Knowledge-based Semantic Enrichment for Semantic Segmentation

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science (M.Sc.)

PARIS-LODRON UNIVERISTY SALZBURG (PLUS)

Faculty of Digital and Analytical Sciences

Department of Geoinformatics

and

UNIVERSITY OF SOUTH BRITANNY (UBS)

Faculty of Sciences and Engineering Sciences

Mathematics, Computer Science, Statistics Department

Data Science Specialization

PLUS Supervisor:
Assoc. Prof. Dirk Tiede, **Felix Kröber**

UBS Supervisors:
Prof. Sébastien Lefèvre

submitted by
Rama Kamala Rajeswari Parasa

Salzburg & Vannes, June 2024

Abstract

The incorporation of prior knowledge into deep learning models is expanding across various fields. In remote sensing, this often involves using spectral indices and incorporating physical principles into the models' loss functions. Yet, the approach of embedding domain-specific semantic information—detailed descriptions at the pixel level in satellite images—in deep learning settings has not been explored. This study seeks to establish foundational benchmarks for integrating such semantically enriched data, investigating three different approaches. Additionally, the study examines the potential of using semantically enriched data in the pretraining phase, which is particularly beneficial when labelled data is scarce. The findings indicate that the most detailed semantic data achieves the best performance in a fully supervised learning setup. Furthermore, models combining semantic and multispectral data as inputs surpass all others in supervised and pretraining settings. However, due to design limitations, such as the minimal use of unlabeled data for pretraining, the benefits of integrating semantic data in pretraining tasks are not definitively proven. This is among the initial studies in this area, so employing simple network architectures and training strategies was necessary. Considering the ongoing advancements in self and semi-supervised learning, future research is encouraged to utilise more sophisticated pretraining approaches.

Keywords: semantic enrichment, semantic segmentation,pretraining, multi-task learning

Acknowledgments

This journey would have been less rewarding without the wise guidance and invaluable support of my supervisors, Prof Dirk Tiede, Prof Sebastien Lefevre, and Felix Krober. Their trust in my abilities to lead this research and their mentorship during challenging times have been fundamental to completing this thesis. A special note of appreciation goes to Felix Krober, whose insightful advice and collaborative spirit have been incredibly inspiring when facing obstacles. I also thank Thomas Fillon for patiently clarifying my doubts throughout the semester about how best to use the cluster computational resources.

I am deeply grateful to all my lecturers at UBS and PLUS, whose constant encouragement has supported me through these transformative two years. My heartfelt thanks also extend to my peers in the CDE Masters program—your warmth and brilliance have made this academic journey a delightful experience. I am particularly thankful to Khizer Zakir and Maria Paula Rodriguez, whose partnership in brainstorming sessions has been invaluable. To my friends, both near and far, your unwavering emotional support during some of the most trying times of this thesis and master's has not gone unnoticed. I am blessed to have the kindest parents and an incredibly supportive brother, whose love and encouragement have been my constant. Additionally, my gratitude extends to Sai, my strongest pillar of strength over these past years. Lastly, I am thankful to EMCDE for its generous scholarship and the opportunity to grow under the guidance of some of the most esteemed professionals in the field.

Contents

List of Figures

1 Introduction

1.1 Leveraging domain knowledge in deep learning

There is an ongoing debate about how much, or even, whether the integration of prior knowledge is desirable for deep learning models (DLs) [31]. The reduced use of pre-existing knowledge is often intentional in training deep learning models, as they aim to gain all the necessary insights for the problem by comprehending the connections between inputs and outputs [31]. However, exploring techniques incorporating domain-specific knowledge has seen an upward trend and has proven useful when applying DL methods. Studies across domains make at least two main arguments that merit the integration of domain knowledge in deep learning models — first, inadequate labelled data available in the respective domains to train the data-hungry DLs, and second, the need for increased algorithmic explainability of neural networks. [33] propose domain adaptive neural networks highlighting the challenges encountered in acquiring representative training data for modelling physical, chemical or biological processes. For instance, it is unsafe to collect data at sensitive infrastructure such as energy facilities; when possible, it is either prohibitively expensive or of poor quality. In the domain of medical diagnostics, [53] highlight similar challenges to data acquisition for deep learning models — the high cost and labour-intensive nature of collecting medical images, the limited annotation of these images requiring expertise from experienced doctors, and the difficulty in obtaining balanced datasets due to the rarity of some diseases. [23] and [58], argue for integrating prior finance knowledge in DLs to increase the explainability of the models and better alignment with advanced finance theories, for their uptake in a sensitive sector like finance. Such incorporation of prior knowledge in deep neural networks has taken at least three forms – feature engineering of raw data before inputting into the model, embedding the domain laws into the loss function or placing domain-specific constraints, and finally, through architectural design choices tailored using domain knowledge [10].

The domain of remote sensing too, is no stranger to the DL story described above— acquiring large annotated datasets labelled at pixel-level and improving the explainability of DL ‘black box’ remain critical challenges [54]. Hence, continuous innovation is needed in both techniques and concepts. Self-supervised learning has shown promise in tackling the first challenge by reducing the need for human annotation and enhancing model performance in low-data contexts [51]. Additionally, integrating domain knowledge, such as using spectral indices as inputs to neural networks [30], [25] and incorporating physical laws as constraints [54], helps address the latter challenge. To further these dual goals, the present study proposes an integration of semantically enriched satellite data in DL neural networks.

The semantic enrichment of satellite data involves translating raw data into meaningful symbols that represent stable concepts. The semantic enrichment used in this study was created using the Satellite Image Automatic Mapper (SIAM) [5]. SIAM provides automatic and near real-time semantic enrichment of multi-spectral satellite imagery. It uses a knowledge-based decision tree that categorises reflectance values into a predefined set of semi -symbolic spectral categories without requiring user-defined parameters. As a result, each pixel has a defined semantic meaning (e.g., dense vegetation or bare soil). Given the simplification of complex multispectral data offered by semantic enrichment and thereby noise reduction, it is well-positioned to be leveraged in DL models.

This study chooses land cover classification as the task to investigate the model performance improvements upon incorporation of SIAM enrichment. Landcover classification is an important tool for understanding our environment and planning our land-based resources better. [54] in their comprehensive review of DL in environmental remote sensing highlight that despite the impressive results achieved in land cover classification using DL, the inadequacy of labelled datasets for this task limit the widespread application of DL for land cover mapping. Another reason making land cover classification a natural choice for preliminary studies, in in-troducing SIAM to DL, is the semantic proximity of well-defined high-level land cover classes to the semi-symbolic spectral categories produced by SIAM.

1.2 Research questions

This study reveals that while SIAM-based semantic enrichment of satellite data is common in deterministic remote sensing analyses, its effectiveness has not been systematically evaluated in deep learning applications, such as land cover classification. The study hypothesises that using SIAM-based semantic enrichment, which encapsulates extensive domain knowledge about the spectral behaviour of Earth’s surface, simplifies complex multispectral data and reduces noise. This is expected to guide the models based on physical knowledge, resulting in improved explainability of the results, quicker convergence and higher accuracy in deep learning tasks for land cover classification. Against this backdrop, the study poses an overarching research question – does the use of semantically enriched satellite data result in improved performance of deep learning models for a land cover classification task?

The research explores this question through two distinct investigative lines based on the supervision setup. The first line examines the impact of using SIAM enrichment as raw input in a fully supervised task trained from scratch. This approach establishes a baseline for model comparison and identifies the most effective SIAM granularity. The second line considers the potential of SIAM enrichment in training pretext tasks, aiming to enhance current self-supervised approaches that rely heavily on noisy multispectral data. These approaches typically lack detailed spectral behaviour knowledge. Furthermore, SIAM semantic enrichment, applicable to any sensor data corrected to at least top-of-atmosphere reflectance levels, allows for automatically generating spectral category maps for each multispectral image patch used in supervised and self-supervised training. This capability significantly enhances the volume of task-agnostic pretraining data, enriching it with comprehensive domain knowledge essential for learning robust data representations.

Furthermore, SIAM semantic enrichment, applicable to any sensor data corrected to at least top-of-atmosphere reflectance levels, allows for automatically generating spectral category maps for each multispectral image patch used in supervised and self-supervised training. This capability significantly enhances the volume of task-agnostic pretraining data, enriching it with comprehensive domain knowledge essential for learning robust data representations.

A) SIAM enrichment as input in supervised models trained from scratch

This set of questions explores whether using SIAM enrichment, alone or combined with multispectral data, improves performance on a test set compared to a baseline model that uses only multispectral data.

* How do models trained with only SIAM data as input perform compared to those using only multispectral data?

* Among the four levels of SIAM granularity, which one achieves the best performance?

* Does combining SIAM with multispectral data outperform using only multispectral data?

B) SIAM enrichment in a pretext task

This set of questions investigates whether SIAM enrichment enhances the supervision quality in pretraining tasks, used alone or alongside a reconstruction task like in an autoencoder.

* Does incorporating SIAM in a pretext task lead to better downstream task performance than using image reconstruction alone?

* Does a multitask learning setup, involving both image reconstruction and predicting SIAM labels, yield better results than using image reconstruction alone in the pretext task?

This paper is structured as follows. The Introduction Chapter discusses the importance of leveraging domain knowledge in deep learning and outlines the research gaps and questions. The Literature Review synthesises existing studies that use semantic enrichment in remote sens-ing, supervised semantic segmentation for land cover classification, and self-supervised learning for semantic segmentation. The Data Chapter describes the benchmark Semantic World dataset and the specific subset used for the study. The Methods Chapter details the integration of SIAM into neural networks, including training from scratch and pretraining for semantic segmentation. The Experimental Design outlines the cross-validation and ensembling techniques, losses and evaluation metrics, hardware specifications, and methodological simplifications. The Results Chapter presents the outcomes of hyperparameter tuning, the overall performance of the models, convergence analysis, the tradeoff between model performance and convergence, the dependence of each model on the amount of labelled data, class-wise performance comparison, and visual comparisons of segmentation maps. Finally, the Discussion and Conclusions Chapter analyses the findings and their implications. References and additional details are provided in the Appendix.

2 Literature Review

The literature review aims to comprehensively review the current state of research in to two key areas directly related to this study's topic. The first section looks at studies from the remote sensing domain that have employed semantic enrichment. The next section looks at some pretraining tasks designed to be used in downstream semantic segmentation. This review will contextualise the research gaps and questions addressed in this thesis.

2.1 Semantic enrichment in remote sensing

2.1.1 Use and representation of semantic information

Semantic enrichment of satellite data means enhancing image patches by adding context and meaning. This enrichment can be achieved through various methods. In remote sensing and image-related tasks that use deep learning, studies have explored techniques including multi-source data fusion [56] (also see [55]), its combination with multi-temporal data (see [13]), inte-grating deep features [17], [52], and fusion of data layers containing context-specific descriptors [3] and [24].

This study is particularly interested in the last method—context-specific descriptors. Studies have examined using such data input in its vector or object form. For example, a body of literature from Volunteer Geographic Information (VGI) uses OSM data to provide contextual information and enrich the input data's semantics. OSM data contains semantically rich high-level geographic information crowdsourced by volunteers. [57] use semantic elements from OSM data as input to an Object-based CNN for urban scene understanding and achieve high accuracies for complex urban scenes. Most studies that use semantically rich descriptive data in a multi-class image or pixel form instead of object-based or vector form end up using it to provide supervision. [26], show that using OSM data as labels for segmenting roads and buildings can effectively reduce efforts and costs incurred in manual annotation for obtaining ground truth. [19] developed a semi-automatic approach to create training data for land cover classification of aerial imagery and elevation. They use OSM data to generate labels. Very few studies explore using this data as pixel-level inputs rather than as labels.

A study of high relevance in this discussion is [3]. They propose using semantically rich OSM data with remotely acquired optical data to build semantic maps. They study two methods of representing the OSM data in the neural network. The first method represents the data as a sparse tensor to encode the data discretely. Each channel in the tensor corresponds to a raster class. The channel contains binary information indicating whether or not the respective class is present in that pixel. The second method is a continuous representation called the signed distance transform (SDT). In this representation as well, the channels of the tensor correspond to the raster classes; however, each channel contains a distance transform associated with that class such that the distance 'd' from the respective class is, d<0 appears if the pixel is inside the class and d>0 if it is outside it. They explore two architectural variations for the fusion technique based on the use case: one, a fusion that uses residual correction if the use case is to detect the classes that can be directly inferred from OSM, such as roads or buildings, and two, FuseNet for use cases wherein the target classes can be indirectly inferred from OSM raster classes, such as settlement type based on the buildings. The binarised representation gives slightly better results than the signed distance transform, potentially due to much less diffused information than the distance transform. Another study that binarises OSM data is [22], wherein residential and industrial buildings and highways are binarised and stacked together before inputting into a multi-branch CNN that processes three data sources, PAN, multispectral, and OSM data parallelly to predict an urban land use class.

[24] also represent context-specific descriptive enrichment in pixel form. They utilise Google Maps rasters as three-channel inputs within their framework of conditional GANs, proposing a ver-satile solution for image-to-image translation tasks. This approach facilitates transformations such as converting sketches into realistic photos, changing black-and-white images to colour, and altering daytime scenes to nighttime. Although the paper does not focus extensively on the specifics of input data representation, its use of RGB channel-based input for conveying semantic information is relevant to this study.

