**Technische Universität Berlin**

Faculty of Electrical Engineering and Computer Science
Dept. of Computer Engineering and Microelectronics
**Remote Sensing Image Analysis Group**

# Multi-Modal Vision Transformers For Crop Mapping From Satellite Image Time Series

Master of Science in Computer Science

February, 2024

**Theresa Follath**

Matriculation Number: 374592

**Supervisor:** Prof. Dr. Begüm Demir

**Advisor:** David Mickisch

## Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, 06.02.2024

T. Follath

.......................................

*Theresa Follath*

# Acknowledgements

First, I would like to thank Prof. Dr. Begüm Demir for introducing me to the topic of Remote Sensing and her supervision of my thesis. Second, I would like to thank my advisor David Mickisch for his guidance throughout the research process.

Further, I am thankful for the team at RSiM for their support during the writing of my thesis. I would like to thank especially Barış Büyüktaş, Jakob Hackstein and Julia Henkel for their feedback.

Finally, I would like to thank my family and friends for their continuous support and encouragement.

# Abstract

Using images acquired by different satellite sensors has been shown to improve classification performance in the framework of crop mapping from satellite image time series (SITS). Current state-of-the-art architectures utilize self-attention to process the temporal dimension and convolutions for the spatial dimensions of SITS. Motivated by the success of purely attention-based architectures in crop mapping from single-modal SITS, we introduce several multi-modal, multi-temporal architectures based on the single-modal Temporo-Spatial Vision Transformer (TSViT). In order to enable it to incorporate features from multiple modalities to produce a single prediction, our architectures use either a modified token embedding or a modified temporal encoder. To assess their effectiveness, we compare them with each other as well as to single-modal baselines and two existing architectures from the literature. Experiments are conducted on the EOekoLand dataset for multi-modal multi-temporal crop mapping. The data contains two optical modalities of different spatial, spectral and temporal resolutions as well as a third modality consisting of synthetic aperture radar (SAR) images. Results show that our proposed architectures achieve clear improvements over single-modal baselines. Moreover, we find that directly deriving tokens from the fused input achieves better results than using a separate token embedding for each modality and fuse later in the TSViT architecture. Among proposed architectures with separate token embeddings for each modality, those with modality-specific temporal encoders outperform architectures that use a single temporal encoder. In a detailed ablation study we investigate the effect of hyperparameters on the performance of our most successful architectures. Finally, it is shown that the proposed architectures outperform two other crop mapping architectures from the literature which further affirms their effectiveness for multi-modal crop mapping.

# Zusammenfassung

Die automatisierte Kartierung von Kulturpflanzen mithilfe von Deep Learning basierend auf Satellitenbild-Zeitreihen trägt dazu bei, die globale Verfügbarkeit von Nahrungsmitteln akkurat einzuschätzen. Vergangene Studien haben gezeigt, dass die Nutzung von Satellitenbildern, die mit unterschiedlichen Verfahren erzeugt wurden, die automatisierte Kartierung von Kulturpflanzen verbessert. Der kürzlich vorgestellte Temporo-Spatial Vision Transformer (TSViT), welcher ausschließlich self-attention für die Klassifizierung von Kulturpflanzen von rein optischen Satelliten-Zeitreihen verwendet, konnte andere aktuelle Architekturen übertreffen. In diesem Zusammenhang erforscht diese Arbeit mehrere Möglichkeiten den TSViT für die Fusion multi-modaler Satellitenbild-Zeitreihen für eine verbesserte Klassifizierung anzupassen. Die vorgestellten Architekturen werden mit der Leistung der TSViT Architektur verglichen, wenn diese auf den einzelnen Modalitäten individuell trainiert wird. Außerdem werden die Architekturen untereinander sowie mit zwei weiteren Architekturen aus der Literatur verglichen. Für die Auswertung wird der EOekoLand Datensatz verwendet, welcher zwei multispektrale, optische Modalitäten sowie eine Modalität bestehend aus Radardaten beinhaltet. Satellitenbild-Zeitreihen sind im Rahmen dieses Datensets für zwei verschiedene geographische Regionen verfügbar. Es beinhaltet außerdem Referenzdaten für zwei bzw drei verschiedene Jahre für die beiden Regionen. Die Ergebnisse zeigen große Verbesserungen der vorgestellten Architekturen im Vergleich zum auf den einzelnen Modalitäten trainierten TSViT. Des Weiteren kann gezeigt werden, dass sie bessere Ergebnisse als die beiden der Literatur entnommenen Architekturen erzielen. In einer detailierten Ablationsstudie wird außerdem der Einfluss verschiedener Hyperparameter auf die Leistung der vorgestellten Archtiekturen ausgewertet.

# Contents

# List of Acronyms

| | |
|---|---|
| CA | Cross Attention |
| CNN | Convolutional Neural Network |
| HSI | Hyperspectral Imaging |
| IACS | Integrated Administration and Control System |
| LiDAR | Light Detection and Ranging |
| LSTM | Long Short-Term Memory |
| L-TAE | Lightweight Temporal Attention Encoder |
| MA | Mean Accuracy |
| mIoU | Mean Intersection over Union |
| MLP | Multi-layer Perceptron |
| MM TSViT | Multi-modal Temporo-Spatial Vision Transformer |
| MODIS | Moderate Resolution Imaging Spectroradiometer |
| MTC | Modality Token Concatenation |
| NDVI | Normalized Difference Vegetation Index |
| NIR | Near Infrared |
| NLP | Natural Language Processing |
| OA | Overall Accuracy |
| RBF | Radial Basis Function |
| ReLU | Rectified Linear Unit |
| RGB | Red Green Blue |
| RNN | Recurrent Neural Network |
| RVI | Radar Vegetation Index |
| SAR | Synthetic Aperture Radar |
| SCT | Synchronized Class Token |
| SITS | Satellite Image Time Series |
| SM TSViT | Single-modal Temporo-Spatial Vision Transformer |
| SVM | Support Vector Machine |
| SWIR | Short Wave Infrared |
| TSViT | Temporo-Spatial Vision Transformer |
| VH | Vertical transmit and horizontal receive |
| VIIRS | Visible Infrared Imaging Radiometer Suite |
| ViT | Vision Transformer |
| VV | Vertical transmit and vertical receive |

# List of Figures

# List of Tables

# 1 Introduction

Monitoring the surface of the earth is vital in the light of current challenges related to climate change such as extreme weather events or droughts [18]. Moreover, the increase in global population leads to a higher demand for agricultural products [56]. These conditions threaten food security all around the globe. The United Nations have formulated the goal to end hunger by 2030 as the second of their sustainable development goals [36]. Therefore, policies for increasing agricultural productivity especially in poorer countries are developed. To evaluate the effectiveness of these policies, crop mapping from remote sensing imagery is a valuable strategy to monitor the state of crops and food availability [39].

Crops are mostly characterized by their growth cycle over the year. They differ in the time of their growth stages such as sowing, flowering, harvest and senescence. The crops can be conveniently monitored with spectral reflectance values obtained by satellites. Based on these, a simple measure for a plant's state or "greenness" is the Normalized Difference Vegetation Index (NDVI) which is calculated from the Near Infrared (NIR) and Red reflectance values:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{1.1}$$

This simple measure was successfully used to find differences in growth patterns between crop types as shown in Figure 1.1 and even to classify crop types [58], which shows that satellite image time series (SITS) are a valuable means to produce accurate annual crop maps.

Due to the decreasing cost of sensors and satellite missions, the availability and temporal resolution of satellite images has continuously increased over the past years [43]. With the missions Sentinel-1 and Sentinel-2 ESA's Copernicus program provides synthetic aperture radar (SAR) and optical satellite images free of charge. Due to a small revisit time of 6-10 days and a spatial resolution of 10-60m per pixel, both missions produce a large amount of data every day [1]. The high-quality and freely available satellite image resources have sparked a high interest in deep learning applications to process these images.

Operating on SITS poses new challenges for machine learning models. Due to the high temporal and spatial resolution, processing SITS requires large storage and computational resources. With increasing power of modern graphics processing units (GPUs) and decreasing cost of storage, the computational capabilities of modern hardware have increased over recent years. This allows processing of high resolution SITS. Due to previous limitations of available hardware capabilities, architectures that incorporate the temporal dimension for crop mapping are a recent field of research. Inspiration is mostly drawn from research in the field of computer vision as video frames and SITS have many properties in common. Current state-of-the-art models include recurrent neural networks (RNN) [44, 57], convolutional neural networks (CNN) [38, 20] and attention-based models [13, 51].

