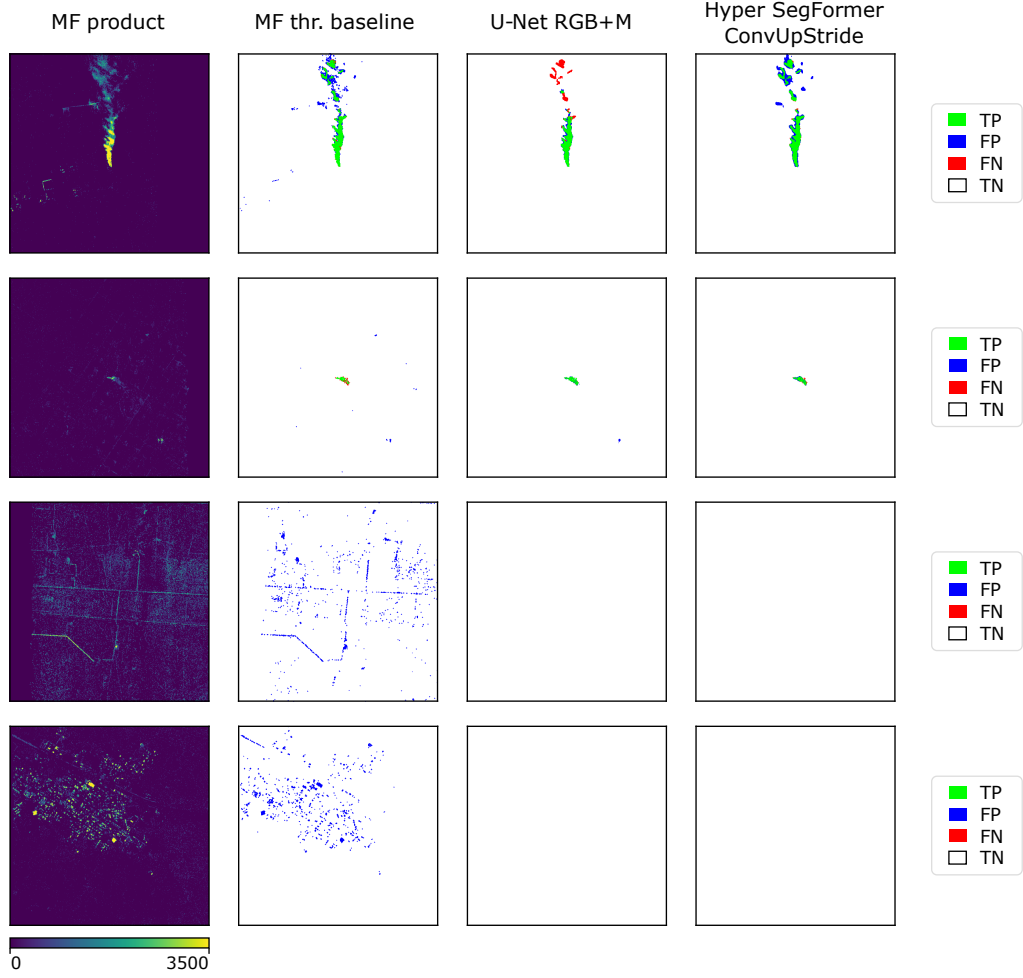


Figure 5.11: Qualitative results comparing our new models on the benchmark STAR-COP dataset showing the computed matched filter in the first column, the matched filter baseline, U-Net based model from [Růžička *et al.*, 2023b] and our proposed Hyper SegFormer model (“ConvUpStride” variant). MF refers to the mag1c matched filter product.

5.7.4 Inference times

Table 5.7 shows the model inference times needed to process a full EMIT scene capture of 1280x1242 px. These measurements show the IO time needed to load the hyperspectral datacube, time to calculate the MF product for methods depending on it and finally the model inference time. These are shown separately and only for the Unibap flatbed machine on Figure 5.13. We note that as the SegFormer models crashed the Myriad X VPU, we used the CPU of the Unibap flatbed machine instead (there the models ran without any crashes). We also don’t report the numbers for the EfficientViT model, as we lost access to this deployment platform. It should be taken into consideration, that the other devices haven’t been tested in the space

Table 5.6: Quantitative results on the mineral dataset OxHyperMinerals. We show the average of training 5 runs of our models. We report the F1, precision and recall metrics for the aggregated scores (weighted by the number of pixels belonging to each class) and the average F1 scores for individual classes (G = Goethite, H = Hematite, K = Kaolinite). SF refers to SegFormer and EV to EfficientViT architecture.

	F1	Precision	Recall	F1 G.	F1 H.	F1 K.
SF base	81.77 $\pm$ 1.5	77.33 $\pm$ 3.6	86.92 $\pm$1.8	79.31	80.88	85.79
SF ConvUp	82.99 $\pm$ 0.9	80.55 $\pm$ 1.0	85.64 $\pm$ 2.3	79.36	81.47	89.15
SF ConvUpStride	84.70 $\pm$0.7	83.42 $\pm$ 2.7	86.14 $\pm$ 2.0	81.19	83.59	90.35
EV base	77.08 $\pm$ 0.7	79.37 $\pm$ 3.7	75.28 $\pm$ 4.0	69.88	77.89	85.25
EV ConvUp	79.73 $\pm$ 0.9	82.66 $\pm$ 2.3	77.11 $\pm$ 2.4	74.59	79.74	86.55
EV ConvUpStride	79.18 $\pm$ 1.2	84.03 $\pm$2.7	74.96 $\pm$ 2.2	73.57	77.93	87.24

Table 5.7: Timing of models on different devices. We estimate the time in seconds for processing a full granule of EMIT data (76.8 km x 74.52 km area corresponding to 1280x1242px, tiled into 100 tiles of 128x128px). IO time to load the hyperspectral tiles is kept separate in the last row. SF refers to SegFormer and EV to EfficientViT architecture.

	Unibap CPU	Rasp3 CPU	Rasp4 CPU	Xavier CPU	X. GPU (onnx)	X. GPU (trtexec)
HyperSTARCOP MF+RGB	203.3	1025.25	286.57	377	371.17	370.27
SF Base	15.3	38.82	9.19	7.87	1.62	0.49
SF UpConv	30.1	105.43	28.7	13.92	1.97	0.64
SF UpConvStride	119.5	399.76	107.87	48.38	2.95	1.37
EV Base	N/A	39.96	9.45	9.39	2.36	0.37
EV ConvUp	N/A	47.73	13.81	9.99	2.00	0.55
EV ConvUpStride	N/A	91.07	21.26	18.09	2.10	0.5
IO load time	4	39.99	12.88	6.13	6.13	6.13

environment. On NVIDIA Jetson AGX Xavier board we report results from running the models with several different python libraries. Runtime on the GPU is measured first using the ONNX format and the Onnxruntime library. We also report the times when using the Tensor RT engine which compiles the models directly on-board of the specific hardware of the device. Namely, this uses the “trtexec” tool with post training optimization of mixed precision quantification (allowing float32 and float16), which is similar to the steps reported in the codebase of [Cai *et al.*, 2023].

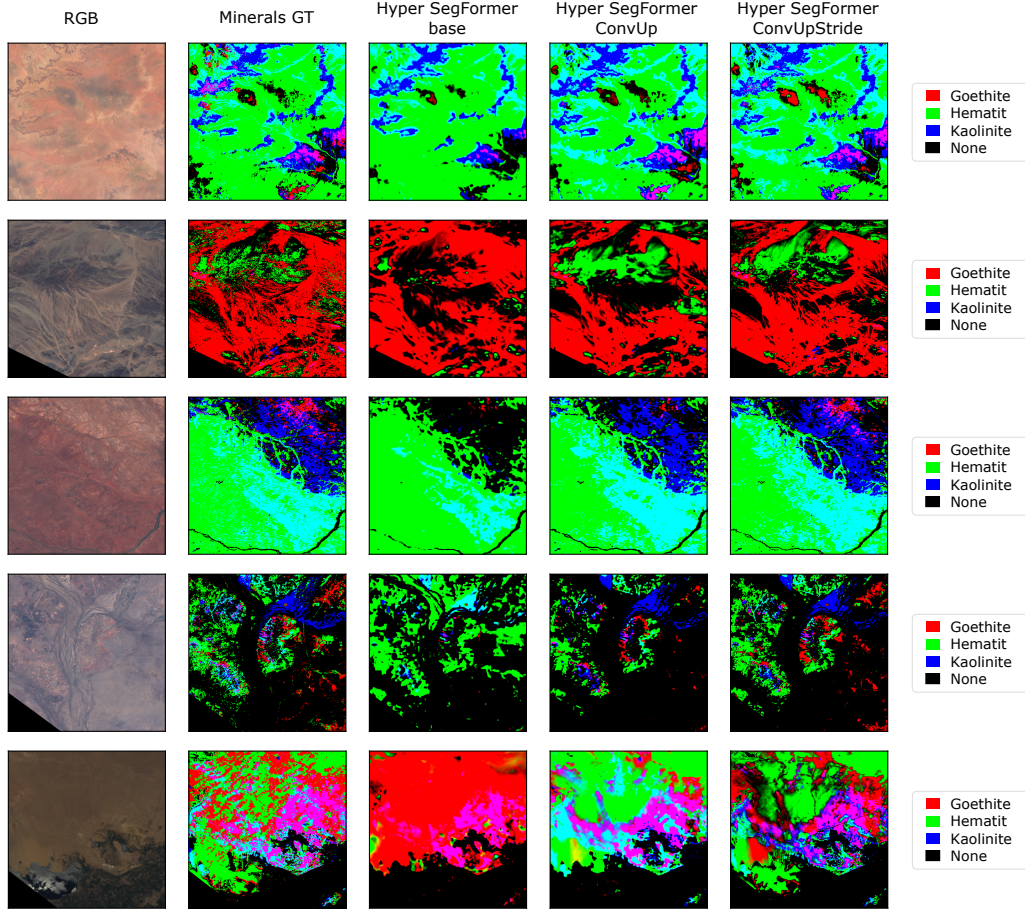


Figure 5.12: Qualitative results comparing our new models on the mineral dataset OxHyperMinerals showing our proposed Hyper SegFormer model variants (with output resolution degradation in default $\frac{1}{4}x$, “ConvUp” $\frac{1}{2}x$, and “ConvUpStride” full $1x$). The three mineral classes “Goethite”, “Hematite” and “Kaolinite” are mapped onto as the “Red”, “Blue” and “Green” channels for visualisation, model output is not thresholded and matches the saturation. Best viewed in color and zoomed in.

5.8 Discussion

When considering results of all of the conducted experiments, we confirm that end-to-end machine learning models are capable of outperforming all the other tested variants - including classical baseline methods and classical machine learning models depending on the products of these baselines. The basic versions of the end-to-end architectures are however not enough, on several occasions the vanilla version of these models doesn’t even reach the performance of the U-Net based HyperSTARCOP models using “MF+RGB”.

Our proposed HyperspectralViT adaptations are needed - they address identified limitations in the capability of processing spectral information of the data and by

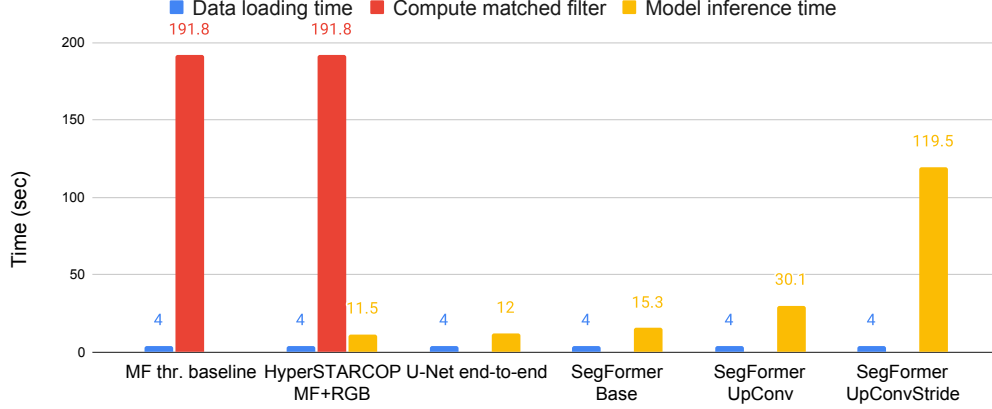


Figure 5.13: Detailed timings of different methods for processing a full granule of EMIT data (1280x1242px tiled into 100 tiles of 128x128px) decomposed into approximate IO load time, pre-processing time and the model inference.

maintaining spatial resolution. Often multiple combined adaptations are required for the best results.

5.8.1 Methane detection

On the task of methane detection, we report the new best performing models. In Table 5.2, on the newly created dataset of synthetic methane leak events in the EMIT data, our proposed HyperSegFormer (ConvUpStride variant) gains over 43.85% in the IoU score (and 27.88% in F1 and 20.7% in AUPRC) over the HyperSTARCOP MF+RGB model. Our model also achieves over 155% in IoU score (and 96.9% in F1 score) over the classical matched filter baseline. These improvements hold over all measured metrics. We see increasing scores when implementing proposed adaptations in both the SegFormer (from 60.26 F1 score to 74.27) and EfficientViT (from 68.40 F1 score to 72.26) models. We note that in Tables 5.3 and 5.4, we experienced even better performance when using larger encoder networks, however these might not be best suited for on-board deployment given their increased number of parameters.

Interestingly, when evaluating the models pre-trained on the dataset with synthetic events on a new dataset of real methane leak events collected by EMIT, we observe that the zero-shot generalisation performance is low (only two model variants reach the MF baseline). On Figure 5.9 we however see that with fine-tuning for 5-10 epochs, we are able significantly outperform both the classical baselines and all model variants trained from scratch. Namely the HyperSegFormer (ConvUpStride variant) improves the performance of the MF baseline by 12.9% in F1 score (and 17.1% in

IoU) and the same version of the model trained from scratch by over 6.9% in F1 score (and 8.8% in IoU). The HyperEfficientViT (ConvUp variant) outperforms the MF baselines by more than 23% in F1 (31.5% in IoU) and the same model variant trained from scratch by 19.2% in F1 (and 25.6% in IoU). We observe that while training from scratch for longer helps, it doesn’t reach the performance of using the models pre-trained on the synthetic dataset. Interestingly, the ConvUp variant slightly outperforms the ConvUpStride variants on this dataset with the EfficientViT models. Overall, we see, that on the task of detecting real methane leak events in EMIT data, we benefit from pre-training our models on a dataset of synthetically created events (especially in cases similar to ours, where the dataset of real events is relatively small).

