Master's Thesis
Course of Studies Data Science

Integrating Satellite Image Time Series in Multi-Modal Foundation Models for Enhanced Land Use and Land Cover Segmentation

Reiko Lettmoden

Registration number 4945013

December 13, 2024

Institute of Geodesy and Photogrammetry
Prof. Dr.-Ing. Markus Gerke

Institute of Computer Graphics
Prof. Dr.-Ing. Martin Eisemann

Supervisors:
Dr. Pedro M. Achanccaray Diaz
Dr. rer. nat. Ksenia Bittner

Statement of Originality

This thesis has been performed independently with the support of my supervisor/s. To the best of the author's knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text.

Braunschweig, December 13, 2024

Contents

1. Introduction **1**

2. Related Works **5**

3. Background **11**
3.1. Active and Passive Sensors 11
3.2. Image Recognition in Remote Sensing 12
3.3. Dynamic One For All (DOFA) 14
3.4. Spatial Decoders 15
3.5. Temporal Encoders 16
3.6. Spatio-Temporal Architecture 18
3.7. Multi-Modal Fusion 18
3.8. Multi-Modal and Multi-Temporal Regularization 19

4. Material and Methods **21**
4.1. Dataset 21
4.2. Methodology 25

5. Experiments **29**
5.1. Implementation 29
5.2. Experimental Protocol 30
5.3. Results 33
5.3.1. Quantitative Results 33
5.3.2. Qualitative Evaluation 50

6. Discussion **53**

7. Conclusion **57**

Bibliography **59**

A. Appendix **63**
A.1. Auxiliary Supervision 63
A.2. Additional Experiments 63
A.3. Additional Visualizations 64

List of Figures

Figure 3.1. Vision Transformer (ViT) architecture by Dosovitskiy et al. [8] . . . 13
Figure 3.2. DOFA by Xiong et al. [41] . . . 14
Figure 3.3. Masked Autoencoder by He et al. [18] . . . 15
Figure 3.4. UPerNet architecture by Xiao et al. [38] . . . 16
Figure 3.5. L-TAE by Garnot and Landrieu [11] . . . 17
Figure 3.6. Progression of operations for the U-TAE[12] model . . . 18
Figure 3.7. Progression of operations for the U-BARN[9] model . . . 18
Figure 3.8. Mid Fusion visualized by Garnot et al. [13] . . . 19

Figure 4.1. The covered Harz region . . . 22
Figure 4.2. Cloudy Sentinel-2 sample. . . . 22
Figure 4.3. PlanetScope data from 2020 with invalid stripes . . . 23
Figure 4.4. Visualization of VH from Sentinel-1 with missing data. . . . 24
Figure 4.5. Labels for the years 2020 and 2021. . . . 24
Figure 4.6. The label distribution of the Harz dataset. . . . 25
Figure 4.7. Methodology of this work . . . 26
Figure 4.8. Forward flow of the model based on the fusion method and spatio-temporal architecture . . . 27

Figure 5.1. Training graphs for mIoU across epochs with ablations in: a) Learning Rate b) Augmentations c) Temporal Dropout . . . 34
Figure 5.2. Training graph of the mIoU along epochs for early fusion (red) and late fusion with concatenation (green and blue) . . . 37
Figure 5.3. Class-wise IoU and mIoU for each distinct modality and the combination of all modalities . . . 39
Figure 5.4. Test mIoU scores for different temporal encoders across modalities 43
Figure 5.5. Class wise IoU scores and mIoU across different temporal encoders for different modalities: a) Sentinel-1 b) PlanetScope c) Sentinel-2 d) All. Note, that the *L-TAE* configuration is not stated in Figure 5.5d but instead the configuration *L-TAE S1 & PlanetScope/S2 Temporal Max-Pool* . . . 44
Figure 5.6. Confusion matrices normalized in vertical (a) and horizontal (b) dimension. The class *tree* is treated as a background class during training . . 45
Figure 5.7. The model trained with different weight initializations of the DOFA[41]. ImageNet[7] weights only initialize the ViT-B[8] backbone. . . . 46
Figure 5.8. mIoU scores for models initialized with DOFA[41] weights (straight lines) or no pre-trained weights (dotted) in relation to different amounts of available data . . . 47

Figure 5.9. Effectiveness of modalities measured in FLOPs in relation to the test mIoU for each configuration with color codes based on different subsets of modalites 48
Figure 5.10. Qualitative visualization for *L-TAE/Max-Pool All* configuration. Class labels for *built-up* are more fine-granular than the predicted label map of the model 50
Figure 5.11. Qualitative visualization for *L-TAE/Max-Pool All* configuration. Labels of the class *grass* mark country roads and field paths which are hard to learn for the model 51
Figure 5.12. Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangles, noisy labels and predictions of the class *Crop* are visualized 51
Figure 5.13. Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangles, differences in labels and prediction of the class *dead tree* are visualized 52
Figure 5.14. Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangle, differences in labels and predictions regarding the class *grass* are visualized 52

Figure A.1. IoU values per class obtained in the test set with a comparison of the main heads trained with no auxiliary head (purple) and auxiliary heads (red) along with each auxiliary head's prediction different λ^{S1} values: a) 0.5 and b) 0.25 64
Figure A.2. Auxiliary losses obtained in the train and validation set for different auxiliary heads with different λ^{S1} values: a) 0.5 and b) 0.25 66
Figure A.3. Effectiveness of modalities measured in FLOPs in relation to the test mIoU for each configuration with color codes based on different subsets of modalites. Added experiments *ViT-S* and *ViT-S* with down-scaled parameters 67
Figure A.4. Qualitative visualization for *L-TAE/Max-Pool All* configuration. Label inconsistencies for the class *dead tree* are shown in the red boxes 67
Figure A.5. Qualitative visualization for *L-TAE/Max-Pool All* configuration. Label inconsistencies for the class *dead tree* are shown in the red boxes 68
Figure A.6. Qualitative visualization for *L-TAE/Max-Pool All* configuration. Class labels for *built-up* are more fine-granular than the predicted label map of the model, as visualized in the blue boxes 68

List of Tables

Table 2.1. Remote sensing datasets for tree and land cover classification adapted from Garioud et al.[10]. The spatial resolutions of the single-date images and labels, and the temporal resolutions of time series are provided in parentheses. S1/S2 denotes Sentinel-1 and 2, Planet denotes PlanetScope . 5

Table 3.1. Comparison of Sentinel-1, Sentinel-2 (Bold indicates that the band is used in this work), and PlanetScope Satellite Data Characteristics 11

Table 4.1. Available data of the multi-modal and multi-temporal Harz dataset including PlanetScope, Sentinel-2, and Sentinel-1 data 21

Table 5.1. GitHub repositories and links for various models, frameworks, and components used in the thesis 29
Table 5.2. Ablation studies conducted in this work 32
Table 5.3. Experiment Results: mIoU and Loss for Train, Validation, and Test Sets. For every experiment, the experiment above in the table is set as baseline with the first experiment being the standard configuration. * LR: Learning Rate 33
Table 5.4. Comparison of different configurations using auxiliary heads with varying λ^{S1} regarding their class-wise IoU (stated in %) scores and evaluation metrics on the test set 35
Table 5.5. Comparison of model architectures and their performance. The spatial encoder requires 333 GFLOPs for all modalities. * Parameters per modality except for the shared decoder. For the *DOFA* configuration, the decoder includes the parameters of the neck and UperNet[38] ** Partial double means a double block is employed for S1 and S2, while a single block is used for PlanetScope. 36
Table 5.6. Comparison of fusion methods in test mIoU and computational requirements measured in FLOPs and parameters. To measure the effect of few Sentinel-1 data caused by early fusion, a configuration with high Sentinel-1 dropout $p_{S1} = 0.9$ was added 36
Table 5.7. Comparison of the spatio-temporal UBARN[9] and U-TAE[12] architecture in computational resources and test mIoU 38
Table 5.8. Comparison of different combinations of modalities regarding computational resources and test mIoU 40
Table 5.9. Comparison of ViT[8] configurations in the DOFA[41] as spatial encoder regarding test mIoU and computational resources 41

Table 5.10. Comparison of temporal configurations applied to different sets of modalities regarding their computational resources and test mIoU. *ST All - Modality* denotes the model trained on all modalities and tested on the specified modality. *L-TAE/Max-Pool* denotes temporal max-pooling for Sentienl-2 and PlanetScope and L-TAE[11] used with Sentinel-1 data. * Total GFLOPs can be optimized to 486.9 (=S1+S2+Planet) 41
Table 5.11. Comparison of ablations in the spatial decoder, fusion method, spatio-temporal architecture, temporal encoder, modalities, weight initialization and spatial encoder size w.r.t. computational requirements and test mIoU. *L-TAE/Max-Pool* denotes temporal max-pooling for Sentienl-2 and PlanetScope and L-TAE[11] used for Sentinel-1. * Total GFLOPs can be optimized to 486.9 (=S1+S2+Planet) 49

Table A.1. Comparison of different model weight initialization in combination with different ViT[8] configurations regarding test mIoU and computational requirements. Regarding temporal feature encoding, L-TAE[11] is applied to Sentinel-1 and max-pooling to Sentinel-1 and PlanetScope . . . 65
Table A.2. ViT[8] configurations with spatial decoder, L-TAE[11] and segmentation head hyperparameters 65
Table A.3. Experiment Results for mIoU and loss for train, validation, and test Sets across different ViT[8] backbone variants trained from scratch. The ViT-Ti configuration barely overfits in comparison to the larger configurations 65

1. Introduction

Climate change and its associated extreme weather conditions have become increasingly pressing concerns in recent years, with consequences for local vegetation and ecosystems. In Europe, climate change causes a change in precipitation patterns, resulting in more extreme intensities[34]. This results in floods or heatwaves causing damage to human lives, ecosystems and infrastructure[34, 26].

To monitor these, remote sensing (RS) provides useful data to generate land use and land cover maps (LULC) on a large scale. As tasks in remote sensing commonly include fine-granular localization or a pixel-wise classification, e.g. semantic segmentation, techniques from computer vision are widely adapted for image recognition[29, 2]. Therefore, machine learning aims to learn and recognize patterns based on data without the explicit programming of patterns. Due to the availability of large datasets for training, powerful models can generalize well on unseen data.

As a sub-domain of machine learning, deep learning leverages deep neural networks containing millions[41] to billions[15] of parameters to learn complex features from imagery[21, 17]. In particular, Convolutional Neural Networks (CNNs)[17] and transformer[8] models have significantly improved the accuracy of land cover classifications[2, 29]. Recent advancements in deep learning have been achieved through self-supervised learning, which allows to train computer vision backbones with vast amounts of cheap unlabelled data and thus initialize models for efficient fine-tuning on down-stream tasks such as semantic segmentation[15, 1]. For fine-tuning, pre-trained foundation models require fewer training resources such as data and training time while achieving a better performance[9, 15].

Remote sensing data is collected by a variety of missions such as Sentinel-1 and PlanetScope deploying sensors with different characteristics. Passive sensors receive a broad electromagnetic spectrum which is especially crucial to distinguish vegetation[13]. Passive sensors are commonly deployed with different amounts of wavelengths and capture imagery at a different ground sample distance (spatial resolution). Active sensors, on the other side, have the advantage of being weather-independent by emitting and measuring their own energy. To utilize all data available provided by each sensor (modality), multi-modal models were introduced to learn joint feature representation across modalities[1, 15].

While multi-modal approaches help integrate different sensor types, weather phenomena present challenges for data acquisition, as optical sensors can be impaired by cloud cover and other atmospheric conditions[12]. Additionally, vegetation changes significantly over growing seasons, such that multi-temporal acquisitions provide both meaningful features for the task and reduce the dependency on correct data acquisitions.

In the recent year, multi-modal and multi-temporal models[15, 1] emerged, but are either too light-weight in the spatial encoder to extract complex features from high-resolution modalities like the OmniSat[1] model or are too computationally expensive to be fine-tuned on consumer hardware like the SkySense[15] model. To reduce computational requirements by training a single powerful spatial encoder across multiple modalities, multi-modal foundation models such as the DOFA[41] and msGFM[16] were recently proposed. While they are trained across multiple modalities, they are currently limited to single-temporal data.

Hence, this thesis enhances the multi-modal foundation model DOFA[41] with satellite image time series. Methods for multi-modal and multi-temporal regularization are evaluated to reduce overfitting. Ablation studies regarding the model efficiency are conducted to provide both recommendations for lightweight and efficient models. Therefore, experiments are conducted with different fusion methods, spatio-temporal architectures, spatial decoders, spatial encoder configurations and temporal encoders.

As a case study, the model is trained and evaluated on a dataset based on the Harz region. A more extreme distribution of precipitation in combination with global warming causes more challenges for the ecosystem of the Harz[26]. In the Harz forest, 70% of the forested area is covered by Norway Spruce, which is especially vulnerable to the bark beetle. Due to climate change, summer becomes drier and the growing season longer due to global warming[26]. This causes the pests to spread faster, and further outbreaks are predicted for the years to come, endangering the Harz ecosystem. For this work, all available data in the scope of the EXDIMUM[1] project is used for the model. In particular, Synthetic Aperture Radar (SAR) from Sentinel-1, multi-spectral data from Sentinel-2, and high-resolution PlanetScope data are fused in a unified model to make a single semantic segmentation prediction per growing season. Sentinel-1 data is given in high-temporal sequence length per growing season, while for optical data, only short-time sequences are available due to a focus on the growing season. This differs from previous work in which temporal features are encoded across long temporal sequences[1, 13]. The semantic labels are provided for the growing seasons 2020 and 2021 by ESA using random forests[4]. To distinguish between coniferous and deciduous trees, labels generated by a random forests classifier are provided by Blickendorfer et al.[3], while dead trees are annotated manually. LULC maps are produced with semantic segmentation to measure the bark beetle's influence on the tree population. Hence, this work aims to measure the impact of climate change on the ecosystem of the Harz region in Germany.

The key contributions of this work can be summarized in the following three points:

- Enhancing the multi-modal DOFA[41] with temporal feature extraction and integrating different fusion methods in a modular framework
- Conducting a case study on a regional dataset Harz dataset for land use and land cover classification with a focus on tree species and dead trees

[1] https://www.exdimum.org/

- Investigating the impact of spatio-temporal architecture, fusion method, spatial decoder, each modality, temporal feature extraction and parameter scaling with regard to FLOPs, model size and achieved mIoU

This thesis is structured as follows: First, an overview of related work for multi-modal and multi-temporal foundation models is given in Chapter 2. The necessary background of the DOFA[41] model, multi-modal fusion and temporal feature extraction applied in this work is introduced in Chapter 3. The Harz dataset as area of interest and the methodology of this work is proposed in Chapter 4. Implementation, experimental protocol for the ablations and both quantitative and qualitative evaluations are given in Chapter 5. The limitations and possible areas for future work are discussed in Chapter 6. This work and the findings are summarized and concluded in Chapter 7.

2. Related Works

In this Chapter, related works are introduced relevant to multi-modal and multi-temporal foundation models in LULC. First similar datasets to LULC with a focus on tree classification are introduced. Foundation models for remote sensing and recent progress in multi-temporal and multi-modal extensions of foundation models are discussed.

LULC Datasets Voelsen et al.[36] investigate the whole area of Lower Saxony based on multi-temporal Sentinel-2A data containing 4 spectral bands. However, they don't distinguish in tree classes but focus on land cover in general.

Sani et al. [29] trained instance segmentation for dead trees in the Bavarian Forest National Park on optical images with a ground sample distance (GSD) of 0.2m. However, the dataset is not publicly available and is limited to mono-temporal and mono-modal images.

FLAIR is a landcover dataset based in France and comprises over 20 billion manually annotated pixels at 0.2m GSD, very high-resolution images (0.2m) and multi-temporal Sentinel-2 (10m) sequences[10]. It distinguishes trees in coniferous and deciduous trees similar to the Harz dataset in this work but does not contain any dead tree labels. Common works also provide Sentinel-1 imagery to be independent of cloud cover [10].

An overview of the datasets is given in Table 2.1.

Dataset	Modalities		Labels
	images (single date)	time series	
Voelsen et al.[36]	S2 (10m)	-	land cover (10m)
Sani et al.[29]	aerial (0.2m)	-	dead tree detection (bounding boxes)
FLAIR[10]	aerial (0.2m)	S2 (20-114 / year)	land cover (0.2m)
PASTIS-HD[10]	SPOT 6-7 (1.5m)	S1/S2 (30-140 / year)	agriculture (10m)
TreeSatAI-TS[10]	aerial (0.2m)	S1/S2 (10-70 / year)	forestry (60m)
Harz	-	Planet(3-4/year), S1(21-30/year), S2(3-4/year)	land cover (10m)

Table 2.1.: Remote sensing datasets for tree and land cover classification adapted from Garioud et al.[10]. The spatial resolutions of the single-date images and labels, and the temporal resolutions of time series are provided in parentheses. S1/S2 denotes Sentinel-1 and 2, Planet denotes PlanetScope

Mono Temporal and Modal Models Sun et al.[30] constructed a large dataset with a spatial resolution ranging from 0.1 to 30m and proposed a novel masking strategy. The strategy helps reconstruct detailed objects by preserving a few pixels from masked patches which outperforms other (pre-trained) models across multiple tasks. Their model is however pre-trained mostly in the visible light spectrum which limits use cases for hyperspectral or multi-spectral Sentinel-2 data used in this work.

Tang et al. [31] proposed Cross-Scale MAE, which employs a scale augmentation technique in combination with generative and contrastive losses. The framework does not rely on aligned multi-scale data while learning cross-scale features. However, their work is limited to RGB channels which prevents the use of common multi-spectral remote sensing data.

Cong et al. [6] introduced the SatMAE model which is self-trained as a masked autoencoder (MAE)[18] on multi-spectral data given by Sentinel-2. Images are embedded based on channel groups which are split according to their spatial resolution and wavelength similarity. They showed consistent improvements on down-stream tasks for the foundation model weights in comparison to a model trained from scratch, however do not compare against a ViT[8] pre-trained on ImageNet[7]. Additionally, their methodology requires further supervision by dividing the data into groups ahead of training.

Single Temporal Multi-Modal Models Han et al. [16] followed a similar approach to the DOFA[41] spatial encoder applied in this work. They proposed the msGFM model with a MoE-Swin-Transformer[24] as a spatial feature extractor and self-train it based on SIMMIM [40]. They promoted joint representations by cross-decoding patches, e.g. reconstructing patches for different modalities than the input modality. In contrast to the DOFA with a general wave-length encoder module (Chapter 3.3) for arbitrary modalities, they followed a simpler approach and train a patch embedding for each modality. To the date of writing this thesis (24.10.24), the code and foundation model weights are not available despite announcement[1]. For downstream tasks, they implemented multi-modality by concatenating modalities (early fusion as described in Chapter 3.7). However, the Swin-Transformer requires an adaption of the code to process multi-resolution data to allow an independent forward of modalities for other fusion approaches.

Temporal Encoders Rußwurm and Körner[28] applied the transformer architecture to temporal sequences over multiple self-attention layers and utilized global max-pooling in the temporal dimension. They showed that their temporal feature extraction performs better than convolutional models but, on the other side, relies on expensive self-attention across multiple layers.

Garnot et al. [14] proposed the temporal attention encoder (TAE) to apply efficient attention by calculating attention scores and using the input vector as a value (Chapter 3.5). They showed, that their approach is more efficient in computational resources while achieving better quantitative results than previous state-of-the-art (SOTA) of recurrent neural networks [14]. The TAE was further improved with the Lightweight Temporal Attention Encoder (L-TAE) which is used in this work and was proposed by Garnot et al. [11]. Computations and parameters are reduced compared to TAE by making the query trainable instead of calculating the query from the input with an additional MLP. A shortcoming

[1] https://github.com/boranhan/Geospatial_Foundation_Models

of L-TAE is that it is tailored to multi-temporal data and requires an extension to fuse multi-modal data.

Single Modal Multi-Temporal Models Garnot et al. [12] introduced L-TAE to a lightweight UNet[27] to process long temporal sequences. In particular, they calculate L-TAE attention masks on the features of the last block and apply them to the features. The attention masks are then upscaled to the spatial resolution of the other blocks and multiplied with the features.

Dumeur et al. [9] trained a lightweight multi-temporal foundation model by masking feature maps in the temporal dimension and trying to reconstruct them. They showed across multiple datasets that the proposed U-BARN architecture outperforms the U-TAE, however, the model benefits from pre-training for only one of the two evaluated datasets. The spatiotemporal architecture has more costs, which become costly with larger spatial decoders. However, these costs are mostly insignificant in the case of the convolutional UNet with a million parameters.

The Prithvi[19] foundation model proposed by Jakubik et al. extends the Masked Autoencoder (MAE)[18] approach to the temporal dimension. They proposed a lightweight and efficient decoder for segmentation (Chapter 3.4), however, the spatio-temporal encoder requires many computational resources due to a large spatial image resolution of 224×224 and applying self-attention across all temporal tokens. This makes the model inefficient and unsuitable for high-resolution spatial and temporal data.

Bastani et al. introduced SatLasNet[2], a multi-temporal dataset, and trained a foundation model based on a Swin-Transformer. As the temporal sequence is short with 8-12 temporal acquisitions, they apply temporal max-pooling. While the model achieves great performance in downstream tasks compared to the ImageNet[7] initialized model, studies and temporal encoding for longer time sequences are not conducted. Their dataset compromises high-resolution aerial data and multi-spectral data, however they trained models only on a single modality.

Multi-Modal Multi-Temporal Models Images can be also segmented by pixel-wise classification, in which modalities and time sequences can be easily fused by concatenating all available pixel data. For multi-modal and multi-temporal classification, random forest[4] can be used, which combines multiple decision trees. Each node in a tree marks a decision based on a selected feature and the leaf node marks the predicted class. The final prediction for classification is aggregated by majority voting of all trees, e.g. the class with the most votes is chosen. Trained as a foundation model, Presto was proposed by Tseng et al. [33] and applies a transformer encoder and decoder. It compromises less than 1 million parameters and thus can be trained and inferenced on CPUs and GPUs efficiently. A major disadvantage of pixel-based classification is, that they can't include the local bias of image data, as pixels in images are locally correlated[21].