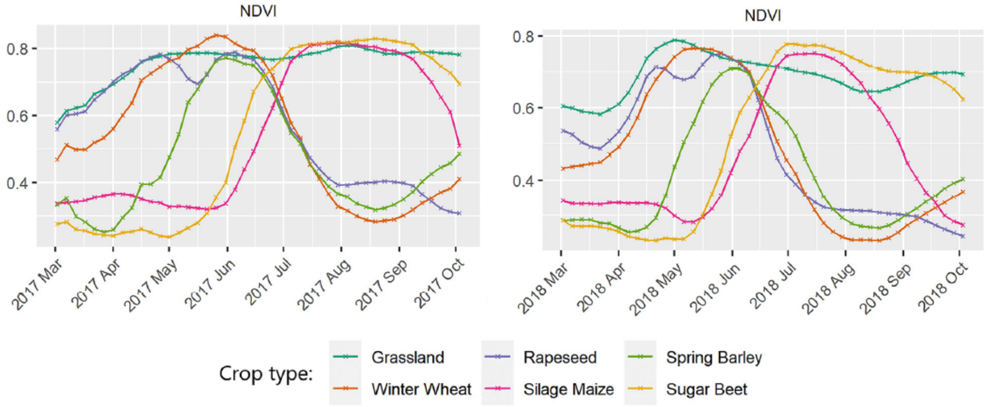


Figure 1.1: Differences in NDVI calculated from spectral reflectance values between crop types. A high NDVI corresponds to a high greenness and therefore to a high growth of the crop. The differences between crops in their growth pattern over one season can be clearly seen [3].

Although most existing literature focuses on crop mapping from Sentinel-2 or Landsat-8 time series [20, 38, 44, 51], multi-spectral data has the disadvantage that it can be obstructed by clouds [21]. Other imaging processes such as radar use larger wavelengths which enable them to measure reflectances through clouds. These images can be incorporated into the classification to provide additional information [3, 55]. Moreover, combining data from different modalities such as optical and radar data can also provide complementary information and increase the performance compared to single-modal data. Furthermore, commercial products such as ”Planet Fusion” [25] or ”Planet Scope” [26] by Planet Labs with a high spatial and temporal resolution contribute valuable information as well. With these products Planet Labs offers ready-to-use databases that have been successfully used for tillage practice mapping [33] and for crop mapping from time series by themselves [46] or in combination with Sentinel-1 and Sentinel-2 imagery in a multi-modal context [24].

Learning from multi-modal data requires dedicated fusion models that are able to extract information from multiple modalities and fuse them to make a single prediction. Current state-of-the-art fusion strategies apply CNNs to the spatial dimensions and attention-based methods to the temporal dimension [15] or do not process the spatial dimension at all [54]. However, due to recent success of purely attention-based methods in Semantic Segmentation such as Segmenter [48] and Swin Transformer [31], it seems beneficial to process the spatial dimension with a Transformer encoder as well. This has been done for crop mapping from Sentinel-2 data with the Temporo-Spatial Vision Transformer (TSViT) architecture [51] with great success but has yet to be explored for multi-modal fusion. In this thesis, we will therefore investigate the effectiveness of several multi-modal fusion methods in the TSViT architecture. To this end, we compare their performance with each other and to single-modal baselines.

This thesis is separated into seven chapters. In Chapter 2, we introduce relevant background knowledge regarding both Remote Sensing data and the Transformer architecture based on which we develop multi-modal fusion architectures. Next, in Chapter 3, we present previous work in the fields of crop mapping, both from single and multiple modalities, and explore existing multi-modal fusion architectures that rely on Transformers. Chapter 4 formally introduces the problem statement and our proposed methods in detail. This includes motivation and advantages of each method as well as computational complexity and possible disadvantages. In Chapter 5 we present the data set and experimental setup used for our experiments. Our experimental results are presented in Chapter 6 along with further ablation studies and their analysis. Finally, Chapter 7 provides our conclusion and an outlook on future work.

# 2 Background

This chapter briefly introduces the types of satellite data used in this thesis. Specifically, we introduce the types of sensors used to acquire that data. Moreover, we summarize the Transformer architecture that is adapted for multi-modal fusion in this thesis.

## 2.1 Remote Sensing Fundamentals

Remote sensing is the acquisition of information about an object or scene from a great distance with a satellite or a plane and specifically without physical contact with the object. There is active and passive remote sensing which capture different aspects of the observed scene and are often utilized as complementing modalities.

### Passive Sensors

Passive sensors do not emit any radiation of their own but capture light from a separate light source, usually sunlight that is reflected back to space from the surface of the earth. The sensors measure light in different wavelength ranges, referred to as spectral bands. These typically range form the visible spectrum over near infrared (NIR) to short wave infrared (SWIR). The measured data is referred to as optical data and more specifically multispectral data if few wide bands are captured or hyperspectral if several narrow bands are captured.

Table 2.1: Sentinel-2 bands with central wavelengths, spatial resolution and interpretation [9].

| Band | Resolution | Central Wavelength | Description |
|---|---|---|---|
| B1 | 60 m | 443 nm | Ultra Blue (Coastal and Aerosol) |
| B2 | 10 m | 490 nm | Blue |
| B3 | 10 m | 560 nm | Green |
| B4 | 10 m | 665 nm | Red |
| B5 | 20 m | 705 nm | Visible and Near Infrared (VNIR) |
| B6 | 20 m | 740 nm | Visible and Near Infrared (VNIR) |
| B7 | 20 m | 783 nm | Visible and Near Infrared (VNIR) |
| B8 | 10 m | 842 nm | Visible and Near Infrared (VNIR) |
| B8a | 20 m | 865 nm | Visible and Near Infrared (VNIR) |
| B9 | 60 m | 940 nm | Short Wave Infrared (SWIR) |
| B10 | 60 m | 1375 nm | Short Wave Infrared (SWIR) |
| B11 | 20 m | 1610 nm | Short Wave Infrared (SWIR) |
| B12 | 20 m | 2190 nm | Short Wave Infrared (SWIR) |

The most popular missions that capture optical data are NASA's Landsat and MODIS programs as well as the Sentinel-2 mission as part of ESA's Copernicus Programme [8] whose data is used in this thesis. The Sentinel-2 mission consists of two satellites, Sentinel-2A and Sentinel-2B, both in a sun-synchronous orbit with a 180 degree phase difference between each other. Both satellites have a revisit time of at most 10 days. Due to the near-polar orbit, swaths are overlapping at locations of higher latitude which results in multiple observations of these locations within 10 days. However, they are observed with different viewing angles and can therefore not be directly used with each other. Using two satellites achieves a total revisit time of 5 days. The sun-synchronous orbit ensures that the local time and therefore the illumination angle of a point on the surface is nearly the same every time the satellite passes over it. This minimizes differences in shadows and ensures similar light conditions which is especially important when analyzing time series data. The Sentinel-2 satellites capture reflected light in 13 different bands which are listed in Table 2.1. The first three bands, red, green and blue (RGB), can be used to produce a so called true color image, which show the surface of the earth as closely as possible to human perception. Moreover, false color images can be created by selecting different bands as RGB channels of the output image. With NIR, red and green, vegetation is shown in red for example. An example for both types of images for each season is shown in Figure 2.1

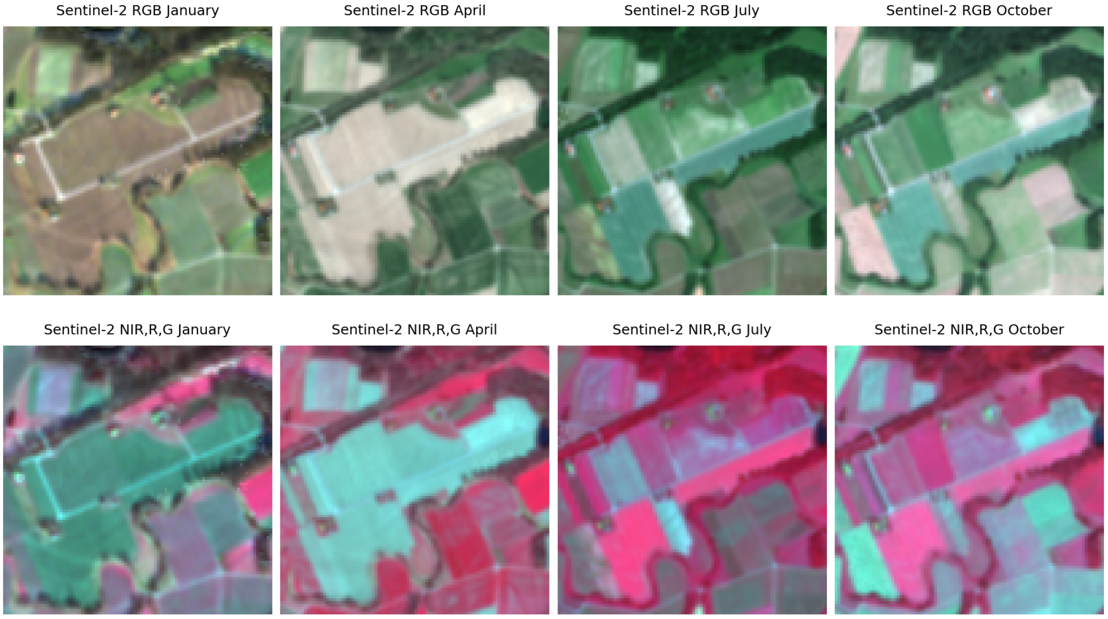


Figure 2.1: Sample images from Sentinel-2 data of our evaluation dataset for each season as a true color image in the first row and as a false color image which shows vegetation in red in the second row.