When training our proposed models on the previously created benchmark dataset of real methane leak events in AVIRIS-NG data, we see some interesting mixed outcomes in Table 5.5. While the performance of our models drops on F1 score of strong methane leak events in contrast to the HyperSTARCOP “MF+RGB” model, on the metrics that aggregate all event sizes together, we see improvements. Namely, the HyperSegFormer (ConvUpStride variant) outperforms the U-Net MF based model by 12.9% in F1 score and by 18% in IoU score. The threshold-less AUPRC metric is also improved by over 17%. Similarly as in the case of the results presented in Table 5.2, while the EfficientViT model outperforms SegFormer in the base variant (without our adaptations for hyperspectral data) - the best results are still achieved by the fully adapted SegFormer model. The high performance of previous STARCOP models on strong methane leak events could be explained by the fact that the used matched filter method, mag1c, works better for large plumes (as it is iteratively increasing the contrast of positive pixels - this could however lead to lowered performance in weaker events).

We have encountered very large gains on the synthetic and real EMIT datasets, however smaller ones on the benchmark dataset of AVIRIS-NG data. This may be due to the smaller number of bands (and their corresponding spectral range) selected when training on the AVIRIS-NG data (60 bands, a choice made to match the bands used by mag1c filter). It has since been demonstrated also in Roger *et al.* [2023], that having access to more bands even outside of the typical methane visibility is beneficial. We also note that better versions of the MF product could lead to stronger baselines also in the EMIT data. For example [Foote *et al.*, 2021] explores scene specific adaptation of MF products, and similarly there could be some EMIT specific improvements for calculating MF.

5.8.2 Mineral identification

When exploring the Table 5.6 results of models trained on the dataset of minerals, we note smaller but sustained gains in performance due to the proposed adaptations. Namely the HyperSegFormer (ConvUpStride variant) improves the F1 score by over 3.5% in contrast to the vanilla model (and precision by 7.9%). The HyperEfficientViT demonstrates weaker performance for both base and adapted versions, however the used adaptations do improve this performance - the ConvUp variant scores 3.4% improvement in the F1 score.

Perhaps better improvement can be seen on the qualitative comparison between the models on Figure 5.12, where we show the effect of gradually improving output resolution with the HyperSegFormer model. This is visible namely in the presence of high frequency details in the predictions. The model variants using “Up” and “Stride” adaptations (and their combination) show sharper predictions than the vanilla version of the model (as that one is limited to the output size reduced by the factor of 4).

We note that our experiments with mineral identification are initial, as we had to rely on pseudo-ground truth labels generated by Tetracorder. But given other labelled datasets, our models are ready, and they can be re-trained.

5.8.3 On-board deployment

We have tested our model variants on a range of devices using different processing units. We highlight, that from all the tested devices only the environment available on the Unibap flatbeds has been tested in space, other are being proposed for space only for non-critical applications.

First, when considering model inference times measured only on the Unibap flatbed in Figure 5.13, we see that the matched filter computation is the slowest step influencing both the classical baselines and the HyperSTARCOP models. Our proposed end-to-end model variants avoid this step gaining vastly reduced inference time. Namely the HyperSegFormer ConvUp variant reduces the required time by over 85%, while the ConvUpStride variant reduces the time by over 41% - both compared against the HyperSTARCOP model and effectively also the classical MF baseline. We note that the HyperSegFormer models couldn’t run on the Myriad X VPU, which is why we used the device’s CPU instead.

One could also feasibly use non-iterative variants of the matched filter products, which could lead to faster computation, however this wouldn’t match the detection scores of our end-to-end models.

When considering other devices in Table 5.7, the MF calculation remains to be the bottleneck and the end-to-end models are faster. While using the CPU of the Raspberry Pi 3B+ the runtime is slower than on the Unibap flatbed, Raspberry Pi 4B reaches similar performance. Finally, the most powerful CPU of the NVIDIA’s Jetson AGX Xavier board shows the fastest runtimes of the tested CPU devices.

If we use our proposed HyperSegFormer ConvUp model on the Jetson AGX Xavier GPU, we achieve over 47x times speed-up over the Unibap’s machine CPU (e.g. 0.64 instead of 30.1 seconds per capture) and 317x times speed-up over the HyperSTAR-COP model (203.3 seconds per capture). Using the GPU of the Jetson AGX Xavier board provides the fastest runtimes from all tested devices. We observed further speedup when using the Tensor RT engine and compiling the neural network directly on the device with allowed mixed precision - in the case of the HyperSegFormer ConvUp, this is by over 70% in contrast to using the Onnxruntime library (without compilation and without mixed precision mode).

When using the Tensor RT inference engine, we also experience faster inference times when using the adapted EfficientViT architecture instead of the adapted SegFormer architecture - this is consistent with the fact the EfficientViT was designed to be fast in [Cai *et al.*, 2023]. Namely for the ConvUpStride variants, the HyperEfficientViT reduces the runtime by about 64% in contrast to the HyperSegFormer models. Similar reduction can also be observed on the CPU of the Raspberry Pi devices.

We consider these results as encouraging for showing the potential speedups that can be obtained when using GPUs in space, or when using the very cheap consumer hardware of Raspberry Pi’s with similar processing speeds as the dedicated chips. We however reiterate, that the deployment of these devices in the harsh environment of space has yet to be fully tested and if this processing pipeline becomes one of the critical mission goals, robust and radiation protected hardware would likely be needed.

For reference, the EMIT sensor collects up to 300 granules per day (and usually much less), while the planned SBG mission will produce an estimate of about 10x more. These granules have 1280x1242 px and can be tiled into 100 tiles of 128x128 which we used as reference when measuring model runtimes. With the estimated 300 granules and using the measurements from the Unibap machines, the mag1c matched filter baseline would run for almost 17 hours (with 203 seconds for each capture), while the proposed ConvUp variant of our HyperSegFormer would take just 2.5 hours (with 30 seconds for each capture). We highlight that with our model, one scene can

be processed in around half a minute (6.7x times speedup against the baseline). This time makes it more feasible to catch up with the estimated rate of data capture, we also assume that data loading could be handled on another thread as the author of this thesis explored previously in [Růžicka & Franchetti, 2018]. Further reduction to just estimated 3.2 minutes (with 0.64 seconds for each capture) could be achieved when using the GPU on the Jetson Xavier board. This outlines the potential speed-up future space missions could leverage if using high-power compute units (even if these are turned on just for short periods of time).

5.9 Summary

In summary, we propose **HyperspectralViTs**, new general machine learning architectures adapted to work with hyperspectral (imaging spectroscopy) data with end-to-end processing and three custom made architectural changes. We rigorously evaluate these different versions of adaptations applied on two modern Transformer based architectures. We demonstrate the generality of our approach by training the proposed models on two distinct tasks and three newly created **OxHyper** datasets from the hyperspectral sensor EMIT. We also confirm this on previously created benchmark dataset with data from other hyperspectral sensors. We identify limitations in the current landscape of semantic segmentation architectures and propose adaptations that address informational bottlenecks created from high spectral dimensionality of the data and the degradation of spatial resolution.

Our models have significantly improved the performance of methane leak detection in high resolution data, addressing an important real-world problem. We demonstrate this on a synthetic dataset of simulated events and confirm it on a newly collected dataset of real methane leak events observed by the EMIT instrument. We show that pre-training models on synthetic data can boost performance on small datasets of real events.

Finally, our proposed models provide a path towards autonomy on-board of low compute devices. We design model architectures with fast inference times and measure them on several target hardware configurations, including an environment which serves as a proxy of a real-world satellite. Our models could be used for a fast on-board detection of methane leak events. These could be followed with sending alerts via priority channels, downlink prioritization of relevant data and also for automatically triggering follow-up observations of other satellites. Time delay between the event observation and actual detection could be reduced from days to just minutes.

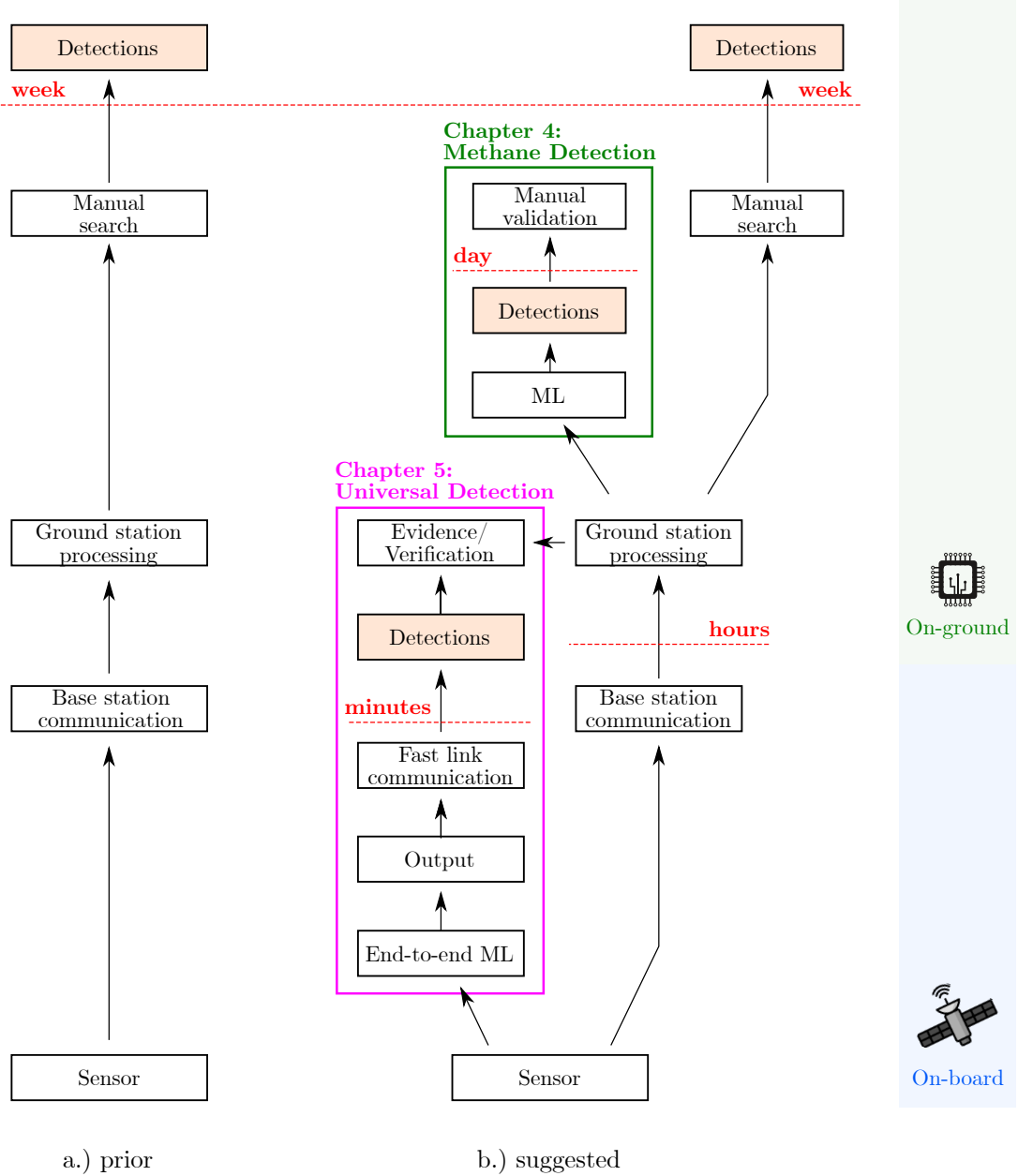


Figure 5.14: Illustration of the methane detection system deployed either on-board of the satellite, or with an on-ground element and the potential gained speed-ups in detections in contrast with the current traditional pipelines. Desired output of the system (here shown as “Detections”) can be obtained in vastly reduces timeframes, from current week(s) to just days or minutes depending on the implemented pathway. We note that these pathways are not exclusive, on-board detection system can be accompanied by the previously used traditional pipeline.

An illustration of the envisioned system is shown on Figure 5.14. Collecting more evidence due to autonomously scheduled observations could maximize the amount of

scientifically relevant data and could lead to much more accurate estimates of the behaviour of methane emissions.

To promote research in the domain of machine learning models used for hyperspectral data, we open source all our codes, and publish all created datasets. We also release all of the trained machine learning models, which we hope, could be used in real-world methane detection pipelines. One such example could be for example the United Nation’s Methane Alert and Response System (MARS) [Guanter *et al.*, 2023]. For practical deployment of our models, we also note that paradigms from life-long learning of semantic segmentation models would be useful (e.g. having a model which is compute efficiently updated weekly or monthly given newly detected methane leaks) - as is also noted by the survey of [Li *et al.*, 2024], this is an underexplored research direction, previously tackled in other disciplines [Mitchell *et al.*, 2018].

We identify several shortcomings of the presented experiments. Firstly, the collected dataset of real methane leak events could be extended with more manual labour (adding plume annotations in scenes which were missed in the original initial labelling), and by downloading more recent data captures. We however note, that having a small dataset was also useful to demonstrate how much can be gained from pre-training on synthetic data. Our models can also be trained on any of the future larger datasets. Secondly, the selected explored adaptations of machine learning architectures are not exhaustive. We present several directions and evaluate them in an ablation study, however more techniques could be implemented in the future. We note that the so far proposed adaptations are modular and will likely be of use in combination with any future ones (such as for example the recently published method by [Liu *et al.*, 2024] for semantic segmentation models). Thirdly, our created global dataset for mineral identification uses only initial pseudo-ground truth labels generated using the Tetracorder system and subsequential aggregation of few selected minerals. We however note that our proposed architectures are general, and as such, given future larger and potentially improved or manually validated datasets of mineral maps, they could also be re-trained.

We divide the potential future research directions into three dimensions: a.) research in the design of machine learning models, b.) application specific research for improved methane detection and c.) systems-based research in on-board deployment of the proposed models.

Firstly, there are more exciting directions to explore in the search for improved architecture design for processing hyperspectral data. This data domain significantly

differs from other examples in the discipline of Computer Vision, and techniques such as 3D convolutions [Yu *et al.*, 2020], sequential modelling [Zhao *et al.*, 2020] or other could be explored. Other machine learning architectures should also be explored, for example the modern members of the family of convolutional networks such as ConvNeXt [Liu *et al.*, 2022] or other models proposed directly by the Remote Sensing community. We however highlight that this research needs to be evaluated on large and ideally global datasets, we support this by publicly releasing all our used data as new benchmark datasets.

Additionally, using the newly available hyperspectral data from the near-global sensor EMIT, there are many potential tasks of interest (for example those relevant to goals of the SBG mission [Cawse-Nicholson *et al.*, 2021]) which could lead to creation of new multi-task datasets and proposal of new multi-task models. Performance on a single task of interest could be boosted by other similar tasks, as was demonstrated in Computer Vision datasets [Kokkinos, 2017]. Pre-training of large unsupervised machine learning models, or usage of so-called Foundation models [Lu *et al.*, 2024; Hong *et al.*, 2024] should also be explored. Namely the archive of EMIT data alone contains more than 131k raw data captures (in February 2025) at high resolutions with large number of spectral bands. Large pre-trained models could help with many downstream tasks.