Garnot et al. [13] enhanced the U-TAE[12] to a multi-modal model and researched different fusion strategies and techniques to regularize the model. They proposed a temporal dropout and auxiliary supervision to each modality for regularization. Sentinel-1 data is split into an ascending and descending orbit, such that they investigate with Sentinel-2 data in a total of 3 modalities showing the benefits of merging multiple modalities. On the other hand, recent works often mix the three common modalities of high-resolution data, multi-spectral data and data from active sensors such as SAR [15, 1].

For their proposed FLAIR dataset, Garioud et al. [10] proposed a U-T&T model, which includes a U-Net[27] spatial encoder based on a ResNet-34[17] backbone and decoder for very high-resolution (VHR) data and a U-TAE model for multi-temporal Sentinel-2 data. However, the model benefits from multi-modal data, increasing only from 54.7% mIoU with VHR data to 56.9% mIoU with additional multi-temporal Sentinel-2 data.

SkySense proposed by Guo et al. [15] achieves SOTA on most remote sensing tasks based on their multi-modal and multi-temporal approach. The model is trained as a foundation model through multiple contrastive losses on a dataset of 21.5 million training samples. A Swin-Transformer[24] encodes high-resolution imagery, while separate ViT's[8] are trained for Synthetic Aperture Radar (SAR) and multi-spectral optical data. A simple fusion is employed by adding a trainable positional time encoding and a concatenation of spatial features of all modalities along the temporal dimension for the transformer. Guo et al. [15] added geo-context to encoded features. Therefore, the globe is split into regions and prototypes are learned for each region. The standard configuration of the model is trained on over 80 A100-80GB GPUs and the weights and code are not openly available to this date, preventing the utilization of the foundation model. In its standard configuration, the SkySense model has 2.06 billion parameters, of which almost 1 billion are in the spatial encoder. This makes it infeasible to train and infer the model on a consumer GPU in a multi-modal and multi-temporal scenario. This argument can also be applied to the multi-modal and multi-temporal fusion by the transformer, which is expensive in computation and memory in contrast to other temporal encoders such as L-TAE[12].

OmniSAT[1] was released during the development of this thesis as an efficient multi-modal and multi-temporal model. The spatial encoder is lightweight and efficient. It contains 3.6 million parameters, significantly fewer than the DOFA based on a ViT-B, which has 111 million parameters in total. They apply L-TAE for temporal time sequences and utilize a cross-attention fusion module for multi-modal data. The whole model is trained with a contrastive loss to learn joint multi-modal feature representations and reconstruction loss to learn visual features. Cross-attention further helps to fuse modalities by extracting cross-modal features in comparison to previous work that fuses modalities by concatenation of features[13]. In their experimental comparison, Astruc et al.[1] distort the comparison of their model by not applying the same temporal and multi-modal architecture to other models like the DOFA[41] model. This would be interesting to measure the influence and capabilities of the spatial encoder, as recent attention-based encoders like a ViT[8] or Swin-Transformer[24] should outperform simple convolutional encoders,

especially on high-resolution data, in which some image dependencies require a larger receptive field[8].

Previous multi-modal and multi-temporal work is computationally either too expensive due to separate large spatial encoders for each modality as proposed in the SOTA SkySense model [15] or apply light-weight convolutional spatial encoders[1] restricted to the receptive field[8]. This work bypasses the issue of large frameworks by utilizing a shared spatial encoder across all modalities based on the multi-modal DOFA[41] model while still maintaining a powerful spatial encoder based on the ViT[8]. Further, the model size is reduced with a light-weight spatial decoder proposed by Jakubik et al. [19]. To scale the model up to higher temporal resolutions and reduce the required computational costs similar to the OmniSat[1] model, the proposed framework can be scaled down to smaller spatial encoder configurations.

3. Background

This Chapter introduces the basics of this work. Active and passive sensors are covered in Section 3.1. The basics for image recognition through neural networks are given in Section 3.2. The Dynamic One For All (DOFA) model is explained as a multi-modal spatial encoder in Section 3.3 and spatial encoders used in this work are introduced in Section 3.4. Methods for temporal feature encoding are presented in Section 3.5 and spatio-temporal architectures are discussed in Section 3.6. Fusion architectures are introduced in Section 3.7 and methods to regularize multi-modal and multi-temporal models in Section 3.8 respectively.

3.1. Active and Passive Sensors

Parameter	**Sentinel-2**	**PlanetScope**	**Sentinel-1**
Spatial Resolution	10m (B02, B03, B04, B08) 20m (B05, B06, B07, B8A, B11, B12) 60m (B01, B09)	3m	100m
Spectral Bands	13 bands: B01: Coastal aerosol (442.7nm) **B02**: Blue (492.4nm) **B03**: Green (559.8nm) **B04**: Red (664.6nm) **B05-B07**: Vegetation Red Edge **B08**: NIR (832.8nm) **B8A**: Narrow NIR (864.7nm) B09: Water vapour (945.1nm) **B11-B12**: SWIR (1613.7nm, 2202.4nm)	4 bands: **B1**: Blue (465-515nm) **B2**: Green (547-585nm) **B3**: Red (650-680nm) **B4**: NIR (845-885nm)	C-band SAR: - **VV** polarization - **VH** polarization
Data Type	Optical	Optical	Radar

Table 3.1.: Comparison of Sentinel-1, Sentinel-2 (Bold indicates that the band is used in this work), and PlanetScope Satellite Data Characteristics

Active sensors transmit energy and measure the return signal after reflection from the Earth's surface[23]. Hence, they have the advantage of operating both day and night regardless of the weather conditions since they provide their own source of illumination. Radar systems like Synthetic Aperture Radar (SAR) can be configured either as monostatic, using a single antenna for both transmission and reception or bistatic, employing separate transmitting and receiving antennas. Passive sensors detect reflected sunlight and primarily operate within the optical spectrum. Unlike active sensors, optical passive sensors are generally limited to daytime operation and are more susceptible to atmospheric con-

ditions and cloud cover. All sensors and their characteristics in the wavelength spectrum are stated in Table 3.1.

3.2. Image Recognition in Remote Sensing

Deep Learning is a subset of Machine Learning that aims to train a predictor for a task based on data. Predictors for complex tasks such as classification in Computer Vision can thus be obtained by end-to-end learning without explicit programming of feature extractors[17].

Given the Labels y with C classes and features x, the learning objective is to train a classifier h_θ, which maps from feature space to a label with a minimal loss L. The empirical risk approximates the true risk with a large enough sample size. Hence, the best parameters $\theta*$ are obtained by minimizing the empirical risk over all available samples, as given in Equation 3.1.

$$\theta^* = \min_\theta \frac{1}{m} \sum_{i=1}^{m} L(h_\theta(x^{(i)}), y^{(i)}) \tag{3.1}$$

In a classification problem, a Cross-Entropy loss (Eq. 3.2) is commonly applied to align the predicted distribution and ground-truth label distribution[37].

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{3.2}$$

As for the optimization problem in Equation 3.1, a high-dimensional non-linear deep neural network can be used as a hypothesis class, this problem has to be solved with iterative optimization techniques like gradient descent (Eq. 3.3).

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta L(\theta_t) \tag{3.3}$$

This process is called backpropagation for deep neural networks, as the gradients are calculated from the back to the front of the model [22]. The model parameters are optimized until the convergence of the loss and due to the non-convex nature of this optimization problem, local minima are found.

With the development of bigger GPU memories and more computation power available, Deep Convolutional Neural Networks revolutionized the Computer Vision domain achieving state-of-the-art performance in image recognition tasks[21, 17]. Inspired by breakthroughs in Natural Language Processing (NLP) through the transformer architecture[35], self-attention was later applied in Computer Vision architectures[24, 8]. Self-attention was first introduced to NLP by Vaswani et al.[35]. The self-attention mechanism is applied globally and comes with a computational complexity of $\mathcal{O}(n^2)$. Attention scores between the query (Q) and key (K) are calculated with matrix multiplication and by applying the $softmax$ operation (Eq. 3.4). A denominator $\sqrt{d_k}$ is added for training stability [35]

and the calculated weights are multiplied with the values (V). For H heads, self-attention is applied, and the results are concatenated and multiplied with a weight matrix W^0 (Eq. 3.5). Query, key and value for the attention mechanism are calculated in the transformer encoder from the input through a linear projection with learned weights W_i^Q, W_i^K and W_i^V (Eq. 3.5). One big advantage is, that each head can be computed independently in parallel[35].

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.4}$$

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{3.5}$$

Dosovitskiy et al. [8] successfully applied self-attention to computer vision with their proposed Vision Transformer (ViT), which is visualized in Figure 3.1. They solve the problem of continuous image data by splitting images into non-overlapping 16×16 patches, which are flattened and linearly projected into tokens. A positional encoding is added for spatial information to the projected patch. A transformer layer consists of multi-head attention and an MLP repeating for L layers. The global attention mechanism reduces the inductive bias compared to the limited receptive field of CNNs, making the ViT computationally more costly but theoretically more powerful with enough training data[8].

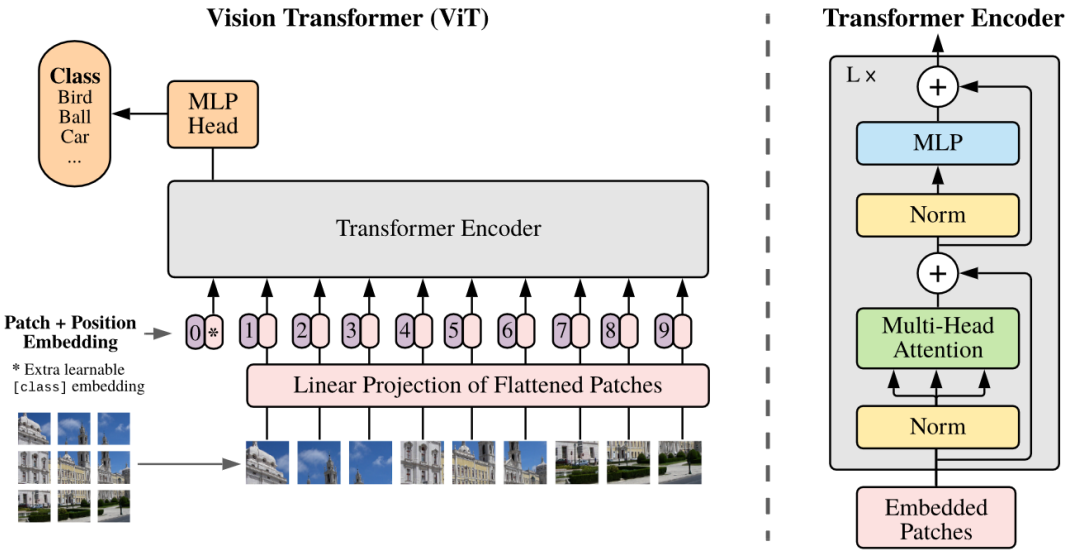


Figure 3.1.: Vision Transformer (ViT) architecture by Dosovitskiy et al. [8]

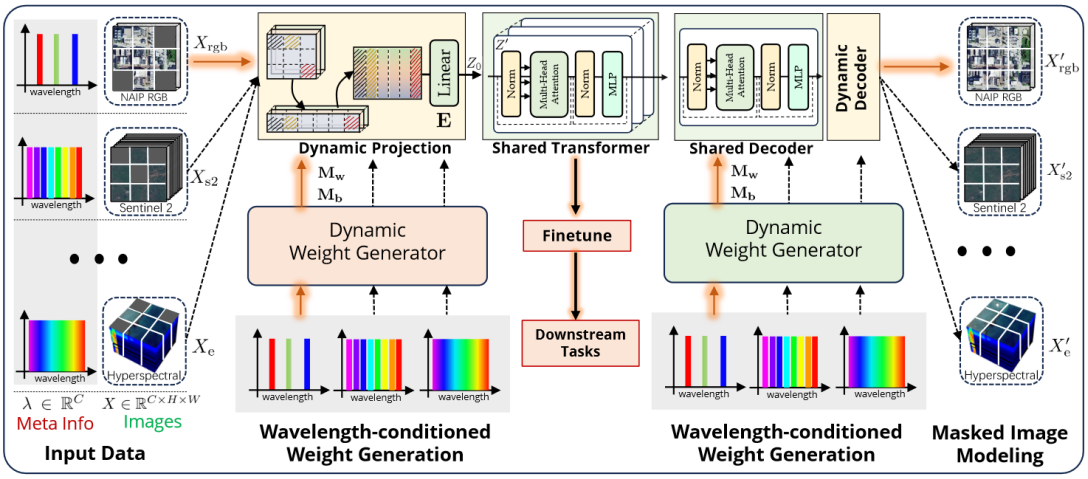


Figure 3.2.: DOFA by Xiong et al. [41]

3.3. Dynamic One For All (DOFA)

Dynamic One For All (DOFA)[41] is introduced with a wave-length encoder for arbitrary multi-spectral data. It utilizes a ViT[8] as a backbone and is trained self-supervised as a masked auto-encoder[18] on multi-modal remote sensing data, as visualized in the overview of the model in Figure 3.2.

The key novelty of the DOFA model is the wave-length encoder allowing to learn patch embeddings for any spectral image $X \in \mathbb{R}^{C\times H\times W}$[41], where C are the channels, H the height and W the width of the image. Given a representation $\lambda \in \mathbb{R}^C$ of the wavelengths, the channels of each centered wavelength are first encoded into a d_λ dimensional feature vector $\mathbf{V}_\lambda \in \mathbb{R}^{C\times D_\lambda}$ (Equation 3.6) with 1D sine-cosine embeddings (Equation 3.7) as positional encodings (PE).

$$\mathbf{V}_\lambda = \mathrm{PE}(\lambda) \in \mathbb{R}^{C\times D_\lambda} \tag{3.6}$$

$$\begin{aligned} \mathrm{PE}(\lambda_i, 2k) &= \sin\left(\frac{\lambda_i}{10000^{2k/D_\lambda}}\right) \\ \mathrm{PE}(\lambda_i, 2k+1) &= \cos\left(\frac{\lambda_i}{10000^{2k/D_\lambda}}\right) \end{aligned} \tag{3.7}$$

Two linear layers with a ReLU activation are applied and added with a residual skip connection to $\mathbf{V}_\lambda$, as given in Equation 3.8 to obtain feature vector $\mathbf{V}'_\lambda$.

$$\mathbf{V}'_\lambda = \mathrm{ReLU}(\mathcal{F}_2(\mathrm{ReLU}(\mathcal{F}_1(\mathbf{V}_\lambda)))) + \mathbf{V}_\lambda, \tag{3.8}$$

Learnable tokens $\mathbf{Q}_w$ and $\mathbf{Q}_b$ for the patch embeddings are concatenated with $\mathbf{V}'_\lambda$, and processed through a Transformer encoder layer with 4 heads (Eq. 3.9).

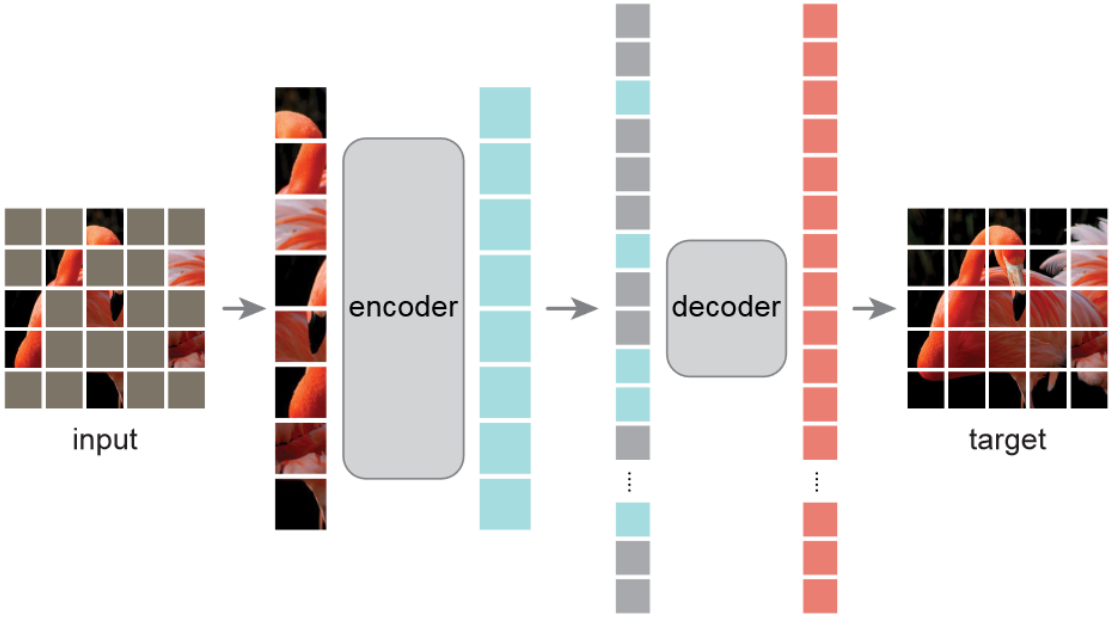


Figure 3.3.: Masked Autoencoder by He et al. [18]

$$\mathbf{V}'' = \text{TransformerEncoder}(\text{Concat}(\mathbf{V}'_\lambda, \mathbf{Q}_w, \mathbf{Q}_b)) \tag{3.9}$$

The learned weights and biases tokens are extracted from $\mathbf{V}''$ and each applied to an MLP. The weight vector is reshaped to a convolutional filter of shape $X \in \mathbb{R}^{D\times C\times P\times P}$, where P is the patch size, C the input channels of the image, and D the embedding dimension. The weight and bias vector are then used to extract non-overlapping patch embeddings from the original image.

The DOFA model is trained with self-supervision as a masked auto-encoder[18]. Therefore, some of the non-overlapping patches of the image are masked out and the encoder receives the remaining patches. The decoder then tries to reconstruct the masked patches with the limited information available, as visualized in Figure 3.3. By applying high masking ratios such as 75%, the model can learn visual features without the availability of any labels [18, 41].

3.4. Spatial Decoders

In this subsection, the UperNet[38] decoder used in the DOFA[41] model is introduced. As light-weight alternatives the spatial decoders proposed in the SegFormer[39] and Prithvi[19] are also presented.

The UperNet[38] extends the UNet [27] with a Pyramid Pooling Module (PPM) head and fuse module. Xiao et al. [38] state, that the PPM helps to overcome the limitations of the empirical receptive field, which is smaller than the theoretical receptive field. The PPM

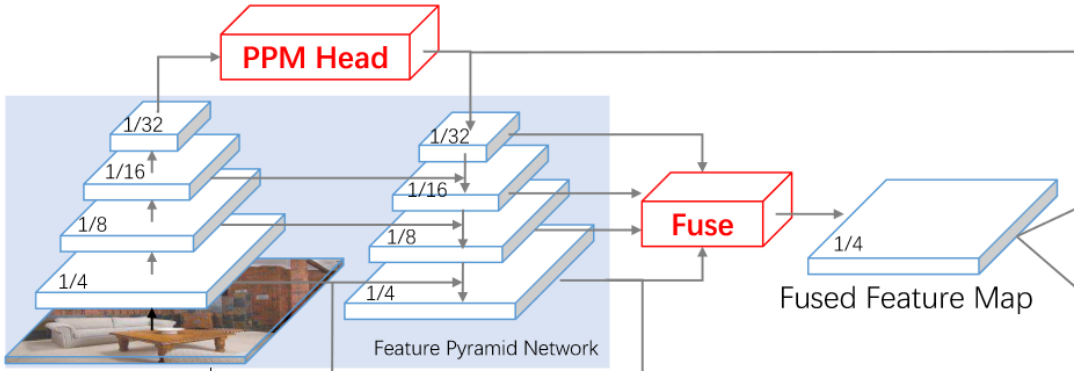


Figure 3.4.: UPerNet architecture by Xiao et al. [38]

is applied to the features of the last backbone layer before giving it as input to the top-down branch in the feature pyramid network, as visualized in Figure 3.4. The fuse module resizes the feature pyramids to the same spatial resolution, concatenates them, and then applies a convolutional layer to reduce feature dimensions.

Global and local features are efficiently merged in the SegFormer decoder MLP with few parameters[39]. Equation 3.10 describes the decoder of SegFormer, in which features $f_{E,i}$ of each encoder block i are reshaped to the same spatial resolution $H/4xW/4$, concatenated, and then applied to an MLP [39].

$$f_{SegFormer}(f_E) = \text{MLP} \circ \text{concat}(\text{reshape}(f_{E,i}))_{i=1}^{4} \tag{3.10}$$

Jakubik et al. [19] propose a simple and lightweight spatial decoder for their ViT-based foundation model. The decoder stacks two blocks of two ConvTranspose2D layers with a block shown in Equation 3.11. Thus, the decoder upsamples patchified 16x16 patches to their original resolution, and the segmentation head classifies on the full resolution in contrast to other segmentation heads[38, 39], that operate on $H/4xW/4$ of the image resolution. The spatial decoder is only applied to the output of the last ViT layer in contrast to the two previously introduced spatial decoders, which process a feature pyramid generated by each encoder block to merge global and local features [27].

$$f_{prithvi,Block}(f_E) = \text{conv\_transposed} \circ \text{norm} \circ \text{activation} \circ \text{conv\_transposed}(f_E) \tag{3.11}$$

3.5. Temporal Encoders

Bastani et al. [2] utilize temporal max-pooling in their SatlasPretrain foundation model based on A Swin Transformer [24]. For a spatial feature $X^l \in \mathbb{R}^{T \times d \times h \times w}$ of a spatial encoder layer l, the maximum value is calculated over T time steps, as given in Equation 3.12.

$$D_{E,Max\_Pool}(X^l_{:,i,j,k}) = \max(\{X^l_{1,i,j,k}, X^l_{2,i,j,k}, ..., X^l_{T,i,j,k}\}) \tag{3.12}$$

Lightweight Temporal Attention Encoder (L-TAE) is introduced to efficiently extract temporal features from satellite image time series (SITS) [11]. For a feature vector e of dimensions $E \times T$, the vectors are split into H heads along the dimension E, such that e_h is the feature vector of head h (Eq. 3.13). A positional embedding $p^{(t)}$ based on a day-of-the-year (doy) encoding in sin waves is calculated to the (Eq.3.14). The key $k_h^{(t)}$ is obtained by adding the positional embedding $p^{(t)}$ to the initial vector $e_h^{(t)}$ and projecting the vector into dimension $K \times T$ (Eq.3.15). Self-attention is calculated on key $k_h^{(t)}$ and a learnable query vector q_h of dimensions $K \times 1$ (Eq.3.16). The attention masks are multiplied with the input vectors as values for each head h to obtain vector o_h (Eq.3.17). All resulting vectors o_h of each head h are concatenated and an MLP is applied to obtain output vector o (Eq.3.18).

$$e_h^{(t)} = [e^{(t)}[(h-1)E' + i]]_{i=1}^{E'} \tag{3.13}$$

$$p^{(t)} = [\sin(\text{day}(t)/\tau^{\frac{i}{E'}})]^{E'} \tag{3.14}$$

$$k_h^{(t)} = FC_h(e_h^{(t)} + p^{(t)}) \tag{3.15}$$

$$a_h = \text{softmax}(\frac{1}{\sqrt{K}}[q_h \cdot k_h^{(t)}]_{t=1}^{T}) \tag{3.16}$$

$$o_h = \sum_{i=1}^{T} a_h[t](e_h^{(t)} + p^{(t)}) \tag{3.17}$$

$$o = \text{MLP}([o_1, \cdots, o_H]) \tag{3.18}$$

An overview of all L-TAE operations is visualized in Figure 3.5.