The raw Sentinel-2 images can be skewed by atmospheric conditions. To mitigate this effect, atmospheric correction is applied. Moreover, cloud and water vapor masks are provided to indicate which parts of the image are obstructed.

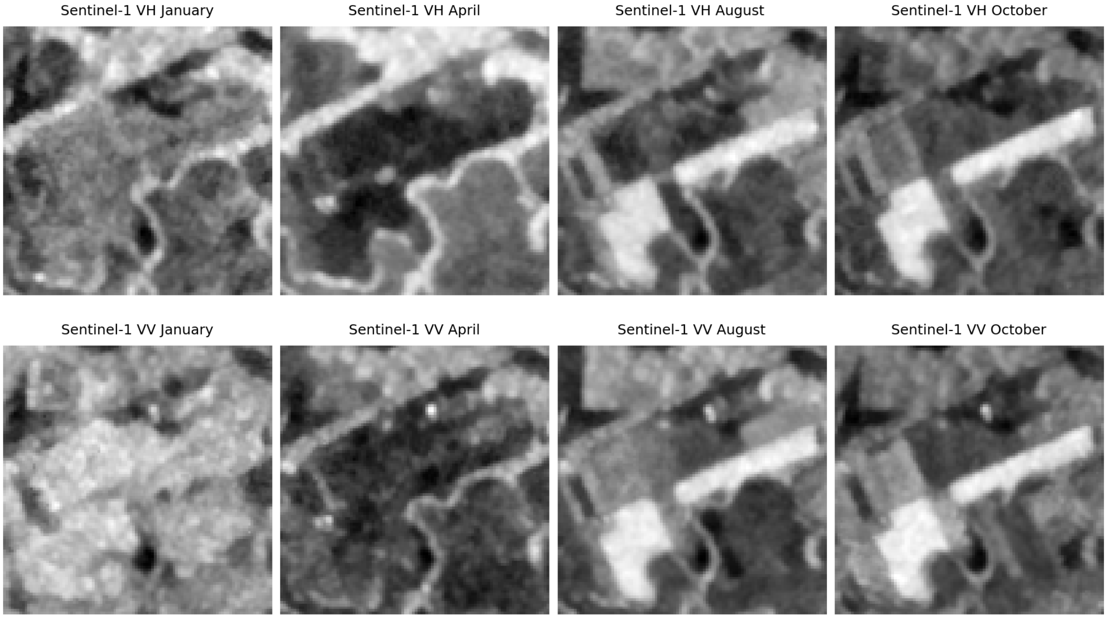


Figure 2.2: Sample images from Sentinel-1 data of our evaluation dataset with VV and VH polarization for each season.

### Active Sensors

Active sensors such as light detection and ranging (LiDAR) and radar systems emit radiation and capture the reflectance. This principle is used by the Sentinel-1 mission of ESA. The two Sentinel-1 satellites use synthetic aperture radar (SAR) with a revisit time of 6 days [1]. The wavelength used by SAR sensors is typically longer than the those measured by passive sensors. Sentinel-1 uses a wavelength of 5.5 cm. The longer wavelength allows the rays to pass through clouds and produce images on cloudy days which is something Sentinel-2 is not able to provide. Moreover, active sensors do not rely on an external energy source such as the sun. Therefore, images can also be acquired at night.

The spatial resolution of the captured image is directly related to the ratio of the wavelength and the length of the antenna. With 6 cm wavelength the antenna would have to be impractically long to get a reasonable spatial resolution. Therefore, a larger antenna is simulated. This is achieved by taking multiple acquisitions one after the other. Due to the time delay between acquisitions the antenna has moved a little further. This allows combining multiple observations to a single high resolution image and to simulate a larger antenna. The technique is called synthetic aperture radar.

SAR allows capturing structure and elevation on the ground. Signals are emitted with a specific magnitude in either horizontal or vertical polarization. Polarization refers to the orientation of the wave's oscillation in the coordinate system of the antenna. The sensor measures the magnitude of the reflected signal in either the same or a different polarization. Vertical emission and measurement is referred to as VV polarization, whereas vertical emission and horizontal measurement is called VH polarization. The reflected signal is referred to as backscatter. Depending on the surface that the emitted signal interacts with, the backscatter varies in intensity. If the surface is perfectly flat, all rays are reflected in the same direction. In the rare case that the surface is perpendicular to the incoming signal, all reflected rays are captured by the sensor and the surface appears bright. Otherwise, the intensity of the captured signal is low and the surface appears dark. Thus, the incidence angle affects the reflected signal. For instance, this angle is different with ascending and descending orbit of the satellite which means that the images captured of the same area but a different orbit can be very different. Sentinel-2 satellites do not suffer from this issue as the ascending part of the orbit is always on the night side of the planet.

However, the surface is usually not perfectly flat. Then, the signal is reflected in various directions and only a fraction is captured by the sensor. The measured intensity gives information about the structure of the land surface. Example images are shown in Figure 2.2 with an image for both VV and VH polarization for each season. The images show the same area as the Sentinel-2 images in the previous section.

### Planet Fusion Product

In addition to observations from a single sensor, there are also enhanced products that combine images from multiple sensors to improve their quality. The Planet Fusion product by Planet Labs, for example, is composed from Landsat-8, Sentinel-2 and Planet Scope images. The latter are acquired by Planet Labs with a constellation of more than 100 Dove satellites in a sun-synchronous orbit. They provide daily images in eight bands Red, Green, Blue, NIR, Red Edge, Yellow, Green I, and Coastal Blue in a spatial resolution of 3m per pixel [26]. Only the first four bands are used in the Planet Fusion product. Planet Labs use the FORCE data cube [10] to fuse Landsat-8 and Sentinel-2 observations. Then, the fused RGB and NIR bands of Landsat and Sentinel images are used for filling gaps due to cloud cover in PlanetScope observations while also utilizing Moderate Resolution Imaging Spectroradiometer (MODIS) and Visible Infrared Imaging Radiometer Suite (VIIRS) data for reference sampling and calibration [25]. The resulting Planet Fusion images contain daily observations with a spatial resolution of 3m per pixel and contain Red, Green, Blue and NIR bands as shown in Table 2.2.

Table 2.2: Planet Fusion bands with wavelength range and interpretation. All bands have a spatial resolution of 3m per pixel and are available daily [25].

| Band | Wavelength Range | Description |
|---|---|---|
| Band 1 | 450 - 510 nm | Blue band |
| Band 2 | 530 - 590 nm | Green band |
| Band 3 | 640 - 670 nm | Red band |
| Band 4 | 850 - 880 nm | NIR band |

The high spatial resolution makes the product suitable for detecting small structures and borders. Planet Scope, for example, has been used for sharpening Sentinel-2 images [24] for crop mapping with a Random Forest classifier. The regular, cloud free images of Planet Fusion also allow for Crop Classification from SITS [46]. Moreover, the high temporal resolution enable the detection of rapid changes such as the time of harvest or the ponding of rice fields [4].

## 2.2 Transformers

In 2017, the Transformer architecture was introduced by Vaswani et al. [59] initially for the domain of Natural Language Processing. It was soon adapted by Dosovitskiy et al. [7] for image classficatiaton. In the following, there were also Transformer architectures developed for SITS classification [43, 61, 51]. The following chapter first introduces the Transformer architecture with its core, the attention mechanism, then Dosovitskiy et al.'s Vision Transformer architecture (ViT) [7] and finally the Temporo-Spatial Vision Transformer (TSViT) by Tarasiou et al. [51], a Transformer architecture for pixel-based classification on SITS that is used in this thesis.

### Transformer Encoder

The original Transformer architecture was introduced for Natural Language Processing (NLP) tasks such as machine translation where it replaced recurrence-based models such as RNNs and LSTMs. It consists of an encoder and decoder. The encoder is used to map the input sequence of words to a sequence of continuous representations, called tokens. Tokens are vectors, derived from one element in the input sequence. For NLP tasks a token is usually derived from a word. The decoder does the same for the target sequence and then applies cross-attention to represent each element in the target token sequence by a weighted combination of the elements in the input token sequence. The weights encode how relevant each input token is to an element of the target sequence.