Secondly, future directions can be explored in application specific research for improving methane detection and all associated tasks surrounding it. A much larger dataset of real methane plumes could be collected from the EMIT data, extended, and annotated with other labels of interests (such as for example plume quantifications, source point localisations and others). These could be added as auxiliary tasks learned by the models. Models allowing for uncertainty quantification of their predictions would be highly usable in real-world scenarios as they would provide better insight behind the decision these models are making. Methods such as model ensembles, or Monte Carlo Dropout could easily be added to the proposed semantic segmentation models, as previously explored in [Růžička *et al.*, 2020]. Sensor agnostic detection of methane leak events across different imaging spectroscopy instruments would also enable easier tracking of events worldwide and creation of more complex datasets.

Thirdly, real-world deployment of the proposed models on the current generation of hyperspectral satellites could enable increased autonomy and intelligent decision-making on-board. Delay between observation and action could be dramatically reduced, also reducing the amount of leaked methane gas. Similar general machine

learning architectures could be trained on other tasks of interest available for detection in hyperspectral data and could also be deployed. This could greatly increase the on-board capabilities of future hyperspectral satellites and deployed imaging spectroscopy instruments. The measured inference speeds reported in our work could be used to inform the performance on different target hardware configurations of future missions. Further significant speed-ups could be reached by follow-up deployment of our models on bespoke hardware and FPGA boards (as we encountered with works of previous chapters).

6

Conclusions

In the conclusion of this thesis we summarize all conducted research and connect it to the research questions outlined in Section 1.3. First, in Section 6.1, we explore achievements and highlight encountered limitations for each research question. Second, in Section 6.2, we use the lessons learned to outline some potential future work directions.

6.1 Discussion

In total, we presented three projects in this thesis connected to the topic of Machine Learning enabled intelligence on-board of communication constrained devices, in particular satellites. We started with general task of universal change detection in Chapter 3. There we deploy the proposed model on-board of a real satellite, gaining valuable insights into what is possible with current generation of accelerators used in space environments. We followed with a very specific application of methane leak detection in Chapter 4, where we explored data from different remote sensing sensors and evaluated their suitability for this task. Importantly, we have identified limitations of the current methods and use these to inform our work in the last chapter. Finally, in Chapter 5, we return to a general system which can be trained for a variety of downstream tasks, from which we explore two. During our work we create several new benchmark datasets needed for evaluation of our newly proposed models. We leverage the knowledge about low compute systems and design our models specifically to be compatible for on-board deployment.

6.1.1 Change detection on-board (Q1)

For unsupervised change detection on-board of satellites, we used the variational autoencoder (VAE) architecture, which proved to work well on a wide range of disaster events. Furthermore, the obtained feature encoding network allows retrieval of latent vectors which are useful for robust latent space comparisons relevant for change detection, but also several other tasks. Firstly, using the compressed features, we lower the storage requirements of the proposed system. Secondly, latent vectors can be used for communication between satellites in a constellation. In fact, a more advanced versions of autoencoder models are being proposed for data compression on-board in [Gómez & Meoni, 2024]. Thirdly, the lower dimensional representation allows us to train tiny classification models directly on-board of the satellites using just few labelled samples – this enables potential on-board adaptation and creation of whitelists and blacklists for further filtering of detected changes. We also observe that the models are by default serving as denoising autoencoders, and as such the latent encoding is also more robust to potential noise arising from low processing levels of data available on-board.

Potential limitation of the proposed system is the reliance on multi-temporal data available on-board. Even though on-board storage is relatively cheap, data synchronisation and sample to sample co-registration could prove to be costly. Furthermore, in scenarios where the detected event is clearly visible, a single image system might be more suitable, such as for example with floods in [Mateo-Garcia *et al.*, 2023] and fire events with L0 data in [Meoni *et al.*, 2023].

Lessons learned when addressing this research question are mainly coming from the fact that the proposed system was tested on-board of a real satellite. This gives us insights into the capabilities of hardware which is compatible with the harsh space environment, and which has been used in a number of similar satellite missions. We directly use some of these to inform our work on following research questions.

6.1.2 Methane detection in hyperspectral data (Q2)

We chose methane detection as an example of detection task which requires very accurate and specific sensors. The existing methods exhibit large amount of false positive and noisy detections and need manual intervention of domain experts. The proposed machine learning models improve the classical methods and allow for automated detections of methane leaks. To this end, we created a new benchmark

dataset with refined labels and several different supported sensor modalities, allowing a comparison between methods using multispectral and hyperspectral data.

We however observed that the performance of the proposed models has space for improvements, namely in contrast with classical computer vision tasks, where massive and highly qualitative benchmark datasets existed for years. We hope that future research works can leverage our data to improve on this task. We have discovered several bottlenecks which could be the cause of the lower performance. Firstly, we observed that to compute classically used methane enhancement products, particularly when using their improved versions which are more robust to in-scene noise, we require prohibitively long inference times. This prevents easy on-board deployment of methods depending on these classical products but is also relevant for future missions and the expected deluge of data. Secondly, we note that models solely relying on these classical products are limited by their quality, and while they can reduce potential false positive detections, they cannot extract new information from the source data. We use the insights learned about these two bottlenecks in our next research question.

6.1.3 Universal detectors (Q3)

We chose the data from imaging spectrometers, or hyperspectral instruments, as a suitable target for universal detection models. The wide VSWIR spectral range supports many downstream tasks and as such is also selected by several recent and upcoming missions. Using the lessons learned in the previous chapters, we propose **HyperspectralViT architectures**, modern machine learning models adapted for end-to-end learning with hyperspectral data. We evaluate these on three newly created **OxHyper datasets**, in detailed ablation studies on two different example tasks, methane detection and mineral identification. Excitingly, we discover that we are able to boost performance on real-world events when using models pre-trained on synthetic data. This is relevant for improving generalisability across diverse samples (with potentially poor sampling in some regions) and for any new sensors.

Finally, we explore the on-board deployment of the proposed models, showing large improvements over classical methods, but also the influence of the chosen base machine learning architecture. We deploy these models on low compute hardware previously tested in space environments, but also on a selection of commercial off-the-shelf (COTS) devices. These exhibit further potential speedups and have started to be explored for space missions in for example [Swope *et al.*, 2022].

We encounter limitations arising from smaller sizes of the used datasets. However, we have built these ourselves from the recently made public large archive of data from the hyperspectral sensor EMIT. These form the first truly diverse benchmark datasets, and we expect that their extension is possible and would further aid the performance of our proposed models. Annotations for new types of tasks would further extend the generality of these universal detector systems. During our work we also identify needs for potential other outputs relevant for the tasks of methane detection. These could be implemented as simple extensions of the current systems (e.g. methods for model uncertainty estimation and the task of quantification).

6.2 Future Work

Following the encountered limitations, here we propose several directions that could improve our methods and enable new capabilities.

Research conducted in Chapter 3 could be followed with more advanced models or different unsupervised learning approaches. Modern variants of the autoencoder architecture such as the VQ-VAE [Van Den Oord *et al.*, 2017] could lead to highly improved reconstruction quality. Models such as the MAE [He *et al.*, 2022] based on Transformer encoder, or specialised architectures proposed for processing Remote Sensing data such as [Reed *et al.*, 2023; Cong *et al.*, 2022] could also improve the quality of the latent vector embeddings, perhaps at a cost of increased computational requirements. Same small models, as those tested on-board of the satellite in our work, could also be trained using contrastive learning paradigms. We conducted some initial study for this in [Zhang *et al.*, 2022] and the work of [Herec, 2024] proposes a hybrid version between autoencoders and contrastive learning using our RaVAEn model architecture. The work of [Zhou *et al.*, 2024] also implements our model on FPGA boards, which may be more suitable for space environment with faster inference speeds and reduced power consumption. As was shown with our work on training machine learning models on-board satellites [Růžička *et al.*, 2023a], unsupervised models like RaVAEn could serve as general feature extractor systems used for other downstream tasks now enabled on-board.

Our research conducted on the task of automated methane leak detection from Chapters 4 and 5 creates some of the first large and diverse dataset benchmarks with near-global coverage. We expect these datasets to promote research in machine learning models used for this domain, and to be expanded with more annotated samples in the future. The sensors used to make these datasets are still operational

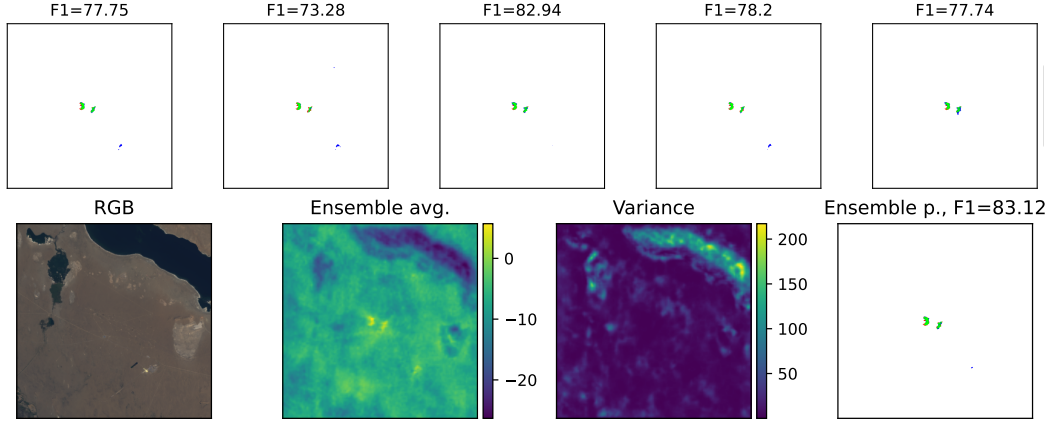


Figure 6.1: Visualisation of the uncertainty estimation method used with an ensemble of trained models. We compare the predictions of the 5 HyperEfficientViT models (which started with different initial weights) finetuned on the dataset of real methane leak events and we show the output of the Ensemble Variance method (e.g. used in [Beluch *et al.*, 2018; Růžička *et al.*, 2020]). First, the average of ensemble predictions can provide more robust detection [Izmailov *et al.*, 2018], majority vote could also be used to improve the score over samples (here shown with the F1 score over the predictions). We note that instead of average for example the *mode* operation could also be used. Second, variance of the predictions can also be used as a measure of uncertainty and agreement over the predicted scene. For example, on this scene we see highlighted region corresponding to area with the lake. None of the models in the ensemble predicted plume pixels there, but some amount of disagreement can be observed in the computed variance.

and provide up to 300 new observations each day. New sensors, such as SBG, are estimated to increase this data capture rate up to ten-fold. Our research from Chapter 5 confirms that simulated data can be used to boost the detection performance on real events. This would be highly relevant for maintaining generalisability of trained machine learning models in regions with otherwise low amount of real event samples, and for preparing pre-trained machine learning models for newly deployed sensors.

Furthermore, as illustrated by the Figure 2.6 in the introductory sections and by related background literature from Section 2.4, additional types of tasks relevant for the practical deployment of methane detection pipelines should be considered. Better semantic segmentation provided by our models can be used to aid follow-up tasks such as detected plume leak quantification, or an automated attribution to a known facility. Simple adaptations of the already trained machine learning models could also enable model uncertainty estimation, as previously illustrated with semantic segmentation models by the author of this thesis in [Růžička *et al.*, 2020] and on Figure 6.1. Our proposed machine learning model adaptations are modular and can

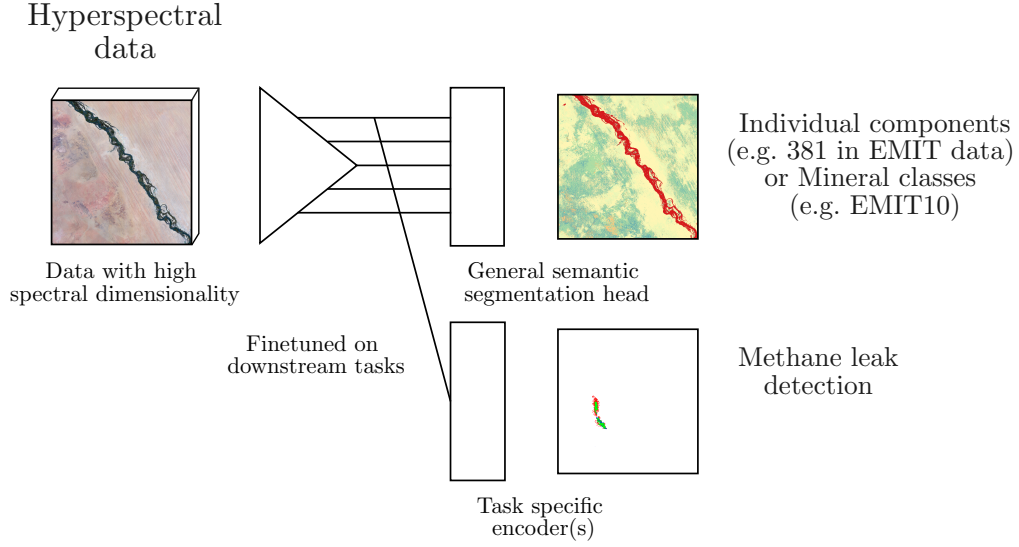


Figure 6.2: Proposal of pre-training Foundation models on general tasks such as fine-grained mineral identification and later using them for other specific task. The model could reuse frozen weights of in the encoder and train only the decoder network specifically tuned for target tasks. Alternatively, multi-task training regime could also be leveraged with several targeted downstream tasks.

be added to new architectures alongside other future adaptations. For on-the-ground detection pipelines larger models trained on future large datasets could also reach new state-of-the-art results.

Large archives of data from recent and upcoming hyperspectral sensors provides exciting opportunities for research in Foundation models extending our work from Chapter 5 on Universal detection models. Illustration on Figure 6.2 shows an example of pre-training models on general task of fine-grained detection of individual chemical signatures (generated for example as pseudo ground-truths using methods like Tetracorder [Clark *et al.*, 2003] as we show in Chapter 5). These would be later leveraged on other downstream tasks of interest (e.g. methane detection). The major current bottleneck is the lack of large, annotated datasets. As of now (February 2025), the EMIT data portal contains over 131k samples (estimated as around 263 TB of data) alongside with initial mineral class labels. These Universal detection models could be later adapted for multiple tasks (tapping into methodology from multi-task learning) or on data from multiple sensors allowing sensor agnosticity. Similarly as data harmonisation, this could effectively reduce the revisit time by processing data across future hyperspectral sensors such as SBG and CHIME, and potentially future aerial missions such as AVIRIS-3 and AVIRIS-4.

Our proposed models are prepared for on-board deployment on real hyperspectral

imaging missions and tested on typically used space compliant hardware. Future autonomous imaging spectroscopy missions could be enabled for Earth Observation, but also for deep space missions such as Europa Clipper, or missions to the Moon, Mars, or other planets. Further reductions of inference times can be obtained with additional methods for machine learning model compression, and by deployment on specialised boards or FPGAs.