2.1.2 SIAM and its use in deterministic studies

The Satellite Image Automatic Mapper (SIAM™), introduced by [5], autonomously generates semantic enrichments without requiring user-defined parameterisation or training data. Operating as an expert system, SIAM™ employs a per-pixel physical spectral model-based decision tree on images calibrated to at least top-of-atmosphere reflectance levels. This setup facili-tates automatic, near real-time multi-spectral discretisation utilising pre-existing knowledge. The tool's patent review discusses the operational limitations of contemporary remote sensing image understanding systems, focusing on automation, efficiency, and robustness to parameter changes. Some examples of such systems include ATCOR [38], which is semi-automatic and requires user intervention and site-specific settings, and eCognition [14], which employs object-based image analysis concepts that currently lack methodological consensus. Against these limitations, it highlights the need for a fully automated, knowledge-driven, decision-tree-based tool.

[6], outline the interdisciplinary foundation of predetermined colour naming within cognitive science, spanning from linguistics to computer vision. They also review existing research on using static colour names in multispectral (MS) imaging. They explain that when every pixel of a multispectral image is mapped to a colour space, partitioned as a set of mutually exclusive and collectively exhaustive hyper polyhedra (equivalent to a predefined-and-pre agreed upon dictionary of colour names), then a semi-symbolic multi-level colour map is generated auto-matically. The authors also highlight how such an approach to the partitioning of a measure-ment space into hyperpolyhedra is synonymous with vector quantisation in inductive machine-learning-from-data; and also with the process of deductively transforming a numerical variable into fuzzy sets within a framework of logic. This study utilises the comprehensive Semantic World dataset, whose semantic layers are derived from the SIAM tool, as detailed in Chapter 3.

Key analyses performed using semantic enrichment generated from SIAM, so far, are pri-marily classical knowledge-based approaches. [44] used SIAM data to automatically pre-classify optical Earth Observation (EO) images into semantic information layers for surface water detection for flood assessment in Somalia using water-related SIAM spectral categories like 'turbid-water like', 'deep-water like', etc.

[49] use SIAM spectral category-based cloud masks and compare them with cloud metadata provided by ESA's Sentinel-2 products to find that cirrus cloud cover is overestimated in regions with high altitudes. They argue for better algorithms to generate cloud-related metadata for satellite products that several users and applications use. [16] used SIAM's spectral categories like 'clouds', 'shadow area with vegetation', and 'thin clouds over vegetation' to create multitemporal cloud filters over the grassland patches in the study area. The filters are used to select images for their study to monitor temporal mowing events over grasslands, which is highly relevant for preserving grassland biodiversity. [20] combine SIAM's 'water-like' categories to create water masks to identify water bodies suitable for floating photovoltaics. The study develops an approach to a detailed water body stability and size analysis over time, providing valuable data for spatial planning and renewable energy projects.

2.2 Self-supervised pretraining for semantic segmentation

Pretraining tasks in deep learning involve neural networks that learn useful representations from input data, which can be categorised into supervised [1] or self-supervised learning (SSL) tasks ([43], [32], [41]). In supervised pretraining, models initially train on established labelled datasets such as ImageNet [11], with downstream tasks—related to specific areas of user interest—subsequently finetuned using weights from these pretrained networks, a process known as transfer learning. While such an approach is largely beneficial for downstream tasks, it might come with some limitations, primarily because the labels in the established datasets may not closely match the requirements of the downstream tasks, which can adversely affect the performance [34]. In SSL approaches, models derive insights from vast amounts of unlabeled input data. These learning tasks are meticulously crafted to be either neutral or specifically aware of the downstream tasks. Compared to supervised methods starting from scratch, self-supervised approaches are particularly effective in contexts with scant labelled data.

The research on SSL techniques is swiftly evolving, offering a broader range of training designs than supervised methods. The remote sensing domain has also started to sharpen the focus on these approaches in light of the increasing amounts of remotely sensed data. Studies conducted by [50], [47], [7] and [51] have made significant contributions to this field by attempting to consolidate the use of SSLs in the Remote Sensing domain. [51] extensive review of SSL approaches categorises self-supervised techniques into predictive, generative and contrastive methods within the remote sensing domain.

This section leverages the taxonomy presented in [51] Still, it limits its scope to generative and contrastive self-supervised techniques that are either designed for or tested on task for creating a segmented map, either through classical approaches after pretraining or finetuning with a supervised downstream task.

Self-supervised techniques vary widely. Among them, the methods that focus on capturing the lowest level of pixel detail are the generative methods that focus on ways of reconstructing the input. Among generative methods, Autoencoders [4] have been widely used for pretraining and learning representations from multispectral remote sensing data. They comprise an encoder-decoder architecture and the model is trained to learn the representation of inputs by reproducing it back. [42] demonstrate the use of Autoencoders in compressing high dimensional vegetation indices data and then use a Random Forest classifier for the classification of pixels to vegetation types.

Another group of generative methods are Generative Adversarial Networks (GANs) [15] wherein two models are forced to compete against each other in the learning process. The comprising models are called generator and discriminator. While a generator accepts a random vector as input and learns to produce a fake output, the discriminator learns to decide whether the generated output is fake or not. In this manner, both models learn to optimise their losses and eventually, the generator learns to produce outputs that the generator cannot deem as fake. [43] deploy an adversarial training scheme in which a coach network and an inpainting network are pitted against each other. The coach network with is trained to predict increasingly difficult masks to crop out of input images, which then, are filled in by the inpainting network by reproducing the input. Another interesting contribution of this study is its proposal of getting rid of the fully connected bottleneck layer in the encoder-decoder architecture in order to pre-serve the spatial structure of the image data, relevant for downstream segmentation tasks. They highlight an important observation relevant to the present study, most of the time, pretraining tasks focus only on training the encoder, which is insufficient and inefficient when they're used for a downstream task. Hence, they propose to train both encoder and decoder through the coach-inpainting training and find it to be effective on a downstream land cover segmentation task.

Contrastive methods are another fast-evolving set of self-supervised approaches. They focus on learning features by contrasting different views of the same input, for example spatially or spectrally augmented views of the same image. Negative sampling is one such approach but aims at forcing the model to learn not only from different views of the same image but different images altogether. The study by [32], used SimCLR [9], an innovation in negative sampling-based contrastive approaches, to enhance feature extraction for classifying land cover using SAR and multispectral imagery. Their self-supervised approach, called Spatial-Spectral Context Learning (SSCL), effectively learns features aligned with land cover classes without using labelled data. SSCL combines SimCLR for encoder training with a disturbance-resistant autoencoder for decoder training. During pretraining, the model contrasts feature vectors from two different types of data (e.g., SAR-optical, optical-optical, or SAR-SAR) to reduce contrastive loss. This method aims to harmonize different imaging modalities by aligning their feature vectors, thereby making the features modality-agnostic. The results show that this approach not only improves the integration of SAR and multispectral data but also surpasses fully supervised methods, pretraining with ImageNet, and other self-supervised techniques, offering a powerful solution for land cover classification t asks. Similarly, [41] use SimCLR alongside Swin Transformers to contrast optical and SAR Sentinel patches. They fine-tune the model on two separate tasks, a classification task and a segmentation t ask. They find that their approach outperforms a UNet trained from scratch for the segmentation task of land cover classification. They also note that the model performs better as a finetuning task downstream than with a frozen backbone, potentially because of the skip connections in the segmentation head that combines multiscale features. [29] propose a self-supervised contrastive learning method, Global Style Local Contrastive Learning Network (GLCNet), to enhance semantic segmentation of remote sensing images. By addressing the need for both global and local feature learning, they introduce a local matching contrastive loss, applied between patches within a single image, alongside the traditional global contrastive loss used between different images. This approach enhances the learning of both overall image-level features and also representations from local regions crucial for downstream semantic segmentation tasks.

Finally, clustering techniques are key data mining techniques that involve grouping a set of objects in a way that objects in the same group (called a cluster) are more similar to each other than to those in different groups. One of its implementations in deep learning is the deep clustering [8] approach that involves iteratively grouping features extracted from a neural network into clusters. Then, it uses the assignments from these clusters as pseudo-labels to train the network itself. By combining k-means clustering on the output of the feature by the convolutional layers of a deep network with standard backpropagation for learning, DeepCluster effectively leverages the representation power of deep neural networks without requiring labelled data. [40] deploy self-supervised deep clustering for the joint segmentation of multitemporal high-resolution images. This method involves clustering pixels based on their deep feature representations derived from a convolutional neural network, followed by an iterative refinement process to maintain spatial and temporal consistency across image sequences. This enables the automatic generation of temporally coherent semantic segmentation maps. The approach allows for consistent and accurate segmentation of land cover changes over time, making it highly applicable for environmental monitoring and change detection in remote sensing datasets.

3 Data

3.1 Semantic World

The Semantic World dataset, developed by [Felix Kröber1], is an advanced benchmark for semi-supervised semantic segmentation in remote sensing. It builds on the Dynamic World [46] dataset by employing the SIAM to enrich patches of scenes from Sentinel-2 imagery. SIAM's rule-set-based decision trees translate reflectance values into semi-symbolic spectral categories, resulting in a dataset of 57.4K semantically enriched Sentinel-2 patches (510 x 510 pixels each), with 21.4K labelled, that is, containing Dynamic World class label per each pixel and 36.0K unlabelled patches. The division between training and test data, with 57.0K patches for training and 0.4K patches for testing, follows the original split of the Dynamic World dataset.

The dataset includes reflectance values from all 10m and 20m Sentinel-2 bands for L1C and L2A processing levels, with SIAM categorisation available in granularities of 18, 33, 48, and 96 categories, along with the automated scene classification (SCL) layer. The dataset comprises patches spanning over 9K Sentinel-2 scenes across various global biomes, and years from 2017 to 2019, offering a comprehensive spectral, spatial and temporal representation. 3.1 shows the spatial distribution of the images in the Semantic World dataset. This diversity enables robust training and benchmarking of deep learning models.

The dataset is structured into three size-based subsets. The smallest set contains about 72 patches and is intended for prototyping tasks. The next bigger set contains about 6k patches (50 GB), and the largest set contains about 57k patches (450 GB). This study uses the second or medium-sized set for all the experiments.

3.2 Subset of the Semantic World

This study uses the L1C level patches from the medium-sized set of the Semantic World dataset. It comprises 3601 unlabelled patches and 2480 labelled patches. Of the labelled patches, 2102 patches are reserved for training and 378 patches are set aside for inference and reporting of the model performance, it remains completely unseen by the models during their training phase. Figure 3.2 showcases the spectral categories found in SIAM-96 granularity.

3.2.1 Preprocessing

A standard normalisation technique of scaling the spectral band values to [0,1] is applied. For a band in each image, the pixel value at the 99th percentile is assigned a value of 1 and a pixel value at the 1st percentile is assigned a value of 0. The pixel values outside the mentioned percentile limits are clamped to 0 or 1. Wherever SIAM categories are passed as input to the networks as RGB bands (discussed in Chapter 4), all pixel values are reduced to [0,1] scale by dividing all the pixel values uniformly by a value of 255.