The Transformer encoder architecture is not bound to NLP problems but can be used to process arbitrary sequences. Consequently, it has found several applications in other fields such as image classification. The input to the Transformer encoder is always a list of tokens, i.e. vectors of dimension $d$. These are derived from the input sequence with a domain specific embedding such as word embeddings in NLP tasks. Unlike recurrent or convolutional architectures, the attention-based Transformer encoder is not able to retain the order of the elements in the sequence. Because this context is important for the meaning of the sentence, a positional encoding is added to each token.

The core of the Transformer encoder architecture is the Scaled dot-Product Attention module. It computes pairwise attention between each pair of tokens in a sequence of length $L$. For this, a key, value and query vector are derived from each token by linear projection. The output of this module for each token is then a weighted sum of all derived value embeddings.

The attention weight between two tokens is the dot product between the key of one token and the query of the other. As such it can be interpreted as a similarity measure of both vectors, i.e. the weight is high if the query is similar to the key. For the $j$-th and $i$-th tokens with derived query $q_j$, key $k_i$, the attention weight $a_{ij}$ is computed as follows:

$$a_{ij} = softmax(\frac{q_j \cdot k_i}{\sqrt{d}}) \tag{2.1}$$

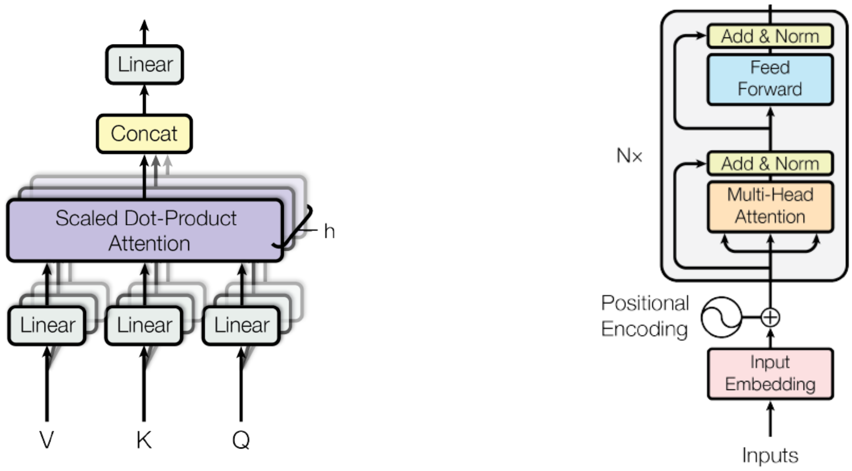


Figure 2.3: Left: Multi-head attention module as introduced in [59]. Each of the $h$ heads computes pairwise attention from the input matrices $V$, $K$ and $Q$. Right: Transformer encoder with $N$ layers each consisting of multi-head attention and an MLP as a feed-forward module [59].

Specifically, $a_{ij}$ defines how relevant token $i$ is to token $j$. The weight then determines how much the value $v_i$ of token $i$ influences the output for token $j$ in the weighted sum:

$$out_j = \sum_{i=1}^{L} a_{ij} v_i \tag{2.2}$$

If token $i$ is relevant to token $j$, the corresponding value embedding $v_i$ contributes a lot to the output token $out_j$. The final new token $out_j$ is thus a weighted sum of the values from all other tokens where the weights are computed from the query of this token and the keys of all the other tokens.

To make computation faster, key, value and query for all tokens are stored in matrices K, V and Q, respectively (see Equation 2.3). As mentioned before, these are derived from the input token sequence $X \in \mathbb{R}^{N \times d}$ by linear projection with $K = XW_K$, $Q = XW_Q$ and $V = XW_V$ where $W_K$, $W_Q$ and $W_V$ are learned weight matrices and form the so-called attention head.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2.3}$$

By using multiple attention heads as shown in Figure 2.3, each can learn different key, query, value representations of the input. Thereby, diversity is increased and the model is more reliable. The resulting tokens of each attention head are concatenated and linearly projected back to the token dimension $d$.

A Transformer encoder layer consists of a Multi-Head Attention and a Feed Forward module, each with a residual connection followed by a normalization step. The output of each layer is again a sequence of tokens of the same dimensions as the input sequence. In total, a Transformer encoder consists of $N$ such layers as shown on the right in Figure 2.3.

The output sequence is of the same shape as the input sequence, but the tokens are weighted based on their relevance to the task. Depending on the task, it can be processed further. In the standard NLP task of language translation, for example, this sequence is passed to the decoder for the translation of the target sequence.

### Vision Transformer

The Vision Transformer architecture (ViT) [7] shown in Figure 2.4 utilizes the previously introduced Transformer encoder to perform image classification. The tokens are derived from image patches instead of words. The image is divided into $16 \times 16$ patches which are projected to the token dimension $d \in \mathbb{N}$ and put into sequence. Because the spatial location of each patch in the original image is lost in this process, a position encoding is added to each token to keep this spatial information. As first introduced in [5], a class token is prepended to the sequence before passing it to the encoder. The encoder computes self-attention, which enhances tokens that are more relevant to the classification and condenses this information into the class token. This class token is retrieved afterwards and projected to class probabilities.

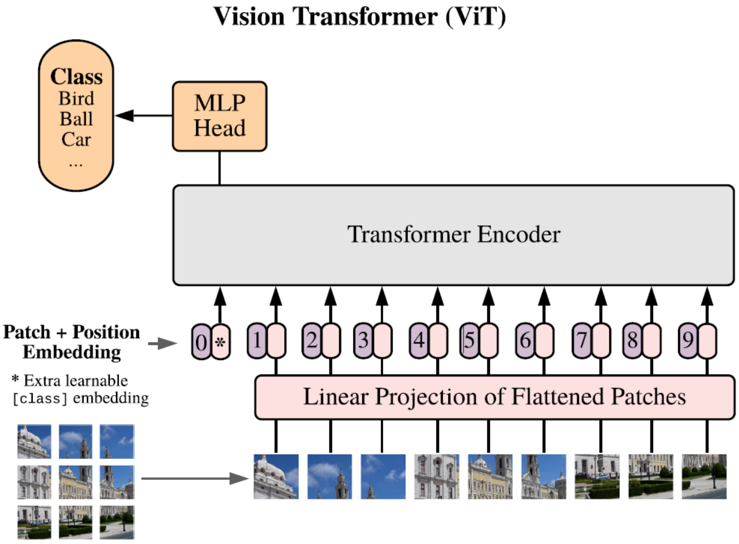

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 2.4: Vision Transformer architecture as introduced by Dosovitskiy et al. [7].

Formally, the token sequence is derived from an input image $X \in \mathbb{R}^{H \times W \times C}$, with $H \in \mathbb{N}$ being the height of the image, $W \in \mathbb{N}$ the width and $C \in \mathbb{N}$ the number of channels. The image is split in the spatial dimension into patches of size $h \times w \times C$. This results in $N = \frac{H}{h} \cdot \frac{W}{w}$ patches. These patches are consequently flattened and linearly projected to the token dimension $d$. A learnable class token $z_0^0 \in \mathbb{R}^d$ is prepended to the sequence and will be used to predict class probablities. During training the model learns to map the most important features from the $N$ feature tokens to the class token. The resulting token sequence $z_0 \in \mathbb{R}^{(1+N) \times d}$ is passed to the first layer of the Transformer encoder. In the attention module, the tokens are modified according to their relevance to each other. The features that are most relevant for the image classification task are mapped to the class token. Each layer first normalizes the input and computes pairwise attention between the tokens as described in the previous section. After further normalization the output of the multi-head attention module is passed through a Multi-Layer Perceptron (MLP).

Finally, the class token is retrieved from the output sequence of the final Transformer layer and passed through an MLP to map to class probabilities.

### Temporo-Spatial Vision Transformer

The Temporo-Spatial Vision Transformer (TSViT) is based on the Vision Transformer introduced in the previous section but has been modified to be applied to SITS for crop mapping. In order to do so its core elements include not only a spatial but also a temporal encoder (see Figure 2.5, left).