6.2.1 Deploying machine learning models

Finally, real-world deployment of our models into existing pipelines could aid automation and help reduce the latency between observations and plume leak detections (anecdotally, the fastest reported detection with EMIT took 5 days between the data capture on 5.4.2024 and the identification by the NASA’s EMIT team on 9.4.2024, this might however be an optimistic scenario). Both on-the-ground and on-board automation could dramatically speed this process up. As was shown in prior cases with detected plume leaks, operators and governments can be contacted through established channels [Guanter *et al.*, 2023], and the broken part of the infrastructure can be fixed. Every saved day would reduce the quantity of greenhouse gasses emitted into the atmosphere, a task highly relevant for Climate Change goals limiting Global Warming [Lee *et al.*, 2023].

Figure 6.3 shows example deployment of our models from Chapter 4 inside pipelines and applications developed and used in the United Nations Methane Alert and Response System (MARS) unit [Guanter *et al.*, 2023]. Top image shows the user interface of the PlumeViewer application with a full tile prediction. Bottom image details the predictions inside this image. Red rectangles denote monitoring locations, red and orange dots information about the infrastructure. Note that this screenshot is from the development branch of the project and that the shown results are only in work-in-progress state.

As it currently stands, the typical processing chain used in this group starts with ingesting images from multispectral and hyperspectral satellites into the Plume Viewer system. For hyperspectral data, there wasn’t any machine learning model in place yet, a method based on thresholding of matched filter products was used. Various methane enhancement products (including the wide matched filter of Roger *et al.* [2023] and mag1c product of Foote *et al.* [2020]) are computed and saved as a visualisable layer. For multispectral data (namely Sentinel-2 and Landsat data), the model proposed by Vaughan *et al.* [2024a] is used on a list of manually created monitoring locations. After the initial processing, manual inspection by a team of domain

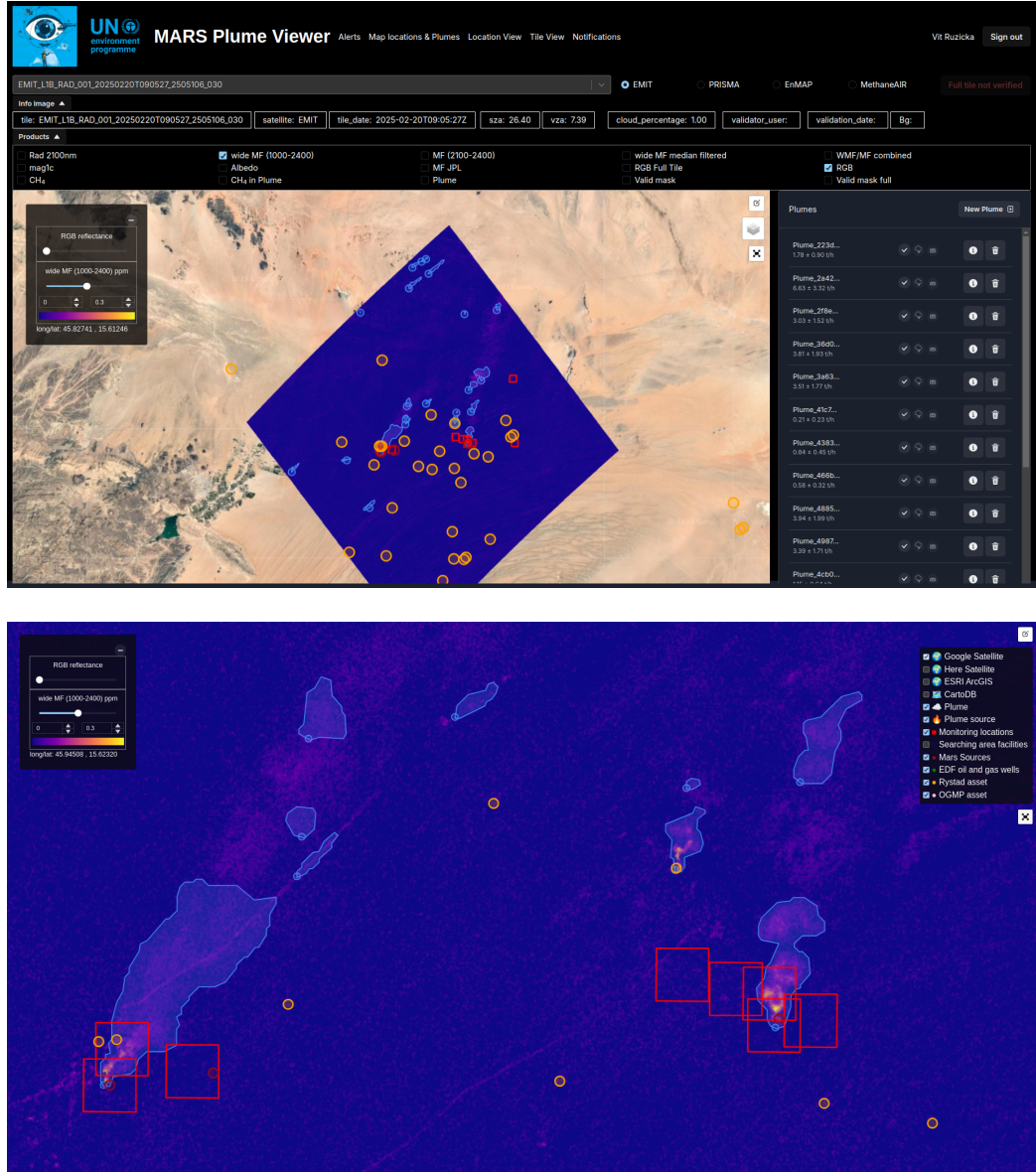


Figure 6.3: Deployment of the STARCOP models inside applications used by United Nations. We are showing the prediction of our work-in-progress models on the recent EMIT file “EMIT_L1B_RAD.001_20250220T090527_2505106_030.nc”. The shown methane enhancement product is the wide matched filter. The vectors outlining potential predicted plumes are computed from the predictions of our models (vectorised semantic segmentation masks).

experts is conducted. They have at their disposal alerts generated by automated methods and can also visually inspect all pre-computed methane enhancement products alongside a high resolution base map and information about the wind sourced from reanalysis products.

Finally, the validated plume detections are communicated through official notifications to the relevant actors (operators or governments) in the area and eventually (after a period of 30 calendar days after detection) reported in the public portal at <https://methanedata.unep.org/>. There, data can be accessed alongside with a document explaining the metadata and definitions used to classify these methane plumes. The portal reports plumes in data from a variety of sensors - multispectral (Sentinel-2, Landsat) and hyperspectral (EMIT, PRISMA, EnMAP) and also the geostationary GOES and low-spatial resolution TROPOMI satellites.

With the deployment of models proposed in Chapters 4 in real-world pipelines, we have experienced a large amount of false positive detection (such as the ones shown on Figure 6.3, still less than when using threshold baselines however). This is encountered more in cases, where we want to produce model predictions over the full granules of satellite data - in contrast to using these only in previously selected monitoring locations (these cover small and usually fixed area around locations with previously detected methane leaks). Models using the entire spectrum of satellite data, such as those proposed in Chapter 5 might bring reduction of these false positive detections. Additionally, recent pre-print of Xiang *et al.* [2024] explores methods to reduce the false positive detections given the knowledge of already predicted plumes - methods like these could be used in addition to existing models as a post-processing step.

Bibliography

- Aybar, Cesar, Ysuhuaylas, Luis, Loja, Jhomira, Gonzales, Karen, Herrera, Fernando, Bautista, Lesly, Yali, Roy, Flores, Angie, Diaz, Lissette, Cuenca, Nicole, *et al.* 2022. CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2. *Scientific data*, **9**(1), 782. 14, 32
- Aybar, Cesar, Mateo-García, Gonzalo, Acciarini, Giacomo, Růžicka, Vít, Meoni, Gabriele, Longépé, Nicolas, & Gómez-Chova, Luis. 2024. Onboard Cloud Detection and Atmospheric Correction With Efficient Deep Learning Models. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1–13. 20
- Ball, John E, Anderson, Derek T, & Chan, Chee Seng. 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of applied remote sensing*, **11**(4), 042609–042609. 36
- Ballé, Johannes, Laparra, Valero, & Simoncelli, Eero P. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*. 68
- Banerjee, Amit, Burlina, Philippe, & Diehl, Chris. 2006. A support vector method for anomaly detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **44**(8), 2282–2291. 53
- Barentsen, Geert, Harris, Joseph, Bourne, Hannah, Haag, Justin, Martin, Emily, Pritchett, Caroline, Giuliano, Paul, LeDuc, Dominic, Roth, Keely, Collison, Alan, *et al.* 2024. *Planet’s Hyperspectral Constellation: First Light Campaign and On-Orbit Calibration*. 100
- Barry, Pamela. 2001. EO-1 Hyperion Science Data User’s Guide, Level 1B. *TRW Space, Defence and Information Systems, Redondo Beach, CA*. 41
- Baumgardner, Marion F., Biehl, Larry L., & Landgrebe, David A. 2015 (Sep). *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. 77

- Bégaint, Jean, Racapé, Fabien, Feltman, Simon, & Pushparaja, Akshay. 2020. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*. 68
- Beluch, William H, Genewein, Tim, Nürnberger, Andreas, & Köhler, Jan M. 2018. The power of ensembles for active learning in image classification. *Pages 9368–9377 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 147
- Berger, Richard W, Bayles, Devin, Brown, Ronald, Doyle, Scott, Kazemzadeh, Abbas, Knowles, Ken, Moser, David, Rodgers, John, Saari, Brian, Stanley, Dan, *et al.* 2001. The rad750/sup tm/-a radiation hardened powerpc/sup tm/processor for high performance spaceborne applications. *Pages 2263–2272 of: 2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, vol. 5. IEEE. 4
- Bhattacharya, Bimal K, Green, Robert O, Rao, Sadasiva, Saxena, M, Sharma, Shweta, Kumar, K Ajay, Srinivasulu, P, Sharma, Shashikant, Dhar, D, Bandyopadhyay, S, *et al.* 2019. An overview of AVIRIS-NG airborne hyperspectral science campaign over India. *Current Science*, **116**(7), 1082–1088. 17
- Blaney, Diana L, Hibbitts, Karl, Diniega, Serina, Davies, Ashley Gerard, Clark, Roger N, Green, Robert O, Hedman, Matthew, Langevin, Yves, Lunine, Jonathan, McCord, Thomas B, *et al.* 2024. The mapping imaging spectrometer for Europa (MISE). *Space Science Reviews*, **220**(7), 80. 100
- Bovolo, Francesca, Camps-Valls, Gustavo, & Bruzzone, Lorenzo. 2010. A support vector domain method for change detection in multitemporal images. *Pattern Recognition Letters*, **31**(10), 1148–1154. 53
- Braham, Nassim Ait Ali, Albrecht, Conrad M, Mairal, Julien, Chanussot, Jocelyn, Wang, Yi, & Zhu, Xiao Xiang. 2024. SpectralEarth: Training Hyperspectral Foundation Models at Scale. *arXiv preprint arXiv:2408.08447*. 103
- Brigot, Guillaume, Colin-Koeniguer, Elise, Plyer, Aurélien, & Janez, Fabrice. 2016. Adaptation and evaluation of an optical flow method applied to coregistration of forest remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **9**(7), 2923–2939. 111

- Brodrick, Philip G., Clark, Roger. N., Swayze, Gregg A., Kokaly, Raymond, Meyer, John, Ehlman, Bethany, Keebler, Abigail, Thompson, David R., & Green, Robert O. 2023 (May). *EMIT L2b Algorithm: Mineral Detection and Related Products at the Pixel Scale [ATBD v1.0]*. Distributed by NASA. Accessed 2024-08-16. 107, 108
- Brown, Christopher F, Brumby, Steven P, Guzder-Williams, Brookie, Birch, Tanya, Hyde, Samantha Brooks, Mazzariello, Joseph, Czerwinski, Wanda, Pasquarella, Valerie J, Haertel, Robert, Ilyushchenko, Simon, *et al.* 2022. Dynamic World, Near real-time global 10 m land use land cover mapping. *Scientific Data*, **9**(1), 251. 57
- Bruno, Jack H, Jervis, Dylan, Varon, Daniel J, & Jacob, Daniel J. 2024. U-Plume: automated algorithm for plume detection and source quantification by satellite point-source imagers. *Atmospheric Measurement Techniques*, **17**(9), 2625–2636. 35, 36
- Buchwitz, Michael, Rozanov, Vladimir V, & Burrows, John P. 2000. A near-infrared optimized DOAS method for the fast global retrieval of atmospheric CH₄, CO, CO₂, H₂O, and N₂O total column amounts from SCIAMACHY Envisat-1 nadir radiances. *Journal of Geophysical Research: Atmospheres*, **105**(D12), 15231–15245. 34
- Cai, Han, Li, Junyan, Hu, Muyan, Gan, Chuang, & Han, Song. 2023. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. *Pages 17302–17313 of: Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22, 26, 27, 37, 43, 99, 104, 117, 124, 132, 137
- Carion, Nicolas, Massa, Francisco, Synnaeve, Gabriel, Usunier, Nicolas, Kirillov, Alexander, & Zagoruyko, Sergey. 2020. End-to-end object detection with transformers. *Pages 213–229 of: European conference on computer vision*. Springer. 22, 39
- Castano, Rebecca, Mazzoni, Dominic, Tang, Nghia, Greeley, Ron, Doggett, Thomas, Cichy, Ben, Chien, Steve, & Davies, Ashley. 2006. Onboard classifiers for science event detection on a remote sensing spacecraft. *Pages 845–851 of: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 40, 41

- Cawse-Nicholson, Kerry, Townsend, Philip A, Schimel, David, Assiri, Ali M, Blake, Pamela L, Buongiorno, Maria Fabrizia, Campbell, Petya, Carmon, Nimrod, Casey, Kimberly A, Correa-Pabón, Rosa Elvira, *et al.* 2021. NASA’s surface biology and geology designated observable: A perspective on surface imaging algorithms. *Remote sensing of environment*, **257**, 112349. 2, 18, 98, 100, 101, 141
- CCSDS. 2019. *Low-complexity lossless and near-lossless multispectral and hyperspectral image compression*. BLUE BOOK. 68
- Chen, Jieneng, Lu, Yongyi, Yu, Qihang, Luo, Xiangde, Adeli, Ehsan, Wang, Yan, Lu, Le, Yuille, Alan L, & Zhou, Yuyin. 2021. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*. 38
- Chen, Tianqi, Xu, Bing, Zhang, Chiyuan, & Guestrin, Carlos. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*. 45
- Chen, Ting, Kornblith, Simon, Norouzi, Mohammad, & Hinton, Geoffrey. 2020. *A Simple Framework for Contrastive Learning of Visual Representations*. 54, 71
- Chien, Steve, Sherwood, Rob, Tran, Daniel, Cichy, Benjamin, Rabideau, Gregg, Castano, Rebecca, Davis, Ashley, Mandl, Dan, Frye, Stuart, Trout, Bruce, *et al.* 2005. Using autonomy flight software to improve science return on Earth Observing One. *Journal of Aerospace Computing, Information, and Communication*, **2**(4), 196–216. 105
- Chollet, François. 2017. Xception: Deep learning with depthwise separable convolutions. *Pages 1251–1258 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 44
- Chulakadabba, Apisada, Sargent, Maryann, Lauvaux, Thomas, Benmergui, Joshua S, Franklin, Jonathan E, Chan Miller, Christopher, Wilzewski, Jonas S, Roche, Sébastien, Conway, Eamon, Souiri, Amir H, *et al.* 2023. Methane point source quantification using MethaneAIR: a new airborne imaging spectrometer. *Atmospheric Measurement Techniques*, **16**(23), 5771–5785. 34
- Ciampiconi, Lorenzo, Elwood, Adam, Leonardi, Marco, Mohamed, Ashraf, & Rozza, Alessandro. 2023. A survey and taxonomy of loss functions in machine learning. *arXiv preprint arXiv:2301.05579*. 25