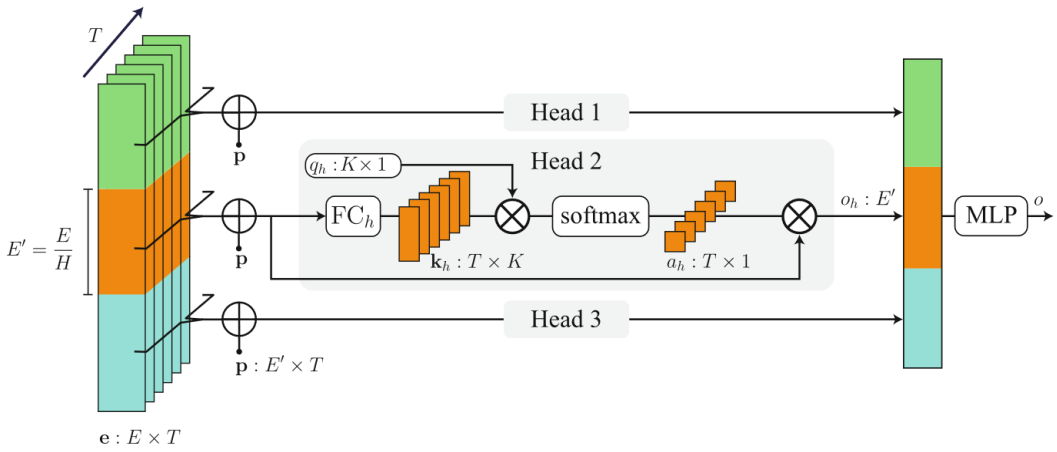


Figure 3.5.: L-TAE by Garnot and Landrieu [11]

$$\left[X(t)\right]_{t=1}^{T} \xrightarrow{E_{spatial}} \left[X_{E,spatial}(t)\right]_{t=1}^{T} \xrightarrow{E_{temporal}} X_{E,temporal} \xrightarrow{D_{spatial}} X_{D,spatial} \xrightarrow{D_{head}} y_{utae}$$

Figure 3.6.: Progression of operations for the U-TAE[12] model

3.6. Spatio-Temporal Architecture

Garnot and Landrieu [12] propose the U-TAE model for segmentation, where they apply the temporal encoder between the spatial encoder and decoder, as stated in Figure 3.6.

In the U-BARN [9] model, all spatial features are extracted (spatial encoder and decoder) first and the temporal encoder is applied afterward, as given in Figure 3.7. For a sequence X of modality m, this requires $\sum_{m=1}^{M} |X_m|$ spatial decoder forward passes in total instead of M in the case of the U-TAE architecture.

$$\left[X(t)\right]_{t=1}^{T} \xrightarrow{E_{spatial}} \left[X_{E,spatial}(t)\right]_{t=1}^{T} \xrightarrow{D_{spatial}} \left[X_{D,spatial}(t)\right]_{t=1}^{T} \xrightarrow{E_{temporal}} X_{E,temporal} \xrightarrow{D_{head}} y_{ubarn}$$

Figure 3.7.: Progression of operations for the U-BARN[9] model

Tarasiou et al.[32] propose a tempo-spatial architecture (TS-Vit), which first applies a temporal transformer encoder and extracts spatial relations afterward. This architecture is not regarded in this work, as Dumeur et al.[9] show the disadvantages of their model in computational constraints and dependency on the batch size. Further, the architecture might prevent the use of pre-trained spatial foundation models, which extract features from continuous optical data.

3.7. Multi-Modal Fusion

Modalities can be fused before the spatial decoder (early fusion), after the spatial encoder and before the temporal encoder (mid fusion), or after the temporal encoder (late fusion).

Garnot et al. [13] propose for early fusion to interpolate missing samples to the longest time sequence available. Let $T \times C_m \times H \times W$ be the images for M modalities before interpolation, where T is the longest time sequence and C_m the channels for modality m. Then, the samples are concatenated in the channel dimension to $Tx\sum_{m=1}^{M} C_m \times H \times W$ and further processed through the spatio-temporal model requiring only one branch for all modalities.

In the case of mid fusion, Garnot et al. [13] process the samples through separate spatial encoders, as visualized in Figure 3.8. A feature vector $T_m \times D \times h \times w$ for every modality m and embedding dimension D of the spatial encoders is obtained and features are concatenated in the temporal dimension resulting in a feature vector $\sum_{m=1}^{M} T_m \times D \times h \times w$.

For late fusion, modalities are first processed independently through a spatial and temporal encoder resulting in feature vector $Dxhxw$ for embedding dimension D. Modalities

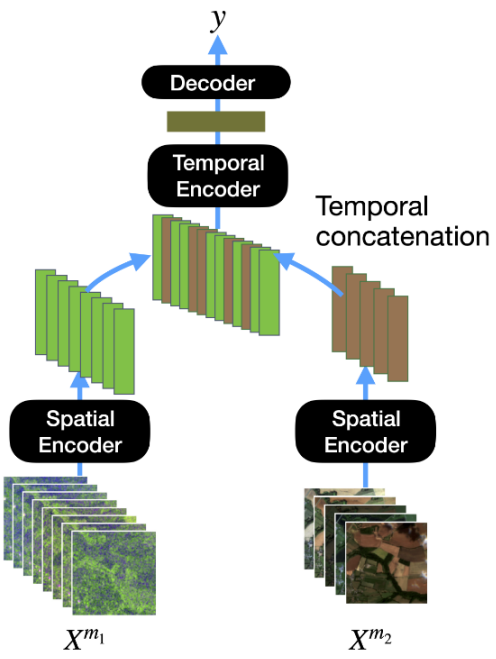


Figure 3.8.: Mid Fusion visualized by Garnot et al. [13]

are fused by summing the channel dimension's features or concatenating all feature vectors, yielding a feature vector $M \cdot D \times h \times w$ before the classification head.

3.8. Multi-Modal and Multi-Temporal Regularization

Temporal dropout is a multi-temporal regularization proposed by Garnot et al. [13] to prevent the model from relying on certain acquisitions. For each modality m, a sample in the sequence is dropped during training with probability $p^m \in [0, 1]$.

Garnot et al. [13] state, that few modalities dominate multi-modal feature fusion. The model learns from the best modalities, which prevents the model from learning inter-modal patterns [13]. Hence, they introduce an auxiliary loss $\mathcal{L}_{aux}$ for additional supervision of each modality $m \in [1, M]$ with weight λ^m, as stated in Equation 3.19.

$$\mathcal{L}_{aux} = \sum_{1}^{m} \lambda^m criterion(y^m, \widehat{y}), \tag{3.19}$$

In the case of late fusion, auxiliary supervision requires additional segmentation heads for each modality, while additional temporal encoders are necessary in the case of mid-fusion.

4. Material and Methods

The Harz dataset is described in Chapter 4.1 and data limitations are discussed. In Chapter 4.2, the general methods of this work are introduced and the forward flow of the proposed model is presented.

4.1. Dataset

The collected LULC Harz dataset contains data from multiple sensors with time sequences. It covers the Harz region in Lower Saxony and partially Saxony-Anhalt, as visualized in Figure 4.1. Images of growing seasons from 2020 and 2021 are available from both passive sensors (Sentinel-2 and PlanetScope) and active sensors (Sentinel-1), as stated in Table 4.1. Sentinel-2 covers a wide light spectrum with 10 bands, PlanetScope captures high-resolution imagery and Sentinel-1 a high temporal resolution.

Sensor	Modality	GSD (m)	# Images/growing season	Selected Features
PlanetScope	Optical	3	4	All (RGB, NIR)
Sentinel-2	Optical	10	3 (2020), 4 (2021)	B2-B8a, B11, B12
Sentinel-1	Radar	10	21 (2020), 30 (2021)	All (polarizations VV, VH)

Table 4.1.: Available data of the multi-modal and multi-temporal Harz dataset including PlanetScope, Sentinel-2, and Sentinel-1 data

Figure 4.2 visualizes a Sentinel-2 sample covered by many clouds, as data in remote sensing is not clean in many cases or real-world applications [1]. The Harz dataset contains 3 acquisitions in 2020 and 4 in 2021, providing at least one cloud-free time-stamp for each region. In contrast to Sentinel-2 data, PlanetScope data is both cloudy and contains invalid stripes, as visualized in Figure 4.3. Invalid stripes are mostly found in 2020 and each patch is guaranteed to be covered by at least one acquisition. For 2021 on the other hand, more images contain clouds. The largest time sequence is available for Sentinel-1, yielding 21 images for 2020 and 30 for 2021. Missing data as visualized in Figure 4.4 occur in approximately half the available samples of each year.

LULC labels were generated with random forests by the ESA from 2020 with 75% overall accuracy [1] and 2021 with 77% overall accuracy [2], making it more challenging in the evaluation and training of the model due to label uncertainty. The tree class is split into *dead*, *coniferous*, and *deciduous* trees. ESA *tree* labels, that are not classified as one of the previous three categories, are preserved and treated as background class for training. Dead trees

[1] https://worldcover2020.esa.int/
[2] https://worldcover2021.esa.int/

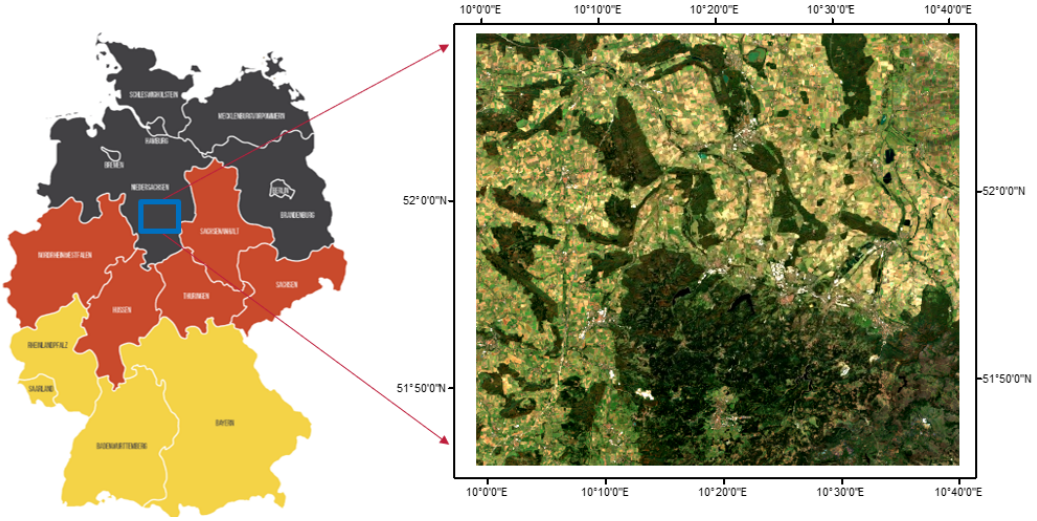
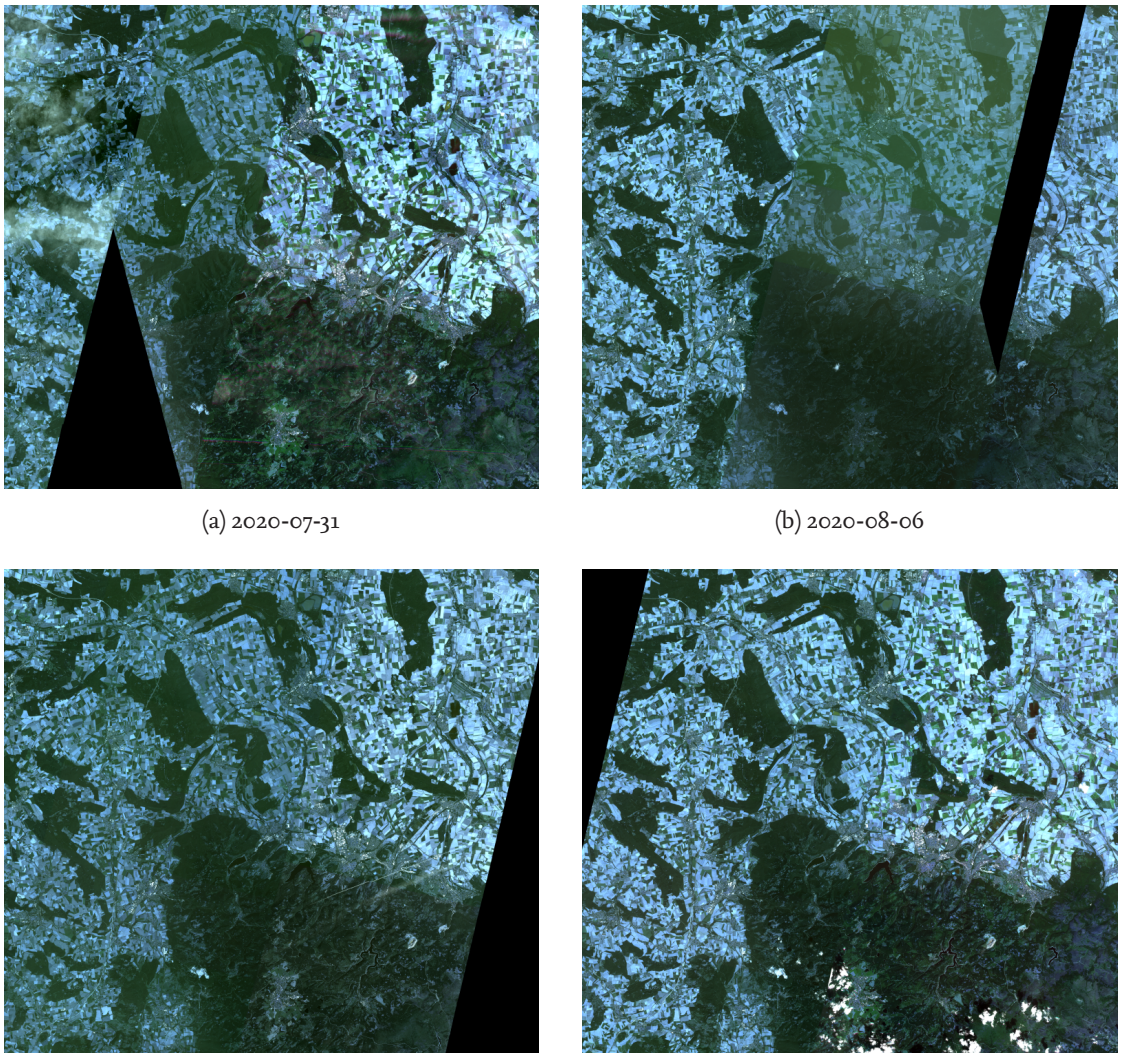


Figure 4.1.: The covered Harz region

Figure 4.2.: Cloudy Sentinel-2 sample.



(a) 2020-07-31 (b) 2020-08-06

(c) 2020-07-30 (d) 2020-08-07

Figure 4.3.: PlanetScope data from 2020 with invalid stripes

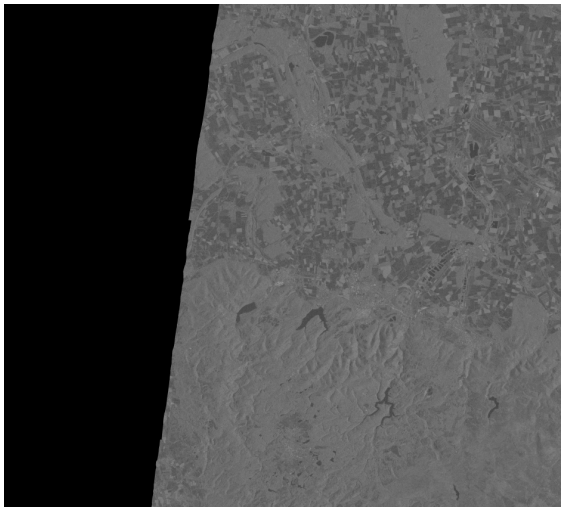
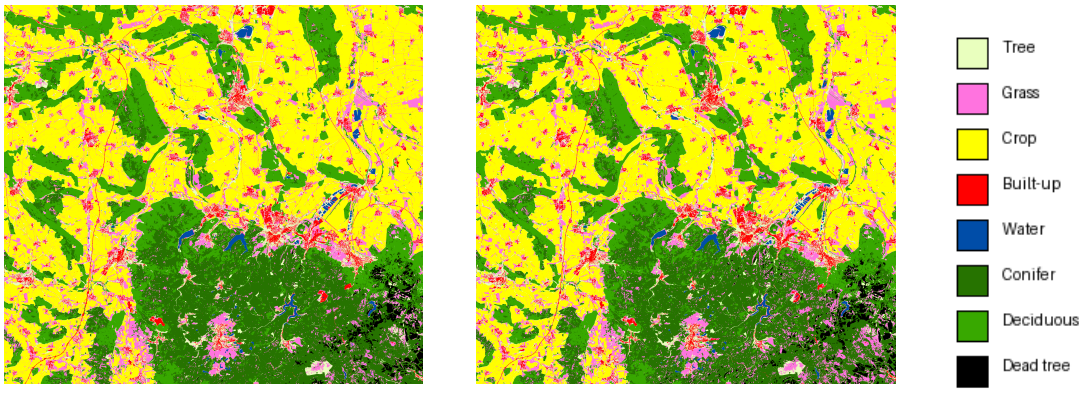


Figure 4.4.: Visualization of VH from Sentinel-1 with missing data.



(a) 2020 (b) 2021

Figure 4.5.: Labels for the years 2020 and 2021.

were manually annotated and tree species labels are based on Blickensdörfer et al.[3] work, where they generated tree species labels in Germany for 2018 using a random forest classifier with 75% accuracy. All generated labels were acquired on the same spatial resolution as Sentinel-2 and Sentinel-1. In total, the following classes will be segmented for LULC: *cropland, grassland, built-up, water, dead tree, coniferous tree, deciduous trees.* Figure 4.5 for 2020 and 2021 shows the complete label maps.

The label distribution is visualized in Figure 4.6. As the Harz region and countryside are handled, the classes are imbalanced towards vegetation classes, dominated with almost 40% of data from *crop*. The tree classes *deciduous* and *conifer* follow with almost 20% each, making up most of the trees in the Harz. A strong imbalance exists towards the class *built-up*, *water*, and *dead tree*, making up less than 5% each.

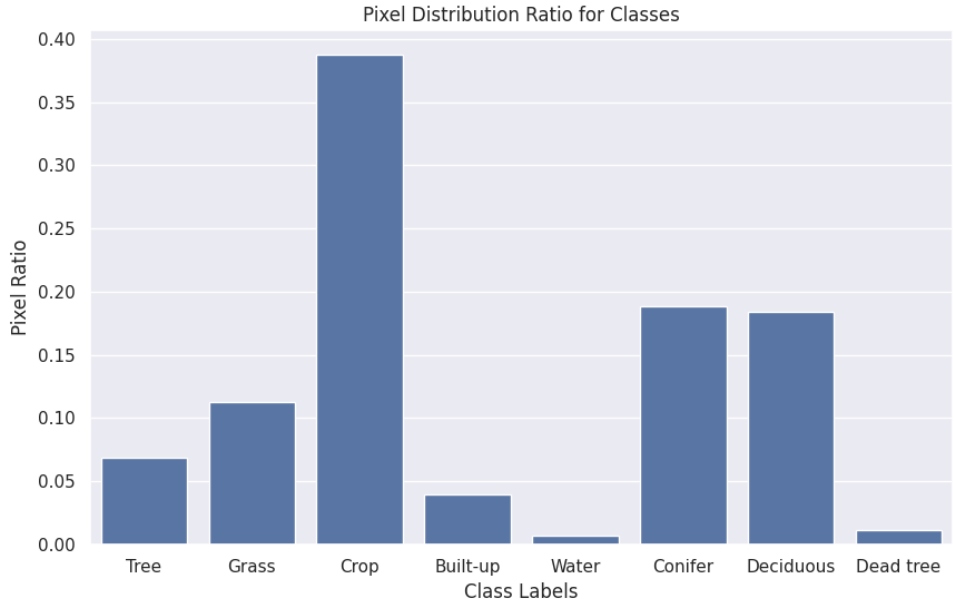


Figure 4.6.: The label distribution of the Harz dataset.

4.2. Methodology

The methodology is presented in Figure 4.7. First, data is preprocessed by patchifying the large .tif image, splitting the data into train-test-validation splits, and calculating the mean and standard deviation on the train-validation splits. During training, valid patches are filtered out, temporal dropout is applied, and then the data is stacked in the temporal dimension for each modality. Augmentations are applied consistently on labels and data from all modalities, and additionally, temporal information for each data sample is loaded. The model is then assessed by training it on the data, and the best model checkpoint with regards to the validation mIoU is stored. The best model checkpoint is then loaded and evaluated quantitatively on the test set based on mIoU and F1-Score, and some samples are visualized for qualitative evaluation.

Figure 4.8 visualizes the forward pass of the framework. First, a subset of the images from all modalities is resized to the same spatial dimension, concatenated in the channel dimension, and then forwarded through the spatial encoder. For other fusion methods, the images are forwarded through the spatial encoder with shared weights across all modalities. Then, spatial features are concatenated along the temporal dimension for mid-fusion. Depending on the spatiotemporal architecture, features are first forwarded through the temporal and then through the spatial decoder. In the case of late-fusion, this happens independently for each modality, such that features are concatenated or summed up afterward, depending on the late-fusion method. The segmentation head predicts, at last, a probability distribution for all classes.