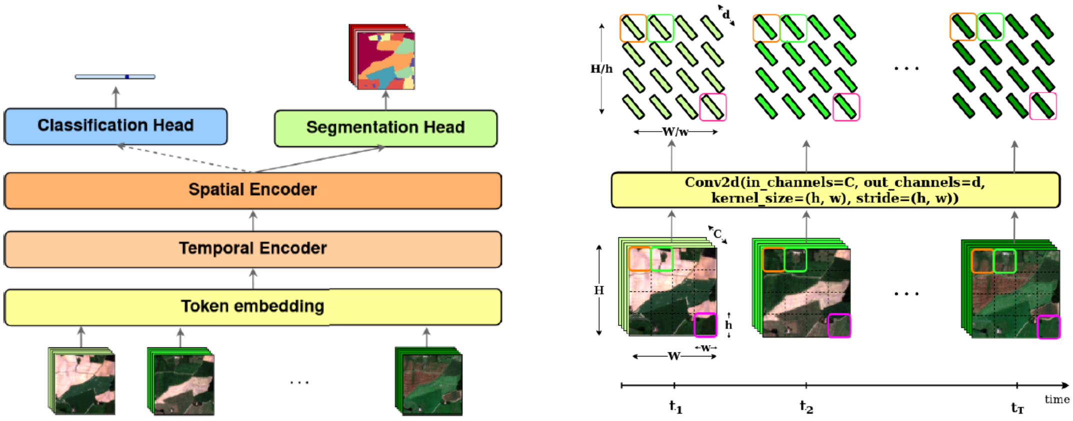
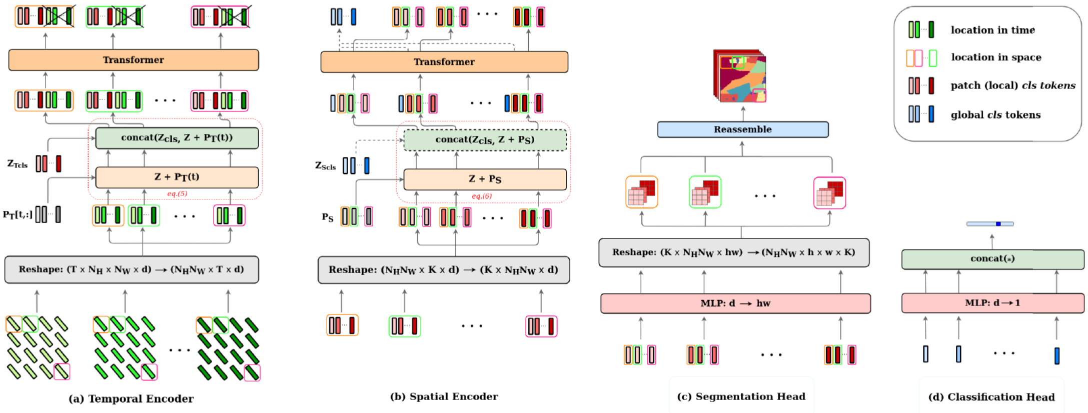


Figure 2.5: Left: Overview of the TSViT architecture [51]. Right: Step 1 - Tokenization: Splitting the image of each time point into patches and projecting them to token dimension $d$.

Let $X \in \mathbb{R}^{T \times C \times H \times W}$ be the input SITS. The first step is to produce a token embedding. Therefore, the SITS is split into $N_T \times N_H \times N_W$ patches of size $t \times h \times w$ with $w, h, t \in \mathbb{N}$ and $N_T = \frac{T}{t}, N_W = \frac{W}{w}, N_H = \frac{H}{h}$. Subsequently, the patches are flattened and projected to token dimension $d$. The authors argue that it is best to choose $t = 1$, i.e. projecting each time point to an individual token, because it simplifies the temporal position encoding and is shown to be most effective in their experimental results. Generally, the smaller $h, w$ and $t$ are chosen, the better the expected performance but also the more tokens are created and consequently the higher the computational complexity. The output of this step is a token time series for each spatial location: $Z_T \in \mathbb{R}^{N_H N_W \times N_T \times d}$.

If $t = 1$ is chosen, the token embedding can be done with a 2D convolutional layer with stride and kernel size of $w \times h$, $C$ input and $d$ output channels (Figure 2.5, right).

Before passing the sequence to the temporal encoder, a temporal position encoding $P_T[t,:$



(a) Temporal Encoder

(b) Spatial Encoder

(c) Segmentation Head

(d) Classification Head

Figure 2.6: (a) Step 2 - Temporal encoder: Adding temporal position encoding and $K$ local class tokens before passing sequence through temporal Transformer encoder. Keep only local class tokens after temporal encoder while discarding feature tokens. (b) Step 3 - Spatial encoder: add spatial position encoding, prepend global class tokens and pass spatial sequence of local class tokens through spatial encoder. (c) Step 4 - Segmentation head: project local class tokens for each spatial location back to pixel patches and reassemble original spatial image extent. This gives the class probabilities for each pixel. (d) Step 5 - Classification head: map global class tokens to class probabilities with MLP. Image taken from [51], here shown for $N_T = T$.

$] \in \mathbb{R}^{N_T \times d}$ is added. This is derived from each day of the year by projecting the corresponding 366 dimensional one hot encoding vector to the token dimension $d$. Then, $K$ class tokens $Z_{Tcls} \in \mathbb{R}^{K \times d}$ are prepended to each temporal token sequence with $K$ being the amount of classes in the dataset. The resulting token sequence $Z_T^0 \in \mathbb{R}^{N_H N_W \times (K+N_T) \times d}$ is put into the temporal Transformer encoder which is built as introduced in Section 2.2.

In the temporal Transformer, the temporal feature tokens influence the class tokens and during training, the model learns which time steps are relevant to each class. This results in the modified token sequences $Z_T^L \in \mathbb{R}^{N_H N_W \times (K+N_T) \times d}$ after the final layer $L$. Out of each of the $N_H N_W$ token sequences only the $K$ local class tokens are kept whereas the temporal feature tokens are discarded.

The local class tokens are transposed to form a spatial token sequence for each class: $Z_S \in \mathbb{R}^{K \times N_H N_W \times d}$. A learned spatial position encoding $P_S \in \mathbb{R}^{N_H N_W \times d}$ is added to each sequence. If image classification is performed as well, a single global class token $Z_{Scls} \in \mathbb{R}^d$ is prepended to the spatial token sequence of each class, resulting in the sequence $Z_S^0 \in \mathbb{R}^{K \times (1+N_H N_W) \times d}$. This is passed through a spatial Transformer encoder which computes spatial attention between local class tokens and learns which spatial locations are relevant for the image classification task by weighting the influence of the local class tokens to the global class token $Z_{Scls}$ in the attention mechanism.

After the spatial encoder, the resulting sequence $Z_S^L \in \mathbb{R}^{K \times (1+N_H N_W) \times d}$ is split into local and global class tokens $Z_{Slocal} \in \mathbb{R}^{K \times N_H N_W \times d}$ and $Z_{Sglobal} \in \mathbb{R}^{K \times d}$. The local class tokens are projected back from the token dimension $d$ to their initial spatial resolution of $h \times w$. This allows reassembling the initial extents $W \times H$ of the image for each of the $K$ classes. The reassembled image $I_{out} \in \mathbb{R}^{H \times W \times K}$ can be interpreted as a class probability map. The global class tokens for each class are projected from token dimension $K \times d$ to $K$ and are also interpreted as class probabilities for the image classification task. In case the task is pixel-based classification as in this thesis, the global class token is not added before passing the local class token sequences to the spatial encoder. Then, its input and output are sequences of shape $\mathbb{R}^{K \times N_H N_W \times d}$.

The TSViT architecture is a straightforward adaptation of the ViT architecture to i) process the additional temporal dimension of the data and to ii) the pixel-based classification task of crop mapping. The inherent modality agnostic properties of the Transformer encoder make it suitable to implement multi-modal fusion approaches. Moreover, the TSViT is a state-of-the-art model for crop mapping from single-modal SITS and the first fully attentional architecture for this task. It has outperformed previous state-of-the-art models which apply convolution to extract spatial features or models that do not process the spatial dimension of the SITS at all. Its success in crop mapping from single-modal SITS makes it a promising architecture to extend to multi-modal fusion to further improve classification performance.

# 3 Related Work

In this section we present existing work related to crop mapping and multi-modal fusion. We first introduce architectures for single-modal crop mapping, then review the state of the art on multi-modal crop mapping. As Transformers are not yet widely used for multi-modal crop mapping, we also review multi-modal Transformer architectures in other domains.

## 3.1 Crop Mapping from SITS

Analyzing time series data has been shown to be beneficial for crop mapping as the changes in phenological features over the year differ between crop types [3]. Due to the increased availability and complexity of high-quality SITS there has been a boost in the development of algorithms for processing this data. The following sections give a brief overview over existing methods in crop mapping first from single- and then multi-modal data. While there are a few publications about single-modal crop mapping from SAR data [6, 17, 40], most focus on optical data only. In our review of the state of the art we will do the same.

### Random Forest

Many approaches to crop mapping from SITS use a Random Forest which operates on the channel information of each individual pixel [55, 46]. In [3] the authors use a Random Forest for crop mapping where features are collected for all time points of a pixel and concatenated to a single feature vector. The pixel feature vectors are then used to train a Random Forest classifier. As the model operates on the pixel level, this is a pixel-based classification model. However, Random Forests can also be used for parcel classification as shown in [23] where the pixel intensities of each field are averaged to form a single feature vector.

### Convolutional Neural Networks

A disadvantage of the random forest classifier is that it does not consider the correct temporal order of the observations [38]. CNNs can be used to convolve the temporal dimension and thereby retaining the temporal order of observations.