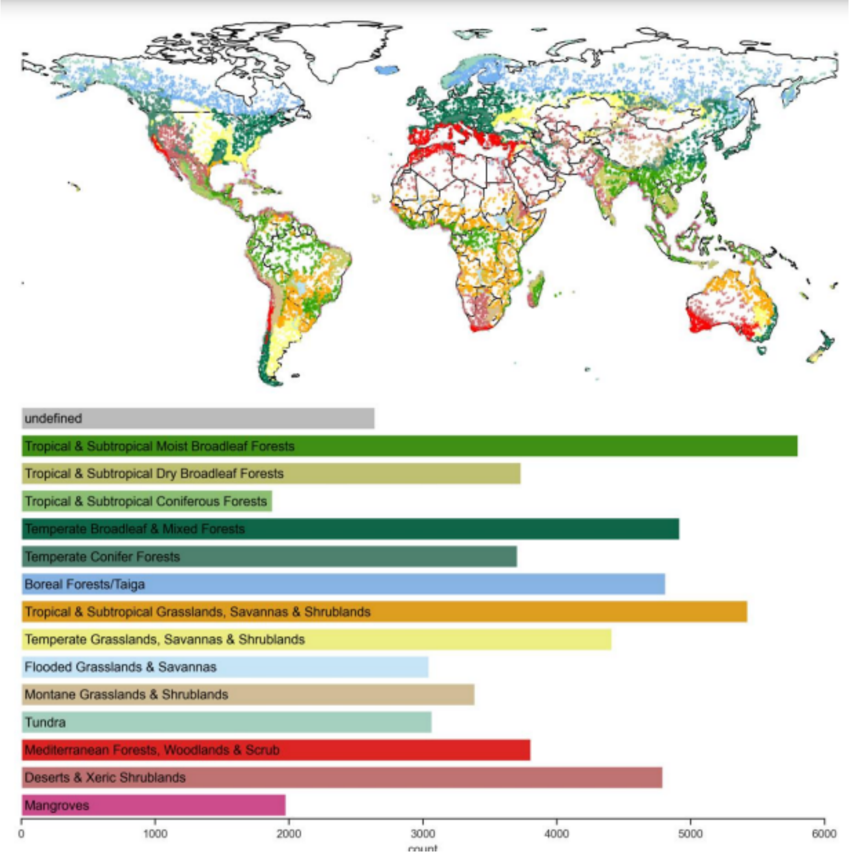
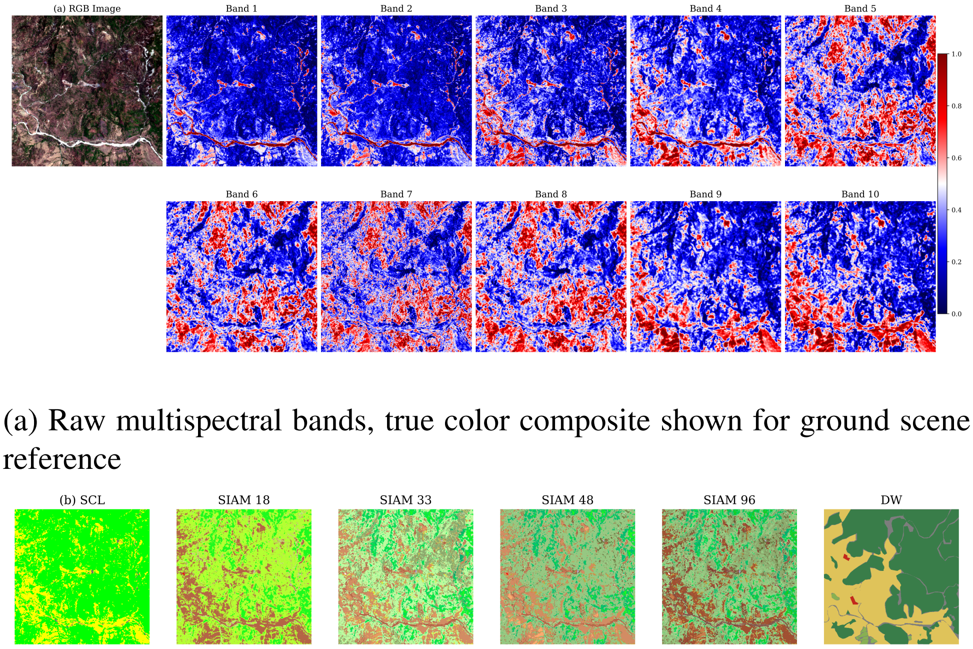

undefined
Tropical & Subtropical Moist Broadleaf Forests
Tropical & Subtropical Dry Broadleaf Forests
Tropical & Subtropical Coniferous Forests
Temperate Broadleaf & Mixed Forests
Temperate Conifer Forests
Boreal Forests/Taiga
Tropical & Subtropical Grasslands, Savannas & Shrublands
Temperate Grasslands, Savannas & Shrublands
Flooded Grasslands & Savannas
Montane Grasslands & Shrublands
Tundra
Mediterranean Forests, Woodlands & Scrub
Deserts & Xeric Shrublands
Mangroves
0
1000
2000
3000
4000
5000
6000
count

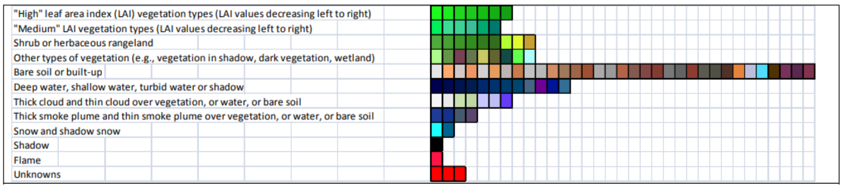
Figure 3.1: Spatial distribution of the Semantic World data along with biome information. Adapted from [Felix Kröber1]



(a) Raw multispectral bands, true color composite shown for ground scene reference

(b) Scene Classification Layer, SIAM layers, and Dynamic World layers

Figure 3.2: A sample tile showcasing all the bands available in Semantic World dataset



"High" leaf area index (LAI) vegetation types (LAI values decreasing left to right)
"Medium" LAI vegetation types (LAI values decreasing left to right)
Shrub or herbaceous rangeland
Other types of vegetation (e.g., vegetation in shadow, dark vegetation, wetland)
Bare soil or built-up
Deep water, shallow water, turbid water or shadow
Thick cloud and thin cloud over vegetation, or water, or bare soil
Thick smoke plume and thin smoke plume over vegetation, or water, or bare soil
Snow and shadow snow
Shadow
Flame
Unknowns

Figure 3.3: Adapted from [6]. Pseudo colors of 92 spectral categories, categories are aggregated along a row if they share the same parent class for better readability

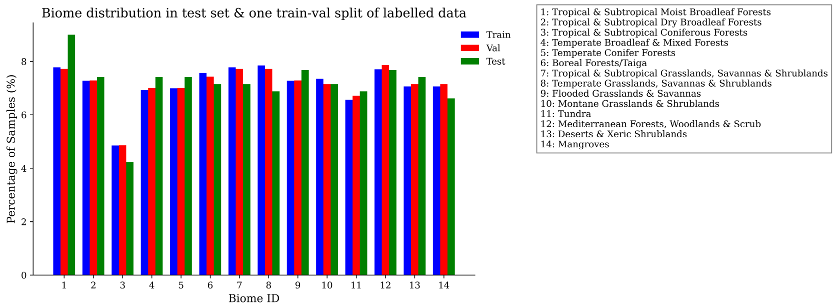


Figure 3.4: Distribution of biomes across train, test and validation sets.

3.2.2 Train-Test-Val splits

The Semantic World dataset provides a test dataset for evaluating the final models. It remains unseen during the training process. The study uses three stratified splits in both supervised and pretraining scenarios (for reasons discussed in Chapter 5). A splitting based on the Biome covariate of each image patch is used. This approach is chosen because biomes broadly reflect the spectral diversity found in the patches, ensuring a consistent spectral distribution across the pixels from these regions. Figure 3.4 illustrates the biome distribution for one stratified split of the labeled patches alongside that of the patches in the test set. It is important to note the difference in biome distributions between the test and validation patches, a factor that must be considered during the interpretation of the model evaluation analysis. Similarly, the unlabelled images are also split using this biome-based strategy. Each labelled split contained 1400 train images and 700 validation images. Each unlabelled split contained 2400 train images and 1200 validation images.

4 Methods

4.1 Introducing SIAM to neural networks

Applying the semantic information representations discussed by [3] to this study might lead to issues due to the inherent difference between the characteristics of the OSM data used in [1] and SIAM data. This is because OSM contains about five raster classes, such as buildings and roads. In contrast, SIAM data includes 18 to 96 classes, depending on the chosen granularity. Utilising binary or signed distance transform representations could result in either too sparse or too diffuse inputs, respectively. The study ultimately adopted a pseudo-RGB representation for each spectral category as the most suitable approach.This method aligns more closely with [24] approach in terms of data structuring, although it differs in that the present study assigns a discrete meaning to each RGB value, unlike the continuous raster image representations of map views derived from GoogleMaps in [24].

The present study leverages the mapping described in the Chapter 2 provided by SIAM for the Sentinel-2 multispectral polyhedra to one multispectral colour name as the basis for transforming the semantic categories into a pseudo-RGB raster input for deep neural networks. Each colour name represents a stable concept that is sensor-agnostic. Incidentally, in SIAM, the predefined pseudo-colour of a spectral category mimics the natural colours of pixels belonging to that spectral category, e.g. dark green for 'dense vegetation' maps to pseudo-RGB values of (30, 250, 30) as per SIAM-48 granularity.

4.2 Training from scratch

In this group of experiments, the models follow a fully supervised training setup with an output task to predict the Dynamic World label. First, a baseline model is trained with only ten bands of Sentinel-2 multispectral data as input. Then, four different models are trained, each with a different SIAM granularity data represented as the 3-band pseudo-RGB input, to predict the Dynamic World label. Finally, after comparing the individual performances of models trained with the four SIAM granularities, the best-performing SIAM granularity was chosen to train a fifth model, with a combined input of multispectral and SIAM data. In this final model, 13 bands are passed as input in total; 10 Sentinel-2 multispectral bands and three pseudo-RGB bands representing the chosen SIAM granularity. Figure 4.1 illustrates the different training scenarios implemented in the study.

Architecture specifications. The study chooses a U-Net [39] architecture with a ResNet-50 [17] encoder to perform the semantic segmentation task. This configuration utilises a deep convolutional network renowned for its robust feature extraction capabilities, which are particularly suitable for image segmentation tasks. The encoder consists of the ResNet-50 structure, starting with an initial convolutional layer (7x7 convolutions, stride 2) followed by a max pooling layer (3x3 pooling, stride 2). This setup reduces spatial dimensions while capturing essential features. The ResNet-50 backbone includes four main stages, each composed of multiple residual blocks that employ skip connections to facilitate the flow of gradients and prevent the vanishing gradient problem. These stages progressively double the number of filters from 64 in the initial stage to 512 in the final stage.

The decoder in the U-Net architecture is specifically designed to reconstruct the segmentation map from the encoded features. It consists of four main blocks, each of which includes transposed convolutions or upsampling operations followed by convolutional layers to refine the features progressively. These operations are designed to gradually restore the spatial dimensions of the output to match those of the original input. Skip connections from corresponding encoder stages are integrated at each level of the decoder, merging features from the downsampled pathway with the upsampled outputs. This approach helps the network localize and refine segmentation outputs effectively by leveraging spatial hierarchies maintained throughout the encoding process. Overall, the network used in the study has about 33 million trainable parameters. In the decoder, the combination of upsampling and feature refinement through convolutional processing, coupled with the integration of skip connections, ensures precise prediction of the Dynamic World label for each pixel in the segmentation map.

4.3 Pretraining

Pretraining strategies are devised to front-load the learning capabilities and generalisation for the downstream tasks. The pretraining phase prepares the model architecture by using unlabelled data to better handle the complexities of a downstream task with limited labelled data before tackling it. This study uses ResNet-50 encoder-based UNet for pretraining tasks as well. Figure 4.2 showcases the three pretraining scenarios tested in this study.

4.3.1 Single task learning as pretraining

Three pretraining tasks are designed. All tasks expect multispectral image patches as input. The first pretraining task is designed as an autoencoder task where the model learns to reproduce all the ten multispectral bands in the input as a regression task. The regression task head has a Sigmoid activation function after the final convolution layer. The model minimises the Mean Squared Error Loss (Chapter 5) used as the reconstruction loss function during learning. The reproduction of images forces the model to learn low-level pixel information the extracted feature representation prepares the model for performing well on the downstream task.

The second pretraining task is designed to predict pixel-wise SIAM-96 spectral categories as labels, as in a semantic segmentation task. Here, the segmentation task head outputs logits corresponding to the 97 classes (including a no-data class). The design of this task aims to utilise SIAM spectral categories to provide a knowledge-guided supervision signal for learning much higher-level features, compared to reconstruction-based pretraining, making the task semantically closer to the downstream land cover class prediction task. The model optimises on a standard segmentation loss function, the Dice Loss (Chapter 5). Finally, the third task is devised as a multi-task learning model, where the model learns to both reproduce the input and also, to output a segmentation map predicting SIAM categories for each pixel.