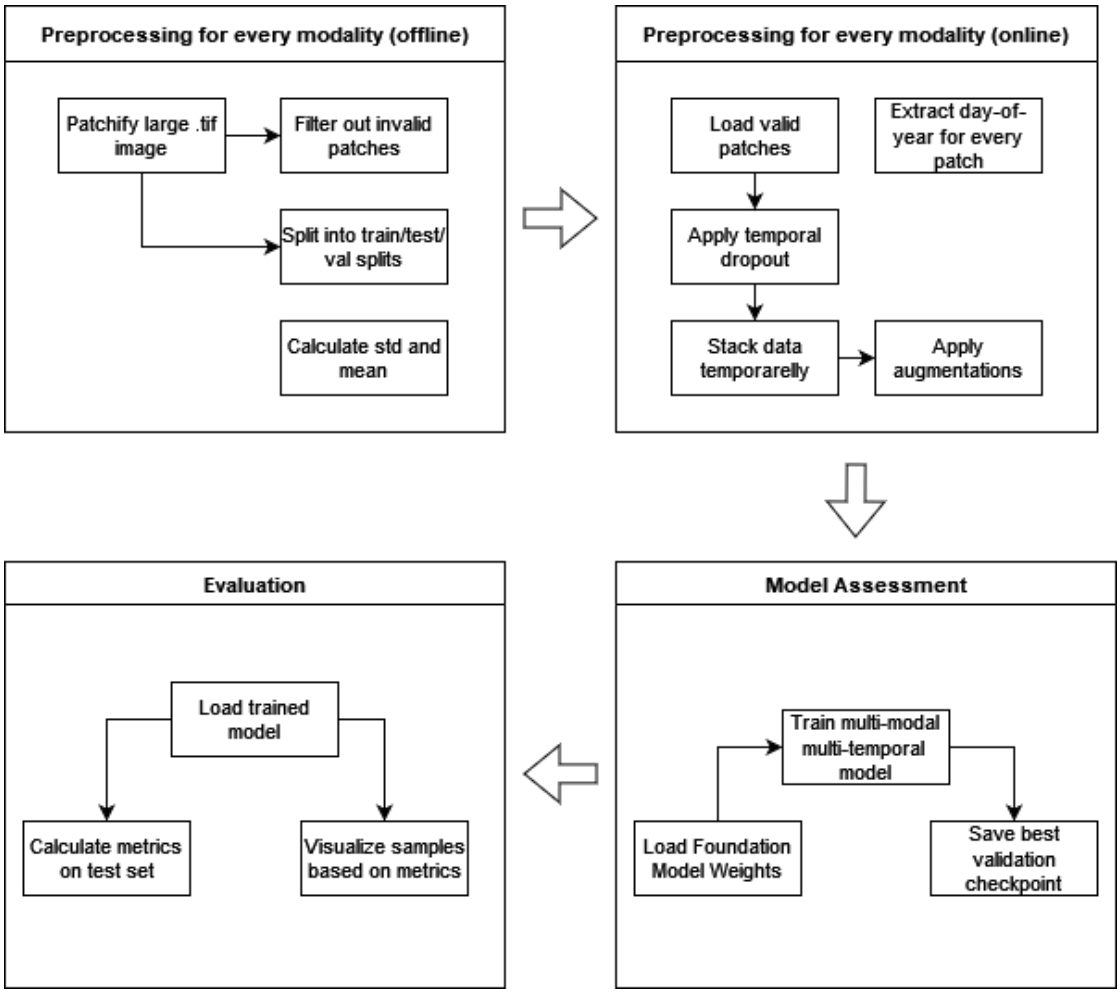


Figure 4.7.: Methodology of this work

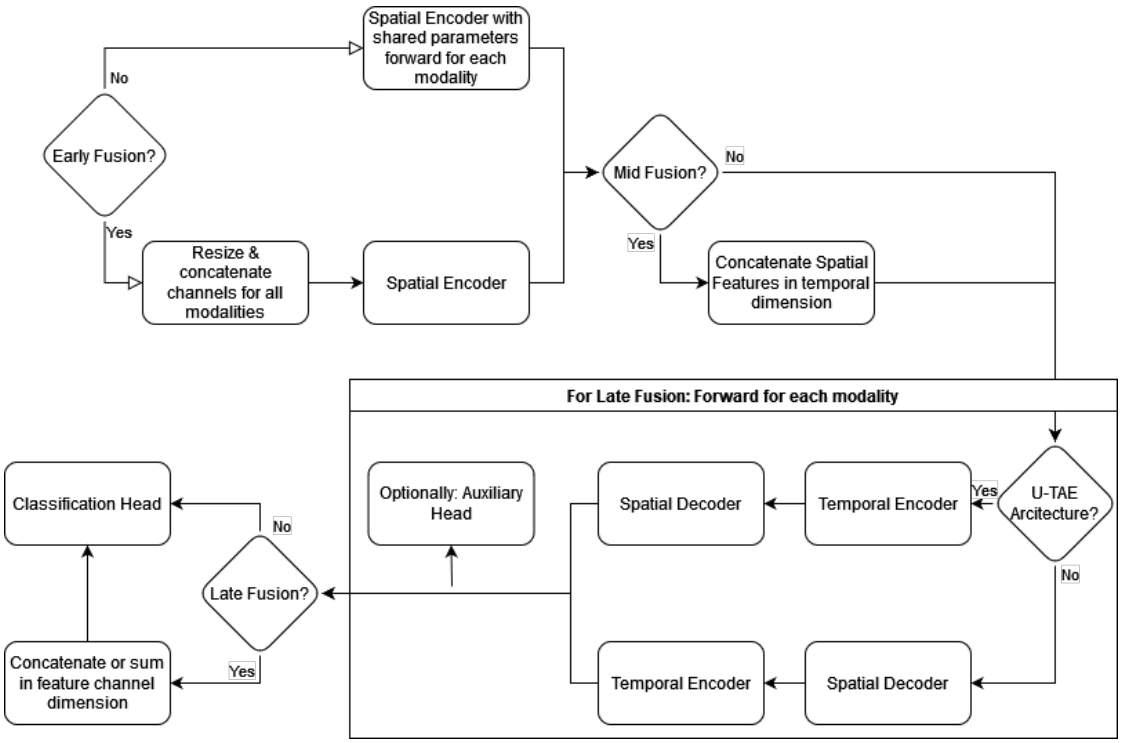


Figure 4.8.: Forward flow of the model based on the fusion method and spatio-temporal architecture

5. Experiments

This chapter first describes the implementation of the methodology in Chapter 5.1. The experimental protocol is given in Chapter 5.2 and the results are evaluated both quantitatively in Chapter 5.3.1 and qualitatively in Chapter 5.3.2.

5.1. Implementation

All experiments are conducted on a single node of the Phoenix-Cluster[1] by the TU Braunschweig. Each node contains 4 NVIDIA Tesla P100 16GB HBM2, 64GB RAM, and 2 CPU INTEL Xeon E5-2640v4. An overview of all used frameworks and the code of the components can be found in Table 5.1. The GeoSeg framework based on PyTorch Lightning was adapted and a modular framework based on configuration files for spatio-temporal encoding was implemented.

Model/Component	**GitHub Repository**
DOFA[41] model, weights, hyperparameters	https://github.com/zhu-xlab/DOFA
UPerNet[38] Spatial Decoder	https://github.com/yassouali/pytorch-segmentation/blob/master/models/upernet.py
Prithvi[19] Spatial Decoder	https://github.com/NASA-IMPACT/hls-foundation-os
FCN Head, FCN Auxiliary Head, SegFormer[39]	https://github.com/open-mmlab/mmsegmentation
L-TAE[11] and U-TAE Architecture [12]	https://src.koda.cnrs.fr/iris.dumeur/ssl_ubarn
fvcore: FLOPs calculation	https://github.com/facebookresearch/fvcore
Framework	
GeoSeg	https://github.com/WangLibo1995/GeoSeg
PyTorch Lightning	https://lightning.ai/docs/pytorch/stable/

Table 5.1.: GitHub repositories and links for various models, frameworks, and components used in the thesis

The initial model yields the same configuration as the spatial encoder and decoder as the DOFA[41], e.g. an enhanced ViT-B[8] encoder and UPerNet decoder[38]. The spatio-temporal architecture implemented by the U-TAE[12] is used as default in combination with temporal max-pooling. Further, late fusion based on sum is used as default fusion method.

The PyTorch Lightning framework does not implement linear scheduling, which is why a CosineAnnealing scheduler is used instead. All models are trained for 100 epochs and the model checkpoint with the best validation mIoU is saved. For the AdamW[25] optimizer, weight decay is set to 0.01, and the learning rate to $6e-5$ and $6e-6$ for the DOFA backbone (wavelength encoder and ViT[8]). A cross-entropy loss is used with *label_smoothing* set to 0.1.

[1] https://www.tu-braunschweig.de/it/hpc

The Sentinel-2 and Sentinel-1 data are split into 128x128 patches. For PlanetScope, the tile is resized to 3 times the resolution of Sentinel data and divided into 384x384 patches. As the same spatial area is covered in two different years, train-validation-test indices are randomly split in the ratios 60-20-20 for one year. The equivalent spatial indices of the other year are distributed in the same splits, so the train-validation-test splits are spatially consistent over time. Invalid patches were computed offline and are defined as patches, that do not contain any data at all. This means, that in an image of the time sequence, all values are *nan* or that the minimum value of the patch equals the maximum value. These invalid patches are dropped during data loading.

The batch size is set to 1 to avoid masking varying lengths of time sequences and running into memory issues. Some patches are missing data completely and are dropped during training. Experiments are conducted on a node with 4 GPUs in distributed data parallel (DDP) mode, such that each GPU gets one batch sample in parallel. Hence, the degeneration of *BatchNorm2d* with batch size 1 is avoided by replacing *BatchNorm2d* with *SyncBatchNorm*.

Flip and crop augmentations were implemented from scratch as they have to be applied to all modalities in the same manner independent of their image resolution. For the crop augmentations, a patch of the image is cropped out at a random location and then resized to the original resolution. The normalization values are computed on the train and validation split of the dataset for every modality.

FLOPs are calculated in this work with the *fvcore* library (Table 5.1). The FLOPs are calculated for all configurations with a sample from the year 2021 with the most data available, e.g. time sequences of length 30 for Sentinel-1, and 4 for Sentinel-2 and PlanetScope.

5.2. Experimental Protocol

Training Hyperparameters First ablations are conducted to regularize the model and fine-tune training hyperparameters. In the first ablation, the backbone learning rate is set to the same learning rate as the whole model's learning rate in comparison to the diminished backbone learning rate proposed in DOFA[41].

Training Regularization Augmentations are commonly used during the training of models in computer vision for regularization to reduce overfitting [17, 21]. Hence, ablations on the effect of augmentations are conducted, in which the model without augmentations only applies normalization. For the crop augmentations, the crop size is set to 0.75. All augmentations are used with a probability of $p = 0.5$ each. Afterward, ablations are done to measure the influence of temporal dropout[13]. The values for temporal dropout are adapted from Garnot et al.[13] and set to $p_{optical} = 0.4$ and $p_{SAR} = 0.2$. An advantage of this regularization is the reduced computational requirements for training, as missing samples do not have to be padded due to a batch size of 1. However, it is guaranteed, that at least one patch exists for every modality due to training errors otherwise. To regularize

the modalities, ablations are conducted regarding auxiliary supervision for each modality[13]. The proposed weights by Garnot et al.[13] are adapted, e.g. auxiliary loss weights $\lambda^m = 0.5$ for every modality m. In the first experiment, the auxiliary losses are dominated by Sentinel-1 (Figure A.2a), such that further experiments with smaller λ^{S1} are conducted.

Spatial Decoder Different spatial decoders are further considered, as in the tempo-modal extension of the model, memory and compute efficiency of a large spatial model matter. The original DOFA model utilizes a neck and UperNet, while the compared SegFormer[39] or Prithvi decoder [19] are lightweight. For the Prithvi decoder, both variants with only one and two upsampling blocks are taken into regard for the ablations. For the double block used with PlanetScope images, the spatial resolution of the features has to be downsampled again from 384×384 to the label resolution of 128×128. Since this might degrade the performance, another configuration *Prithvi Partial Double* is added. In this configuration, a double decoding block is utilized for Sentinel-2 and Sentinel-1, and only a single block for PlanetScope to avoid an overflow in spatial resolution.

Spatio-Temporal Architecture Due to memory and computational complexity, the U-TAE[12] architecture (3.6) for spatio-temporal encoding is set as a baseline. For completeness, it is compared against the U-BARN[9] architecture (3.7).

Fusion For modal fusion, sum (baseline according to Kang et al. [20]) and concatenation fusion are employed for late fusion. Additionally, mid and early fusion are investigated due to the low computational requirements. For early fusion, Garnot et al. [13] interpolate missing samples for each modality to obtain the same amount of samples for each modality. In this work, images are dropped due to the imbalanced sequence lengths for each modality (Table 4.1). There exist more Sentinel-1 images (Table 4.1) with little information for the segmentation task (Paragraph *Modalities* in Chapter 5.3.1), while just a few samples for Sentinel-2 and PlanetScope exist. Additionally, the high resolution for PlanetScope is the most computationally expensive part of the inference pipeline (Table 5.8), yielding too large memory requirements for the interpolation of PlanetScope images.

The early fusion is implemented by concatenating the temporary closest sample for each modality to each sample of the modality with the smallest sequence length. Therefore, the spatial resolution of each modality is resized to the highest spatial resolution e.g. PlanetScope with 384x384 pixels.

Modalities All subsets of modalities are compared to measure the effectiveness of each modality and the underlying approach to extract tempo-modal features concerning compute efficiency.

Temporal Feature Extraction As temporal data is distributed with varying availability for each modality, the effectiveness of temporal feature extraction is studied in ablations.

Therefore, on both each modality and all modalities together, experiments are conducted with max-pooling, L-TAE as feature extractor, and in a single-temporal configuration. For the single temporal configuration, the data stays the same but the model does not extract any temporal features, e.g. the label is duplicated T times for a time series of length T. During evaluation, the approach by Brown et al.[5] to max-pool the class probabilities over time, e.g. utilize the most confident prediction.

Parameter Scaling The ViT-B is replaced with a ViT-L, which increases the parameters of the feature extractor in DOFA from 86 million to 307 million parameters. This increases the computation and memory requirements significantly, so it is only applied to the best-performing modality PlanetScope and modality combination PlanetScope and Sentinel-2.

Overview Table 5.2 gives an overview of all ablations.

Category	Ablations
Training Hyperparameters	1. Spatial encoder learning rate is equal to the model's learning rate 2. Lower spatial encoder learning rate based on DOFA[41](baseline)
Augmentations	1. No augmentations (only normalization, baseline) 2. Augmentations
Temporal Dropout [13]	1. No temporal dropout (baseline) 2. With Temporal Dropout
Auxiliary Head [13]	1. No auxiliary heads (baseline) 2. Auxiliary supervision
Spatial Decoder	1. DOFA[41]: neck and UperNet (baseline) 2. SegFormer [39] 3. Prithvi [19] (one upsampling block) 4. Prithvi [19] (two upsampling blocks)
Spatio-Temporal Architecture	1. U-TAE [12] (baseline) 2. U-BARN [9]
Multi-Modal Fusion	1. Late: Sum (baseline) 2. Late: Concatenation 3. Mid 4. Early
Modalities	All subsets of the modalities PlanetScope, S2, S1
Spatial Encoder Parameters	1. ViT-B[8] (baseline) 2. ViT-L[8]
Temporal Encoder	1. Max-pooling (baseline) 2. L-TAE [12] 3. Single-temporal configuration

Table 5.2.: Ablation studies conducted in this work

5.3. Results

5.3.1. Quantitative Results

Training Hyperparameters and Regularization

Experiment	mIoU (%) ↑			Loss ↓		
	Train	Val	Test	Train	Val	Test
Default LR*	94.20	79.34	79.43	0.5213	0.6287	0.6404
Higher Spatial Encoder LR*	**96.91**	79.91	80.19	**0.5002**	0.6218	0.6282
Augmentations	89.63	81.15	81.34	0.5503	0.6051	0.6105
Temporal Dropout	87.06	**81.38**	**81.59**	0.5693	**0.6017**	0.6109
Auxiliary Head ($\lambda^{S1} = 0.25$)	86.71	81.25	81.55	0.5715	0.6020	**0.6102**

Table 5.3.: Experiment Results: mIoU and Loss for Train, Validation, and Test Sets. For every experiment, the experiment above in the table is set as baseline with the first experiment being the standard configuration. * LR: Learning Rate

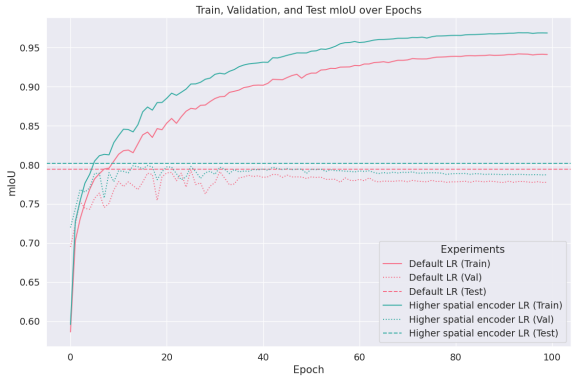
Table 5.3 shows the results for the ablation studies in hyperparameters and regularization methods. To analyze the overfitting of the models, both mIoU and loss for train, validation and test are listed. If the validation mIoU score increases in an ablation study, the new configuration is set as a new baseline for further studies. A higher spatial encoder learning rate achieves the highest train learning rate of all the methods but also shows a huge discrepancy in train and validation mIoU indicating a lot of overfitting. Continuously, each experiment except the *Auxiliary Head* experiment achieves a higher test and validation mIoU score and helps regularize the model indicated by a lower train mIoU. Similar behavior can be seen in the loss, where a large discrepancy indicates overfitting, as given for the configurations *Default LR* and *Higher Spatial Encoder LR*. All other configurations cause a smaller gap in train and validation/test loss indicating a lower overfitting. *Temporal Dropout* and *Auxiliary Head* achieve the lowest validation and test losses with insignificant differences.

Detailed training graphs are shown in Figure 5.1 visualizing the magnitude of overfitting. The influence of a higher learning rate of the spatial encoder is given in Figure 5.1a. The unregularized models both show a steady increase in train mIoU, however an oscillating validation mIoU. The validation mIoU for both experiments starts achieving its highest point around epoch 15 and slightly decreases from epoch 35 caused by heavy overfitting. In general, a higher learning rate causes the model to achieve higher mIoU scores and a lower loss across the train, validation and test set. This indicates that the model can be trained better and faster on a higher spatial encoder learning rate, but also that the model requires a stronger regularization.

As seen in Figure 5.1b, the train mIoU for the model without augmentations keeps in-

creasing while the validation mIoU decreases. The model with augmentations converges to a lower train mIoU while achieving a higher test and validation mIoU. The validation mIoU oscillates less and the validation mIoU starts converging from epoch 50 and does not decrease significantly in contrast to the model without augmentations. The combination of a closer mIoU and loss across splits indicates an effective regularization of overfitting by applying augmentations.

Figure 5.1c shows the effect of temporal dropout. While validation mIou of both approximately converge to the same value, especially the train mIoU is regularized by temporal dropout. While the training loss increases, the validation loss decreases with a consistent test loss and slightly higher test and validation mIoU. Temporal dropout reduces the redundancy in temporal data and requires fewer training resources due to a batch size of 1, as there is no padding of time sequences required. In combination with slightly less overfitting shown in the loss and mIoU, temporal dropout is an effective regularization of the model.



(a) Learning Rate

(b) Augmentations

(c) Temporal Dropout

Figure 5.1.: Training graphs for mIoU across epochs with ablations in: a) Learning Rate b) Augmentations c) Temporal Dropout

Multi-modal regularization with auxiliary heads on the other side, is not as effective, as test mIoU scores of all auxiliary configurations given in Table 5.4 decrease. Of all the

auxiliary configurations, the configuration with weight $\lambda^{S1} = 0.25$ performs best being insignificantly worse in mIoU and F1-Score than the configuration without auxiliary. For some classes like *Conifer* and *Deciduous*, IoU scores can slightly increase due to auxiliary supervision however across most scores, the configuration without auxiliary supervision performs best across most classes and metrics. However, the proposed auxiliary improves the mIoU score only slightly by 0.5% in their work[13]. Based on the experiments in this work, auxiliary supervision shows no positive effect on the test mIoU, requires more training resources and requires further hyperparameter fine-tuning. Hence, adapting multi-modal auxiliary supervision is left open for future work. Additional material is given in the Appendix A.1.

Experiment	Built-up	Conifer	Crop	Dead tree	Deciduous	Grass	Water	mIoU (%) ↑	OA (%) ↑	F1 (%) ↑
With Auxiliary $\lambda^{S1} = 0.5$	77.18	83.03	95.08	65.66	88.86	68.93	89.27	81.14	92.80	89.23
With Auxiliary $\lambda^{S1} = 0.375$	76.96	82.96	95.08	66.52	88.92	68.77	89.42	81.23	92.80	89.30
With Auxiliary $\lambda^{S1} = 0.25$	77.59	**83.19**	95.16	67.36	**89.13**	69.34	89.16	81.55	**92.94**	89.52
With Auxiliary $\lambda^{S1} = 0.125$	76.90	83.11	95.10	66.44	**89.13**	68.86	**89.49**	81.29	92.85	89.33
Without Auxiliary	**77.66**	82.89	**95.22**	**67.51**	89.02	**69.44**	89.36	**81.59**	92.91	**89.54**

Table 5.4.: Comparison of different configurations using auxiliary heads with varying λ^{S1} regarding their class-wise IoU (stated in %) scores and evaluation metrics on the test set

Spatial Decoder

The model size and computational complexity measured in FLOPs are important for each spatial decoder in a multi-modal setting, as efficiency is important due to high memory demand. Table 5.5 shows the computational complexity in FLOPs, amount of parameters of the model and achieved test mIoU. The spatial decoder for PlanetScope costs approximately 10 times more FLOPs for the decoder spatial encoder of PlanetScope than for Sentinel-1 and Sentinel-2, as the spatial resolution of PlanetScope images is 3 times higher. The spatial encoder requires 333 GFLOPS and thus the most compute for almost all listed configurations.