In [38] a temporal CNN is applied for pixel-based classification from optical SITS. The CNN operates on each pixel separately by convolving a one dimensional kernel over the pixel time series of each channel. The feature maps obtained for each individual channel are summed to form the final feature map. The CNN architecture consists of several blocks, each containing a convolutional layer, a batch normalization and a rectified linear unit (ReLU) activation function. Afterwards, a dense layer and a Softmax with a linear layer are applied to retrieve the classification scores. The authors do not apply any pooling to the temporal dimension as pooling decreases the resolution of the data. Especially in crop classification, the exact temporal location and amplitude of a feature is important for the classification as otherwise the time point of vital information such as the peak of greenness or one-time events such as harvest are lost [38]. The authors also evaluate the possibility to use a two dimensional kernel that covers the temporal and the spectral dimension. However, their model does not take any spatial context into account which is shown to lead to salt and pepper noise in the resulting segmentation maps.

A remedy to this salt and pepper noise is using a 3D CNN with a three dimensional kernel that spans in addition to the temporal dimension also the two spatial dimensions [20]. The kernel is applied to each channel of the SITS separately, then the resulting three dimensional feature maps are summed. Pooling is applied after each convolutional layer to reduce the spatial dimension of the feature maps. Similar to the temporal CNN, no pooling is applied to the temporal dimension in order to preserve the whole growth cycle. After the final convolution-pooling layer, the spatial resolution is $1 \times 1$ pixel. Then, a fully connected layer is applied that collapses the temporal dimension. Finally, another fully connected layer maps to a $C$-dimensional vector that represents the class probabilities for each of the $C$ classes.

### Recurrent Neural Networks

RNNs are another way of dealing with the temporal dimension of SITS. In this architecture the classification of a current observation is based on the classification of previous observations. Classical RNNs suffer from the vanishing gradient problem. This occurs when the gradient has to be propagated over many steps but is diminished with each step during backpropagation. Therefore, in 2016 the Long Short-Term Memory cell (LSTM) was introduced. It contains memory cells that store information even for longer sequences. Russwurm et al. [44] use an RNN with LSTM cells for pixel-based crop mapping from SITS. The model operates on sequences of 3x3 image patches, covering the same area at different observation times. Storing information of past observations makes the model more robust to obstruction by clouds or snow.

Another problem from which RNNs, even with LSTM cells, suffer are exploding gradients. This occurs when the gradient is magnified with each step in the sequence. Turkoglu et al. [57] introduce the ConvStar cell that keeps the magnitude of the gradient during backpropagation close to 1 and thereby avoid exploding gradients.

In any case, a general disadvantage of recurrent architectures is that with increasing sequence length the computational demands become very high. Moreover, the inherent sequential nature makes parallelization over samples during training difficult. This can be mitigated by applying self-attention instead which operates on the sequence as a whole.

### Self-attention-based Methods

The adaptation of the Transformer architecture to Vision Transformers as introduced in Section 2.2 has led to a rising of purely attention-based architectures in the Remote Sensing domain. First methods in crop mapping apply self-attention only to the temporal dimension of SITS. Russwurm et al., for example, apply a temporal Transformer encoder for parcel classification [43]. The input is a time series of mean-aggregated spectral features of pixels within the spatial extent of one individual parcel. The Transformer encoder applies attention to the temporal dimension of the feature vector time series. By a global max pooling of the resulting feature token sequence, the temporal dimension is collapsed and the resulting single token can be mapped to class probabilities. Similarly, Garnot et al. apply a temporal attention encoder to a set of features extracted from a subsample of pixels within a parcel [16].

Other approaches combine CNN's with self-attention modules. The U-TAE [13] network contains a 2D spatial convolution U-Net in which each time step is down-sampled separately. At the lowest layer of the U-Net, pixel-wise temporal attention is computed and used to collapse the temporal dimension of the feature maps in the up-sampling branch. Li et al. use a three branch architecture to extract local, global and temporal features from NDVI time series [28]. The first branch contains a Transformer encoder to extract spatial information on a global level, the second extracts local features and finally the third branch extracts temporal features with an LSTM. The three extracted types of features are then merged to a single prediction.

Finally, with the TSViT as introduced in Section 2.2, Tarasiou et al. developed the first purely attentional pixel-based classification method for crop mapping from SITS [51].

## Multi-modal Fusion for Crop Mapping

Practically all of the methods mentioned in the previous section can be adapted for multi-modal data. The simplest approach to achieve crop classification from multiple modalities is Early Fusion which is the concatenation of images from different modalities in the channel dimension. This has been explored for Random Forests from Sentinel-1 and Sentinel-2 data in combination with environmental data [3] as well as with Planet Scope images [60]. Moreover, the previously mentioned single-modal U-TAE has been evaluated with Early Fusion as well as Decision Fusion in which each modality is processed with a separate U-TAE instance and their final predictions are combined [15]. Tseng et al. introduce a lightweight Transformer architecture that processes multi-modal pixel time series [54]. The model is pre-trained on Sentinel-1, Sentinel-2 and environmental data and can be fine tuned on data sets with the same modalities. However, this architecture does not take the spatial dimension into account at all which can lead to noise in the predicted segmentation maps. Similarly, Li et al. introduce a multi-branch architecture in which a Transformer encoder extracts temporal features and a convolution module extracts spatial features from optical-SAR time series [27]. The input of both modalities is again combined by Early Fusion.

In general, Early Fusion requires that the modalities have the same temporal and spatial resolution so that the SITS can be concatenated. Decision Fusion, sometimes also called Late Fusion, is computationally inefficient as a separate instance is needed for each modality. More sophisticated fusion strategies first process each modality separately to extract features and then combine those features in a fusion module. Gadiraju et al. use an CNN branch to extract spatial features from high resolution single-temporal RGB images and an LSTM branch for temporal feature extraction from biweekly NDVI image time series [12]. The resulting features are either concatenated or averaged and consequently passed to a final classification layer that maps to probabilities with either a MLP or a Support Vector Machine (SVM).

## 3.2 Multi-modal Fusion with Transformers

The Transformer architecture has often been adapted to multi-modal fusion. It is particularly suitable for this due to the tokenization step, as this allows to map inputs from different modalities to the same token dimension which does not require the inputs to have the same dimensions. Sun et al. [49] tokenize both modalities separately to the same token dimension $d$ and then concatenate the token sequences before passing them to the Transformer encoder. Further, Li et al. [29] use token sequence concatenation in a sequence to sequence learning task for dance motion prediction from music. Both modalities, 3D motion and audio, are tokenized and first passed through individual Transformers. The resulting token sequences are concatenated and passed through a fusion Transformer which generates new motion sequences to the given audio sequence.

Roy et al. fuse LiDAR and hyperspectral images (HSI) for land cover mapping with an ViT [42]. They extract an $11 \times 11$ pixel patch from each modality to classify the center pixel. To fuse the modalities in the Transformer architecture, they derive the feature tokens from the high-dimensional HSI patch and the class token from the LiDAR patch. As the HSI data has a lot more channels than the LiDAR data, it contains more information and is therefore split into $n$ feature tokens. The LiDAR data on the other hand contains only one channel which is projected to a single token. By using the LiDAR token as the class token, it is not randomly initialized as in the ViT architecture but already carries complementing information that contributes to the classification.

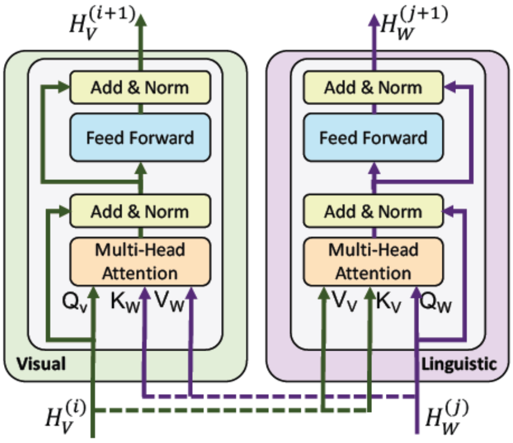


Figure 3.1: Transformer encoder layer with co-attention as introduced in Vilbert [32]. Each modality has its own Transformer encoder and the keys and values are taken from the other modality's encoder.

Another widely used fusion strategy is cross-attention or co-attention which was first introduced by Lu et al. within the Vilbert model [32] to fuse images and natural language. In this method, each modality is tokenized separately and passed through its own Transformer encoder. In each layer the keys and values are exchanged between modalities as shown in Figure 3.1.

This means that in the encoder for one modality only the queries are derived from this modality, whereas key and value are derived from the other.