4.3.2 Multitask learning as pretraining

The relevance of both reconstruction and SIAM segmentation tasks to the downstream land cover classification task is the driving motivation behind proposing a multitask learning strategy

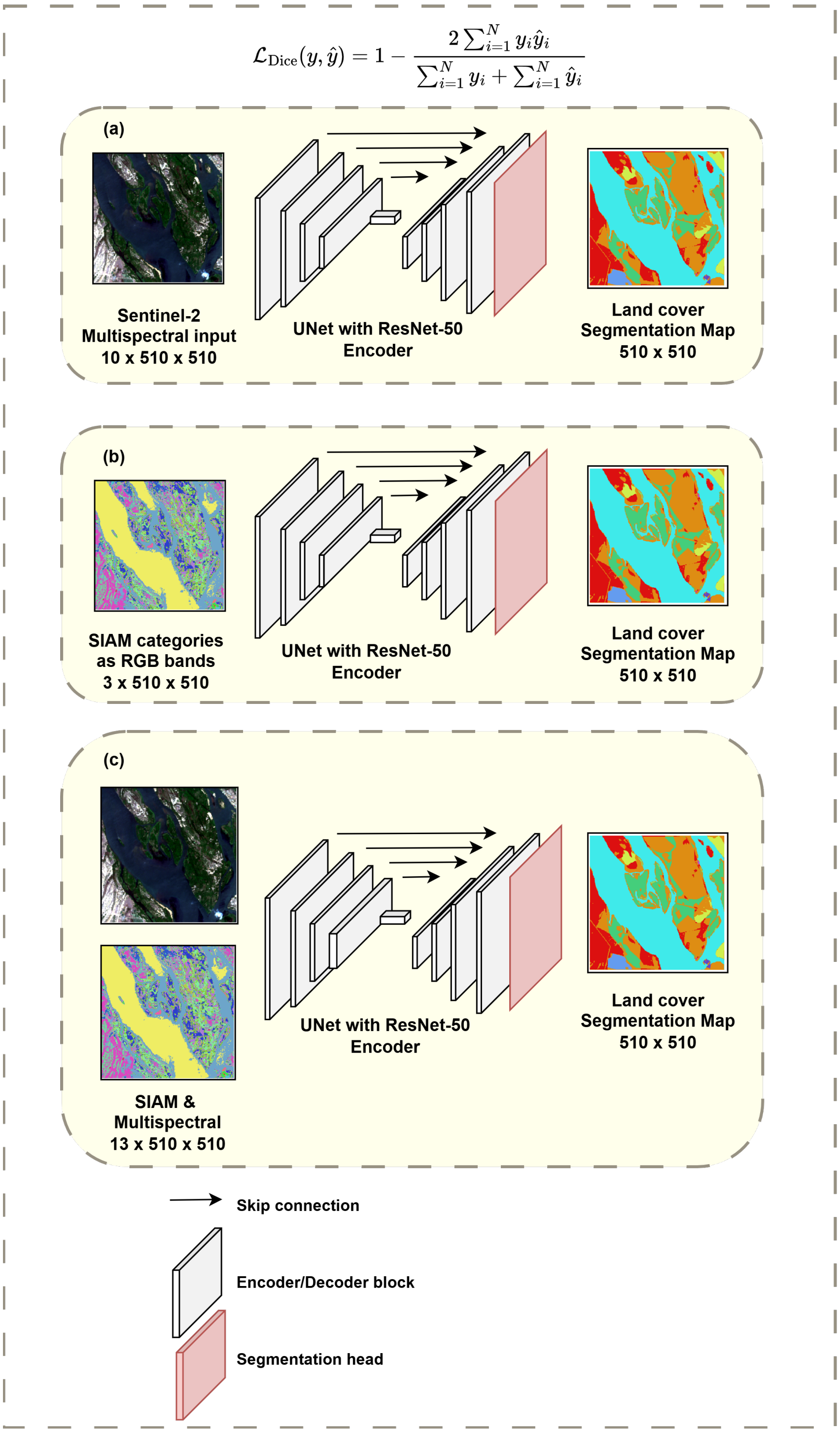


Figure 4.1: Illustration of the different training scenarios for models trained from scratch employed by this study

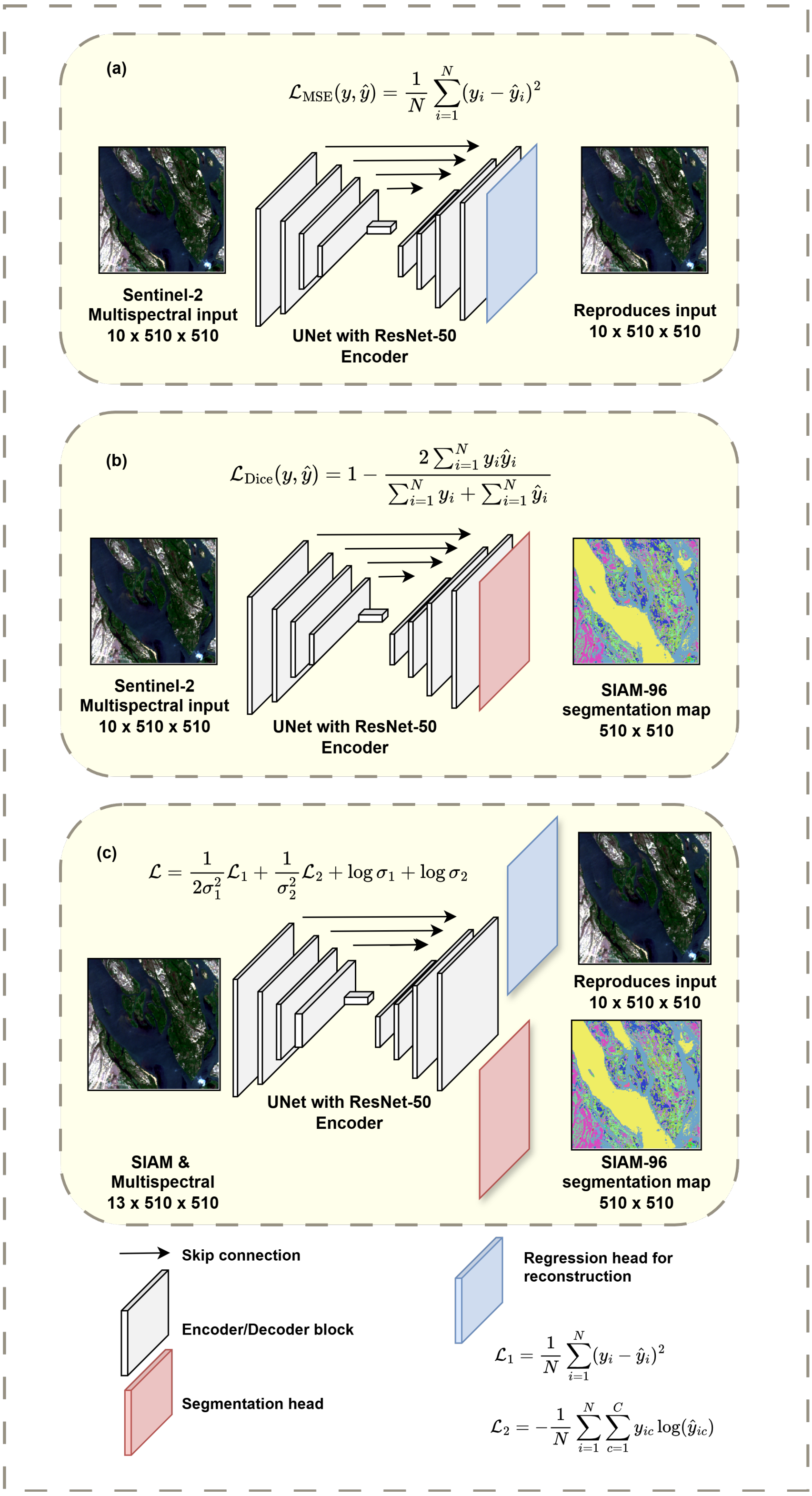

(a)
$\mathcal{L}_{\text{MSE}}(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
Sentinel-2 Multispectral input 10 x 510 x 510
UNet with ResNet-50 Encoder
Reproduces input 10 x 510 x 510
(b)
$\mathcal{L}_{\text{Dice}}(y,\hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i\hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N}\hat{y}_i}$
Sentinel-2 Multispectral input 10 x 510 x 510
UNet with ResNet-50 Encoder
SIAM-96 segmentation map 510 x 510
(c)
$\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_1 + \frac{1}{\sigma_2^2}\mathcal{L}_2 + \log\sigma_1 + \log\sigma_2$
SIAM & Multispectral 13 x 510 x 510
UNet with ResNet-50 Encoder
Reproduces input 10 x 510 x 510
SIAM-96 segmentation map 510 x 510
Skip connection
Encoder/Decoder block
Segmentation head
Regression head for reconstruction
$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
$\mathcal{L}_2 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic})$

Figure 4.2: Illustration of the different training scenarios for pretraining tasks employed by this study

as a pretraining task. The network learns representations of the input multispectral data that are relevant to both tasks, which are in turn relevant to the downstream task; the reproduction of images forces the model to learn low-level pixel information while learning SIAM spectral categories offers a supervision signal for slightly higher-level image understanding making the task semantically closer to the downstream land cover class prediction task.

Architecture specifications. For the third pretraining strategy of multitask learning, both tasks share the same encoder-decoder architecture, however, the architecture splits at the very end with two task-specific heads, a reconstruction head and a segmentation head. Meaning they share the parameters along the architecture, except for the last separate convolution layer. The splitting of task-specific decoders, a common practice in supervised multitask learning, is avoided in this study so that the pretrained decoder can also be used by the downstream task. [43] warn against the popular approach of training only encoders or using an encoder pretrained on an encoder-focused task such as scene classification, for a downstream semantic segmentation task.

Task loss weighting. This study employs the multitask weighting approach developed by [27]. The approach leverages homoscedastic uncertainty to dynamically weigh the loss func-tions of different tasks in a multi-task learning setup. They discuss that homoscedastic uncer-tainty, a type of task-dependent uncertainty varies between tasks reflecting the inherent noise in each task. It doesn't depend on the input data. The approach involves defining a multi-task likelihood, where each task's loss is scaled by the inverse of its uncertainty, thus automatically adjusting the weight based on the task's confidence. This results in more confident tasks to get assigned higher weights in the loss calculation. This method not only eliminates the need for manual tuning of task weights but also improves the overall performance by enabling the model to balance the learning of multiple tasks optimally. The authors demonstrate the efficacy of this approach in visual scene understanding, where a single model simultaneously performs seman-tic segmentation, instance segmentation, and depth regression, outperforming models trained separately on each task. For implementing the multitask learning proposed by [27] the weights are set up as trainable parameters along with the model's architectural trainable parameters. They also advise on training the learnable parameters as log variances of each task, instead of variances themselves, for numerical stability.

4.3.3 Downstream training

For all three pretraining models, two variants of downstream training are implemented. After the pretraining task is completed, the weights from the encoder and decoder of the trained network are transferred to the network for the downstream task. In the first set-up, the encoder is frozen to be used as a feature extractor and the decoder is finetuned on the downstream task. In the second set-up, both the encoder and decoder are finetuned. The loss and evaluation metrics used in the downstream are same as those used in the models trained from scratch.

5 Experimental Design

5.1 Cross-validation and ensembling

This study creates three stratified train-val splits, of both labelled and unlabelled datasets, as described in Chapter 3. The three splits are employed in cross-validation to get a statistically more robust estimate of the model generalisation than training a model using only a single train-test split [36]. Briefly, a model is trained on two folds and the third fold is used as a validation set. This process is repeated thrice wherein each fold acts as a validation set in each run. A cross-validation score is obtained by averaging the validation score from individual runs. This score is used in the tuning of hyperparameters. This study's cross-validation approach of training a model plays two additional roles as described below.

For fully supervised and downstream tasks, the three models validated against the three folds are used in an ensemble learning approach by performing model averaging at the time of inference on the test set. Such an approach captures the model uncertainty better and increases the predictive performance on the test set ([48], [28]). For the models used as pretraining tasks, the model that achieves the lowest minimum for the loss function, among the three models, is chosen for transferring the weights to the downstream task.

5.2 Losses and evaluation metrics

All metrics throughout the experiments conducted in this study are aggregated at pixel-level. In the supervised semantic segmentation task of predicting Dynamic World label, a standard DiceLoss [45] is optimised. The primary evaluation metric is chosen to be Intersection Over Union [37] or IoU. An auxiliary metric F1 score is also monitored. In the pretraining tasks, the reconstruction task optimised Mean Squared Error Loss. R2 score is monitored for the evaluation of the training process. The single task pretraining with SIAM spectral categories as labels also optimises DiceLoss and monitors IoU. However, in the dual task pretraining, this study uses [27]to weight losses of each task to form a combined loss which the model optimises. In this loss, the regression loss used is typically Mean Squared Error and the segmentation loss used is the Cross Entropy Loss. The mathematical equations for the several losses used in the pretraining and supervised scenarios are shared in the Figures 4.2 and 4.1 depending on each training setup.

5.3 Hardware and implementational specifications

All the experiments in this study were conducted on an Intel Xeon 6226R @ 2.90GHz CPU attached to an NVIDIA RTX A6000 GPU node with 48 GB VRAM. All the models were built using PyTorch [2] and Segmentation Models PyTorch [21] libraries. Scikit-learn [35] was also used as an auxiliary library for tasks such as stratified splitting of the datasets. Matplotlib [18] was used to plot all the charts presented in this study. The code for the implementation of all the models presented in this study can be found at– https://github.com/rajesvariparasa/semantic-enrichment-for-semantic-image-segmentation

5.4 Methodological simplifications

Acknowledging the computational time and resource constraints, at least two conscious choices made by the study could have kept the models from achieving their optimal performance. Firstly, a limited number of hyperparameters were tuned and, were tuned manually. Secondly, no data augmentations were performed to artificially increase the volume of the training data which could have improved the generalisability of the models.

On the tuning itself, three critical hyperparameters were chosen for tuning, namely, batch size, learning rate, and gamma, the factor by which the learning rate decays after each epoch for the ExponentialLR scheduler. On a related note, hyperparameters once tuned were kept constant across different inputs and training scenarios. This was mainly for the general comparability of model performances and more specifically, for convergence analysis (section 6.3) which neces-sitated some control over the speed of training across models. SIAM-18 granularity was chosen to perform hyperparameter tuning. The models were trained for 50 epochs and cross-validation scores for selecting the best model. The hyperparameters were tuned iteratively starting with batch size, followed by the initial learning rate and gamma. While tuning a hyperparameter, the other two were kept constant.

6 Results

This chapter summarizes the results obtained from the model selection and evaluation experiments, beginning with an analysis of hyperparameter tuning to identify optimal settings for training. The subsequent analysis compares and contrasts model performances with and with-out the inclusion of SIAM data in the training pipeline, covering scenarios of training from scratch and pretraining. The discussion includes a comprehensive review of overall model per-formance, an assessment of how quickly models converge to optimal performance, an examina-tion of the trade-offs between performance levels and convergence speed, a study on how model performance varies with the availability of labeled data, and a detailed evaluation on individual classes.