- Clark, R. 1999. N., 1999. Spectroscopy of rocks and minerals, and principles of spectroscopy. *Manual of Remote Sensing. John Wiley & Sons, New York*, 3–58. 102
- Clark, Roger N, Swayze, Gregg A, Livo, K Eric, Kokaly, Raymond F, Sutley, Steve J, Dalton, J Brad, McDougal, Robert R, & Gent, Carol A. 2003. Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems. *Journal of Geophysical Research: Planets*, **108**(E12). 101, 104, 107, 109, 115, 116, 148
- Clark, Roger. N., Swayze, Gregg A., Realmuto, Vincent J., Brodrick, Philip G., & Thompson, David R. 2020 (April). *EMIT L2b Algorithm: Mineral Detection and Related Products at the Pixel Scale [ATBD v0.2]*. Distributed by NASA. Accessed 2020-04-15. 108, 109
- Claverie, Martin, Ju, Junchang, Masek, Jeffrey G, Dungan, Jennifer L, Vermote, Eric F, Roger, Jean-Claude, Skakun, Sergii V, & Justice, Christopher. 2018. The Harmonized Landsat and Sentinel-2 surface reflectance data set. *Remote sensing of environment*, **219**, 145–161. 101
- Cong, Yezhen, Khanna, Samar, Meng, Chenlin, Liu, Patrick, Rozi, Erik, He, Yutong, Burke, Marshall, Lobell, David, & Ermon, Stefano. 2022. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, **35**, 197–211. 39, 146
- Connelly, David S, Thompson, David R, Mahowald, Natalie M, Li, Longlei, Carmon, Nimrod, Okin, Gregory S, & Green, Robert O. 2021. The EMIT mission information yield for mineral dust radiative forcing. *Remote Sensing of Environment*, **258**, 112380. 109
- Cusworth, Daniel H., Duren, Riley M., Thorpe, Andrew K., Olson-Duvall, Winston, Heckler, Joseph, Chapman, John W., Eastwood, Michael L., Helmlinger, Mark C., Green, Robert O., Asner, Gregory P., Dennison, Philip E., & Miller, Charles E. 2021. Intermittency of Large Methane Emitters in the Permian Basin. *Environmental Science & Technology Letters*, **8**(7), 567–573. Publisher: American Chemical Society. 78, 79, 103
- Daudt, Rodrigo Caye, Le Saux, Bertr, & Boulch, Alexandre. 2018. Fully convolutional siamese networks for change detection. *Pages 4063–4067 of: 2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 53

- Deborah, Hilda, Richard, Noël, & Hardeberg, Jon Yngve. 2015. A comprehensive evaluation of spectral distance functions and metrics for hyperspectral image processing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **8**(6), 3224–3234. 115
- Del Prete, Roberto, Meoni, Gabriele, Longép  , Nicolas, Graziano, Maria Daniela, & Renga, Alfredo. 2023. First Results of Vessel Detection with Onboard Processing of Sentinel-2 Raw Data by Deep Learning. *Pages 6262–6265 of: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 20
- Dooley, Jonathan F, Minschwaner, Kenneth, Dubey, Manvendra K, El Abbadi, Sahar H, Sherwin, Evan D, Meyer, Aaron G, Follansbee, Emily, & Lee, James E. 2024. A new aerial approach for quantifying and attributing methane emissions: implementation and validation. *Atmospheric Measurement Techniques*, **17**(17), 5091–5111. 17
- Dosovitskiy, Alexey. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. 26, 27
- Du, Andrew, Doan, Anh-Dzung, Law, Yee Wei, & Chin, Tat-Jun. 2024. Domain Adaptation for Satellite-Borne Multispectral Cloud Detection. *Remote Sensing*, **16**(18), 3469. 54
- Duntelman, George H. 1989. *Principal components analysis*. Sage. 52, 58
- Duren, Riley M., Thorpe, Andrew K., Foster, Kelsey T., Rafiq, Talha, Hopkins, Francesca M., Yadav, Vineet, Bue, Brian D., Thompson, David R., Conley, Stephen, Colombi, Nadia K., Frankenberg, Christian, McCubbin, Ian B., Eastwood, Michael L., Falk, Matthias, Herner, Jorn D., Croes, Bart E., Green, Robert O., & Miller, Charles E. 2019. California’s methane super-emitters. *Nature*, **575**(7781), 180–184. Number: 7781 Publisher: Nature Publishing Group. 29, 75
- ED Chaves, Michel, CA Picoli, Michelle, & D. Sanches, Ieda. 2020. Recent applications of Landsat 8/OLI and Sentinel-2/MSI for land use and land cover mapping: A systematic review. *Remote Sensing*, **12**(18), 3062. 98
- Ehlmann, Bethany L, Klima, Rachel L, Seybold, Calina C, Klesh, Andrew T, Au, Mitchell H, Bender, Holly A, Bennett, C Lee, Blaney, Diana L, Bowles, Neil, Calcutt, Simon, *et al.* 2022. NASA’s Lunar Trailblazer Mission: A Pioneering Small

- Satellite for Lunar Water and Lunar Geology. *Pages 1–14 of: 2022 IEEE Aerospace Conference (AERO)*. IEEE. 100
- Estlin, Tara A, Bornstein, Benjamin J, Gaines, Daniel M, Anderson, Robert C, Thompson, David R, Burl, Michael, Castano, Rebecca, & Judd, Michele. 2012. Aegis automated science targeting for the mer opportunity rover. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **3**(3), 1–19. 40, 41
- Eyring, Veronika, Collins, William D, Gentine, Pierre, Barnes, Elizabeth A, Barreiro, Marcelo, Beucler, Tom, Bocquet, Marc, Bretherton, Christopher S, Christensen, Hannah M, Dagon, Katherine, *et al.* 2024. Pushing the frontiers in climate modelling and analysis with machine learning. *Nature Climate Change*, 1–13. 32
- Firat, Hüseyin, Asker, Mehmet Emin, Bayindir, Mehmet İlyas, & Hanbay, Davut. 2022. Spatial-spectral classification of hyperspectral remote sensing images using 3D CNN based LeNet-5 architecture. *Infrared Physics & Technology*, **127**, 104470. 38
- Flordal, Oskar, Synodinos, Aris, Herlitz, Mattias, Magnusson, Henrik, Steenari, David, Förster, Kyra, Castorina, Michele, George, Tom, Troxel, Ian, Reid, Simon, *et al.* 2021. SpaceCloud cloud computing and in-orbit demonstration. *In: European Workshop on On-Board Data Processing (OBDP)*. 42
- Foote, Markus D., Dennison, Philip E., Thorpe, Andrew K., Thompson, David R., Jongaramrungruang, Siraput, Frankenberg, Christian, & Joshi, Sarang C. 2020. Fast and Accurate Retrieval of Methane Concentration From Imaging Spectrometer Data Using Sparsity Prior. *IEEE Transactions on Geoscience and Remote Sensing*, **58**(9), 6480–6492. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. 33, 76, 78, 79, 85, 101, 110, 115, 149
- Foote, Markus D, Dennison, Philip E, Sullivan, Patrick R, O'Neill, Kelly B, Thorpe, Andrew K, Thompson, David R, Cusworth, Daniel H, Duren, Riley, & Joshi, Sarang C. 2021. Impact of scene-specific enhancement spectra on matched filter greenhouse gas retrievals from imaging spectroscopy. *Remote Sensing of Environment*, **264**, 112574. 135
- Francis, Alistair, & Czerkawski, Mikolaj. 2024. Major TOM: Expandable Datasets for Earth Observation. *arXiv preprint arXiv:2402.12095*. 22

- Frankenberg, Christian, Platt, Ulrich, & Wagner, T. 2005. Iterative maximum a posteriori (IMAP)-DOAS for retrieval of strongly absorbing trace gases: Model studies for CH₄ and CO₂ retrieval from near infrared spectra of SCIAMACHY onboard ENVISAT. *Atmospheric Chemistry and Physics*, **5**(1), 9–22. 34
- Frankenberg, Christian, Thorpe, Andrew K, Thompson, David R, Hulley, Glynn, Kort, Eric Adam, Vance, Nick, Borchardt, Jakob, Krings, Thomas, Gerilowski, Konstantin, Sweeney, Colm, *et al.* 2016. Airborne methane remote measurements reveal heavy-tail flux distribution in Four Corners region. *Proceedings of the national academy of sciences*, **113**(35), 9734–9739. 34
- Fuchs, Martin Hermann Paul, & Demir, Begüm. 2023. Hyspecnet-11k: A large-scale hyperspectral dataset for benchmarking learning-based hyperspectral image compression methods. *Pages 1779–1782 of: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 39, 103
- Furano, Gianluca, Meoni, Gabriele, Dunne, Aubrey, Moloney, David, Ferlet-Cavrois, Veronique, Tavoularis, Antonis, Byrne, Jonathan, Buckley, Léonie, Psarakis, Michalis, Voss, Kay-Obbe, *et al.* 2020. Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. *IEEE Aerospace and Electronic Systems Magazine*, **35**(12), 44–56. 3, 39, 40, 43, 45, 46, 47, 54
- Gewali, Utsav B, Monteiro, Sildomar T, & Saber, Eli. 2018. Machine learning based hyperspectral image analysis: a survey. *arXiv preprint arXiv:1802.08701*. 77
- Giuffrida, Gianluca, Diana, Lorenzo, de Gioia, Francesco, Benelli, Gionata, Meoni, Gabriele, Donati, Massimiliano, & Fanucci, Luca. 2020. Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images. *Remote Sensing*, **12**(14), 2205. 40, 49, 54
- Giuffrida, Gianluca, Fanucci, Luca, Meoni, Gabriele, Batič, Matej, Buckley, Léonie, Dunne, Aubrey, Van Dijk, Chris, Esposito, Marco, Hefe, John, Vercruyssen, Nathan, *et al.* 2021. The Φ -Sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, **60**, 1–14. 4, 40, 123
- Gómez, Pablo, & Meoni, Gabriele. 2024. Tackling the Satellite Downlink Bottleneck with Federated Onboard Learning of Image Compression. *Pages 6809–6818 of: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 43, 144

- Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>. 22, 24
- Gorroño, Javier, Varon, Daniel J., Irakulis-Loitxate, Itziar, & Guanter, Luis. 2022. Understanding the potential of Sentinel-2 for monitoring methane point emissions. *Atmospheric Measurement Techniques Discussions*, Oct., 1–25. Publisher: Copernicus GmbH. 110
- Gottfriedsen, Julia, Spichtinger, Andrea, Schoettl, Fabian, Franquinet, Julian, Langer, Martin, & Strebl, Johanna. 2023. Real-time on-orbit fire detection. *Pages NH41E-0775 of: AGU Fall Meeting Abstracts*, vol. 2023. 71
- Green, R. 2022. *EMIT L1B At-Sensor Calibrated Radiance and Geolocation Data 60 m V001*. Distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. Accessed 2023-09-19. 5, 6
- Green, R. 2023. *EMIT L2B Estimated Mineral Identification and Band Depth and Uncertainty 60 m V001*. Distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. Accessed 2024-08-16. 104, 107, 109
- Green, R., Thorpe, A., Brodrick, P., Chadwick, D., Lopez, A., Elder, C., Villanueva-Weeks, C., Fahlen, J., Coleman, R., Jensen, D., Bender, H., Vinckier, Q., Xiang, C., Olson-Duvall, W., Lundeen, S., & Thompson, D. 2023. *EMIT L2B Estimated Methane Plume Complexes 60 m V001 [Data set]*. Distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. Accessed 2024-08-16. 103, 111, 112
- Gretok, Evan W, & George, Alan D. 2021. Onboard multi-scale tile classification for satellites and other spacecraft. *Pages 110–121 of: 2021 IEEE Space Computing Conference (SCC)*. IEEE. 63, 124
- Groshenry, Alexis, Giron, Clement, Lauvaux, Thomas, d’Aspremont, Alexandre, & Ehret, Thibaud. 2022. Detecting Methane Plumes using PRISMA: Deep Learning Model and Data Augmentation. *arXiv preprint arXiv:2211.15429*. 35, 103
- Guanter, Luis, Irakulis-Loitxate, Itziar, Maasakkers, Joannes D, Aben, Ilse, Lelong, Christian, Randles, Cynthia A, Zavala-Araiza, Daniel, Hamburg, Steven P, & Caltagirone, Manfredi. 2023. Methane Alert and response system (MARS): IMEO’s satellite-based system for detection and attribution of methane point

- sources around the world. *Pages EGU-9548 of: EGU General Assembly Conference Abstracts*. 140, 149
- Guerrieri, Andrea, Kashani-Akhavan, Sahand, Lombardi, Pasquale, Belhadj, Bilel, & Ienne, Paolo. 2018. A dynamically reconfigurable platform for high-performance and low-power on-board processing. *Pages 74–81 of: 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE. 42
- Guertin, Steven M. 2022. Raspberry Pis for space guideline. *NASA: Washington, DC, USA*. 40, 123
- Guisiano, Jade E, Lauvaux, Thomas, Cifarelli, Claudio, Moulines, Éric, & Sublime, Jérémie. 2023. O&GProfile: An automated method for attribution of satellite methane emissions detections to oil and gas sites and operators. *In: International Conference on Machine Learning and Data Mining MLDM 2023*. 32
- Guo, Meng-Hao, Lu, Cheng-Ze, Hou, Qibin, Liu, Zhengning, Cheng, Ming-Ming, & Hu, Shi-Min. 2022. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, **35**, 1140–1156. 37
- Hagen, Nathan, & Kudenov, Michael W. 2013. Review of snapshot spectral imaging technologies. *Optical Engineering*, **52**(9), 090901–090901. 18
- Han, Song, Mao, Huizi, & Dally, William J. 2015a. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*. 44
- Han, Song, Pool, Jeff, Tran, John, & Dally, William. 2015b. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, **28**. 44
- Han, Song, Liu, Xingyu, Mao, Huizi, Pu, Jing, Pedram, Ardavan, Horowitz, Mark A, & Dally, William J. 2016. EIE: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, **44**(3), 243–254. 44
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2016. Deep residual learning for image recognition. *Pages 770–778 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 26, 27, 44