The DOFA[41] spatial decoder configuration including a convolutional neck and UperNet[38] is computationally expensive with 750 GFLOPs in total and a total of 60.78 million parameters per decoder for each modality, making up almost 300 million parameters for the whole model. Due to the big model size, the DOFA model requires the most FLOPs of the listed configurations and is the only configuration with more spatial decoder FLOPs than spatial encoder FLOPs. In contrast, the spatial decoder from the Prithvi [19] model with 1 of 2 blocks takes up only a fraction of the total flops. The Prithvi decoder requires with 4.7 million parameters just $\frac{1}{10}$ of the parameters of the DOFA decoder per modality, while on the other side performing better than all other spatial decoders by achieving 82.02 mIoU on the test set. The Prithvi single encoder with shared parameters across all modalities results in the same amount of FLOPs. Still, it requires only 4.72 million parameters for all modalities, e.g. almost 10 million parameters less for 3 modalities. This comes

however with a cost of approximately 0.4% less mIoU in comparison to the configuration with a trainable spatial decoder for every modality.

Scaling up the Prithvi decoder to its original size with 2 upscaling blocks increases the heads FLOPs, as the classification head is applied to the full input size of 128×128 instead of $\frac{128}{4} \times \frac{128}{4}$. Additionally, the GFLOPs for each spatial decoder increase from 0.8 to 12 for Sentinel-1 and Sentinel-2, and even more from 7 to 116 for PlanetScope. For both double-block configurations, the test mIoU stayed as good or decreased as the single-block configuration with fewer parameters and FLOPs.

The SegFormer[39] decoder yields approximately as many parameters and slightly more FLOPs than the single Prithvi[19] block decoder but achieves less mIoU, making the single Prithvi encoder the standard decoder.

Decoder	Head GFLOPs	Decoder S1 GFLOPs	Decoder S2 GFLOPs	Decoder Planet GFLOPs	Total GFLOPs ↓	Decoder Params* (M)	Total Params (M) ↓	Test mIoU (%) ↑
DOFA [41] (Neck + UperNet[38])	16.3	36.5	36.5	328.3	750.3	60.8	295.9	81.59
Prithvi [19] Single	16.3	0.8	0.8	6.8	**357.3**	4.7	127.7	**82.02**
Prithvi [19] Single Shared Decoder	16.3	0.8	0.8	6.8	**357.3**	4.7	**118.3**	81.59
Prithvi [19] Partial Double **	29.0	12.8	12.8	6.8	394.2	9.4	137.2	**82.02**
Prithvi [19] Double	29.0	12.8	12.8	115.6	503.0	9.4	141.9	82.00
SegFormer [39]	16.3	2.6	2.6	23.1	377.3	4.7	127.8	80.75

Table 5.5.: Comparison of model architectures and their performance. The spatial encoder requires 333 GFLOPs for all modalities. * Parameters per modality except for the shared decoder. For the *DOFA* configuration, the decoder includes the parameters of the neck and UperNet[38] ** Partial double means a double block is employed for S1 and S2, while a single block is used for PlanetScope.

Modal Fusion

Fusion	Spatial Encoder GFLOPs	Decoder GFLOPs	Head GFLOPs	Total GFLOPs ↓	Head Params (M)	Total Params (M) ↓	Test mIoU (%) ↑
Early	153.1	6.8	16.3	**176.2**	1.8	**118.3**	82.19
Mid	332.7	6.8	16.3	355.8	1.8	**118.3**	81.89
Late: Sum	332.7	8.3	16.3	357.3	1.8	127.7	82.02
Late: Concat	332.7	8.3	48.9	390.0	5.3	131.3	82.13
Late: Concat with $p_{S1} = 0.9$	332.7	8.3	48.9	390.0	5.3	131.3	**82.22**

Table 5.6.: Comparison of fusion methods in test mIoU and computational requirements measured in FLOPs and parameters. To measure the effect of few Sentinel-1 data caused by early fusion, a configuration with high Sentinel-1 dropout $p_{S1} = 0.9$ was added

Table 5.6 shows the parameters and FLOPs for each fusion method. All experiments lie within a mIoU difference of less than 0.4% such that the given results may not be expressive enough. Early fusion achieves the highest mIoU of 82.19% for all initial fusion methods while having the least amount of parameters and requires half of the FLOPs in comparison to the other fusion methods. This is due to the drop of more than 20 samples

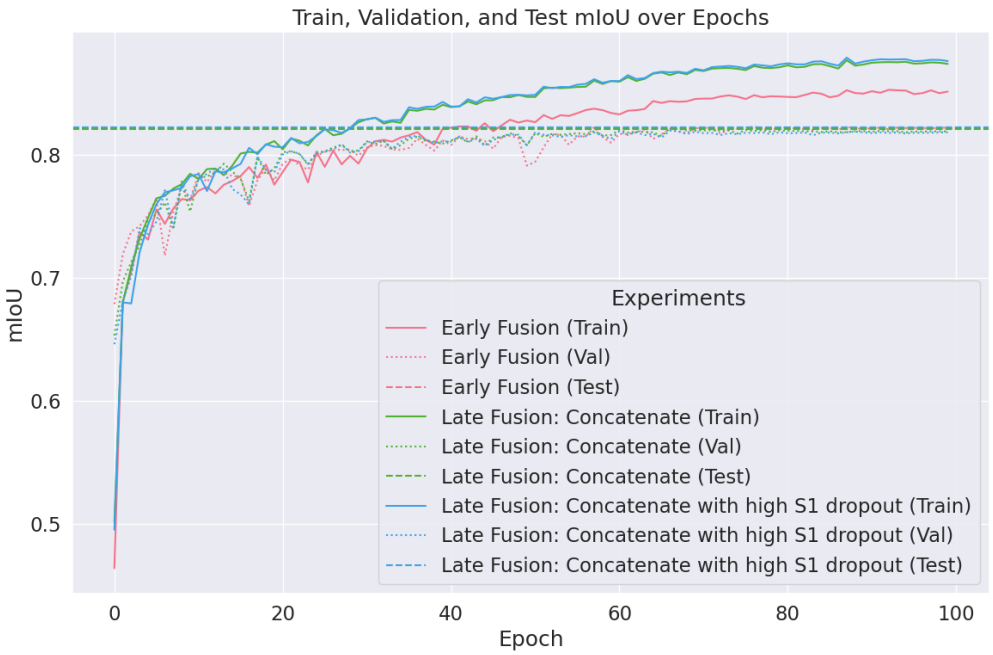


Figure 5.2.: Training graph of the mIoU along epochs for early fusion (red) and late fusion with concatenation (green and blue)

from Sentinel-1, as the smallest sequence size of all modalities is 4 and only the temporal nearest Sentinel-1 samples are concatenated. Since all modalities are initially concatenated, only 4 spatial encoder forward passes are required in contrast to more than 30 in the case of late or mid fusion. Early fusion might perform better due to the reduction of irrelevant Sentinel-1 samples in training. This might reduce the influence of gradients from Sentinel-1 samples in the spatial encoder and hence causes the spatial encoder to better learn features from Sentinel-2 and PlanetScope relevant for the task (see Table 5.8). The proposed concatenation strategy in combination with temporal dropout promotes concatenation diversity during training. However, no samples are dropped during test time, and the same combination of samples is concatenated. Here, future work can investigate how well each modality is diversified and further improve the concatenation strategy.

Mid fusion yields as many parameters as the shared single block Prithvi[19] configuration (Table 5.5) but achieves a slightly higher mIoU with 81.89% in comparison to 81.59%. However, it still performs worst of all the fusion methods. Due to the efficient lightweight spatial decoder, the FLOPs are insignificant lower than a late sum fusion.

Late fusion requires the most computational resources in parameters and FLOPs, as a spatial decoder is trained for each modality. The concatenation fusion is the most expensive, as the head parameters increase from 1.77 to 5.31 million parameters and the total GFLOPs from 357 to 390 respectively. The increase comes with a slight boost in performance from 82.02% to 82.13% mIoU, but is slightly worse than early fusion. Another experiment with high Sentinel-1 dropout $p_{S1} = 0.9$ for the late concatenation fusion was

added to verify the hypothesis that the early fusion might perform better by reducing irrelevant Sentinel-1 samples in training.

Figure 5.2 shows the training graph for early fusion and both concatenation methods. While both concatenation configurations have a very similar course of mIoU scores across all splits, the model with early fusion is heavily regularized visualized by a significantly lower train mIoU in comparison to the concatenation configurations. Despite the lower train difference, all models converge from epoch 60 to a similar validation mIoU.

Hence, the early fusion configuration overfits less than the late fusion configurations while achieving a similar mIoU. The fusion method strongly influences parameters and FLOPs, where early and mid fusion prove to be more efficient while being simpler than models with late fusion. However, late fusion allows the model to learn features for the fusion achieving the best mIoU metrics. The configuration *Late: Concat* is set as a baseline, as a high temporal dropout might prevent methods for temporal feature extraction from training. The results show that the temporal sequence length of each modality is crucial for the regularization of the shared large spatial encoder. Future work can further finetune the hyperparameters for temporal dropout and research the influence of each modality's sequence length on the gradients of the spatial encoder.

Spatio-Temporal Architecture

The change in spatio-temporal architecture causes an increase in FLOPs of the spatial decoder, as for every image, a forward pass through the spatial decoder is required. As seen in Table 5.7, the GFLOPs of the whole model increase from 390 for the U-TAE architecture to 427 for the UBARN architecture. The more expensive architecture achieves a slightly higher test mIoU of 82.27% mIoU compared to the previous 82.13% of the U-TAE architecture. Hence, the UBARN architecture is utilized as a new baseline.

Model	Spatial Encoder GFLOPs	Decoder GFLOPs	Total GFLOPs ↓	Total Params (M) ↓	Test mIoU (%) ↑
U-TAE [12] (baseline)	332.7	8.3	**390.0**	**131.3**	82.13
UBARN [9]	332.7	45.3	427.0	**131.3**	**82.27**

Table 5.7.: Comparison of the spatio-temporal UBARN[9] and U-TAE[12] architecture in computational resources and test mIoU

Modalities

Figure 5.3 shows the class-wise IoU and mIoU for each distinct modality and the combinations of all modalities. Sentinel-1 achieves the lowest IoU scores across all classes and hence the lowest mIoU of all modalities. While Sentinel-1 catches up to the other modalities for the classes *Crop* and *Water*, it falls behind in other cases achieving more than 0.2%

Figure 5.3.: Class-wise IoU and mIoU for each distinct modality and the combination of all modalities

less IoU for each class. Sentinel-2 performs almost as well as PlanetScope for the classes *Crop*, *Deciduous* and *Conifer*. PlanetScope's highest increase in mIoU compared to Sentinel-1 is for the class *Built-Up* which might be due to the higher ground sample distance. Overall, the mIoU and class-wise IoU show a clear progression in performance from Sentinel-1 (lowest) to Sentinel-2, then PlanetScope, with the combination of all modalities yielding the best results. Hence, multi-spectral data yields more information than SAR and high GSD is more relevant than covering a higher light spectrum. The best-performing combination of all modalities suggests that integrating multiple modalities provides the most accurate land cover classification.

Table 5.8 compares the different combinations of modalities regarding computational complexity and test mIoU. Due to late fusion with concatenation, each modality requires a spatial decoder and increases the number of head parameters. Hence, each additional modality increases the model size by approximately 6 million parameters, resulting in the largest model with all 3 modalities yielding 131.28 million parameters in comparison to a single modal model with 118.30 million parameters. The sequence length and model complexity determine the number of FLOPs required by the spatial encoder, decoder and head. Of all the single modal configurations, Sentinal-1 requires the most GFLOPs with 191 due to the high sequence length closely followed by PlanetScope due to the high spatial resolution. Sentinel-2 requires almost 10 times fewer FLOPs due to the low sequence length and low spatial resolution. The combination of modalities is approximately addi-

tive regarding FLOPs (some additional Head FLOPs due to concatenation fusion) such that the most complex model with all modalities has the highest total FLOPs (427 GFLOPs).

Sentinel-1 is the most inefficient modality requiring the most FLOPs while achieving by far the worst mIoU of 61.13% of all modalities. This is due to the highly redundant long temporal sequences which do not carry much information to distinguish vegetation (Figure 5.3). PlanetScope is computationally as efficient as Sentinel-1 but achieves with 81.12% test mIoU the best score of all modalities showing the benefits of high-resolution data yielding most information. Sentinel-2 is the most efficient due to a low spatial resolution and low temporal sequences achieving 78.61% test mIoU while only requiring 21 GFLOPs. When combining different modalities, the best-performing subset provides a lower bound on the test mIoU in all cases. This behavior is expected, as providing more data and diversifying data should improve the model performance and capabilities. The merge of Sentinel-1 and Sentinel-2 causes almost no improvement in test mIoU, which on the other hand might be due to the missing regularization of Sentinel-1 data. The combination of PlanetScope with Sentinel-1 or Sentinel-2 both yield a similar test mIoU score while the combination of PlanetScope and S2 requires 170 fewer GFLOPs. Combining all modalities achieves the best mIoU score of 82.27% but requires most FLOPs and parameters.

Modality	Spatial Encoder GFLOPs	Decoder GFLOPs	Head GFLOPs	Total GFLOPs ↓	Total Params (M) ↓	Test mIoU (%) ↑
S1	167.00	22.70	1.81	191.00	**118.3**	61.13
S2	17.30	2.27	1.81	**21.37**	**118.3**	78.61
Planet	148.72	20.39	16.33	185.45	**118.3**	81.12
S1 Planet	315.41	43.06	32.64	391.10	124.8	81.66
S2 Planet	166.01	22.66	32.64	221.31	124.8	81.69
S1 S2	183.98	24.93	3.63	212.53	124.8	78.97
All	332.70	45.32	48.95	427.00	131.3	**82.27**

Table 5.8.: Comparison of different combinations of modalities regarding computational resources and test mIoU

Spatial Encoder Parameters

Table 5.9 shows a significant increase in GFLOPs when transitioning from a ViT-B to ViT-L spatial encoder. The *MT Planet* configuration's spatial encoder GFLOPs increase from 149 to 525 when moving to the ViT-L version, representing an approximately 3.5-fold increase. Similarly, the spatial encoder GFLOPs of the *MT S2 Planet* configuration increase from 166 to 585 with the ViT-L architecture. This increase in computational complexity is accompanied by a rise in the number of parameters, with the spatial encoder parameters approximately tripling from about 112 to 338 million parameters.

However, despite this high increase in model size and complexity, the improvement in mIoU is almost insignificant with an increase of approximately 0.1% mIoU for both larger encoder ablations. This limited performance increase despite the larger spatial model size and higher computational requirements suggests that the model architecture is fully utilized or more likely that the data quality is not good enough. The dataset is mostly automatically annotated and limits the model performance, as highlighted in Chapter 4.1 and 5.3.2. This can also be seen in more overfitting of the larger model, as the validation mIoU decreases for the ViT-L for Sentinel-2 and PlanetScope configuration. At the same time, the train mIoU increases by 2% mIoU in comparison to its ViT-B counterpart.

Model	Backbone GFLOPs	Decoder GFLOPs	Total GFLOPs ↓	Parameters Spatial Encoder (M)	Total Parameters (M) ↓	Train mIoU (%) ↑	Val mIoU (%) ↑	Test mIoU (%) ↑
MT Planet	149	20	185	111.80	118.30	85.52	80.80	81.12
MT Planet ViT-L	525	36	583	337.76	348.52	86.06	81.00	81.24
MT S2 Planet	166	23	221	111.80	124.79	87.82	81.91	81.69
MT S2 Planet ViT-L	585	40	669	337.76	359.27	89.39	81.57	81.79

Table 5.9.: Comparison of ViT[8] configurations in the DOFA[41] as spatial encoder regarding test mIoU and computational resources

Temporal Encoder

Temporal Configuration	Backbone GFLOPs	Head GFLOPs	Decoder GFLOPs	TE Planet GFLOPs	TE S1 GFLOPs	TE S2 GFLOPs	Total GFLOPs ↓	Total Params (M) ↓	Test mIoU (%) ↑
ST S1	166.7	54.4	22.7	-	-	-	243.8	**118.3**	51.60
ST S2	17.3	5.4	2.3	-	-	-	25.0	**118.3**	76.46
ST Planet	148.7	48.9	20.4	-	-	-	218.1	**118.3**	80.61
ST All	332.7	587.9	45.3	-	-	-	965.9*	**118.3**	76.76
ST All - S1	166.7	54.4	22.7	-	-	-	243.8	**118.3**	49.32
ST All - S2	17.3	5.4	2.3	-	-	-	25.0	**118.3**	70.70
ST All - Planet	148.7	48.9	20.4	-	-	-	218.1	**118.3**	78.84
MT S1	167.0	1.8	22.7	-	-	-	191.0	**118.3**	61.13
MT S2	17.3	1.8	2.3	-	-	-	**21.4**	**118.3**	78.61
MT Planet	149.0	16.3	20.4	-	-	-	185.0	**118.3**	81.12
MT All	332.7	48.9	45.3	-	-	-	427.0	131.3	82.27
L-TAE[11] S1	167.0	1.8	22.7	-	21.3	-	212.0	119.0	64.51
L-TAE[11] S2	17.3	1.8	2.3	-	-	2.1	23.5	119.0	77.88
L-TAE[11] Planet	149.0	16.3	20.4	19.1	-	-	205.0	119.0	81.09
L-TAE[11] All	332.7	48.9	45.3	19.1	21.3	2.1	469.0	133.4	82.33
L-TAE[11]/Max-Pool All	332.7	48.9	45.3	-	21.3	-	448.0	132.0	**82.37**

Table 5.10.: Comparison of temporal configurations applied to different sets of modalities regarding their computational resources and test mIoU. *ST All - Modality* denotes the model trained on all modalities and tested on the specified modality. *L-TAE/Max-Pool* denotes temporal max-pooling for Sentienl-2 and PlanetScope and L-TAE[11] used with Sentinel-1 data. * Total GFLOPs can be optimized to 486.9 (=S1+S2+Planet)

Table 5.10 states the test mIoU and temporal encoder GFLOPs for a single temporal (ST), multi temporal (MT) and L-TAE[11] configuration for each modality and the configuration with all modalities.

The increase in mIoU through temporal max-pooling is most outstanding for Sentinel-1, improving by almost 10% mIoU. The max-pooling configuration achieves significantly higher mIoU while requiring instead of 244 GFLOPs only 191, since for a single prediction only 1 head forward pass is needed instead of nearly 30. L-TAE for Sentinel-1 adds another 21 GFLOPs and 690K parameters, but also further leads to an improvement of 3% mIoU. Hence, by making the model able to learn temporal features for long sequences of Sentinel-1 data, the model is able to improve mIoU by 13% while requiring 15% less GFLOPs.

Multi-temporal max-pooling also improves the mIoU for Sentinel-2 by almost 2.15% accompanied by a decrease of GFLOPs from 25 to 21.4, making it the most light-weight configuration. On the side, adding temporal encoding with L-TAE causes a decrease of 0.5% mIoU with the cost of additional FLOPs and parameters. Moving from the single-temporal to the multi-temporal configuration, the increase of the mIoU is smaller with approximately 0.5% for PlanetScope while requiring 23 GFLOPs less. PlanetScope with L-TAE on the other side even decreases mIoU by 0.03% in comparison to the max-pooling configuration. Similar to Sentinel-2, this decrease in mIoU may be due to the short sequence length of 3-4 samples, as in reference works L-TAE is applied to sequence lengths of 50-100[13, 12]. This short sequence length may prevent the model from learning useful temporal features such that simple temporal max-pooling is more beneficial.

The single temporal model on all modalities degenerates from 82.27% mIoU to 76.76% while requiring 60 GFLOPs[2] more. This decrease in mIoU may be due to the imbalance of data available in modalities and information provided for the task by each modality. Sentinel-1 provides the most data and meanwhile the least information for the classification task (Table 5.8), such that gradients in the model might be dominated by Sentinel-1 samples similar to the findings in the fusion ablations (Table 5.6). Therefore, an evaluation with samples of each modality separately of the trained *ST-ALL* configuration is given in Table 5.10. While the mIoU of the model on Sentinel-1 data only decreases slightly compared to the *ST S1* configuration, a similar drop in mIoU can be found on the evaluation of PlanetScope samples. On the other hand, the mIoU decreases by almost 6% mIoU compared to the *ST S2*, which shows that the model degenerates the least for the modalities providing the most data and information for the segmentation task. All evaluations of the *ST All* configuration generally achieve worse mIoU than the analog single-temporal configuration trained on a single modality. This demonstrates the importance of fusing all modalities in the model, which is computationally less expensive and improves mIoU.