It is important to note that the error bars presented in most charts correspond to the validation sets and represent the standard deviation of the measures shown on the y-axis, derived from three stratified s plits. The length of these error bars allows readers to infer the variability of the respective measure. While these error bars do not directly establish statistical significance, they provide a visual cue for assessing whether differences might be significant. For this r -eason, an analysis of performances on both the validation and test sets is included, ensuring a thorough assessment of model robustness. Further, another note on reading the charts provided in the chapter - the model names are intended to be intuitive, prefixes either refer to the input to the model, or the type of pretraining used. For example, 's2' refers to models trained with Sentinel-2 multispectral data and 'single segsiam' refers to the pretraining with single task in which models use SIAM categories as labels. On the other hand, the suffixes refer to the type of the task itself. So, 'scratch' refers to models trained from scratch, that is, without any pretraining, 'fe' refers to the downstream models in which the encoder from the pretraining task was frozen and the decoder was finetuned. F inally, 'ft' refers to downstream models which were fully finetuned, that is, both encoder and decoder are finetuned.

Additional material is provided in the Appendix, including training curves of all the final models trained in this study, curves showing the evolution of weights for uncertainty-based task weighting in dual task learning, and segmented maps predicted by all models for visual comparison of performance.

6.1 Hyperparameter tuning

Figure 6.1 illustrates the model performances from various hyperparameter tuning tests conducted on the validation set using a cross-validation approach. The results indicate that a batch size of 16 yields the highest IoU, as shown in Figure a. It is also important to note that a batch size of 4 exhibits the least variability in IoU. Regarding the initial learning rate, a setting of 0.0001 not only provides the highest IoU but also maintains relatively low variability, as detailed in Figure b. Additionally, the learning rate decay factor, gamma, set at 0.95, slightly improves the IoU compared to a setting of 0.92, as seen in Figure c; however, it also introduces greater variability. Analysis of the learning curves reveals that a gamma of 0.95 corresponds to a significant divergence between the training and validation curves, in dicating less stable training. In conclusion, the optimal combination of hyperparameters for training the models is

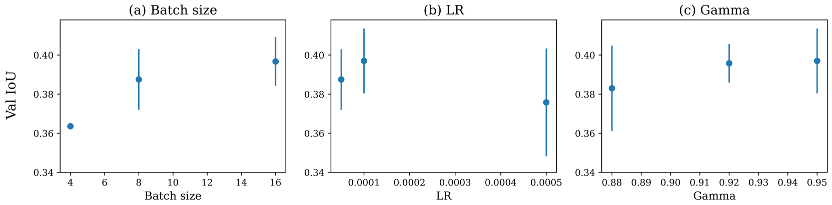


Figure 6.1: The results from the hyperparameter tuning tests are presented. The error bars on the graph represent the standard deviation of performances across the three validation set splits.

determined to be a batch size of 16, an initial learning rate of 0.0001, and a gamma of 0.92. This combination has been selected as the best training setting based on the balance between performance and stability.

6.2 Overall performances

Figure 6.2 summarises the performances of all the models in terms of IoUs and F1 scores. The model trained from scratch with a combined input of multispectral bands and SIAM-96 categories results in the highest IoU and F1 score. This is considerably higher than the baseline model trained only with multispectral data as input. Among the models trained from scratch, with only SIAM categories as input, the 96-class granularity performs the best. However, the 96-class granularity doesn't outperform the baseline model with multispectral input. An inspection of the errobars on the mean performance on validation set, indicate that these observations might be statistically significant.

The performance of the model that combines multispectral and SIAM-96 inputs is closely followed by models where both the encoder and the decoder were fine-tuned for the downstream task. However, the slight differences observed between the baseline and the finetuned models or among the finetuned models, might not be significant considering the variabilities shown by the errorbars.

Noting the initial assessment of SIAM-96 and its superior performance compared to other granularities, it was considered for subsequent SIAM-based models including, the model with combined input and the pretraining tasks.

Two observations confirm our intuition that SIAM based enrichment is perhaps well-suited as a complementary information source to sharpen the supervision signal however, it is not a substitute for multispectral data in land cover classification tasks. These two observations include, the comparable performance of the higher SIAM granularities with the baseline, if not more; and, the superior performance obtained when combining multispectral data with SIAM enrichment. Further, it is contrary to our intuition that the pretrained models do not significantly outperform the baseline. While pretraining is expected to enable the learning of useful features from the input data, allowing downstream tasks to focus on learning task-specific features, this does not appear to be the case here. Another important observation is that the pretrained models are less effective when used as feature extractors, with the downstream task freezing the backbone and only fine-tuning the decoder. [41] note a similar observation regarding the performance of finetuned and frozen-backbone models. They pointed out that this could be because of the skip connections in the architecture used to merge characteristics from the encoder with the decoder at corresponding scales.

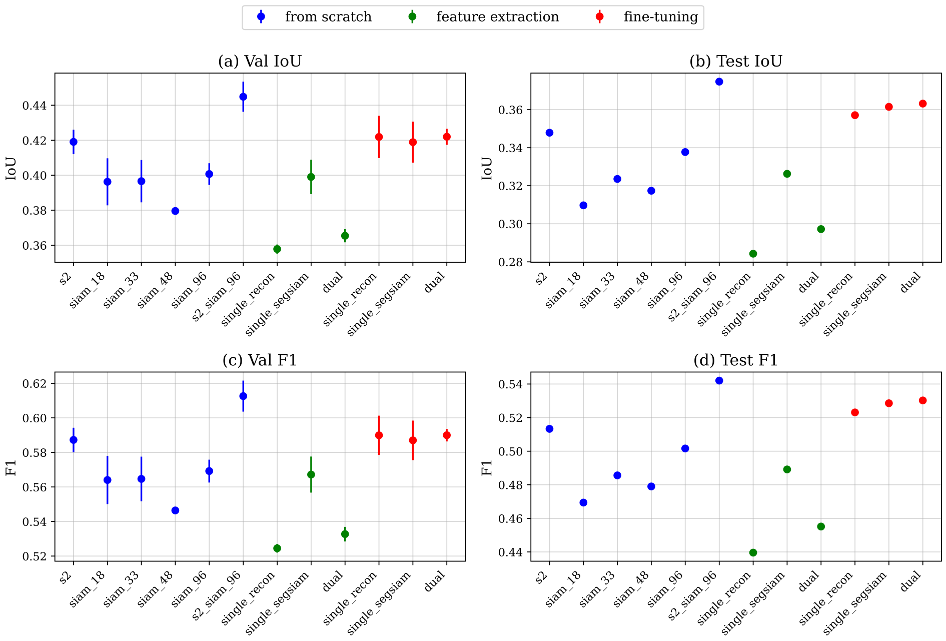


Figure 6.2: Performances using, IoU and F1 score, of all models are summarised here. The errorbars indicate standard deviation obtained from the three val splits

Additionally, the difference in the performances of the model between validation and test sets can be attributed to the inherent differences among the aggregated spectral characteristics of the patches in these sets stemming from the difference in the distribution of biomes among the two sets (an observation previously noted in the Chapter 3.)

6.3 Convergence analysis

In this section, the results from the convergence analysis are presented. The criterion for convergence was set as the epoch at which the model first reaches 98% of its peak performance (given by IoU). The GPU time to complete training until after this epoch was considered the time to reach convergence. The convergence time is calculated as a multiplicative product of the average time taken per epoch and the number of epochs taken to reach convergence.

Figure 6.3 summarises the results from the convergence analysis. The low variability expressed by the error bars for convergence time indicate that the time for convergence might be a more robust criteria for most of the experiments performed in this analysis for comparing convergence, than the epochs taken to converge. It is observed that the combined input model trained from scratch takes the highest number of epochs and the longest to converge. Almost all SIAM granularities converge quickly, this is most distinctively seen with the SIAM granularities 18 and 48. They take about the same number of epochs as the baseline to converge, however, they take far less computational time to reach this convergence. SIAM-based pretraining tasks, single or dual, result in faster convergence compared to the reconstruction tasks, both in terms of epochs and time taken. This behaviour is observed in both the variants of downstream training settings, frozen encoder and complete finetuning. Finally, it can be observed that frozen encoder pretraining tasks converge much faster compared to their finetuning counterparts, while the latter converge around the same time as the baseline.

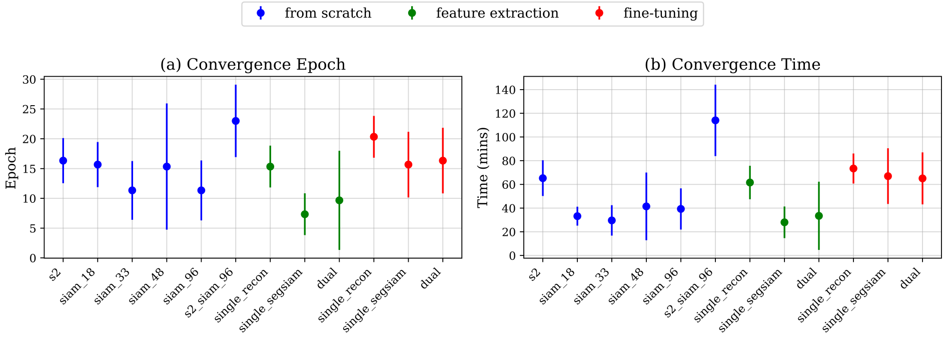


Figure 6.3: Results from the convergence analysis performed for all the models. (a) shows the epoch at which the model reached 98% of peak performance (b) time taken to train until then. The errobars indicate standard deviation obtained from the val splits.

Among the models trained from scratch, it was anticipated that the combined data would require fewer epochs to converge compared to using only multispectral input. This expectation arises because SIAM likely simplifies the prediction of certain land cover categories, while the multispectral input allows the model to to learn finer details. However, this wasn't observed in the results. Although, the long time and large number of epochs taken for the combined input to converge could also potentially be an outcome of the increased data dimensionality to 13 channels from its individual components with 3 or 10 bands. The same logic explains the fast convergence of all of the SIAM granularities compared to the high dimensional multispectral baseline.

Among the pretrained models, while it was expected that downstream tasks with frozen encoder would reach their peak performance fast in lieu of their least number of trainable parameters, it wasn't expected that the fully finetuned models would take about the same time or epochs as the baseline model to converge. This is because, the pretraining is used for initiliazing the downstream task after the architecture has learnt some useful features from the input unlabelled data, so it was supposed that the downstream task would take much less time to undergo task-specific finetuning.

6.4 The tradeoff: IoU vs convergence

Figure 6.4 summarises the combines results from the previous two sections. It is clear that while the combined input model, trained from scratch, yields the best performance, it also takes the longest to reach its peak performance. And while fully finetuned models perform slightly better compared to the baseline, they take about the same time to converge as the latter. All the other models, while they take much less time to converge, are inferior to the baseline in terms of performance. An examination of the errorbars indicate that these observations are likely to be statistically significant.

A combined reading of model performance and convergence analyses is necessary for better discussing the gains from the individual models, over the baseline. Such information has implications for tradeoffs to be considered in implementational settings.

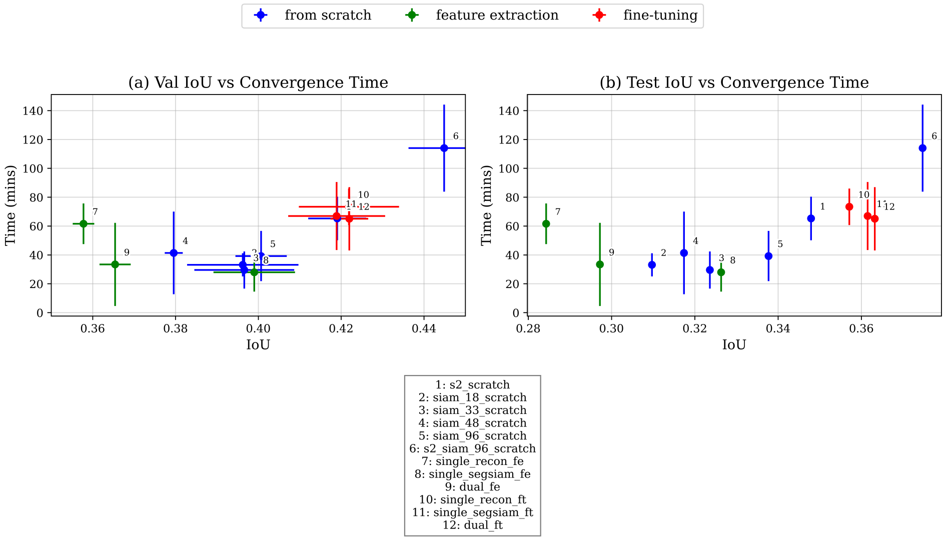


Figure 6.4: Illustration of the tradeoff between model performance and convergence speed. Errorbars indicate standard deviation obtained from the three splits

6.5 Dependence on labelled data

Analysing the dependence on the available labelled data is critical to understanding the model performance in low labelled data contexts. Figure 6.5 summarises the results obtained from training these models using 10%, 25%, 50% and 100% of the labelled data within each stratified split. It is observed that the model trained from scratch with the combined input of multispectral and SIAM-96 data outperforms all the models, including pretrained ones, across volumes of available labelled data. Followed by, the performance of the dual task, and then other pretraining tasks. Finally, the model with multispectral input, trained from scratch performs poorly across volumes of labelled data except for the least amount of labelled data (10%), wherein pretraining based on SIAM alone performs the worst. However, this set of observations are less and less likely to be statistically significant in experiments performed with dereasing amounts of labelled data.