- He, Kaiming, Chen, Xinlei, Xie, Saining, Li, Yanghao, Dollár, Piotr, & Girshick, Ross. 2022. Masked autoencoders are scalable vision learners. *Pages 16000–16009 of: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 24, 27, 54, 71, 146
- He, Yihui, Zhang, Xiangyu, & Sun, Jian. 2017. Channel pruning for accelerating very deep neural networks. *Pages 1389–1397 of: Proceedings of the IEEE international conference on computer vision*. 44
- He, Yihui, Lin, Ji, Liu, Zhijian, Wang, Hanrui, Li, Li-Jia, & Han, Song. 2018. Amc: Automl for model compression and acceleration on mobile devices. *Pages 784–800 of: Proceedings of the European conference on computer vision (ECCV)*. 45
- Herec, Bc Jonáš. 2024. On-Board Change Detection in Satellite Imagery: Leveraging Transfer Learning for Flood Detection. *Master thesis at Masaryk University, Department of Machine Learning and Data Processing*. 71, 146
- Hersbach, Hans, Bell, Bill, Berrisford, Paul, Hirahara, Shoji, Horányi, András, Muñoz-Sabater, Joaquín, Nicolas, Julien, Peubey, Carole, Radu, Raluca, Schepers, Dinand, *et al.* 2020. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, **146**(730), 1999–2049. 34
- Hinton, Geoffrey, Vinyals, Oriol, & Dean, Jeff. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*. 44
- Hochreiter, S. 1997. Long Short-term Memory. *Neural Computation MIT-Press*. 26
- Hoersch, Bianca. 2013. *Sentinel-2 User Handbook*. Accessed on 7.9.2024. 55
- Hong, Danfeng, Han, Zhu, Yao, Jing, Gao, Lianru, Zhang, Bing, Plaza, Antonio, & Chanussot, Jocelyn. 2021. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, **60**, 1–15. 39
- Hong, Danfeng, Zhang, Bing, Li, Xuyang, Li, Yuxuan, Li, Chenyu, Yao, Jing, Yokoya, Naoto, Li, Hao, Ghamisi, Pedram, Jia, Xiuping, *et al.* 2024. SpectralGPT: Spectral remote sensing foundation model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 141

- Hu, Jie, Shen, Li, & Sun, Gang. 2018. Squeeze-and-excitation networks. *Pages 7132–7141 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 38
- Irakulis-Loitxate, Itziar, Guanter, Luis, Maasakkers, Joannes D, Zavala-Araiza, Daniel, & Aben, Ilse. 2022. Satellites detect abatable super-emissions in one of the world’s largest methane hotspot regions. *Environmental science & technology*, **56**(4), 2143–2152. 21, 29
- Izmailov, Pavel, Podoprikin, Dmitrii, Garipov, Timur, Vetrov, Dmitry, & Wilson, Andrew Gordon. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*. 147
- Jahan, Ismot, Mehana, Mohamed, Matheou, Georgios, & Viswanathan, Hari. 2024. Deep Learning-Based quantifications of methane emissions with field applications. *International Journal of Applied Earth Observation and Geoinformation*, **132**, 104018. 35, 36
- Jain, Jitesh, Li, Jiachen, Chiu, Mang Tik, Hassani, Ali, Orlov, Nikita, & Shi, Humphrey. 2023. Oneformer: One transformer to rule universal image segmentation. *Pages 2989–2998 of: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 37
- Johnson, R D, & Kasischke, ES. 1998. Change vector analysis: A technique for the multispectral monitoring of land cover and condition. *International journal of remote sensing*, **19**(3), 411–426. 52
- Jongaramrungruang, Siraput, Thorpe, Andrew K., Matheou, Georgios, & Frankenberg, Christian. 2022. MethaNet – An AI-driven approach to quantifying methane point-source emission from high-resolution 2-D plume imagery. *Remote Sensing of Environment*, **269**(Feb.), 112809. 35, 36, 38, 103, 110
- Joyce, Peter, Ruiz Villena, Cristina, Huang, Yahui, Webb, Alex, Gloor, Manuel, Wagner, Fabien Hubert, Chipperfield, Martyn P., Barrio Guilló, Rocío, Wilson, Chris, & Boesch, Hartmut. 2022. Using a deep neural network to detect methane point sources and quantify emissions from PRISMA hyperspectral satellite images. *EGUsphere*, Nov., 1–22. Publisher: Copernicus GmbH. 32, 35, 36, 103

- Kerner, Hannah R, Wagstaff, Kiri L, Bue, Brian D, Wellington, Danika F, Jacob, Samantha, Horton, Paul, Bell, James F, Kwan, Chiman, & Amor, Heni Ben. 2020. Comparison of novelty detection methods for multispectral images in rover-based planetary exploration missions. *Data Mining and Knowledge Discovery*, **34**(6), 1642–1675. 100
- Kingma, Diederik P, & Ba, Jimmy. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 25, 62
- Kingma, Diederik P., & Welling, Max. 2014. Auto-Encoding Variational Bayes. *In: Bengio, Yoshua, & LeCun, Yann (eds), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 24, 27, 50, 53, 60, 62
- Kirillov, Alexander, He, Kaiming, Girshick, Ross, Rother, Carsten, & Dollár, Piotr. 2019. Panoptic segmentation. *Pages 9404–9413 of: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 32
- Kokkinos, Iasonas. 2017. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *Pages 6129–6138 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 32, 141
- Kothari, Vivek, Liberis, Edgar, & Lane, Nicholas D. 2020. The final frontier: Deep learning in space. *Pages 45–49 of: Proceedings of the 21st international workshop on mobile computing systems and applications*. 39, 45, 46
- Krizhevsky, Alex, Sutskever, Ilya, & Hinton, Geoffrey E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, **25**. 23, 26, 27, 32, 53, 116
- Kuhlmann, Gerrit, Brunner, Dominik, Emmenegger, Lukas, Schwietzke, Stefan, Zavala-Araiza, Daniel, Thorpe, Andrew, Hueni, Andreas, & Röckmann, Thomas. 2023. Quantifying methane super-emitters from oil and gas production in Romania with the AVIRIS-NG imaging spectrometer. *Pages EGU–6751 of: EGU General Assembly Conference Abstracts*. 17

- Kumar, Satish, Kingwill, William, Mouton, Rozanne, Adamczyk, Wojciech, Hupertz, Robert, & Sherwin, Evan D. 2022. Guided Transformer Network for Detecting Methane Emissions in Sentinel-2 Satellite Imagery. *In: NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*. 35
- Kumar, Satish, Arevalo, Ivan, Iftexhar, ASM, & Manjunath, BS. 2023. Methanemap: Spectral absorption aware hyperspectral transformer for methane detection. *Pages 17609–17618 of: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 35, 39, 103
- Kuylenstierna, Johan CI, Michalopoulou, Eleni, & Malley, Chris. 2021. *Global Methane Assessment: Benefits and costs of mitigating methane emissions*. 29, 73, 75
- Lam, Remi, Sanchez-Gonzalez, Alvaro, Willson, Matthew, Wirnsberger, Peter, Fortunato, Meire, Alet, Ferran, Ravuri, Suman, Ewalds, Timo, Eaton-Rosen, Zach, Hu, Weihua, *et al.* 2022. GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*. 33
- Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, & Winther, Ole. 2016. Autoencoding beyond pixels using a learned similarity metric. *Pages 1558–1566 of: International conference on machine learning*. PMLR. 54
- Lebedev, Vadim, Ganin, Yaroslav, Rakhuba, Maksim, Oseledets, Ivan, & Lempitsky, Victor. 2014. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*. 45
- Lee, Hoesung, Calvin, Katherine, Dasgupta, Dipak, Krinner, Gerhard, Mukherji, Aditi, Thorne, Peter, Trisos, Christopher, Romero, José, Aldunce, Paulina, Barret, Ko, *et al.* 2023. IPCC, 2023: Climate Change 2023: Synthesis Report, Summary for Policymakers. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland. 27, 28, 75, 149
- Li, Xiangtai, Ding, Henghui, Yuan, Haobo, Zhang, Wenwei, Pang, Jiangmiao, Cheng, Guangliang, Chen, Kai, Liu, Ziwei, & Loy, Chen Change. 2024. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 38, 118, 140

- Lin, Ji, Zhu, Ligeng, Chen, Wei-Ming, Wang, Wei-Chen, & Han, Song. 2023. Tiny machine learning: progress and futures [feature]. *IEEE Circuits and Systems Magazine*, **23**(3), 8–34. 43, 44, 45, 61
- Liu, Chenxi, Zoph, Barret, Neumann, Maxim, Shlens, Jonathon, Hua, Wei, Li, Li-Jia, Fei-Fei, Li, Yuille, Alan, Huang, Jonathan, & Murphy, Kevin. 2018. Progressive neural architecture search. *Pages 19–34 of: Proceedings of the European conference on computer vision (ECCV)*. 45
- Liu, Zhijian, Zhang, Zhuoyang, Khaki, Samir, Yang, Shang, Tang, Haotian, Xu, Chenfeng, Keutzer, Kurt, & Han, Song. 2024. Sparse Refinement for Efficient High-Resolution Semantic Segmentation. *arXiv preprint arXiv:2407.19014*. 140
- Liu, Zhuang, Mao, Hanzi, Wu, Chao-Yuan, Feichtenhofer, Christoph, Darrell, Trevor, & Xie, Saining. 2022. A convnet for the 2020s. *Pages 11976–11986 of: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 27, 141
- Llaveria, David, Longepe, Nicolas, Meoni, Gabriele, Del Prete, Roberto, & Camps, Adriano. 2023. Band Selection Neural Network-Based Methodology Using L0 Data. *Pages 1879–1882 of: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 20
- Lofqvist, Martina, & Cano, José. 2020. Accelerating deep learning applications in space. *arXiv preprint arXiv:2007.11089*. 40, 124
- Loshchilov, Ilya, & Hutter, Frank. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. 25
- Lu, D., Mausel, P., Brondizio, E., & Moran, E. 2004. Change detection techniques. *International Journal of Remote Sensing*, **25**(12), 2365–2401. 51
- Lu, Siqi, Guo, Junlin, Zimmer-Dauphinee, James R, Nieusma, Jordan M, Wang, Xiao, VanValkenburgh, Parker, Wernke, Steven A, & Huo, Yuankai. 2024. AI Foundation Models in Remote Sensing: A Survey. *arXiv preprint arXiv:2408.03464*. 72, 141
- Ma, Ningning, Zhang, Xiangyu, Zheng, Hai-Tao, & Sun, Jian. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *Pages 116–131 of: Proceedings of the European conference on computer vision (ECCV)*. 43
- Maini, Anil K., & Agrawal, Varsha. 2014. *Satellite Hardware*. Wiley Telecom. Pages 174–228. 4

- Manas, Oscar, Lacoste, Alexandre, Giró-i Nieto, Xavier, Vazquez, David, & Rodriguez, Pau. 2021. Seasonal contrast: Unsupervised pre-training from uncured remote sensing data. *Pages 9414–9423 of: Proceedings of the IEEE/CVF International Conference on Computer Vision*. 54, 71
- Marin, Alessandro, Coelho, César, Deconinck, Florian, Babkina, Irina, Longepe, Nicolas, & Pastena, Massimiliano. 2021. Phi-Sat-2: Onboard AI apps for earth observation. *Proc. Space Artif. Intell.* 4, 40, 42
- Markidis, Stefano, Der Chien, Steven Wei, Laure, Erwin, Peng, Ivy Bo, & Vetter, Jeffrey S. 2018. Nvidia tensor core programmability, performance & precision. *Pages 522–531 of: 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE. 45
- Mateo-Garcia, Gonzalo, Veitch-Michaelis, Joshua, Smith, Lewis, Oprea, Silviu Vlad, Schumann, Guy, Gal, Yarin, Baydin, Atılım Güneş, & Backes, Dietmar. 2021. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific reports*, **11**(1), 1–12. 42, 49, 54, 55, 105
- Mateo-Garcia, Gonzalo, Veitch-Michaelis, Josh, Purcell, Cormac, Longepe, Nicolas, Reid, Simon, Anlind, Alice, Bruhn, Fredrik, Parr, James, & Mathieu, Pierre Philippe. 2023. In-orbit demonstration of a re-trainable machine learning payload for processing optical imagery. *Scientific Reports*, **13**(1), 10391. 4, 40, 42, 144
- Matheou, Georgios, & Chung, Daniel. 2014. Large-eddy simulation of stratified turbulence. Part II: Application of the stretched-vortex model to the atmospheric boundary layer. *Journal of the Atmospheric Sciences*, **71**(12), 4439–4460. 103, 110
- Maxwell, Aaron E, Warner, Timothy A, & Fang, Fang. 2018. Implementation of machine-learning classification in remote sensing: An applied review. *International journal of remote sensing*, **39**(9), 2784–2817. 36
- McInnes, Leland, Healy, John, & Melville, James. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*. 67
- Menghani, Gaurav. 2021. Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. *arXiv preprint arXiv:2106.08962*. 44

- Meoni, Gabriele, Del Prete, Roberto, Serva, Federico, De Beussche, Alix, Colin, Olivier, & Longép , Nicolas. 2023. THRawS: A Novel Dataset for Thermal Hotspots Detection in Raw Sentinel-2 Data. *arXiv preprint arXiv:2305.11891*. 20, 55, 144
- Meoni, Gabriele, M rtens, Marcus, Derksen, Dawa, See, Kenneth, Lighthouse, Toby, S cher, Anthony, Martin, Arnaud, Rijlaarsdam, David, Fanizza, Vincenzo, & Izzo, Dario. 2024. The OPS-SAT case: A data-centric competition for onboard satellite image classification. *Astrodynamics*, 1–22. 2, 40, 54
- Miller, Christopher Chan, Roche, S bastien, Wilzewski, Jonas S, Liu, Xiong, Chance, Kelly, Souri, Amir H, Conway, Eamon, Luo, Bingkun, Samra, Jenna, Hawthorne, Jacob, *et al.* 2024. Methane retrieval from methaneair using the CO2 proxy approach: a demonstration for the upcoming methanesat mission. *Atmospheric Measurement Techniques*, **17**(18), 5429–5454. 20, 100
- Mitchell, Tom, Cohen, William, Hruschka, Estevam, Talukdar, Partha, Yang, Bis-han, Betteridge, Justin, Carlson, Andrew, Dalvi, Bhavana, Gardner, Matt, Kisiel, Bryan, *et al.* 2018. Never-ending learning. *Communications of the ACM*, **61**(5), 103–115. 140
- Molod, Andrea, Takacs, Lawrence, Suarez, Max, Bacmeister, Julio, Song, In-Sun, & Eichmann, Andrew. 2012. *The GEOS-5 atmospheric general circulation model: Mean climate and development from MERRA to Fortuna*. Tech. rept. NASA, God-dard Space Flight Center, Greenbelt, Maryland, USA. 34
- Moons, Bert, Goetschalckx, Koen, Van Berckelaer, Nick, & Verhelst, Marian. 2017. Minimum energy quantized neural networks. *Pages 1921–1925 of: 2017 51st Asilo-mar Conference on Signals, Systems, and Computers*. IEEE. 44
- Moons, Bert, Bankman, Daniel, & Verhelst, Marian. 2019. Embedded deep learning. *Embedded Deep Learning*. 45
- Mrziglod, John, & Francis, Alistair. 2019. *IRIS: Intelligently Reinforced Image Seg-mentation graphical user interface*. 79
- Murshed, MG Sarwar, Murphy, Christopher, Hou, Daqing, Khan, Nazar, Anantha-narayanan, Ganesh, & Hussain, Faraz. 2021. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)*, **54**(8), 1–37. 39, 40