Figure 5.4 visualizes the findings of Table 5.10, where the mIoU increases consistently with more temporal feature encoding due to long-sequence lengths provided by Sentinel-1. PlanetScope and Sentinel-2 benefit from temporal Max-Pooling but achieve less mIoU with L-TAE due to short sequence size. Further, all models benefit from temporal feature

[2]This can be achieved by adjusting the forward flow. In the implementation of the framework, the feature maps from Sentinel-1 and Sentinel-2 samples are resized to the spatial resolution of PlanetScope resulting in tremendous amounts of unnecessary additional head FLOPs

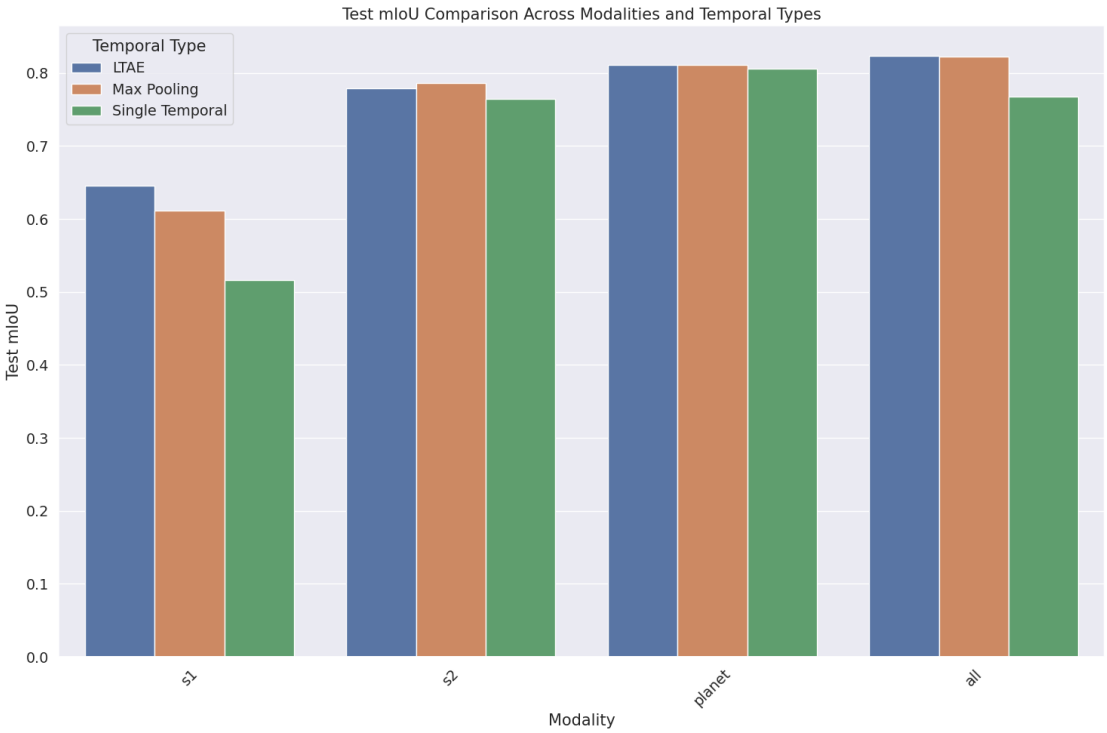
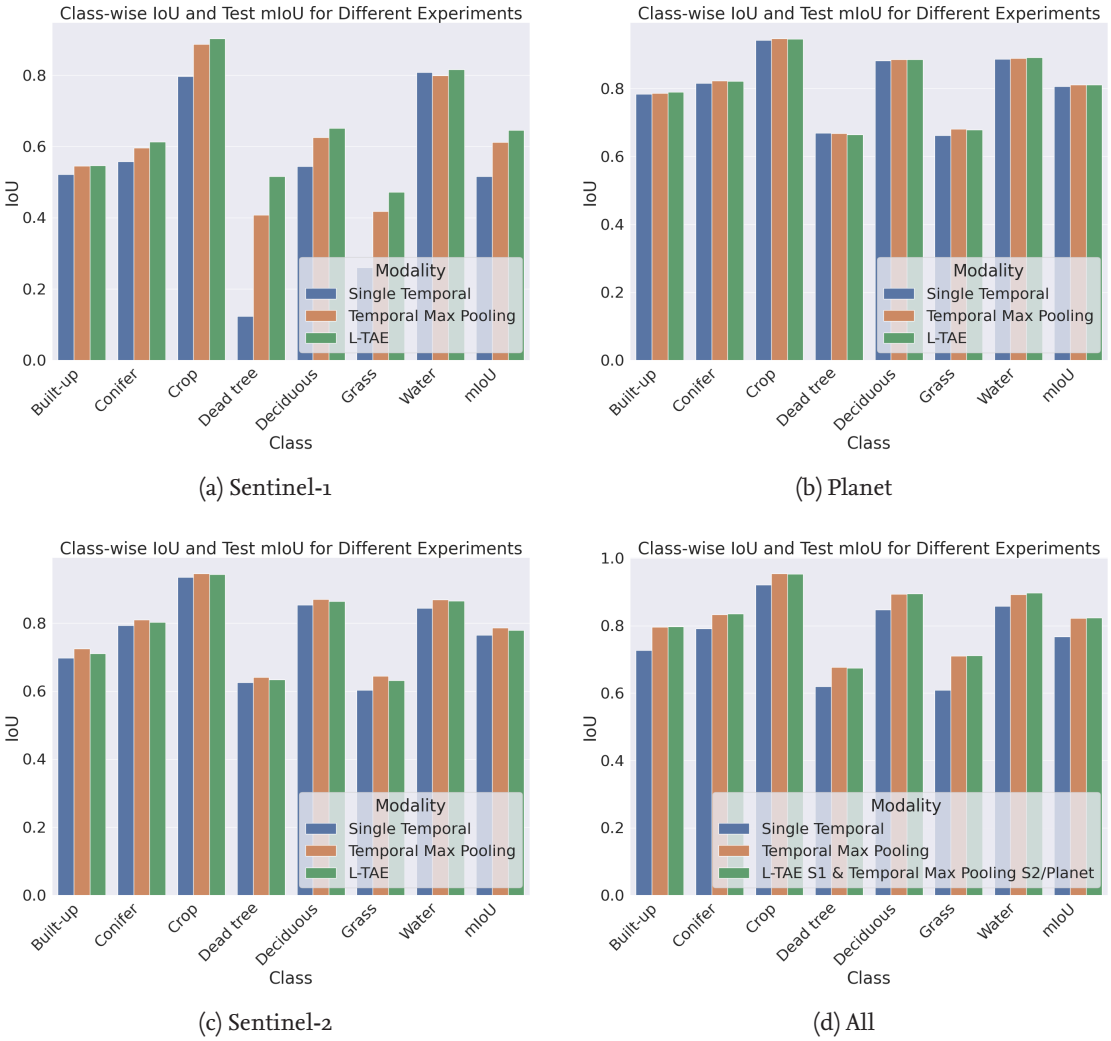


Figure 5.4.: Test mIoU scores for different temporal encoders across modalities

encoding through max-pooling or L-TAE, while requiring fewer FLOPs due to a single head forward pass for each prediction. Hence, another configuration *L-TAE/Max-Pool All* was added with L-TAE only added to the Sentinel-1 modality. This configuration achieves with 82.37 mIoU the highest score of all models in this thesis and requires 21 more GFLOPs than the *MT All* reference.

Figure 5.5 visualizes class-wise IoU for each modality and the combination of all modalities. For each distinct model, the class-wise IoU increases across all classes showing the effectiveness of adding temporal information to the model. Sentinel-1 shows the largest increase from single temporal to temporal max-pooling, with the largest increase for the class *Dead Tree* and further big increases for the classes *Deciduous*, *Grass* and *Crop*. L-TAE encoding result in more increases in IoU across all classes with significant increases for the classes *Dead Tree* and *Grass*, highlighting the efficiency of temporal encoding for large temporal sequences. Sentinel-2 consistently benefits from max-pooling and achieves similar gains in IoU across all classes compared to the single-temporal configuration. L-TAE even causes consistent decreases in IoU, as seen best for the classes *Built-Up* and *Grass*. PlanetScope on the other hand shows more significant increases in mIoU for the classes *Dead Tree*, *Grass* and *Water*. For other classes, the IoU scores just slightly increase. Similar to Sentinel-2, L-TAE causes no improvements in IoU for PlanetScope but there are also no major decreases visible for any classes in contrast to Sentinel-2. The sequence length for Sentinel-2 and PlanetScope is too short to learn any temporal feature, but IoU should



(a) Sentinel-1

(b) Planet

(c) Sentinel-2

(d) All

Figure 5.5.: Class wise IoU scores and mIoU across different temporal encoders for different modalities: a) Sentinel-1 b) PlanetScope c) Sentinel-2 d) All. Note, that the *L-TAE* configuration is not stated in Figure 5.5d but instead the configuration *L-TAE S1 & PlanetScope/S2 Temporal Max-Pool*

on the other hand improve for larger time sequences available as seen for Sentinel-1. For all modalities, the model consistently gains IoU with the most significant gains found in *Dead Tree* and *Grass*. Similar to PlanetScope, the *Dead Tree* IoU decreases only for the best configuration. Some decreases in IoU despite a better mIoU may be due to the label quality, as further analyzed in Chapter 5.3.2 and in the Appendix A.3.

To further analyze misclassifications, both vertically and horizontally normalized confusion matrices are visualized for the best model *L-TAE/Max-Pool* in Figure 5.6. The class *grass* shows both a high horizontal and vertical confusion with the classes *crop*, *built-up* and *conifer*. The high confusion with *crops* may be due to unused farmlands, while the confusion with *conifer* may be due to labeling quality in deforested or recovering areas. The

(a) Vertically normalized confusion matrix L-TAE/Max-Pool

(b) Horizontal normalized confusion matrix L-TAE/Max-Pool

Figure 5.6.: Confusion matrices normalized in vertical (a) and horizontal (b) dimension. The class *tree* is treated as a background class during training

class *crop* performs in both the vertical and horizontal confusion the best, which may be due to its spatial homogenous distribution and discriminability over time. A high vertical and horizontal confusion can be seen between the classes *built-up* and *grass*, which may be due to fine-granular *built-up* labels, which the ViT[8] is unable to capture as visualized in Chapter 5.3.2 due to the 16×16 patch embedding. The class *water* is despite the confusion with the class *tree* almost as well distinguishable as the class *crop*. This is probably due to the spatial homogenous distribution and well visibility in the near-infrared spectrum. For the vertical confusion matrix, conifer and deciduous have a high confusion between each other ranging from 4% to 5%, which may be due to the automatic label generation and similarity of classes. This also holds for the horizontal confusion matrix, while there is an additional high confusion between the class conifer and grass of 3%. For the class dead tree, the largest amount of false positives with approximately 21% are assigned to conifer, while dead trees on the other hand only make up 1% of false positives for conifer. This imbalance is probably due to the imbalance of classes (Figure 4.6). The classifier also finds 83% of dead tree labels, with most false negatives being conifer with 16%. The high confusion between dead trees and conifer is expected due to the difficulty in label uncertainty as shown in Chapter 5.3.2. The background class tree is distributed mostly across the class grass with 40%, deciduous with 28% and built-up with 13%, showing that the labels mostly exist outside the forests due to low confusion with tree classes.

Data & Weight Initialization

Figure 5.7 shows the effectiveness of initializing the spatial encoder weights. The model is trained with the ViT-B[8] spatial encoder initialized with no pre-trained weights (from

scratch), with pre-trained weights based on ImageNet[7] and with the whole DOFA[41] spatial encoder pre-trained as a foundation model. As expected, training the model from scratch yields the lowest mIoU scores across all splits. Initializing the ViT-B with ImageNet weights significantly increases the train and validation mIoU but only slightly the test mIoU. The effectiveness of foundation model weights for the spatial encoder can be seen in a large gap in mIoU scores across all splits. This coincides with previous findings that the ViT requires a large pre-training corpus due to low inductive bias of the model[8, 41]. However, the model also overfits more comparing to the other weight initializations, which hints at improvements with better data.

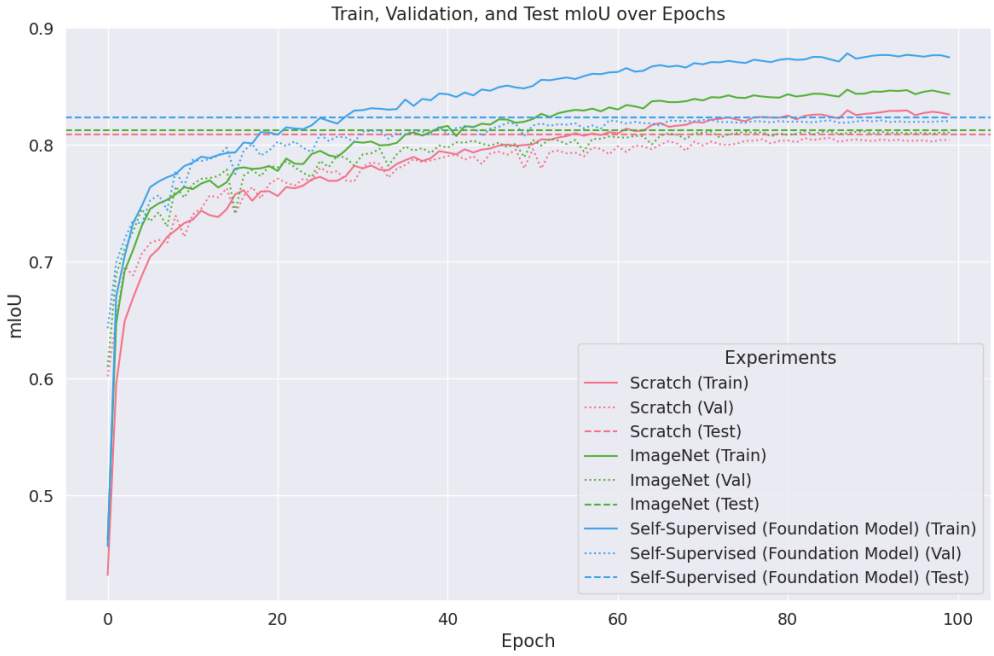


Figure 5.7.: The model trained with different weight initializations of the DOFA[41]. ImageNet[7] weights only initialize the ViT-B[8] backbone.

Figure 5.8 shows the mIoU on the train-test-validations splits for the model initialized with foundation model weights and the model trained from scratch for different amounts of data available. Similar to the findings in Figure 5.7, the model initialized with foundation model weights outperforms the model trained from scratch across all experiments. For both the foundation model and the model trained from scratch, both validation and test mIoU converge to the same value when utilizing all data available. The available data regarding the mIoU score increases logarithmically, starting to saturate with approximately 25% of the available data. However, there are big gaps and fluctuations in validation and test mIoU, especially for a few data used. This can be explained due to the label quality, which causes more inconsistencies across splits with few data.

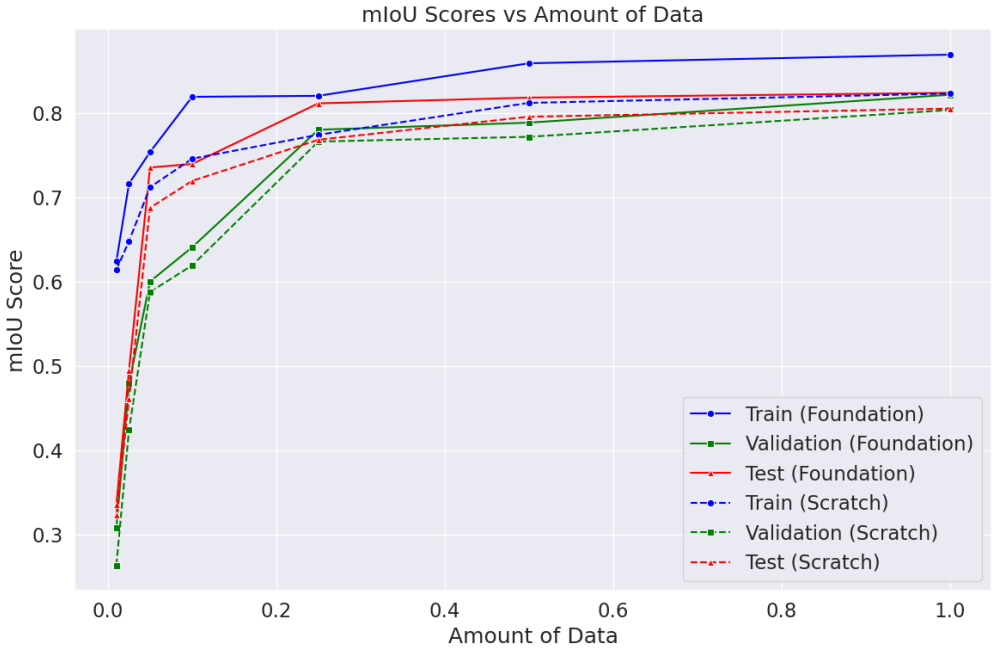


Figure 5.8.: mIoU scores for models initialized with DOFA[41] weights (straight lines) or no pre-trained weights (dotted) in relation to different amounts of available data

Overview

An overview of the experiments is given in Table 5.11, while mIoU in relation to FLOPs is visualized in Figure 5.9.

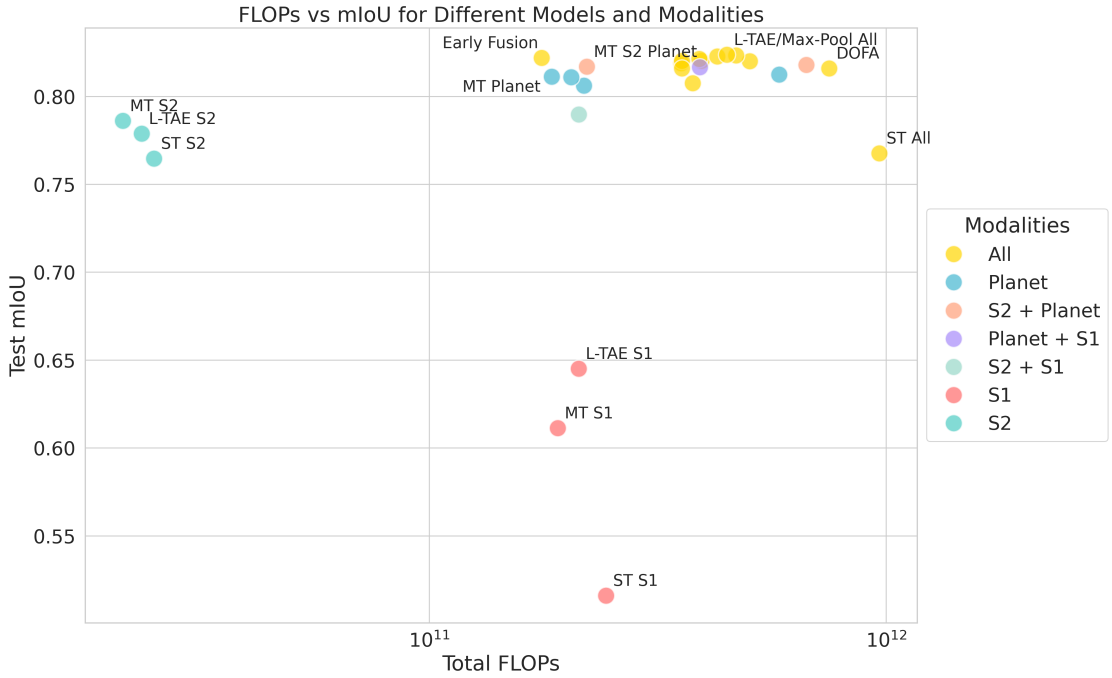


Figure 5.9.: Effectiveness of modalities measured in FLOPs in relation to the test mIoU for each configuration with color codes based on different subsets of modalites

Configuration	Head GFLOPs	Backbone GFLOPs	Decoder GFLOPs	Total GFLOPs ↓	Head Params (M)	Backbone Params (M)	Decoder Params (M)	Total Params (M) ↓	Test mIoU (%) ↑
DOFA[41]	16	333	401	750	1.8	111.8	60.8	295.9	81.59
Prithvi[19] Single	16	333	8	357	1.8	111.8	4.7	127.7	82.02
Prithvi[19] Shared Single	16	333	8	357	1.8	111.8	4.7	**118.3**	81.59
Prithvi[19] Partial Double	29	333	33	394	1.8	111.8	9.4	137.2	82.02
Prithvi[19] Double	29	333	141	503	1.8	111.8	9.4	141.9	82.00
SegFormer[39]	16	333	28	377	1.8	111.8	4.7	127.8	80.75
Early Fusion	16	153	7	176	1.8	111.8	4.7	**118.3**	82.19
Mid Fusion	16	333	7	356	1.8	111.8	4.7	**118.3**	81.89
Late Concatenate Fusion	49	333	8	390	5.3	111.8	4.7	131.3	82.13
UBARN[9]	49	333	45	427	5.3	111.8	4.7	131.3	82.27
S1 Planet	33	315	43	391	3.5	111.8	4.7	124.8	81.66
S2 Planet	33	166	23	221	3.5	111.8	4.7	124.8	81.69
S1 S2	4	184	25	213	3.5	111.8	4.7	124.8	78.97
Planet	16	149	20	185	1.8	111.8	4.7	**118.3**	81.12
S1	2	167	23	191	1.8	111.8	4.7	**118.3**	61.13
S2	2	17	2	**21**	1.8	111.8	4.7	**118.3**	78.61
ST S1	54	167	23	244	1.8	111.8	4.7	**118.3**	51.60
ST S2	5	17	2	25	1.8	111.8	4.7	**118.3**	76.46
ST Planet	49	149	20	218	1.8	111.8	4.7	**118.3**	80.61
ST All*	588	333	45	966	1.8	111.8	4.7	**118.3**	76.76
L-TAE S1	2	167	23	212	1.8	111.8	4.7	119.0	64.51
L-TAE S2	2	17	2	24	1.8	111.8	4.7	119.0	77.88
L-TAE Planet	16	149	20	205	1.8	111.8	4.7	119.0	81.09
L-TAE All	49	333	45	469	5.3	111.8	4.7	133.4	82.33
L-TAE/Max-Pool All	49	333	45	448	5.3	111.8	4.7	132.0	**82.37**
Scratch	49	333	45	448	5.3	111.8	4.7	132.0	80.05
ImageNet	49	333	45	448	5.3	111.8	4.7	132.0	81.23
MT Planet ViT-L	22	525	36	583	2.4	337.8	8.4	348.5	81.24
MT S2 Planet ViT-L	44	585	40	669	4.7	337.8	8.4	359.3	81.79

Table 5.11.: Comparison of ablations in the spatial decoder, fusion method, spatio-temporal architecture, temporal encoder, modalities, weight initialization and spatial encoder size w.r.t. computational requirements and test mIoU. *L-TAE/Max-Pool* denotes temporal max-pooling for Sentienl-2 and PlanetScope and L-TAE[11] used for Sentinel-1. * Total GFLOPs can be optimized to 486.9 (=S1+S2+Planet)

5.3.2. Qualitative Evaluation

Figure 5.10 visualizes the confusion (5.6) between the classes *grass* and *built-up*. While the reference labels are fine-granular due to the random forests generation, the class boundaries generated by the model are spatially smoothed out. Hence, the model is unable to capture details. This causes many ground truth *grass* pixels to be predicted as *built-up* and vice versa.