This goes to show that combining multispectral input with SIAM data, more specifically SIAM-96 granularity data, might tremendously improve the generalisability and the performance of deep learning models for land cover classification.

6.6 Class-wise comparison

Figure 6.6 showcases the class-wise performances of each of the models. It shows that the model trained from scratch with the combined input of multispectral data and SIAM-96, performs the best across most of the land cover classes except for snow and cloud. Here, almost all models trained on SIAM input alone perform the best. However, when combined with multispectral data, the performance falls down on these land cover classes, as shown by the combination input model. Another observation is that SIAM-based pretrained models outperform the baseline model for most land cover classes. This observation is more evident in some classes (water, cloud, bare ground) than others (scrub, crop, tree).

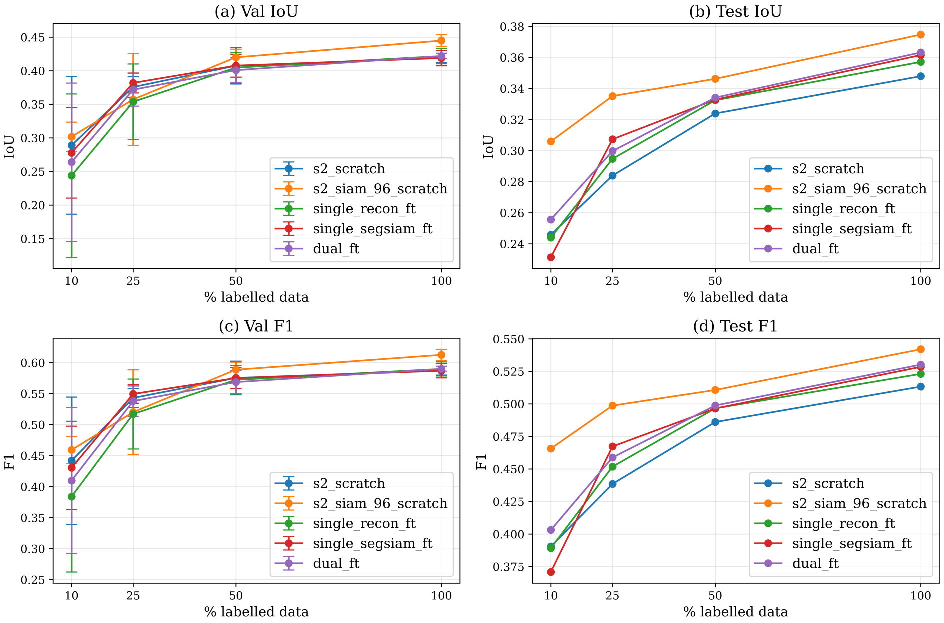


Figure 6.5: Results comparing model performances based on the amount of labelled data used for training.

The visibly better performance of all SIAM granularities on snow and cloud is potentially because all of them have designated spectral categories describing clouds and snow. Hence, this makes it easier for the model to learn to predict the respective categories from the target Dynamic World classes. However, when combined with multispectral data, as in the model trained with combined input, some noise could be introduced, making it difficult to predict these classes.

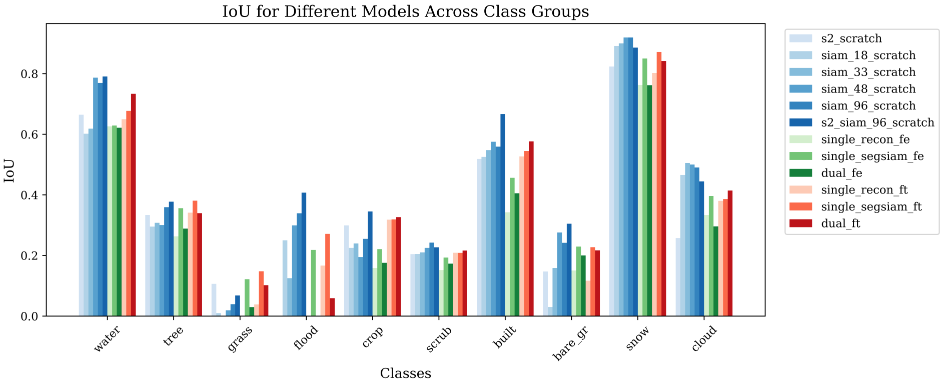


Figure 6.6: Illustration of class-wise performance of all the models.

7 Discussion and Conclusions

This study demonstrates the incorporation of SIAM semantic enrichment in deep learning scenarios for the semantic segmentation task of land cover classification. The results of this study reveal the impact of including SIAM data in the training pipeline for land cover classification models. Models trained from scratch with a combination of multispectral bands and SIAM-96 categories achieved the highest IoU and F1 scores, outperforming the baseline model trained solely on multispectral data. This indicates that SIAM data significantly enriches the model's learning process, enhancing its predictive capabilities. The convergence analysis showed that although the combined input model required the longest time to converge, SIAM granularities (particularly 18 and 48) converged quickly and efficiently, suggesting their potential for rapid model training.

A notable trade-off was observed between performance and convergence time. The highest-performing model, despite taking the longest to reach peak performance, showcased the advantages of integrating multispectral and SIAM data. This trade-off is critical for practical applications where computational resources and time are constrained. Furthermore, the model trained with combined multispectral and SIAM-96 data consistently outperformed others across varying amounts of labeled data, demonstrating its robustness and generalisability. This robustness was particularly evident in low labeled data scenarios, where the combined input model maintained superior performance, while models based solely on SIAM data showed poorer results. Class-wise performance analysis revealed that the combined input model excelled across most land cover classes, except for snow and cloud, where models trained solely on SIAM input performed better, which can be explained by the designated spectral categories for snow and cloud. However, when combined with multispectral data, a fall in performance was observed on these classes hinting at potential noise introduced by the multispectral data.

Pretrained models exhibited interesting behaviors that merit further exploration. While pretrained models were expected to significantly outperform the baseline by leveraging learned features from the input data, this was not always observed. In particular, pretrained models with frozen encoders converged faster due to the reduced number of trainable parameters, yet they did not significantly surpass the performance of fully finetuned models or even the baseline. This suggests that pretraining may not always transfer useful features effectively for downstream tasks in land cover classification. However, it needs to be highlighted that this study used limited data for the pretraining tasks and a fairly simple pretraining design. In context of advanced self-supervised pretraining strategies being extensively researched, some of which were reviewed in this study, it is recommended that future research should delve deeper into optimizing pretraining techniques, perhaps by experimenting with different architectures, pretraining tasks, and data augmentation strategies. Additionally, exploring the potential of SIAM data in other remote sensing applications and investigating how different types of auxiliary data could further enhance model performance will be crucial. Such efforts could lead to more robust, generalisable models that balance performance, convergence speed, and computational efficiency across various contexts and applications.

Bibliography

[1] Agrawal, P., Girshick, R., and Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition.

[2] Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., and Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

[3] Audebert, N., Saux, B. L., and Lefèvre, S. (2017). Joint learning from earth observation and openstreetmap data to get faster better semantic maps.

[4] Ballard, D. H. (1987). Modular learning in neural networks. In *AAAI Conference on Artificial Intelligence*.

[5] Baraldi, A. (2011). Satellite image automatic mapper™ (siam™) - a turnkey software executable for automatic near real-time multi-sensor multi-resolution spectral rule-based preliminary classification of spaceborne multi- spectral images. *Recent Patents on Space Technology*, 1:81–106.

[6] Baraldi, A., Humber, M. L., Tiede, D., and Lang, S. (2017). Stage 4 validation of the satellite image automatic mapper lightweight computer program for earth observation level 2 product generation, part 2 validation.

[7] Berg, P., Pham, M.-T., and Courty, N. (2022). Self-supervised learning for scene classification in remote sensing: Current state of the art and perspectives. *Remote Sensing*, 14(16).

[8] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2019). Deep clustering for unsupervised learning of visual features.

[9] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations.

[10] Dash, T., Chitlangia, S., Ahuja, A., and Srinivasan, A. (2022). A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1).

[11] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Felix Kröber1] Felix Kröber1, Dirk Tiede1, A. B. S. L. Semantic world – a novel benchmark dataset for semi-supervised semantic segmentation.

[13] Ghamisi, P., Rasti, B., Yokoya, N., Wang, Q., Hofle, B., Bruzzone, L., Bovolo, F., Chi, M., Anders, K., Gloaguen, R., Atkinson, P. M., and Benediktsson, J. A. (2019). Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39.

[14] GmbH, T. G. (2021). Trimble documentation ecognition developer 10.1 reference book. *Trimble Germany GmbH*.

[15] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[16] Hartmann, A., Sudmanns, M., Augustin, H., Baraldi, A., and Tiede, D. (2023). Estimating the temporal heterogeneity of mowing events on grassland for haymilk-production using sentinel-2 and greenness-index. *Smart Agricultural Technology*, 4:100157.

[17] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[18] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

[19] Häufel, G., Bulatov, D., Pohl, M., and Lucks, L. (2018). Generation of training examples using osm data applied for remote sensed landcover classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 7263–7266.

[20] Hübl, F., Augustin, H., Sudmanns, M., Tiede, D., and Scholz, J. (2024). Water body detection using sen2cube.at and comparison to open government data - assessing for floating photovoltaics. *AGILE: GIScience Series*, 5:1–5.

[21] Iakubovskii, P. (2019). Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch.

[22] Ienco, D., Ose, K., and Weber, C. (2019). Towards combining satellite imagery and vgi for urban lulc classification. In *2019 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4.

[23] Islam, S. R., Eberle, W., Bundy, S., and Ghafoor, S. K. (2019). Infusing domain knowledge in ai-based "black box" models for better explainability with application in bankruptcy prediction.

[24] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2018). Image-to-image translation with conditional adversarial networks.

[25] Johnson, M. D., Hsieh, W. W., Cannon, A. J., Davidson, A., and Bédard, F. (2016). Crop yield forecasting on the canadian prairies by remotely sensed vegetation indices and machine learning methods. *Agricultural and Forest Meteorology*, 218-219:74–84.

[26] Kaiser, P., Wegner, J. D., Lucchi, A., Jaggi, M., Hofmann, T., and Schindler, K. (2017). Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068.

[27] Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.

[28] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles.

[29] Li, H., Li, Y., Zhang, G., Liu, R., Huang, H., Zhu, Q., and Tao, C. (2022). Global and local contrastive self-supervised learning for semantic segmentation of hr remote sensing images.

[30] Liu, P., Shi, R., Zhang, C., Zeng, Y., Wang, J., Zhu, T., and Gao, W. (2017). Integrating multiple vegetation indices via an artificial neural network model for estimating the leaf chlorophyll content of spartina alterniflora under interspecies competition. *Environmental Monitoring and Assessment*, 189.

[31] Marcus, G. (2018). Innateness, alphazero, and artificial intelligence.

[32] Montanaro, A., Valsesia, D., Fracastoro, G., and Magli, E. (2022). Semi-supervised learning for joint sar and multispectral land cover classification.

[33] Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A., and Ramakrishnan, N. (2018). Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 36–45.

[34] Neumann, M., Pinto, A. S., Zhai, X., and Houlsby, N. (2020). Training general representations for remote sensing using in-domain knowledge. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 6730–6733.

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[36] Raschka, S. (2020). Model evaluation, model selection, and algorithm selection in machine learning.

[37] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression.

[38] Richter, R. (2011). Atmospheric / topographic correction for satellite imagery. *DLR report DLR-IB 565-02/11, Wessling, Germany*, 7.

[39] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

[40] Saha, S., Shahzad, M., Mou, L., Song, Q., and Zhu, X. X. (2022). Unsupervised single-scene semantic segmentation for earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11.

[41] Scheibenreif, L., Hanna, J., Mommert, M., and Borth, D. (2022). Self-supervised vision transformers for land-cover segmentation and classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1421–1430.

[42] Sharma, R. C. and Hara, K. (2021). Self-Supervised learning of Satellite-Derived vegetation indices for clustering and visualization of vegetation types. *J Imaging*, 7(2).

[43] Singh, S., Batra, A., Pang, G., Torresani, L., Basu, S., Paluri, M., and Jawahar, C. V. (2018). Self-supervised feature learning for semantic segmentation of overhead imagery. In *British Machine Vision Conference*.

[44] Sudmanns, M., Tiede, D., Wendt, L., and Baraldi, A. (2017). Automatic ex-post flood assessment using long time series of optical earth observation images.

[45] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing.

[46] Tait, A. M., Brumby, S. P., Hyde, S. B., Mazzariello, J., and Corcoran, M. (2021). Dynamic World training dataset for global land use and land cover categorization of satellite imagery.

[47] Tao, C., Qi, J., Guo, M., Zhu, Q., and Li, H. (2023). Self-supervised remote sensing feature learning: Learning paradigms, challenges, and future works. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–26.