- NASEM. 2019. *Thriving on our changing planet: A decadal strategy for Earth observation from space*. National Academies of Sciences and Division on Engineering and Physical Sciences and Space Studies Board and Committee on the Decadal Survey for Earth Science and Applications from Space. National Academies Press. 3, 30, 75, 101
- Nelson, James, Ames, Alan, Williams, Jason, Patschke, Robert, Mott, Claire, Joseph, James, Garon, Howard, & Mah, Grant. 2012. Landsat data continuity mission (LDCM) space to ground mission data architecture. *Pages 1–13 of: 2012 IEEE Aerospace Conference*. IEEE. 4
- Nemmour, Hassiba, & Chibani, Youcef. 2006. Multiple support vector machines for land cover change detection: An application for mapping urban extensions. *ISPRS Journal of Photogrammetry and Remote Sensing*, **61**(2), 125–133. 53
- Niu, Bowen, Feng, Quanlong, Chen, Boan, Ou, Cong, Liu, Yiming, & Yang, Jianyu. 2022. HSI-TransUNet: A transformer based semantic segmentation model for crop mapping from UAV hyperspectral imagery. *Computers and Electronics in Agriculture*, **201**, 107297. 38
- Njoku, Eni G. 2014. *Encyclopedia of remote sensing*. Springer New York. 16, 20
- Odena, Augustus, Dumoulin, Vincent, & Olah, Chris. 2016. Deconvolution and checkerboard artifacts. *Distill*, **1**(10), e3. Accessed on 1.9.2021. 61
- Pandey, Sudhanshu, van Nistelrooij, Maarten, Maasakkers, Joannes D, Sutar, Pratik, Houweling, Sander, Varon, Daniel J, Tol, Paul, Gains, David, Worden, John, & Aben, Ilse. 2023. Daily detection and quantification of methane leaks using Sentinel-3: a tiered satellite observation approach with Sentinel-2 and Sentinel-5p. *Remote Sensing of Environment*, **296**, 113716. 31
- Paoletti, ME, Haut, JM, Plaza, J, & Plaza, A. 2019. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, **158**, 279–317. 77
- Parr, James, Acciarini, Giacomo, Bridges, Chris, Mateo-García, Gonzalo, Portales-Julia, Enrique, Purcell, Cormac, Ružička, Vít, Spalding, Anne, & Veitch-Michaelis, Josh. 2024. Live Twinning: A Vision of ML Enabled Assets in Leo for Rapid Response to Natural Catastrophes. *Pages 685–688 of: IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*. 14, 39

- Pedivellano, Antonio, Sinn, Thomas, Raharijaona, Ambre, Kringer, Michael, Gruber, Joram, Schmidt, Joachim, Lund, Thomas, Titz, Alexander, Garcia, Diego, Stepanova, Daria, *et al.* 2023. Prototyping and Engineering Model Test Campaign of the 100W 1U PowerCube Deployable Solar Array. *Page 2696 of: AIAA SCITECH 2023 Forum.* 124
- Pham, D Keymeulen T, Klimesh, M, Allen, G, Flesch, G, Valencia, R, Xie, H, Kiely, A, Dolman, D, Roth, K, Crocker, K, *et al.* 2023. High-Performance Embedded System-on-a-Chip for space imaging spectrometer. *Pages 1–10 of: 2023 IEEE Aerospace Conference.* IEEE. 105
- Platt, Ulrich. 1994. *Differential Optical Absorption Spectroscopy (DOAS)*. Air Monitoring by Spectroscopic Techniques, John Wiley, New York. 34
- Platt, Ulrich, & Stutz, Jochen. 2008. *Differential absorption spectroscopy*. Springer. 34
- Poulet, F, Piccioni, G, Langevin, Y, Dumesnil, C, Tommasi, L, Carlier, V, Filacchione, G, Amoroso, M, Arondel, A, D'Aversa, E, *et al.* 2024. Moons and jupiter imaging spectrometer (MAJIS) on jupiter icy moons explorer (JUICE). *Space Science Reviews*, **220**(3), 27. 100
- Radke, Richard J, Andra, Srinivas, Al-Kofahi, Omar, & Roysam, Badrinath. 2005. Image change detection algorithms: a systematic survey. *IEEE transactions on image processing*, **14**(3), 294–307. 51, 53, 58
- Radman, Ali, Mahdianpari, Masoud, Varon, Daniel J, & Mohammadimanesh, Fariba. 2023. S2MetNet: A novel dataset and deep learning benchmark for methane point source quantification using Sentinel-2 satellite imagery. *Remote Sensing of Environment*, **295**, 113708. 35, 36
- Ramanswamy, V, Shine, Keith, Leovy, Conway, Wang, Wei-Chyung, Rodhe, Henning, Wuebbles, Donald J, Ding, M, Lelieveld, Joseph, Edmonds, Jae A, & McCormick, M Patrick. 1991. Radiative forcing of climate. *NASA, Washington, Scientific Assessment of Ozone Depletion: 1991.* 27
- Rankin, Arturo, Maimone, Mark, Biesiadecki, Jeffrey, Patel, Nikunj, Levine, Dan, & Toupet, Olivier. 2020. Driving curiosity: Mars rover mobility trends during the first seven years. *Pages 1–19 of: 2020 IEEE Aerospace Conference.* IEEE. 40

- Rapuano, Emilio, Meoni, Gabriele, Pacini, Tommaso, Dinelli, Gianmarco, Furano, Gianluca, Giuffrida, Gianluca, & Fanucci, Luca. 2021. An fpga-based hardware accelerator for cnns inference on board satellites: benchmarking with myriad 2-based solution for the cloudscout case study. *Remote Sensing*, **13**(8), 1518. 63
- Rasp, Stephan, Pritchard, Michael S, & Gentine, Pierre. 2018. Deep learning to represent subgrid processes in climate models. *Proceedings of the national academy of sciences*, **115**(39), 9684–9689. 33
- Rast, Michael, Nieke, Jens, Adams, Jennifer, Isola, Claudia, & Gascon, Ferran. 2021. Copernicus hyperspectral imaging mission for the environment (chime). *Pages 108–111 of: 2021 IEEE international geoscience and remote sensing symposium IGARSS*. IEEE. 2, 100, 101
- Rastegari, Mohammad, Ordonez, Vicente, Redmon, Joseph, & Farhadi, Ali. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. *Pages 525–542 of: European conference on computer vision*. Springer. 44
- Reed, Colorado J, Gupta, Ritwik, Li, Shufan, Brockman, Sarah, Funk, Christopher, Clipp, Brian, Keutzer, Kurt, Candido, Salvatore, Uyttendaele, Matt, & Darrell, Trevor. 2023. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. *Pages 4088–4099 of: Proceedings of the IEEE/CVF International Conference on Computer Vision*. 71, 146
- Reed, Irving S, & Yu, Xiaoli. 1990. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE transactions on acoustics, speech, and signal processing*, **38**(10), 1760–1770. 53
- Roger, Javier, Guanter, Luis, Gorroño, Javier, & Irakulis-Loitxate, Itziar. 2023. Exploiting the entire near-infrared spectral range to improve the detection of methane plumes with high-resolution imaging spectrometers. *Atmospheric Measurement Techniques Discussions*, **2023**, 1–21. 34, 135, 149
- Rolnick, David, Donti, Priya L, Kaack, Lynn H, Kochanski, Kelly, Lacoste, Alexandre, Sankaran, Kris, Ross, Andrew Slavin, Milojevic-Dupont, Nikola, Jaques, Natasha, Waldman-Brown, Anna, *et al.* 2022. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, **55**(2), 1–96. 36

- Ronneberger, Olaf, Fischer, Philipp, & Brox, Thomas. 2015. U-net: Convolutional networks for biomedical image segmentation. *Pages 234–241 of: International Conference on Medical image computing and computer-assisted intervention*. Springer. 26, 27, 37, 44, 86, 88, 104
- Rothman, Laurence S, Gordon, Iouli E, Babikov, Yury, Barbe, Alain, Benner, D Chris, Bernath, Peter F, Birk, Manfred, Bizzocchi, Luca, Boudon, Vincent, Brown, Linda R, *et al.* 2013. The HITRAN2012 molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, **130**, 4–50. 30, 76, 85
- Rouet-Leduc, Bertrand, & Hulbert, Claudia. 2024. Automatic detection of methane emissions in multispectral satellite imagery using a vision transformer. *Nature Communications*, **15**(1), 3801. 34, 35
- Růžička, Vít, & Franchetti, Franz. 2018. Fast and accurate object detection in high resolution 4K and 8K video using GPUs. *Pages 1–7 of: 2018 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE. 138
- Růžička, Vít, & Markham, Andrew. 2024. HyperspectralViTs: General Hyperspectral Models for On-board Remote Sensing. *arXiv preprint arXiv:2410.17248*. 12, 100
- Růžička, Vít, D’Aronco, Stefano, Wegner, Jan Dirk, & Schindler, Konrad. 2020. Deep active learning in remote sensing for data efficient change detection. *arXiv preprint arXiv:2008.11201*. 53, 104, 141, 147
- Růžička, Vít, Vaughan, Anna, De Martini, Daniele, Fulton, James, Salvatelli, Valentina, Bridges, Chris, Mateo-Garcia, Gonzalo, & Zantedeschi, Valentina. 2022. RaVÆn: unsupervised change detection of extreme events using ML on-board satellites. *Scientific reports*, **12**(1), 16939. 12, 14, 40, 55, 56, 71, 105
- Růžička, Vít, Mateo-García, Gonzalo, Bridges, Chris, Brunskill, Chris, Purcell, Cormac, Longépé, Nicolas, & Markham, Andrew. 2023a. Fast Model Inference and Training On-Board of Satellites. *Pages 2002–2005 of: IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*. 4, 12, 14, 40, 47, 146
- Růžička, Vít, Mateo-Garcia, Gonzalo, Gómez-Chova, Luis, Vaughan, Anna, Guanter, Luis, & Markham, Andrew. 2023b. Semantic segmentation of methane plumes with hyperspectral machine learning models. *Scientific Reports*, **13**(1), 19999. 12, 14, 32, 77, 79, 82, 89, 94, 103, 110, 125, 131

- Sandler, Mark, Howard, Andrew, Zhu, Menglong, Zhmoginov, Andrey, & Chen, Liang-Chieh. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. *Pages 4510–4520 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 43, 86
- Santos, Nuno Duro, & Gonçalves, Gil. 2014. Remote sensing applications based on satellite open data (Landsat 8 and Sentinel-2). *In: Conferencia Nacional de Geodesia Barreiro*. 98
- Schelkens, Peter, Bruylants, Tim, Temmermans, Frederik, Barbarien, Joeri, Doms, Ann, & Munteanu, Adrian. 2009. The JPEG 2000 family of standards. *Page 724802 of: Wavelet Applications in Industrial Processing VI*, vol. 7248. SPIE. 68
- Schmude, Johannes, Roy, Sujit, Trojak, Will, Jakubik, Johannes, Civitarese, Daniel Salles, Singh, Shraddha, Kuehnert, Julian, Ankur, Kumar, Gupta, Aman, Phillips, Christopher E, *et al.* 2024. Prithvi WxC: Foundation Model for Weather and Climate. *arXiv preprint arXiv:2409.13598*. 33
- Schuit, Berend J, Maasakkers, Joannes D, Bijl, Pieter, Mahapatra, Gourav, Van den Berg, Anne-Wil, Pandey, Sudhanshu, Lorente, Alba, Borsdorff, Tobias, Houweling, Sander, Varon, Daniel J, *et al.* 2023. Automated detection and monitoring of methane super-emitters using satellite data. *Atmospheric Chemistry and Physics Discussions*, **2023**, 1–47. 31, 35
- Sentinel Hub, Team. 2021. *s2cloudless: Sentinel Hub’s cloud detector for Sentinel-2 imagery*. Accessed on 4.11.2021. 56
- Serief, Chahira, Ghelamallah, Youcef, & Bentoutou, Youcef. 2023. Deep-Learning-Based System for Change Detection Onboard Earth Observation Small Satellites. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **16**, 8115–8124. 54
- Sharif Razavian, Ali, Azizpour, Hossein, Sullivan, Josephine, & Carlsson, Stefan. 2014. CNN features off-the-shelf: an astounding baseline for recognition. *Pages 806–813 of: Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 45
- Sherwin, Evan D, Rutherford, Jeffrey S, Chen, Yuanlei, Aminfard, Sam, Kort, Eric A, Jackson, Robert B, & Brandt, Adam R. 2023. Single-blind validation of space-based

- point-source detection and quantification of onshore methane emissions. *Scientific Reports*, **13**(1), 3836. 36
- Shirmard, Hojat, Farahbakhsh, Ehsan, Müller, R Dietmar, & Chandra, Rohitash. 2022. A review of machine learning in processing remote sensing data for mineral exploration. *Remote Sensing of Environment*, **268**, 112750. 104, 115
- Shrestha, Ajay, & Mahmood, Ausif. 2019. Review of deep learning algorithms and architectures. *IEEE access*, **7**, 53040–53065. 24, 26
- Shyu, Mei-Ling, Chen, Shu-Ching, Sarinnapakorn, Kanoksri, & Chang, LiWu. 2003. *A novel anomaly detection scheme based on principal component classifier*. Tech. rept. Miami Univ Coral Gables FL Dept of Electrical and Computer Engineering. 53
- Signoroni, Alberto, Savardi, Mattia, Baronio, Annalisa, & Benini, Sergio. 2019. Deep learning meets hyperspectral image analysis: A multidisciplinary review. *Journal of Imaging*, **5**(5), 52. 77, 102, 113
- Simonyan, Karen, & Zisserman, Andrew. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 27
- Slater, Windy S, Tiwari, Nayana P, Lovelly, Tyler M, & Mee, Jesse K. 2020. Total ionizing dose radiation testing of NVIDIA Jetson nano GPUs. *Pages 1–3 of: 2020 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE. 124
- Sumbul, Gencer, Charfuelan, Marcela, Demir, Begüm, & Markl, Volker. 2019. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. *Pages 5901–5904 of: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 53
- Sun, Hao, Zheng, Xiangtao, Lu, Xiaoqiang, & Wu, Siyuan. 2019. Spectral-spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, **58**(5), 3232–3245. 39, 105
- Swayze, Gregg A, Clark, Roger N, Goetz, Alexander FH, Livo, K Eric, Breit, George N, Kruse, Fred A, Sutley, Stephen J, Snee, Lawrence W, Lowers, Heather A, Post, James L, *et al.* 2014. Mapping advanced argillic alteration at Cuprite, Nevada, using imaging spectroscopy. *Economic geology*, **109**(5), 1179–1221. 109