Cross-attention as introduced in Vilbert [32] is only defined for two modalities. However, the concept has been extended to three modalities in different ways: First, in 2019, Tsai et al. [53] use two temporal encoders instead of one for each of the three modalities. Both apply cross-attention with the keys and values from one of the other two modalities. The resulting tokens are concatenated and passed through another Transformer that performs self-attention. The sequences of all modalities are then used for prediction. Then, in 2021, Rahman et al. propose Tribert [41] with one Transformer encoder for each modality where keys and values for cross-attention are created by concatenating the keys and values of the remaining modalities.

The bottleneck fusion method by Nagrani et al. [35] uses a separate Transformer encoder for each modality and prepends a small series of bottleneck tokens to both modality token sequences. After each layer, these tokens are extracted, averaged and prepended again before passing through the next layer. The goal is to pass condensed information from one modality to the other. A similar approach by Hoffmann et al. [19] exchanges the class tokens directly instead of using dedicated bottleneck tokens. After each layer, the class tokens of all modality-specific encoders are extracted, concatenated and projected back to the token dimension $d$. The resulting fused class token is then prepended to the token sequences of all modalities before passing them to the next layer of their respective encoder.

# 4 Proposed Fusion Methods

In this chapter, we introduce the Fusion methods for the TSViT that were developed and evaluated for this thesis. The fusion methods focus on modifying the token embedding or the temporal encoder as fusing early in the architecture adds less computational complexity than fusing at a later point.

Formally, we define the problem statement as follows: Let $\{X_m\}_{m=1}^{M}$ be a set of multi-modal co-registered SITS from $M$ modalities acquired over the same geographic area. An SITS $X_m$ of the $m$-th modality is then defined as $X_m \in \mathbb{R}^{T_m \times H_m \times W_m \times C_m}$ where $T_m, H_m, W_m$ and $C_m$ respectively refer to its number of time steps, height, width and number of channels. A pixel of an SITS $X_m$ is defined as $X_m(h,w) \in \mathbb{R}^{T_m \times C_m}$ with $h \in [H_m]$ and $w \in [W_m]$ being the position of the pixel within the spatial dimension of the SITS.

In the following, crop mapping is considered as a pixel-based classification task. Therefore, we assume for each set of co-registered SITS, a label map, $Y \in [0,1]^{H \times W \times K}$, where $K$ denotes the number of labels for the classification task. Each label map together with the corresponding SITS from different modalities are associated to the same geographical area. Since we assume a single label per pixel, $Y(h,w)$ is a one-hot vector with only a single non-zero entry. We further assume that the spatial resolution of the label map corresponds to the finest spatial resolution among the input modalities, i.e. $W = \max\{W_m\}_{m=1}^{M}$ and $H = \max\{H_m\}_{m=1}^{M}$.

The task of pixel-based classification can then be defined as finding a function that takes a set of co-registered SITS and generates an output, $\hat{Y} \in [0,1]^{H \times W \times K}$, which approximates $Y$ as closely as possible.

In the following sections, we describe our proposed fusion methods. We denote the single-modal TSViT as SM TSViT and multi-modal adaptations of this architecture as MM TSViT.

## 4.1 Early Fusion

The most basic and straight forward fusion method is Early Fusion which has been explored in the literature with several architectures as summarized in Chapter 3. The fusion is done by concatenating the data of different modalities before passing them to the model. For multi-modal crop mapping it can be applied in combination with any model that performs pixel-based classification from single-modal image time series such as the SM TSViT.

The multi-modal SITS $\{X_m\}_{m=1}^{M}$ are concatenated in the channel dimension. This allows treating them like a single-modal SITS that can be passed through the SM TSViT. In order to stack the channels, the SITS from all modalities need to have the same spatial and temporal resolution with $W_m = W_n, H_m = H_n, T_m = T_n$ for all $m,n \in [M]$. Then, we define the fused input

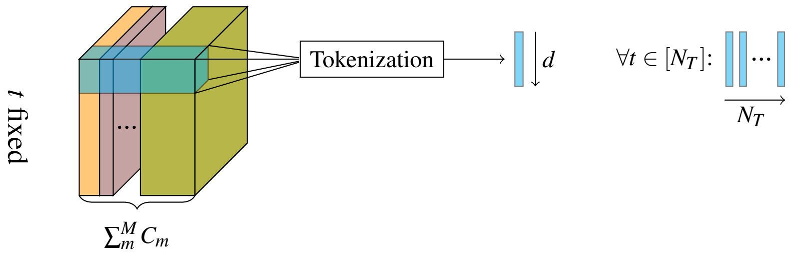


Figure 4.1: In the Early Fusion architecture the modalities are stacked in the channel dimension and tokenized as in the SM TSViT architecture, here shown for $N_T = T$. All channels of a $h \times w$ patch at a specific time point $t$ are projected to token dimension $d$. Afterwards, the tokens of each spatial patch for all time steps $t \in [N_T]$ are assembled and put into sequence.

to the SM TSViT architecture as:

$$\bar{X} = concat(X_1, \dots, X_M) \tag{4.1}$$

with $\bar{X} \in \mathbb{R}^{T \times H \times W \times C}$ with $C = \sum_{m=0}^{M} C_m$ and $T, H$ and $W$ being the equal temporal and spatial dimensions of all $M$ modalities. The resulting multi-modal SITS is passed through the SM TSViT to produce the prediction $\hat{Y}$.

If the spatial resolution of the modalities is different, the images of those with the lower spatial resolution should be up-sampled to reduce information loss in the higher dimensional modalities. As for the temporal dimension, the images should have enough time points to accurately reflect significant changes in crop phenology. Too many time points, i.e daily images, however, increase computational overhead while offering little more information, as spectral features change slowly over time. Therefore, a reasonable middle ground such as weekly to monthly images must be found. If necessary, modalities must be down- or up-sampled by selecting images from time points that correspond to the selected time points or by interpolation.

By combining the data from all modalities as early as possible, all components of the architecture, i.e. patch embedding as well as temporal and spatial encoders contribute to the fusion of the information. As no additional components are added to the architecture and the amount of tokens generated is the same, the computational complexity does not increase.

## 4.2 Modality Token Concatenation

In the Early Fusion method the information of all modalities within one patch is condensed into a single token. This leads to some fusion likely taking place in the linear transformation of the tokenization step. Tokenizing the features of the modalities separately and concatenating the sequences can retain modality-specific features that are enhanced or diminished depending on their relevance in the attention mechanism. This poses a potentially more meaningful fusion than in the linear layer of the tokenization step.

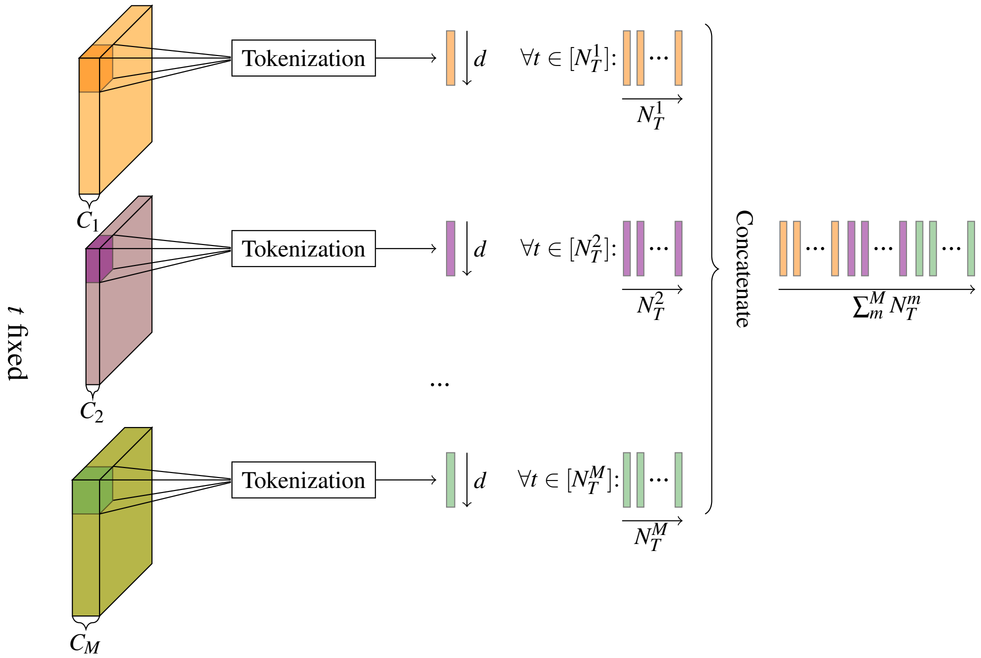


Figure 4.2: In the Modality Token Concatenation architecture, each modality is embedded separately. For a patch of each modality a token time series is produced as in the SM TSViT. These token time series are concatenated to form a sequence of tokens. From there on it is proceeded as in the SM TSViT architecture by prepending the class tokens and passing to the temporal encoder.