[48] Tassi, C., Gawlikowski, J., Fitri, A., and Triebel, R. (2022). The impact of averaging logits over probabilities on ensembles of neural networks.

[49] Tiede, D., Sudmanns, M., Augustin, H., and Baraldi, A. (2021). Investigating esa sentinel-2 products' systematic cloud cover overestimation in very high altitude areas. *Remote Sensing of Environment*, 252:112163.

[50] Wang, D., Zhang, J., Du, B., Xia, G.-S., and Tao, D. (2023). An empirical study of remote sensing pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20.

[51] Wang, Y., Albrecht, C. M., Braham, N. A. A., Mou, L., and Zhu, X. X. (2022). Self-supervised learning in remote sensing: A review.

[52] Wang, Z., Guo, J., Huang, W., and Zhang, S. (2021). High-resolution remote sensing image semantic segmentation based on a deep feature aggregation network. *Measurement Science and Technology*, 32(9):095002.

[53] Xie, X., Niu, J., Liu, X., Chen, Z., Tang, S., and Yu, S. (2021). A survey on incorporating domain knowledge into deep learning for medical image analysis. *Medical Image Analysis*, 69:101985.

[54] Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang, Q., Wang, J., Gao, J., and Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 241:111716.

[55] Zhang, J. (2010). Multi-source remote sensing data fusion: status and trends. *International Journal of Image and Data Fusion*, 1(1):5–24.

[56] Zhao, J., Zhang, D., Shi, B., Zhou, Y., Chen, J., Yao, R., and Xue, Y. (2022). Multi-source collaborative enhanced for remote sensing images semantic segmentation. *Neurocomputing*, 493:76–90.

[57] Zhao, W., Bo, Y., Chen, J., Tiede, D., Blaschke, T., and Emery, W. J. (2019). Exploring semantic elements for urban scene recognition: Deep integration of high-resolution imagery and openstreetmap (osm). *ISPRS Journal of Photogrammetry and Remote Sensing*, 151:237–250.

[58] Zheng, Y., Yang, Y., and Chen, B. (2021). Incorporating prior financial domain knowledge into neural networks for implied volatility surface prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, page 3968–3975, New York, NY, USA. Association for Computing Machinery.

Appendix-I

This Appendix showcases predicted segmentation maps by all the models compared in this study for visual comparison of their performances. (charts from the following page)

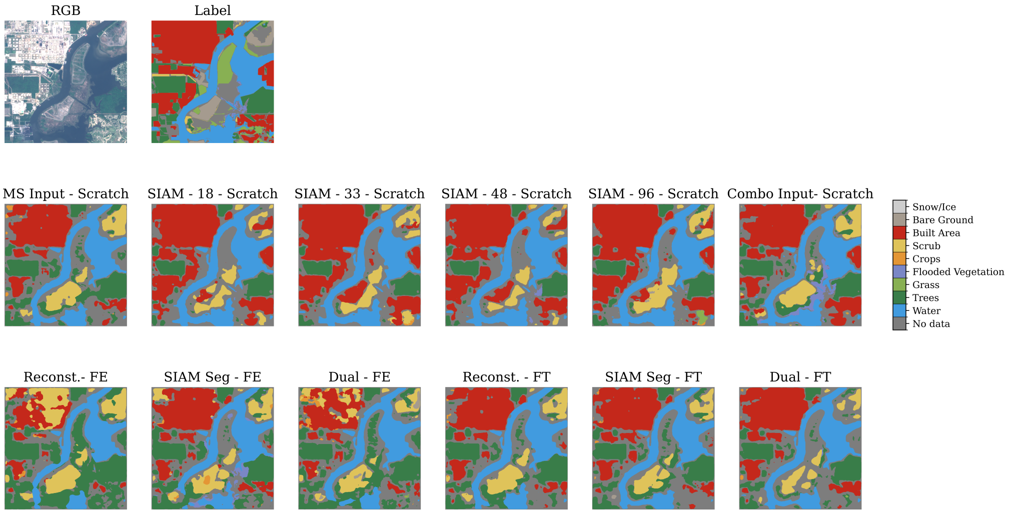


Figure 7.1: Predicted maps by models on sample patch 1

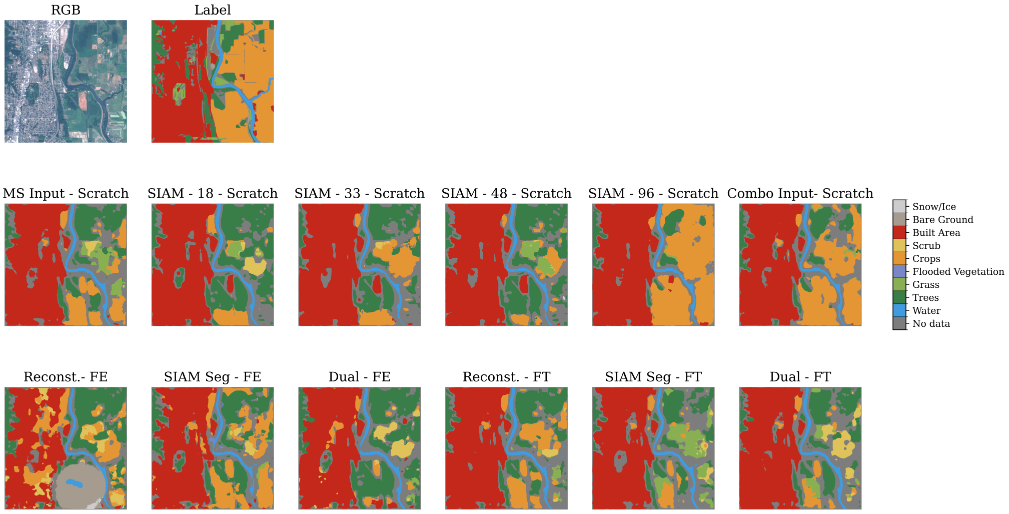


Figure 7.2: Predicted maps by models on sample patch 2

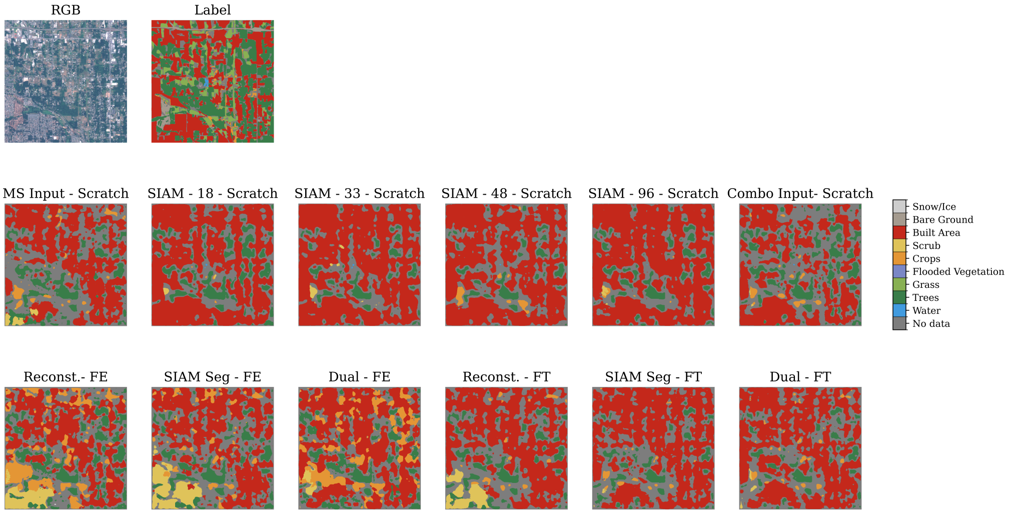


Figure 7.3: Predicted maps by models on sample patch 3

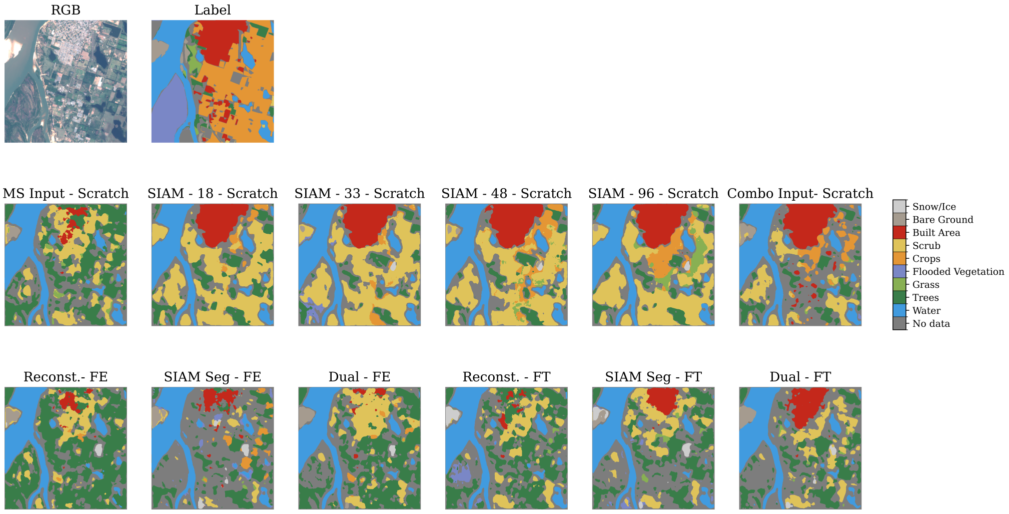


Figure 7.4: Predicted maps by models on sample patch 4

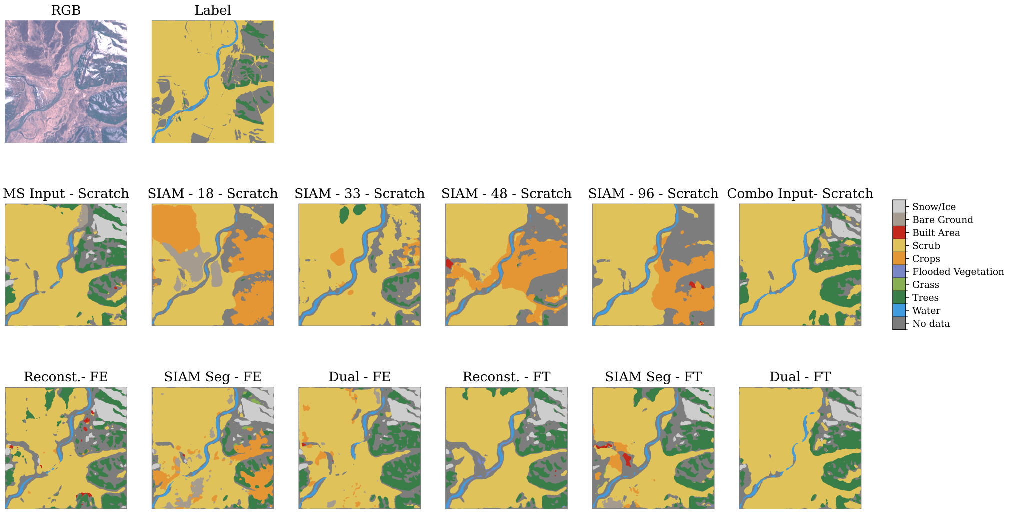


Figure 7.5: Predicted maps by models on sample patch 5

Appendix - II

This Appendix contains the loss curves of all the models trained in this study. These curves serve as a diagnostic tool to monitor the training behaviour of the models (charts from the following page).