- Swope, Jason, Mirza, Faiz, Dunkel, Emily, Towfic, Zaid, Chien, Steve, Russell, Damon, Sauvageau, Joe, Sheldon, Doug, Fernandez, Mark, & Knox, Carrie. 2022. Benchmarking remote sensing image processing and analysis on the snapdragon processor onboard the international space station. *Pages 5305–5308 of: IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 4, 40, 45, 145
- Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, & Rabinovich, Andrew. 2015. Going deeper with convolutions. *Pages 1–9 of: Proceedings of the IEEE conference on computer vision and pattern recognition*. 44
- Sánchez-García, Elena, Gorroño, Javier, Irakulis-Loitxate, Itziar, Varon, Daniel J., & Guanter, Luis. 2022. Mapping methane plumes at very high spatial resolution with the WorldView-3 satellite. *Atmospheric Measurement Techniques*, **15**(6), 1657–1674. Publisher: Copernicus GmbH. 2, 33, 76, 83, 110
- Thompson, D. R., & Brodrick, P. G. 2021. Realizing machine learning’s promise in geoscience remote sensing. *EOS*, **102**(July). 77, 113
- Thompson, David R, Bornstein, Benjamin J, Chien, Steve A, Schaffer, Steven, Tran, Daniel, Bue, Brian D, Castano, Rebecca, Gleeson, Damhnait F, & Noell, Aaron. 2012. Autonomous spectral discovery and mapping onboard the EO-1 spacecraft. *IEEE Transactions on Geoscience and Remote Sensing*, **51**(6), 3567–3579. 4, 40, 41, 105
- Thompson, David R, Altinok, Alphan, Bornstein, Ben, Chien, Steve A, Doubleday, Joshua, Bellardo, John, & Wagstaff, Kiri L. 2015a. Onboard machine learning classification of images by a cubesat in Earth orbit. *AI Matters*, **1**(4), 38–40. 40, 41
- Thompson, DR, Leifer, I, Bovensmann, H, Eastwood, M, Fladeland, M, Frankenberg, C, Gerilowski, K, Green, RO, Kratwurst, S, Krings, T, *et al.* 2015b. Real-time remote detection and measurement for airborne imaging spectroscopy: a case study with methane. *Atmospheric Measurement Techniques*, **8**(10), 4383–4397. 105
- Thorpe, AK, Frankenberg, C, & Roberts, DA. 2014. Retrieval techniques for airborne imaging of methane concentrations using high spatial and moderate spectral resolution: application to AVIRIS. *Atmospheric Measurement Techniques*, **7**(2), 491–506. 34

- Thorpe, Andrew K, Frankenberg, Christian, Aubrey, AD, Roberts, DA, Nottrott, AA, Rahn, TA, Sauer, JA, Dubey, MK, Costigan, KR, Arata, C, *et al.* 2016. Mapping methane concentrations from a controlled release experiment using the next generation airborne visible/infrared imaging spectrometer (AVIRIS-NG). *Remote Sensing of Environment*, **179**, 104–115. 36
- Thorpe, Andrew K, Frankenberg, Christian, Thompson, David R, Duren, Riley M, Aubrey, Andrew D, Bue, Brian D, Green, Robert O, Gerilowski, Konstantin, Krings, Thomas, Borchardt, Jakob, *et al.* 2017. Airborne DOAS retrievals of methane, carbon dioxide, and water vapor concentrations at high spatial resolution: application to AVIRIS-NG. *Atmospheric Measurement Techniques*, **10**(10), 3833–3850. 17, 34
- Thorpe, Andrew K., Green, Robert O., Thompson, David R., Brodrick, Philip G., Chapman, John W., Elder, Clayton D., Irakulis-Loitxate, Itziar, Cusworth, Daniel H., Ayasse, Alana K., Duren, Riley M., Frankenberg, Christian, Guanter, Luis, Worden, John R., Dennison, Philip E., Roberts, Dar A., Chadwick, K. Dana, Eastwood, Michael L., Fahlen, Jay E., & Miller, Charles E. 2023. Attribution of individual methane and carbon dioxide emission sources using EMIT observations from space. *Science Advances*, **9**(46), eadh2391. 30
- Thorpe, Andrew K., Green, Robert O., Thompson, David R., Brodrick, Philip G., Chlus, Adam, Fahlen, Jay E., Willow Coleman, Red, Chadwick, K. Dana, & Eastwood, Michael L. 2024. Attributing Methane and CO₂ Plumes by Emission Sector with the EMIT and AVIRIS-3 Imaging Spectrometers. *Pages 324–330 of: IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*. 20
- Thuillier, G., Hersé, M., Labs, D., Foujols, T., Peetermans, W., Gillotay, D., Simon, P.C., & Mandel, H. 2003. The Solar Spectral Irradiance from 200 to 2400 nm as Measured by the SOLSPEC Spectrometer from the Atlas and Eureka Missions. *Solar Physics*, **214**(1), 1–22. 81
- Tiemann, Enno, Zhou, Shanyu, Kläser, Alexander, Heidler, Konrad, Schneider, Rochelle, & Zhu, Xiao Xiang. 2024. *Machine Learning for Methane Detection and Quantification from Space - A survey*. 20, 34, 36
- Tuia, Devis, Schindler, Konrad, Demir, Begüm, Zhu, Xiao Xiang, Kochupillai, Mri-nalini, Džeroski, Sašo, van Rijn, Jan N, Hoos, Holger H, Del Frate, Fabio, Datcu, Mihai, *et al.* 2024. Artificial Intelligence to Advance Earth Observation: A review

- of models, recent trends, and pathways forward. *IEEE Geoscience and Remote Sensing Magazine*. 24, 36
- Tuli, Shikhar, Dasgupta, Ishita, Grant, Erin, & Griffiths, Thomas L. 2021. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*. 26
- Tzanetos, Theodore, Aung, MiMi, Balaram, J, Grip, Havard Fjrer, Karras, Jaakko T, Canham, Timothy K, Kubiak, Gerik, Anderson, Joshua, Merewether, Gene, Starch, Michael, *et al.* 2022. Ingenuity mars helicopter: From technology demonstration to extraterrestrial scout. *Pages 01–19 of: 2022 IEEE Aerospace Conference (AERO)*. IEEE. 40, 42
- Van Den Oord, Aaron, Vinyals, Oriol, *et al.* 2017. Neural discrete representation learning. *Advances in neural information processing systems*, **30**. 71, 146
- Vanselow, Steffen, Schneising, Oliver, Buchwitz, Michael, Reuter, Maximilian, Bovensmann, Heinrich, Boesch, Hartmut, & Burrows, John P. 2024. Automated detection of regions with persistently enhanced methane concentrations using Sentinel-5 Precursor satellite data. *Atmospheric Chemistry and Physics*, **24**(18), 10441–10473. 34
- Varon, Daniel J, Jacob, Daniel J, McKeever, Jason, Jervis, Dylan, Durak, Berke OA, Xia, Yan, & Huang, Yi. 2018. Quantifying methane point sources from fine-scale satellite observations of atmospheric methane plumes. *Atmospheric Measurement Techniques*, **11**(10), 5673–5686. 34
- Varon, Daniel J., Jervis, Dylan, McKeever, Jason, Spence, Ian, Gains, David, & Jacob, Daniel J. 2021. High-frequency monitoring of anomalous methane point sources with multispectral Sentinel-2 satellite observations. *Atmospheric Measurement Techniques*, **14**(4), 2771–2785. Publisher: Copernicus GmbH. 31, 33, 34, 76, 81, 83, 93
- Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, & Polosukhin, Illia. 2017. Attention is all you need. *Advances in neural information processing systems*, **30**. 26, 37
- Vaughan, Anna, Mateo-Garcia, Gonzalo, Irakulis-Loitxate, Itziar, Watine, Marc, Fernandez-Poblaciones, Pablo, Turner, Richard E, Requeima, James, Gorroño, Javier, Randles, Cynthia, Caltagirone, Manfredi, *et al.* 2024a. AI for operational

- methane emitter monitoring from space. *arXiv preprint arXiv:2408.04745*. 34, 35, 149
- Vaughan, Anna, Mateo-García, Gonzalo, Gómez-Chova, Luis, Ružička, Vít, Guanter, Luis, & Irakulis-Loitxate, Itziar. 2024b. CH4Net: a deep learning model for monitoring methane super-emitters with Sentinel-2 imagery. *Atmospheric Measurement Techniques*, **17**(9), 2583–2593. 14, 35, 81
- Verhelst, Marian, & Murmann, Boris. 2020. Machine learning at the edge. *NANO-CHIPS 2030*, 293–322. Chapter 18. 44
- Verma, Vand, Maimone, Mark W, Gaines, Daniel M, Francis, Raymond, Estlin, Tara A, Kuhn, Stephen R, Rabideau, Gregg R, Chien, Steve A, McHenry, Michael M, Graser, Evan J, *et al.* 2023. Autonomous robotics is driving Perseverance rover’s progress on Mars. *Science Robotics*, **8**(80), eadi3099. 40, 42, 46
- Viola, Paul, & Jones, Michael. 2001. Rapid object detection using a boosted cascade of simple features. *Pages I–I of: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee. 116
- Volpi, Michele, Tuia, Devis, Bovolo, Francesca, Kanevski, Mikhail, & Bruzzone, Lorenzo. 2013. Supervised change detection in VHR images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation*, **20**, 77–85. 53
- Wagstaff, Kiri L, Doran, Gary, Davies, Ashley, Anwar, Saadat, Chakraborty, Srija, Cameron, Marissa, Daubar, Ingrid, & Phillips, Cynthia. 2019. Enabling onboard detection of events of scientific interest for the Europa Clipper spacecraft. *Pages 2191–2201 of: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 40, 42, 100
- Wan, Yu-qing, Fan, Yu-hai, & Jin, Mou-shun. 2021. Application of hyperspectral remote sensing for supplementary investigation of polymetallic deposits in Huaniushan ore region, northwestern China. *Scientific Reports*, **11**(1), 440. 104
- Wang, Di, Hu, Meiqi, Jin, Yao, Miao, Yuchun, Yang, Jiaqi, Xu, Yichu, Qin, Xiaolei, Ma, Jiaqi, Sun, Lingyu, Li, Chenxing, *et al.* 2024. HyperSIGMA: Hyperspectral Intelligence Comprehension Foundation Model. *arXiv preprint arXiv:2406.11519*. 103

- Wang, Qipeng, Xu, Mengwei, Jin, Chao, Dong, Xinran, Yuan, Jinliang, Jin, Xin, Huang, Gang, Liu, Yunxin, & Liu, Xuanzhe. 2022. Melon: Breaking the memory wall for resource-efficient on-device machine learning. *Pages 450–463 of: Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. 45
- Wang, Sheng-Yu, Wang, Oliver, Zhang, Richard, Owens, Andrew, & Efros, Alexei A. 2020. CNN-generated images are surprisingly easy to spot... for now. *Pages 8695–8704 of: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 61
- Wang, Wenhai, Xie, Enze, Li, Xiang, Fan, Deng-Ping, Song, Kaitao, Liang, Ding, Lu, Tong, Luo, Ping, & Shao, Ling. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Pages 568–578 of: Proceedings of the IEEE/CVF international conference on computer vision*. 37
- Xiang, Chuchu, Thompson, David R, Green, Robert O, Fahlen, Jay E, Thorpe, Andrew K, Brodrick, Philip G, Coleman, Red Willow, Lopez, Amanda M, & Elder, Clayton D. 2024. Identification of False Methane Plumes for Orbital Imaging Spectrometers: A Case Study with Emit. *Available at SSRN 5006644*. 151
- Xiao, Guangxuan, Lin, Ji, Seznec, Mickael, Wu, Hao, Demouth, Julien, & Han, Song. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. *Pages 38087–38099 of: International Conference on Machine Learning*. PMLR. 44
- Xie, Enze, Wang, Wenhai, Yu, Zhiding, Anandkumar, Anima, Alvarez, Jose M, & Luo, Ping. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, **34**, 12077–12090. 22, 27, 37, 99, 104, 117
- Yang, Sherry, Nachum, Ofir, Du, Yilun, Wei, Jason, Abbeel, Pieter, & Schuurmans, Dale. 2023. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*. 6, 24, 72
- Yu, Chunyan, Han, Rui, Song, Meiping, Liu, Caiyu, & Chang, Chein-I. 2020. A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial-spectral fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **13**, 2485–2501. 105, 141

- Yuhas, Roberta H, Goetz, Alexander FH, & Boardman, Joe W. 1992. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm. *In: JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop.* 115
- Zagoruyko, Sergey, & Komodakis, Nikos. 2015. Learning to compare image patches via convolutional neural networks. *Pages 4353–4361 of: Proceedings of the IEEE conference on computer vision and pattern recognition.* 53
- Zhan, Yang, Fu, Kun, Yan, Menglong, Sun, Xian, Wang, Hongqi, & Qiu, Xiaosong. 2017. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters*, **14**(10), 1845–1849. 53
- Zhang, Beichen, Wang, Huiqi, Alabri, Amani, Bot, Karol, McCall, Cole, Hamilton, Dale, & Růžicka, Vít. 2022. *Unsupervised Wildfire Change Detection based on Contrastive Learning.* 13, 71, 146
- Zhang, Jiyuan, Franchetti, Franz, & Low, Tze Meng. 2018. High performance zero-memory overhead direct convolutions. *Pages 5776–5785 of: International Conference on Machine Learning.* PMLR. 45
- Zhao, Huijie, Deng, Kewang, Li, Na, Wang, Ziwei, & Wei, Wei. 2020. Hierarchical spatial-spectral feature extraction with long short term memory (LSTM) for mineral identification using hyperspectral imagery. *Sensors*, **20**(23), 6854. 105, 141
- Zheng, Sixiao, Lu, Jiachen, Zhao, Hengshuang, Zhu, Xiatian, Luo, Zekun, Wang, Yabiao, Fu, Yanwei, Feng, Jianfeng, Xiang, Tao, Torr, Philip HS, *et al.* 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *Pages 6881–6890 of: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 37
- Zhou, Hongda, Tao, Weilin, Diessel, Oliver, & Purcell, Cormac. 2024. Accelerating RaVÆn for Real-time Satellite Image Change Detection. *In: Proceedings of the FPT 2024 Conference.* 71, 146
- Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, & Efros, Alexei A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Pages 2223–2232 of: Proceedings of the IEEE international conference on computer vision.* 101

Zortea, Maciel, Almeida, João Lucas De Sousa, Klein, Levente, & Junior, Alberto Costa Nogueira. 2023. Detection of methane plumes using Sentinel-2 satellite images and deep neural networks trained on synthetically created label data. *Pages 3830–3839 of: 2023 IEEE International Conference on Big Data (BigData)*. IEEE.

35