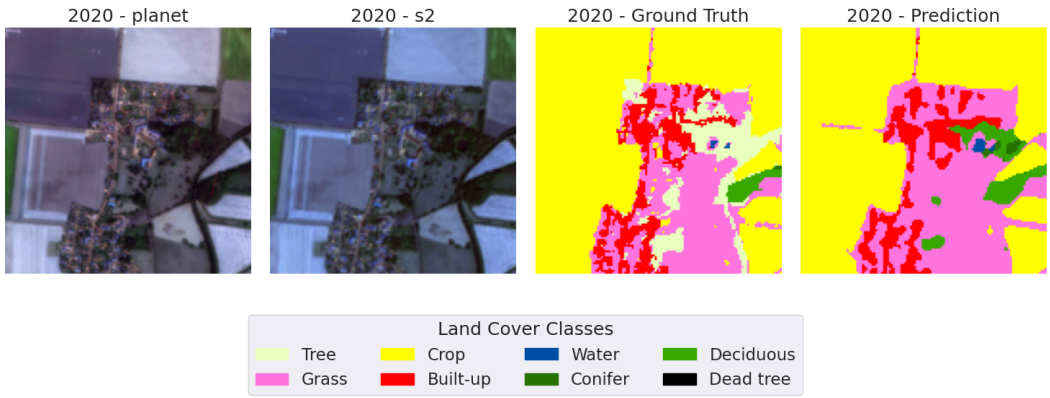


Figure 5.10.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. Class labels for *built-up* are more fine-granular than the predicted label map of the model

Issues in model and label quality can be seen in Figure 5.11. The labels for small country roads and field paths are inconsistently marked as the class *grass*, making it hard for the model to properly learn from the data, as *crop* is the most dominant class in the dataset (Figure 4.6). Hence, the model is biased towards the class *crop*, but may also be more likely to generate false *grass* predictions for country roads, where the automatically generated labels may be missing.

The issue in model prediction and fine-granular auto-generated labels comes also with other classes, as seen on the right with mixed-forests in Figure 5.12, There are some small clusters of conifer trees, which the model predicts partially. Hence, in the case of small fine-granular shapes, pixel-based classifiers outperform large models requiring only fractions of computational resources (given correct labels). In this figure, for the class *crop*, the labels and predictions don't seem plausible because the small shapes near a forest contradict each other.

For the class *dead tree*, the labels were manually annotated and as visualized in Figure 5.13, As seen in the ground truth map on the left, there are some annotations issues, where a large area of dead trees is not correctly annotated. The model however learns to classify these correctly and learns smooth boundaries in contrast to the rectangular annotation shapes. Another issue in the ESA labels can be found in noisy *built-up* pixels in areas with dead trees, which the model learns to ignore. This labeling issue only appears for the year

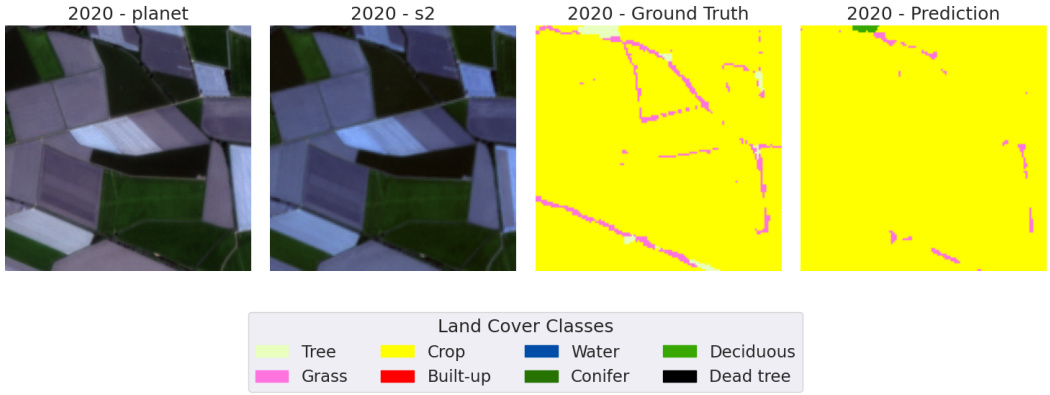


Figure 5.11.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. Labels of the class *grass* mark country roads and field paths which are hard to learn for the model

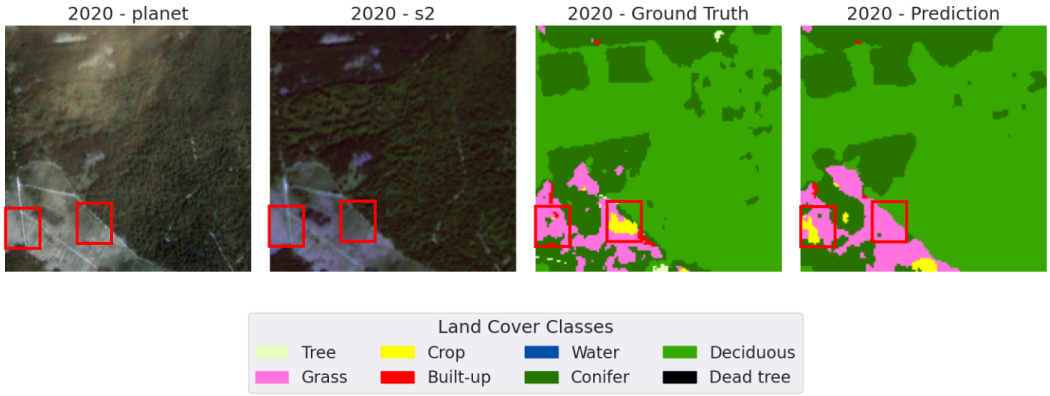


Figure 5.12.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangles, noisy labels and predictions of the class *Crop* are visualized

2020, as seen in Figure 4.5 when zooming into the Harz region.

Figure 5.14 visualizes the confusion between *conifer* and *grass*. The deforested area near the top left is falsely labeled as *conifer*. Here, the model can classify the area as vegetation but fails to classify the small clusters of deforested spots.

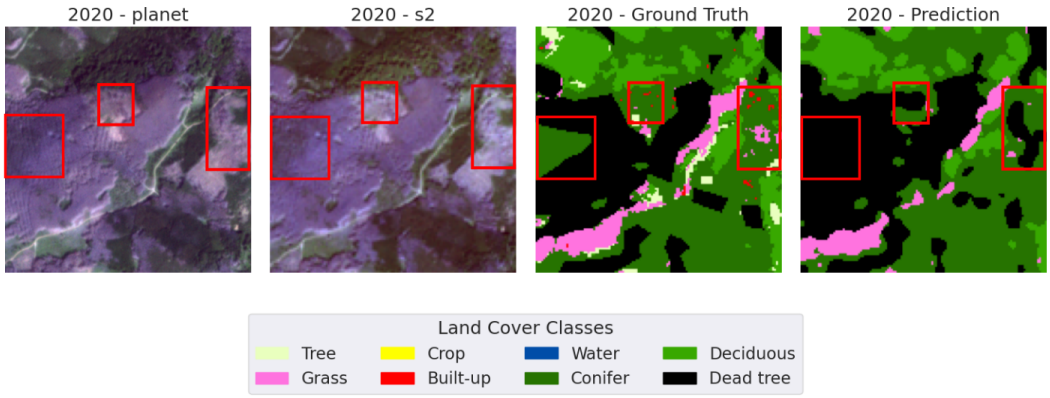


Figure 5.13.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangles, differences in labels and prediction of the class *dead tree* are visualized

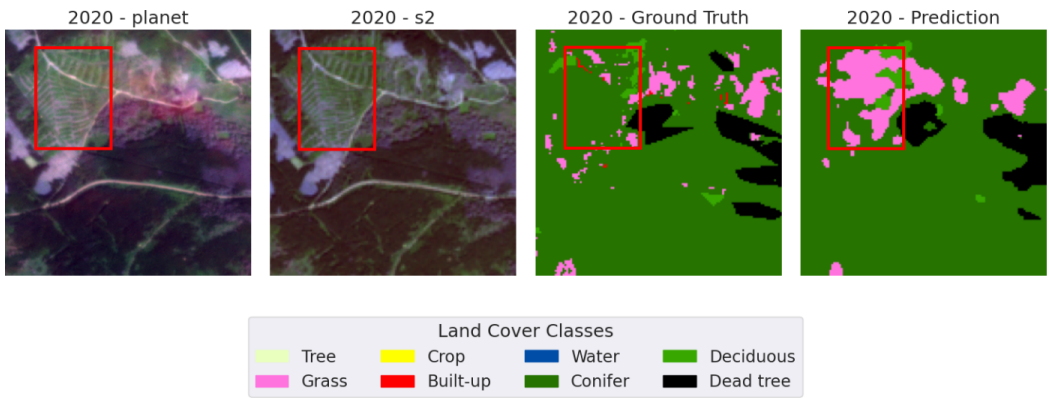


Figure 5.14.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. In the red rectangle, differences in labels and predictions regarding the class *grass* are visualized

6. Discussion

Augmentations show a strong regularizing effect on the model and since the model is still overfitting (see Figure 5.1c), more and stronger regularization techniques can be evaluated. Some may be a photometric distortion on spectral data or a stronger spatial regularization such as a random resize ratio in the random resize crop augmentation.

Our findings contradict the findings by Garnot et al. [13] of the benefits of auxiliary supervision for multi-modal training. This may be due to wrong hyperparameters or the position of the auxiliary head. Additionally, a significant difference compared to their work is, that the spatial encoder shares parameters across all modalities. Hence, the imbalance towards Sentinel-1 data dominates the spatial encoder during training which can decrease the model's performance with an auxiliary loss for Sentinel-1. The model might benefit from further supervision through auxiliary heads, but fine-tuning hyperparameters is necessary. However, adding auxiliary heads allows the model to be trained for inference with limited modalities available during test time (Appendix A.1). It might be interesting to freeze a trained model (or parts of the model) while modality-specific heads are trained only.

The uncertainty in the data due to the automatic label generation distorts the results obtained by this work. Since the data was automatically generated with an overall accuracy of, 75% for 2020 and 77% for 2021 (plus tree labels with 75% overall accuracy), the metrics can measure model quality to some extend only. One disadvantage of the model is that it is unable to generate fine-granular labels, which are important for roads, buildings and mixed forests. This may be due to the ViT's patchifying process or a loss of fine-granular information throughout the ViT, as only features of the last ViT layer are utilized in the spatial decoder. The class *dead tree* class is the most difficult to learn, as the labeling shows inconsistencies (see Figures 5.13 and A.4). Another difficult class below 80% IoU is *grass* which shows similarities with the class *crop* and is also inconsistently labeled in forests (see Figures 5.14 and A.5). Hence, better label quality should decrease overfitting and result in more consistent predictions, especially for the classes *Grass* and *Dead Tree*.

High-resolution images yield the most information for the segmentation task and the model learns multi-resolution features without loss in performance (see Table 5.8), which may be due to the close GSD of PlanetScope (3m) and Sentinel-2/1 (10m). Multi-modal spatial encoders with shared weights across modalities might however degenerate for larger discrepancies in GSD, as related works train separate encoders for very high-resolution (0.2m-0.3m) and multi-spectral Sentinel-2 (10m) data [15, 1].

DOFA[41] was pre-trained on 5 different modalities with variations in GSD and spectral bands covered. The difference in performance and efficiency of training separate spatial encoders instead of a jointly trained shared spatial encoder may be an interesting area

for future work. Both approaches can be evaluated in terms of efficiency of computational requirements and achieved mIoU. The results (Table A.1) in the appendix A show that small configurations of the model achieve slightly lower mIoU scores by less than 1.5% mIoU while requiring significantly fewer parameters. It might be more beneficial to employ small independent encoders instead of one large multi-modal shared encoder. However, no foundation model weights are available for smaller configurations[1].

Early fusion strongly regularizes the model, and further experiments with higher temporal dropout (Table 5.6) show that too many samples from Sentinel-1, in combination with temporal max-pooling, decrease the model performance. The hyperparameters for temporal dropout in this work were adapted by Garnot et al.[13] and should be further evaluated in future work. Mid fusion could further benefit from L-TAE, as the sequence gets long due to the concatenation of all sequences. To distinguish features by modalities, a group encoding[6] could be added in addition to the day-of-year encoding. Other recent works like SkySense[15] or OmniSAT[1] utilize attention to fuse modalities, which is worth to be further explored.

This work further showed that the sequence size of each modality encourages different types of temporal encoders. For this work, only a few optical data are available in contrast to other works with temporal encoders with at least 50 samples per sequence available[13, 12]. There is only a few optical data available for this work, as it focuses on the growing season as otherwise dead trees are more similar to soil and cropland. The effect of long-sequence optical data by gathering data from the whole would be interesting. However, high-resolution data is computationally expensive, and a good trade-off must be found between performance and computational requirements.

Integrating SITS in large foundation models is computationally expensive, but training models with temporal data consistently increases mIoU. Hence, more memory is essential to scale up the model and data. In contrast to this work, the spatial encoder and decoder of other works are lightweight with approximately 1-2 million parameters[12, 9]. The effectiveness of utilizing the large DOFA[41] spatial encoder with 112 million parameters in contrast to lightweight spatial feature extractors has to be further studied. Alternatively, the FLOPs can be significantly reduced by replacing the spatial encoder with a more lightweight spatial encoder and smaller ViT configurations (Appendix A.2) or by reducing the patch size to 64×64 or 96×96.

As the difference between a spatial encoder pre-trained as a foundation model and training from scratch only varies between 1.5 and 1% mIoU compared to ImageNet weights, investigating other more recent spatial encoder architectures might be more beneficial to achieve the best performance for the segmentation task. The ViT foundation model can be exchanged with other spatial encoder architectures such as the MoE-Swin-Transformer[24], which was also trained as a foundation model for remote sensing applications by Han et al.[16]. To the date of writing this thesis, the code and weights of their foundation model

[1] Last checked (29.11.24): https://huggingface.co/XShadow/DOFA/tree/main

are not available yet, however, the code for the MoE-Swin is publicly available[2]. It may be however challenging to adapt the Swin-Transformer to multi-resolution input, as the image size is hard-coded through the model due to the window partitions.

Alternatively, the model can be scaled up in parameter size in the spatial encoder and decoder. The spatial feature extractor in DOFA can be exchanged from a ViT-B to a ViT-L, which takes up approximately 200 million more parameters. The experiments in this work did not show any improvements (Table 5.9) with a larger spatial encoder which may be due to architecture limitations or due to the data uncertainty. More parameters in the spatial decoder do not necessarily result in performance, as the UperNet[38] decoder in combination with a convolutional neck performs worse than the Prithvi[19] decoder. The Prithvi[19] single block decoder has proven itself lightweight and efficient, however, we could not improve the model by applying the whole decoder. This could be due to the sum-fusion as the initial fusion method or the difference in spatial resolution of the features. The Prithvi decoder consisting of two blocks is expected to perform better on Sentinel-1 and Sentinel-2 as classification is done on the original label resolution. Additionally, some further ablations can fine-tune hyperparameters such as the embedding dimension of the spatial decoder to optimize the model.

Since training takes up a maximum of half a day for each configuration and the models are trained on 4 GPUs in parallel, this work was implemented with a batch size of 1. Dumeur et al. [9] showed that the batch size can influence the model quality, which can be further evaluated. Larger batch sizes might improve the model quality but require more computational resources. However, to allow larger batch sizes and thus transfer this approach to larger model sizes and datasets, padding and masking each modality's temporal sequence might be required due to variable sequence sizes. This would, on the other hand, make the whole data pipeline and inference even more complicated and computationally expensive.

To this date, *mmseg* is not designed to handle multi-modal and multi-temporal data, such that at the beginning, it was easier to implement the data pipeline and model from scratch than to adapt the *mmseg* framework. Due to debugging purposes, the control of modules and variations in forward passes based on fusion methods, I decided to implement the framework from scratch based on *GeoSeg* and *PyTorch Lightning*. In hindsight, a multi-temporal and multi-modal wrapper for *mmseg* might be more beneficial for future work due to popularity in research. This, however, requires some in-depth knowledge about the framework and is more difficult to debug. Some modules from the *mmseg* framework are adapted, but implementations or hyperparameters might not be fully correct, as the original DOFA[41] architecture with UperNet[38] performed worse while having almost 3 times as many parameters as the model with a spatial Prithvi[19] decoder.

The importance of sequence size for each modality, batch size, and data quality suggests that this work's findings should be verified on another dataset[3], which requires

[2] https://github.com/microsoft/Swin-Transformer/blob/main/models/swin_transformer_moe.py
[3] Further datasets can be explored here: https://earthnets.github.io/

more multi-modal and multi-temporal datasets to be publicly available. Additionally, the model's scalability should be further verified and compared to other models.

7. Conclusion

This work contributes to the research of multi-modal and multi-temporal models by integrating satellite image time series into the multi-modal DOFA[41] encoder. Many ablation studies are conducted to improve the model architecture while measuring the effectiveness regarding computational complexity in FLOPs and mIoU. As a case study, the Harz region in Germany is investigated for land use and land cover, focusing on the classification of tree species and dead trees. With early fusion, the lightweight modification of the Prithvi[19] decoder and U-TAE[12] spatio-temporal architecture, the model achieves 82.19% mIoU while requiring less than half the FLOPs in comparison to the best model (Table 5.11). The best model utilizes late fusion and employs L-TAE[11] for Sentinel-1 and max-pooling for Sentinel-2 and PlanetScope. Being limited to the data quality, it achieves a mIoU of 82.37% and an overall accuracy of 93.27%.

The multi-modal and multi-temporal model without any regularization is overfitting heavily, which was reduced in the first section by applying augmentations and temporal dropout. Adding auxiliary heads for each modality allows the model to be trained for inference with limited data or modalities available. However, in this work, it did not improve the model in contrast to previous work [13]. The spatial Prithvi[19] decoder has proven to be effective in computational resources regarding the mIoU score. For efficient late fusion, it is recommended to utilize the single block Prithvi decoder due to scalability in modalities.

The ablations in fusion methods show that early fusion has a strong regulative effect and is the most effective fusion method regarding FLOPs and mIoU. However, based on a simple nearest-neighbor approach, early fusion is best suited for short time sequences available across all modalities or when computational effectiveness matters. Mid fusion is effective in parameters when utilizing all temporal data available, but needs further research in the utilization of the long-time sequence. Late fusion allows the model to learn more modality-specific features and hence offers more freedom with the cost of additional computational costs.

The spatio-temporal architecture with UBARN[9] brought slight improvements. For heavy spatial decoders and long temporal sequences per modality, the overhead to performance ratio has to be evaluated and the U-TAE[12] architecture might be better suited.

Each modality was further evaluated separately, and high-resolution imagery by PlanetScope achieved the best results, followed by multi-spectral imagery by Sentinel-2. Finally, SAR from Sentinel-1 data significantly decreased effectiveness, achieving 20% mIoU less than the model trained on PlanetScope data. Merging each set of modalities yields improvements in mIoU with the largest gains combined with PlanetScope data. This however comes with a trade-off in computational costs.

Temporal ablations showed, that temporal encoding with L-TAE[11] for long temporal sequences given in Sentinel-1 data, while temporal max-pooling performs best for the modalities PlanetScope and Sentinel-2 with short time sequences available. The model did not show any improvements from scaling up the DOFA from a ViT-B to a ViT-L, which may be due to data or architecture limitations. The DOFA initialized as a foundation model performs better than the ViT in DOFA initialized with ImageNet[7] weights, which on the other hand performs better than randomly initialized weights. Experiments with fractions of data available showed a consistently better DOFA initialized as a foundation model compared to the whole model trained from scratch.

The data showed limitations in both automated and manual annotation, such that models can be evaluated quantitatively only to a certain degree. There is randomness in training the models, such that insignificant differences in metrics have to be regarded with care. Due to the data and GPUs available with 16GB VRAM only, this work was unable to scale up the complete model in parameters. Additionally, the model may be too big for long time series, as the proposed architecture with a ViT-B is computationally too expensive in comparison to other multi-temporal models[12, 1]. Alternatively, the patch size has to be reduced to run the model on consumer graphic cards or the model downscaled to smaller configurations with a small loss in mIoU (Appendix A).

Future work can apply the proposed method to a new dataset to verify the findings or integrate methods for addressing uncertainty in data. A public framework for multi-modal and multi-temporal computer vision models would further accelerate research in this area, as the data pipelines and training become more difficult to implement and standardize with increasing complexity. Implementations of this work have to be extended to handle larger batch sizes to scale up the proposed model and train it on larger datasets.

Bibliography

[1] Guillaume Astruc et al. "Omnisat: Self-supervised modality fusion for earth observation". In: *European Conference on Computer Vision*. Springer. 2025, pp. 409–427.

[2] Favyen Bastani et al. "Satlaspretrain: A large-scale dataset for remote sensing image understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 16772–16782.

[3] Lukas Blickensdörfer et al. "National tree species mapping using Sentinel-1/2 time series and German National Forest Inventory data". In: *Remote Sensing of Environment* 304 (2024), p. 114069.

[4] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[5] Christopher F Brown et al. "Dynamic World, Near real-time global 10 m land use land cover mapping". In: *Scientific Data* 9.1 (2022), p. 251.

[6] Yezhen Cong et al. "Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 197–211.

[7] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[8] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.

[9] Iris Dumeur, Silvia Valero, and Jordi Inglada. "Self-supervised spatio-temporal representation learning of Satellite Image Time Series". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024).

[10] Anatol Garioud et al. "FLAIR: a country-scale land cover semantic segmentation dataset from multi-source optical imagery". In: *Advances in Neural Information Processing Systems* 36 (2024).

[11] Vivien Sainte Fare Garnot and Loic Landrieu. "Lightweight temporal self-attention for classifying satellite images time series". In: *Advanced Analytics and Learning on Temporal Data: 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers 6*. Springer. 2020, pp. 171–181.

[12] Vivien Sainte Fare Garnot and Loic Landrieu. "Panoptic segmentation of satellite image time series with convolutional temporal attention networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4872–4881.

[13] Vivien Sainte Fare Garnot, Loic Landrieu, and Nesrine Chehata. "Multi-modal temporal attention models for crop mapping from satellite time series". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 187 (2022), pp. 294–305.

[14] Vivien Sainte Fare Garnot et al. "Satellite image time series classification with pixel-set encoders and temporal self-attention". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12325–12334.

[15] Xin Guo et al. "Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 27672–27683.

[16] Boran Han et al. "Bridging remote sensors with multisensor geospatial foundation models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 27852–27862.

[17] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[18] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16000–16009.

[19] Johannes Jakubik et al. "Foundation Models for Generalist Geospatial Artificial Intelligence". In: *arXiv preprint arXiv:2310.18660* (2023).

[20] Wenchao Kang et al. "CFNet: A cross fusion network for joint land cover classification using optical and SAR images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 1562–1574.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[22] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[23] Thomas Lillesand, Ralph W Kiefer, and Jonathan Chipman. *Remote sensing and image interpretation*. John Wiley & Sons, 2015.

[24] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[25] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019.