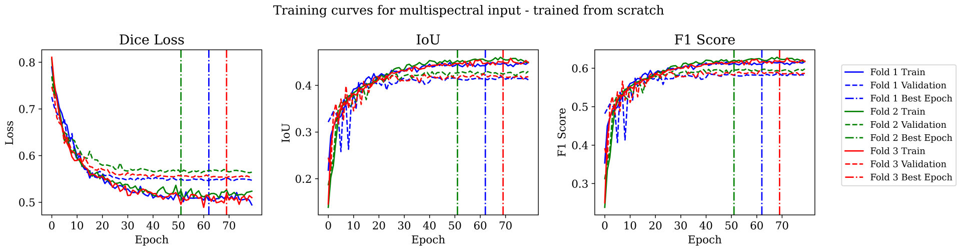


Figure 7.6: Training curves of the model trained from scratch with multispectral input

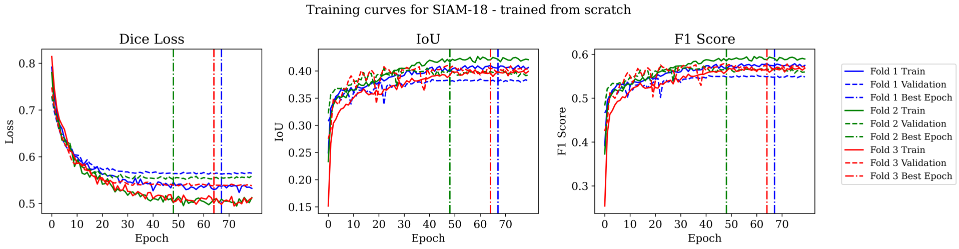


Figure 7.7: Training curves of the model trained from scratch with SIAM-18 input

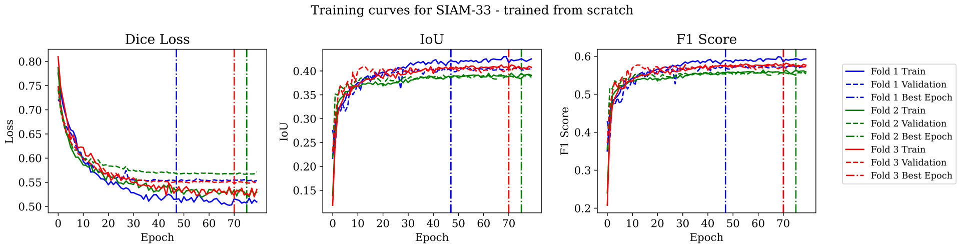


Figure 7.8: Training curves of the model trained from scratch with SIAM-33 input

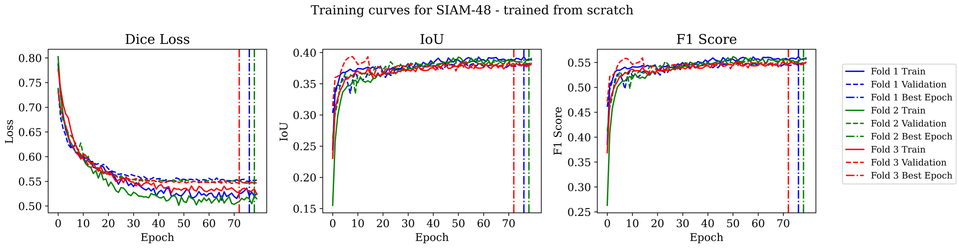


Figure 7.9: Training curves of the model trained from scratch with SIAM-48 input

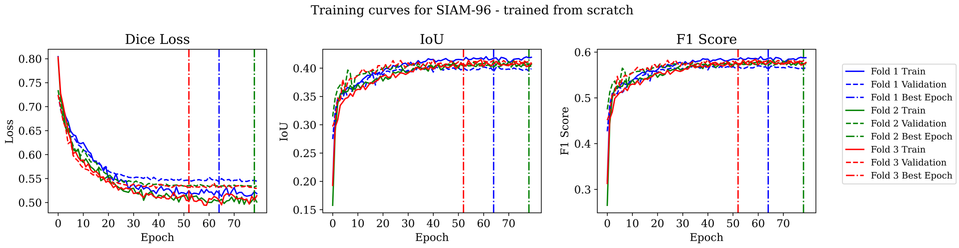


Figure 7.10: Training curves of the model trained from scratch with SIAM-96 input

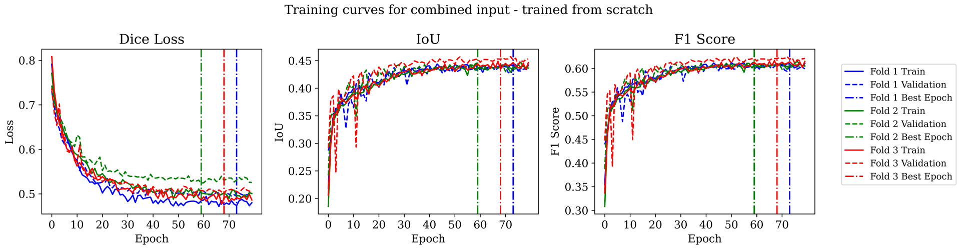


Figure 7.11: Training curves of the model trained from scratch with combined input of multispectral and SIAM-96 data

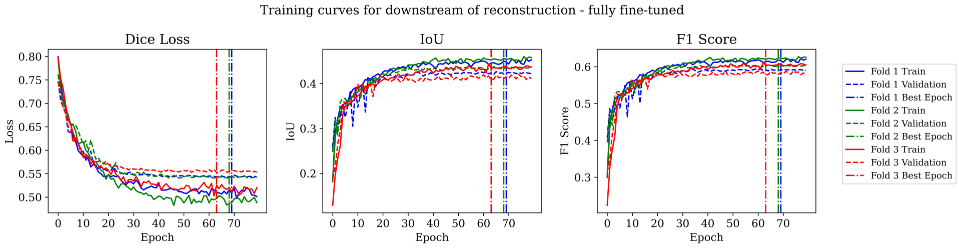


Figure 7.12: Training curves of the downstream task finetuned after reconstruction based pretraining

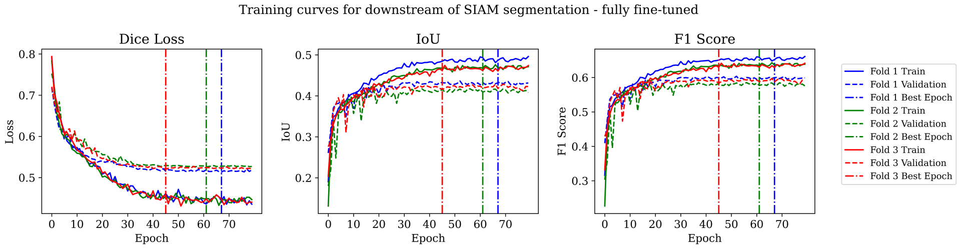


Figure 7.13: Training curves of the downstream task finetuned after SIAM-96 based pretraining

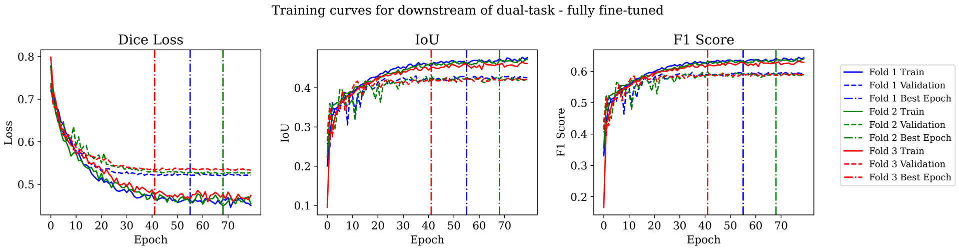


Figure 7.14: Training curves of the downstream task finetuned after dual task pretraining

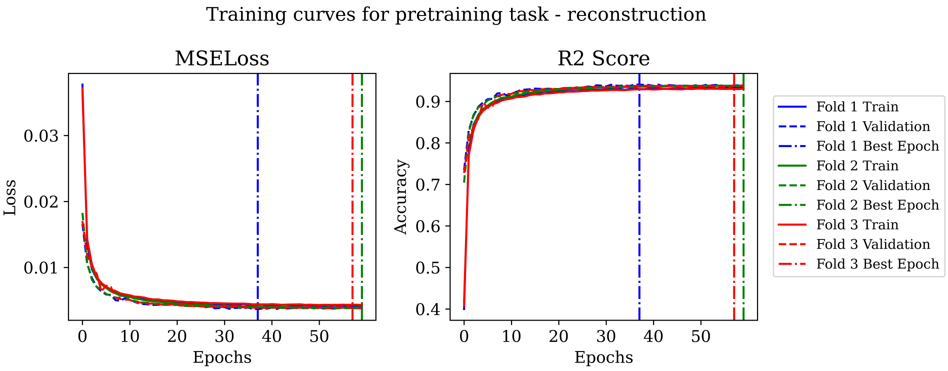


Figure 7.15: Training curves of the reconstruction based pretraining

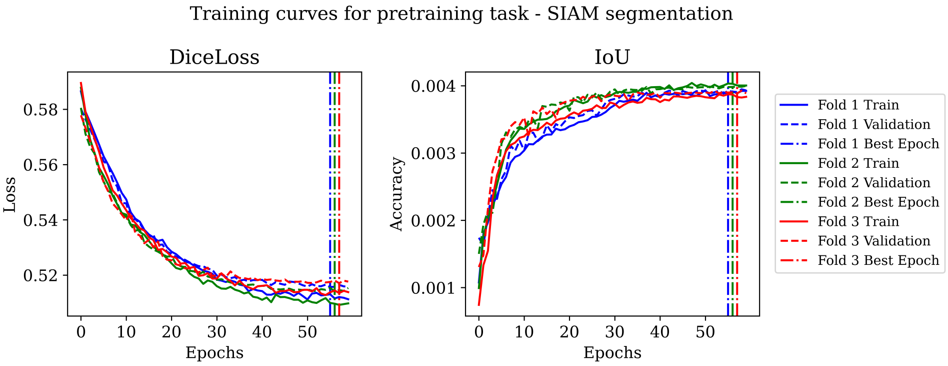


Figure 7.16: Training curves of the SIAM-96 based pretraining

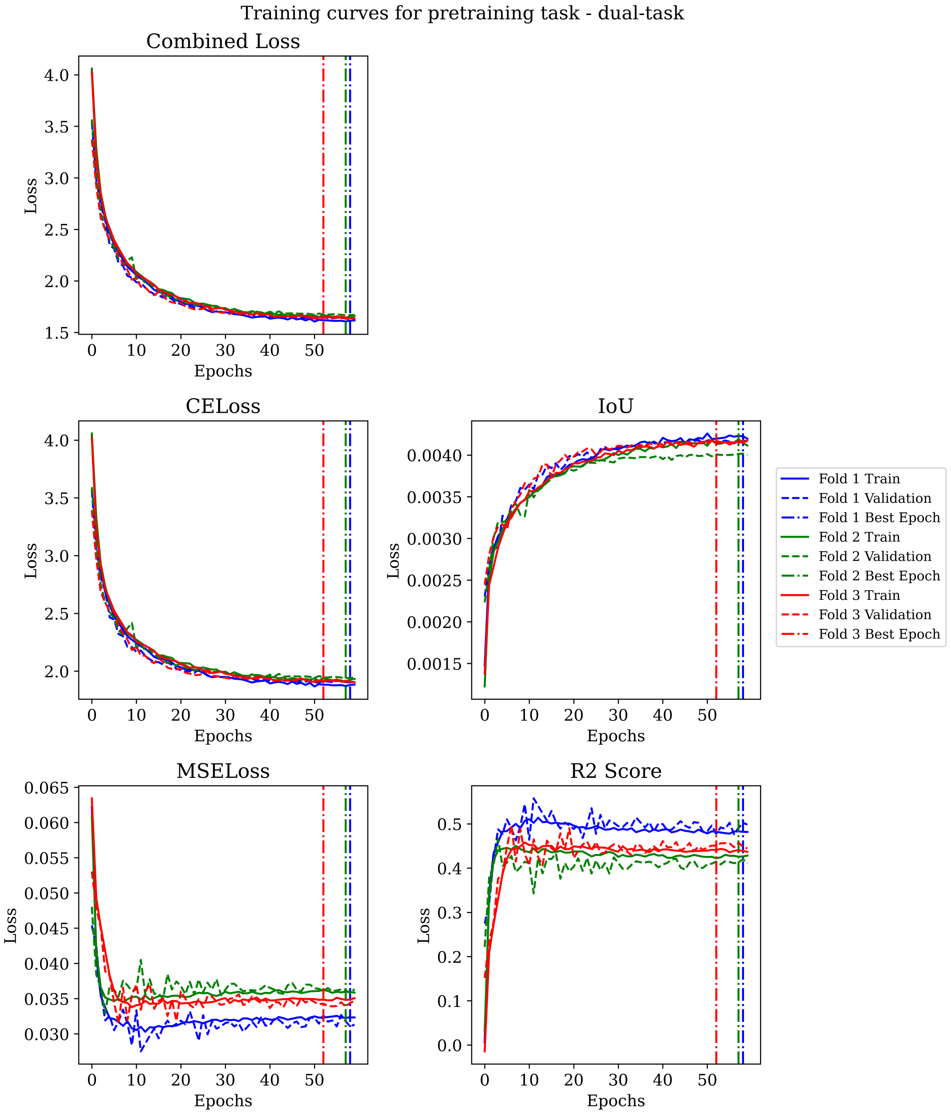


Figure 7.17: Training curves of the dual task pretraining, one task reconstructs output and the other predicts a segmented map for SIAM categories

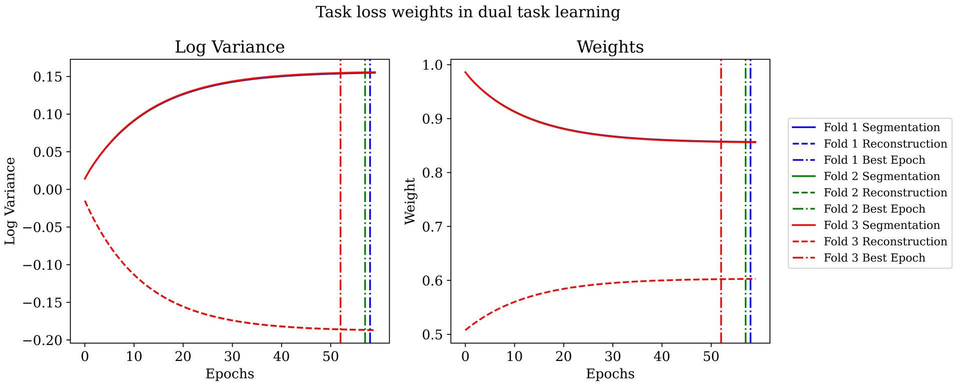


Figure 7.18: Evolution of loss weights for each task in dual task learning