[26] Marc Overbeck and Matthias Schmidt. "Modelling infestation risk of Norway spruce by Ips typographus (L.) in the Lower Saxon Harz Mountains (Germany)". In: *Forest Ecology and Management* 266 (2012), pp. 115–125.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[28] Marc Rußwurm and Marco Körner. "Self-attention for raw optical satellite time series classification". In: *ISPRS journal of photogrammetry and remote sensing* 169 (2020), pp. 421–435.

[29] Abubakar Sani-Mohammed, Wei Yao, and Marco Heurich. "Instance segmentation of standing dead trees in dense forest from aerial imagery using deep learning". In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* 6 (2022), p. 100024.

[30] Xian Sun et al. "RingMo: A remote sensing foundation model with masked image modeling". In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2022), pp. 1–22.

[31] Maofeng Tang et al. "Cross-Scale MAE: A tale of multiscale exploitation in remote sensing". In: *Advances in Neural Information Processing Systems* 36 (2024).

[32] Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. "Vits for sits: Vision transformers for satellite image time series". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 10418–10428.

[33] Gabriel Tseng et al. "Lightweight, pre-trained transformers for remote sensing timeseries". In: *arXiv preprint arXiv:2304.14065* (2023).

[34] Maarten K Van Aalst. "The impacts of climate change on the risk of natural disasters". In: *Disasters* 30.1 (2006), pp. 5–18.

[35] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[36] Mirjana Voelsen, Franz Rottensteiner, and Christian Heipke. "Transformer models for Land Cover Classification with Satellite Image Time Series". In: *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 92.5 (2024), pp. 547–568.

[37] Qi Wang et al. "A comprehensive survey of loss functions in machine learning". In: *Annals of Data Science* (2020), pp. 1–26.

[38] Tete Xiao et al. "Unified perceptual parsing for scene understanding". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 418–434.

[39] Enze Xie et al. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in neural information processing systems* 34 (2021), pp. 12077–12090.

[40] Zhenda Xie et al. "Simmim: A simple framework for masked image modeling". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 9653–9663.

[41] Zhitong Xiong et al. "Neural plasticity-inspired foundation model for observing the earth crossing modalities". In: *arXiv preprint arXiv:2403.15356* (2024).

A. Appendix

In this chapter, additional material regarding auxiliary supervision is given in Section A.1 and experiments to scale down the model in parameters are presented in Section A.2. More visualizations of the *L-TAE/Max-Pool All* configuration are stated in A.3.

A.1. Auxiliary Supervision

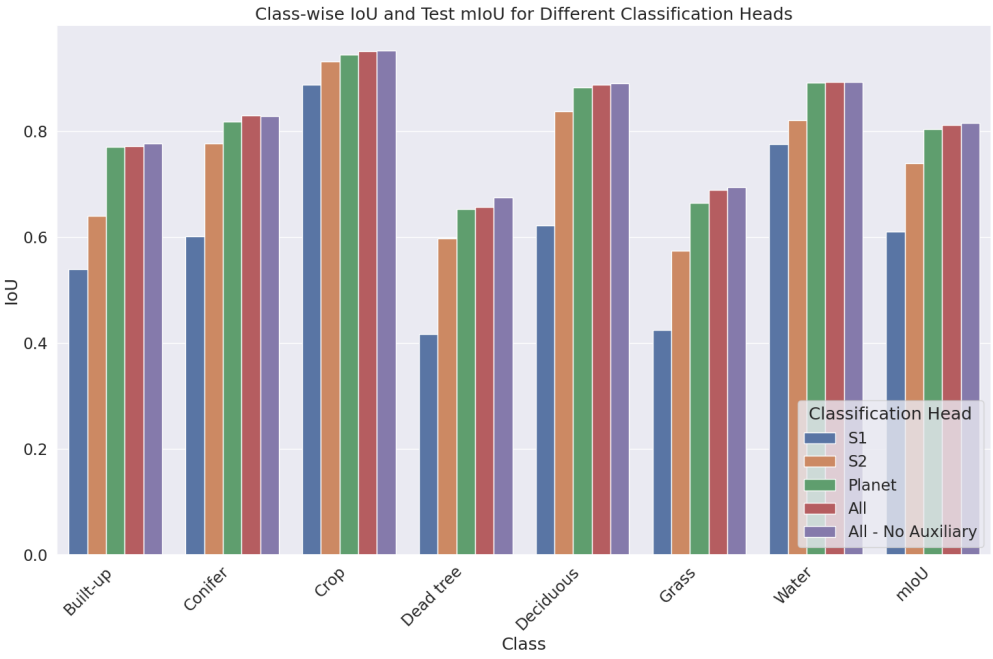
Figure A.1 shows the IoU score for each auxiliary head. The difference between each modality behaves similarly to Figure 5.3, however, a direct comparison is not possible due to different spatial decoders (UperNet[38] vs Prithvi[19]). Sentinel-1 consistently achieves the lowest IoU scores, followed by Sentinel-2 and then Planet. The IoU scores for the Sentinel-1 head do not decrease despite a lower λ^{S1}, when comparing Figure A.1b with Figure A.1a. As given in Table 5.4, the IoU scores of the main classifier get closer to the model trained without auxiliary heads instead.

The loss of each head is visualized in Figure A.2. In Figure A.2a, the loss of all auxiliary heads is dominated by Sentinel-1 and overfits most of all heads. A lower λ^{S1} set to 0.25 causes Sentinel-1 to be the lowest of all losses, as seen in comparison in Figure A.2b. Additionally, a lower λ^{S1} causes the validation and train loss for Sentinel-1 to be closer.

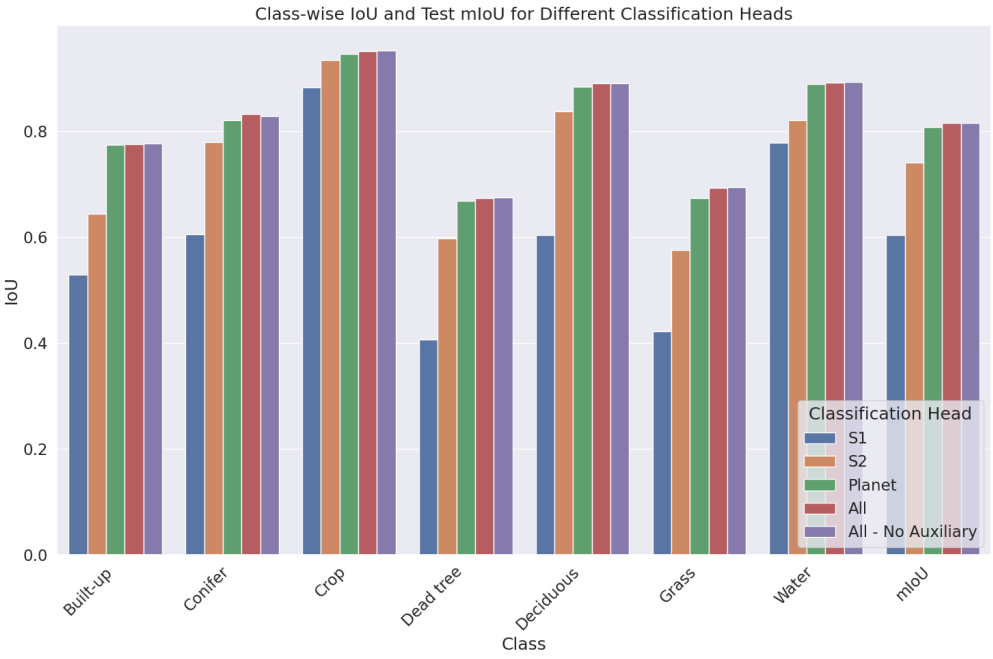
A.2. Additional Experiments

Table A.1 shows the model performance when scaling down the DOFA parameters to a more light-weight ViT[8] backbone. The configurations are adapted from [1] (detailed hyperparameters given in Table A.2 based on L-TAE for Sentinel-1 and max-pooling for Sentinel-1 and PlanetScope) and future work can further boost performance by converting ImageNet[7] weights, which are not available in PyTorch. The experiments show that more lightweight configurations achieve slightly worse mIoU than the base model, while the most lightweight configuration requires approximately 10 times less compute and parameters with a loss of 1.4% mIoU. Due to data limitations, training separate spatial encoders for each modality for each modality might be as beneficial as training a joint encoder which can be further investigated by future work. Further, the added ViT configurations regarding FLOPs and mIoU scores are visualized in Figure A.3. The small gap in mIoU between *MT S2* and *ViT-Ti* shows the need for foundation model weights for small ViT configurations.

[1] https://github.com/google-research/vision_transformer/blob/main/vit_jax/configs/models.py



(a) Auxiliary loss weight $\lambda^{S1} = 0.5$



(b) Auxiliary loss weight $\lambda^{S1} = 0.25$

Figure A.1.: IoU values per class obtained in the test set with a comparison of the main heads trained with no auxiliary head (purple) and auxiliary heads (red) along with each auxiliary head's prediction different λ^{S1} values: a) 0.5 and b) 0.25

A.3. Additional Visualizations

Configuration	Head GFLOPs	Backbone GFLOPs	Decoder GFLOPs	Total GFLOPs ↓	Head Params (M)	Backbone Params (M)	Decoder Params (M)	Total Params (M) ↓	Test mIoU (% ↑)
Foundation Model (ViT-B)	48.9	332.7	45.3	448.0	5.3	111.8	4.7	132.0	**82.37**
ImageNet (ViT-B)	48.9	332.7	45.3	448.0	5.3	111.8	4.7	132.0	81.23
Scratch (ViT-B)	48.9	332.7	45.3	448.0	5.3	111.8	4.7	132.0	80.88
Scratch (ViT-S)	24.5	84.2	5.7	125.3	2.7	35.0	1.2	41.4	80.46
Scratch (ViT-Ti)	12.3	21.6	1.4	**38.1**	1.3	12.5	0.3	**14.8**	79.46

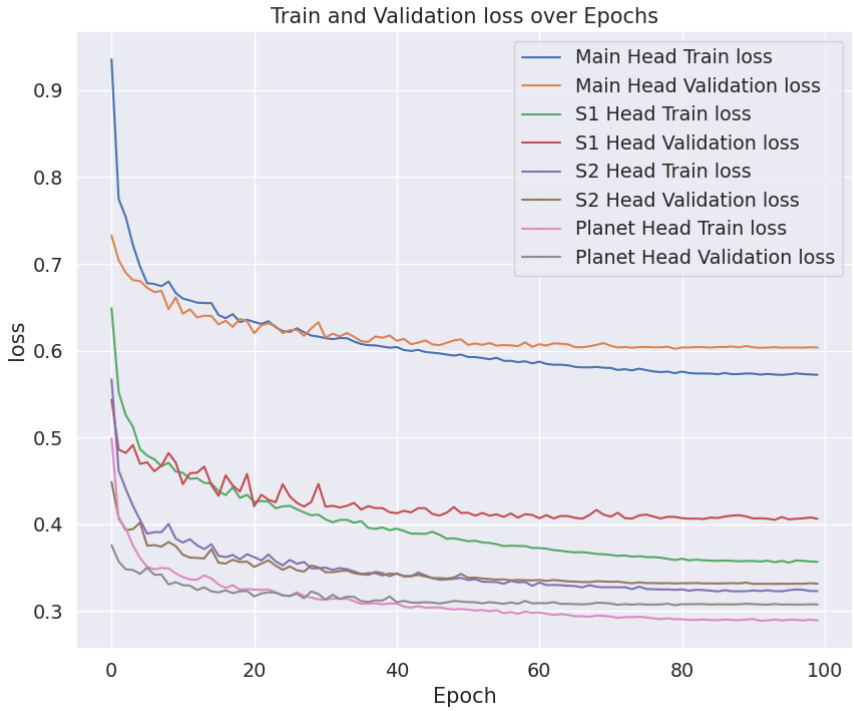
Table A.1.: Comparison of different model weight initialization in combination with different ViT[8] configurations regarding test mIoU and computational requirements. Regarding temporal feature encoding, L-TAE[11] is applied to Sentinel-1 and max-pooling to Sentinel-1 and PlanetScope

Parameter	ViT-Ti	ViT-S	ViT-B
ViT Hidden Size	192	384	768
ViT Num Layers	12	12	12
ViT Num Heads	3	6	12
Hidden Size Spatial Decoder	192	384	768
L-TAE Hidden Size	192	384	768
L-TAE d_k	4	4	8
L-TAE Num Heads	8	16	32
Hidden Size Segmentation Head	576	1152	2304

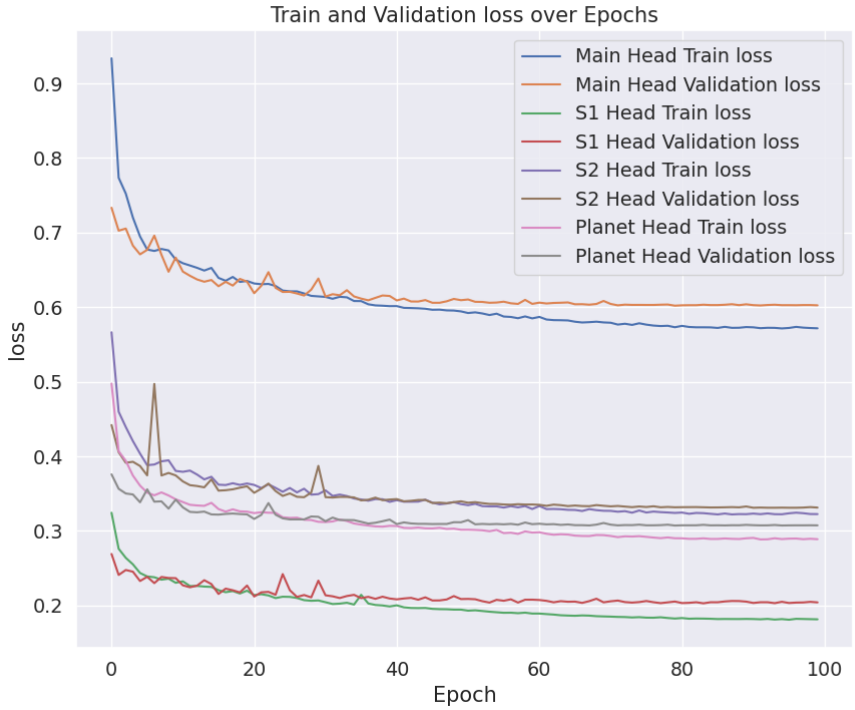
Table A.2.: ViT[8] configurations with spatial decoder, L-TAE[11] and segmentation head hyperparameters

Experiment	mIoU (%) ↑			Loss ↓		
	Train	Val	Test	Train	Val	Test
ViT-B	**82.95**	**80.60**	**80.88**	**0.5995**	**0.6055**	**0.6134**
ViT-S	82.00	80.16	80.46	0.6088	0.6098	0.6177
ViT-Ti	80.41	79.31	79.46	0.6248	0.6183	0.6268

Table A.3.: Experiment Results for mIoU and loss for train, validation, and test Sets across different ViT[8] backbone variants trained from scratch. The ViT-Ti configuration barely overfits in comparison to the larger configurations



(a) Auxiliary losses with $\lambda^{S1} = 0.5$



(b) Auxiliary losses with $\lambda^{S1} = 0.25$

Figure A.2.: Auxiliary losses obtained in the train and validation set for different auxiliary heads with different λ^{S1} values: a) 0.5 and b) 0.25

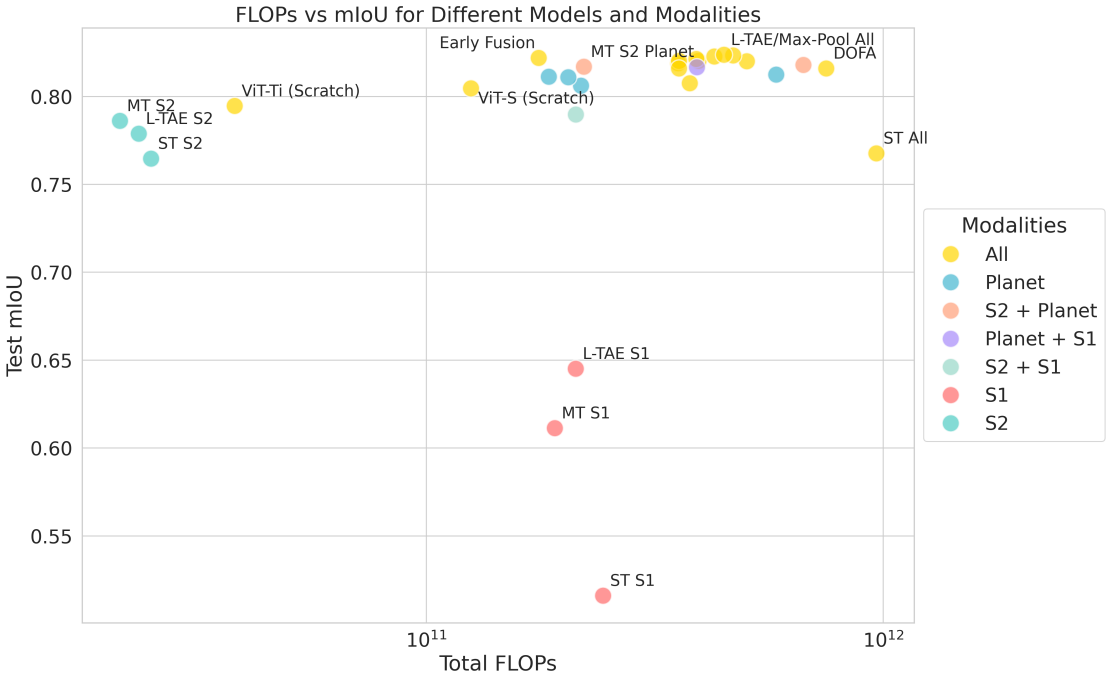


Figure A.3.: Effectiveness of modalities measured in FLOPs in relation to the test mIoU for each configuration with color codes based on different subsets of modalites. Added experiments *ViT-S* and *ViT-S* with down-scaled parameters

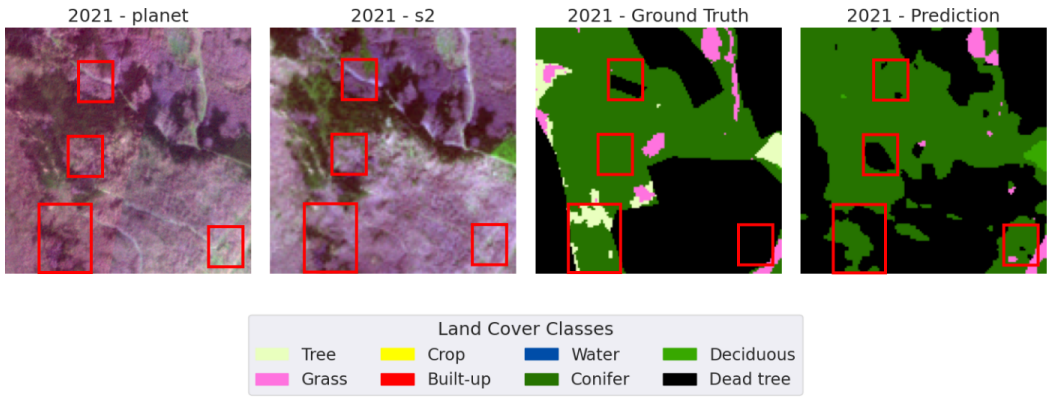


Figure A.4.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. Label inconsistencies for the class *dead tree* are shown in the red boxes

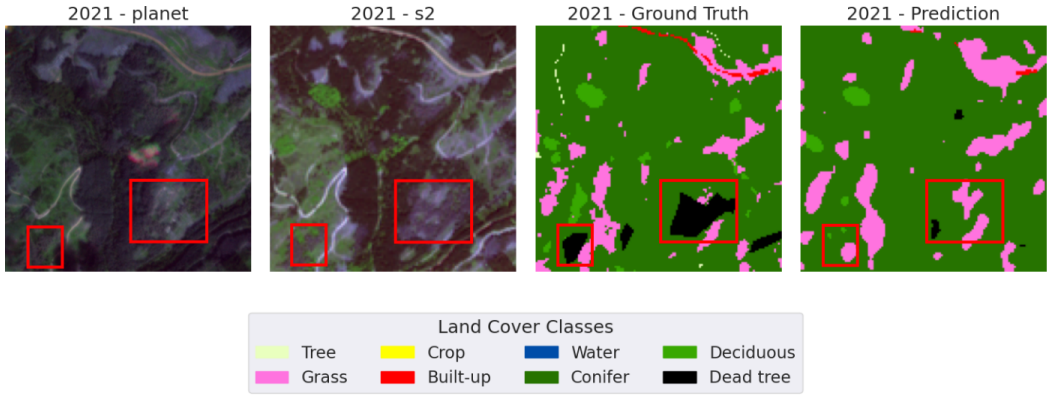


Figure A.5.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. Label inconsistencies for the class *dead tree* are shown in the red boxes

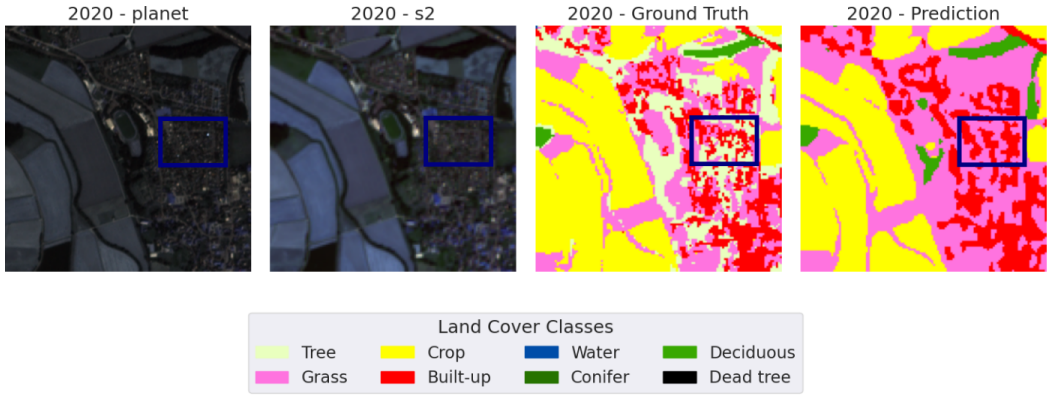


Figure A.6.: Qualitative visualization for *L-TAE/Max-Pool All* configuration. Class labels for *built-up* are more fine-granular than the predicted label map of the model, as visualized in the blue boxes