An advantage of this Modality Token Concatenation (MTC) Fusion over Early Fusion is that the modalities do not necessarily need to have the same temporal dimensions. As each modality is tokenized separately, one can have more time steps and thus more tokens than the other. Further, adding the temporal position encoding to each modality-specific token sequence separately, ensures that the time of observation in the year is preserved for each modality. Moreover, the spatial dimensions can also be different as long as for each patch with spatial dimensions $h_m \times w_m$ from modality $m$, there is a patch with $h_n \times w_n$ from modality $n$ that covers the exact same geographic area. By allowing the spatial and temporal dimensions to be different, it is not necessary to apply interpolation which is often accompanied by information loss.

Formally, the input to the architecture is given as a set of multi-modal SITS $\{X_m\}_{m=1}^M$. The SITS of each modality $m \in [M]$ is cut into $h_m \times w_m \times t_m \times C_m$ sized patches, which are flattened and then passed through a modality specific linear layer to project them to the token dimension $d$. This results in $M$ temporal token sequences with $\{Z_T^m\}_{m=1}^M$ with $Z_T^m \in \mathbb{R}^{N_H N_W \times N_T^m \times d}$ for $m \in [M]$. The modality specific temporal position encoding $P_T^m \in \mathbb{R}^{N_T^m \times d}$ is added to each of the sequences. Then, the sequences are concatenated to a sequence of length $\sum_m^M N_T^m$ resulting in a $N_H N_W$ token time series $Z_T^0 \in \mathbb{R}^{N_H N_W \times \sum_m^M N_T^m \times d}$. After prepending $K$ class tokens to each of token series, they are passed through the temporal encoder. From here on, the procedure is the same as in the SM TSViT architecture.

The complexity of the MTC Fusion architecture is higher than that of the SM TSViT as the token time series that is passed to the temporal encoder is $M$ times longer and the attention computation's complexity is quadratic in the length of this input sequence. Moreover, a linear tokenization layer is needed for each modality.

## 4.3 Channel Encoding

The different modalities utilized for multi-modal fusion contain complementary information. Individual channels of each modality contain complementary information as well, such as e.g. the different wavelengths of optical data. For optical data it is shown that crop types have differences in their spectral pattern which can be learned and by which they can be distinguished. Extending this idea to the combined channels of all modalities, the goal of multi-modal fusion is to learn the importance of the channels of all modalities for the identification of each crop type. However, the amount of channels in each modality is usually different. Then, mapping a patch of each modality to the same token dimension gives each channel in the modality with less bands proportionally more relevance than each channel in the modality with more bands.

To treat this imbalance, we develop a method that maps each channel of a patch to one token, thereby giving each channel equal relevance. This has already been explored by Bao et al. [2] who create a token for each patch and each channel. They concatenate all of them before passing the sequence to the Transformer encoder, similar to the previously introduced Modality Concatenation Fusion. Thereby, the sequence length is increased by factor $C$, the amount of channels. However, applying this before the temporal encoder on time series data with weekly or monthly images and several channels from multiple modalities would result in a sequence of potentially more than a hundred tokens. As the attention mechanism's complexity is quadratic in the input sequence length, this method would be infeasible for most SITS data.

Instead, we introduce another Transformer encoder to the architecture that computes attention over the channel tokens from all modalities for each time point individually as shown in Figure 4.3. Thereby, the channels which are most important for the classification out of all modalities are emphasized. By taking the mean of each channel token sequence after passing it through the encoder, we map the condensed information to a single token. These are then assembled for each image patch and sorted by acquisition time so that they can be passed to the temporal encoder. This requires that the modalities all have the same temporal dimension, i.e. $T_m = T_n$ for all $m, n \in [M]$. In the following, we denote the temporal dimension as $T$ for all modalities.

Formally, let $\{X_m\}_{m=1}^{M}$ with $X_m \in \mathbb{R}^{H_m \times W_m \times T \times C_m}$ for $m \in [M]$ be a set of images of $M$ modalities. The first step, the tokenization, is done separately for all modalities. Additionally to splitting in the spatial and temporal dimensions, we also split the SITS by channels so we obtain $N_H \cdot N_W \cdot N_T \cdot C_m$ patches of shape $h_m \times w_m \times t$. These are subsequently projected to the token dimension $d$ and concatenated by spatial and temporal location. This modified token embedding results in a channel token sequence $Z_C \in \mathbb{R}^{N_H \cdot N_W \cdot N_T \times C \times d}$ with $C = \sum_m^M C_m$.

In order to extract the most relevant information from these channels, we add another Transformer encoder which computes *channel attention*. After adding a learnable channel position

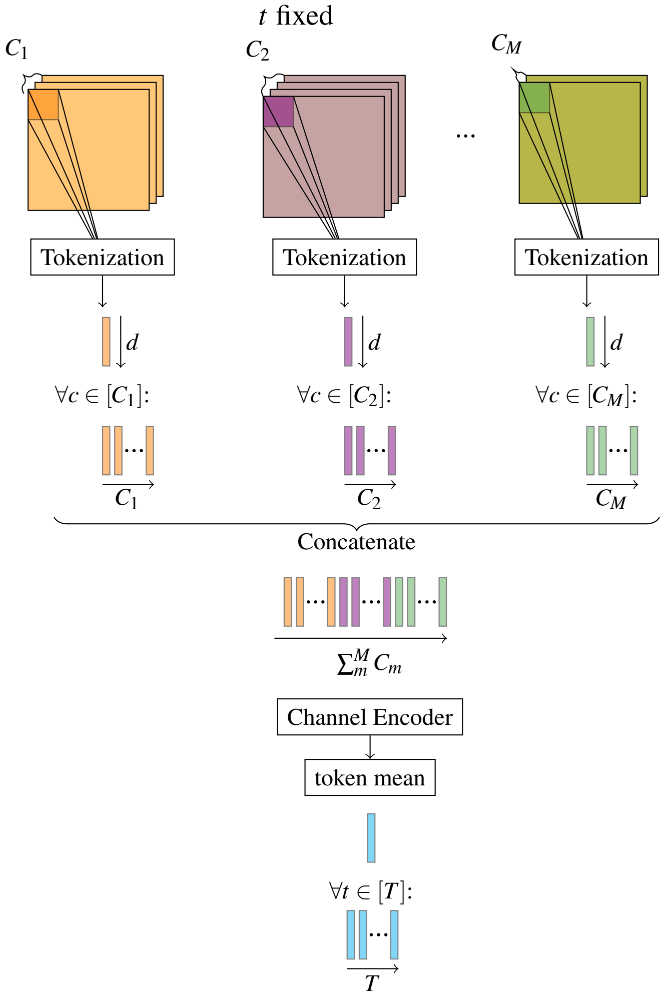


Figure 4.3: Channel Encoding is done by tokenizing each channel of each modality separately and concatenating to a channel token sequence per spatial and temporal location of length $\sum_m^M C_m$. The sequence is passed through a channel encoder and subsequently averaged. After assembling the resulting tokens to a temporal token sequence per spatial location, the sequences can be passed to the temporal encoder as in the SM TSViT.

encoding the sequence is passed through the layers of the encoder. Afterwards, the mean of the token sequence after the final layer $Z_C^L$ is taken for each spatial and temporal location:

$$(Z_C^L)^{mean} = \frac{1}{C}\sum_c^C Z_C^L[:,c] \tag{4.2}$$

which yields $(Z_C^L)^{mean} \in \mathbb{R}^{N_H \cdot N_W \cdot N_T \times d}$. In order to proceed with the temporal encoder, we separate the temporal from the spatial location by reshaping to $Z_T \in \mathbb{R}^{N_H \cdot N_W \times N_T \times d}$. This results in a temporal token sequence for each spatial location which allows for temporal attention computation in the temporal encoder. From this point on, the procedure is the same as in the SM TSViT.

Due to mapping each channel of a patch to a separate token, the amount of tokens created in the tokenization step is increased by factor $C$. The amount of token series processed in parallel in the channel encoder is by a factor $N_T$ larger than in the temporal encoder in the original architecture. This increases the storage requirements significantly. With the additional encoder, the computational requirements are also increased. However, as the input sequence of the channel encoder is generally shorter than for the temporal encoder, the computation of the channel attention is less complex than the temporal attention.

## 4.4 Cross Attention Fusion

In multi-modal learning it is expected that the modalities complement each other, i.e. information that is not contained in one modality can be found in the other. In the Remote Sensing context for example SAR offers complementary information to optical data. Moreover, it is expected that a pair of pixels belonging to the same crop type show similar SAR backscatter as well as similar multi-spectral reflectance values. This can be exploited in the attention mechanism. If for example crop type specific features of a sample are prevalent in the SAR data but less prevalent in the optical data, they can be enhanced in the attention mechanism by deriving the queries from the SAR data. This technique is called cross-attention.

In standard self-attention, as described in section 2.2, attention weights are computed between each pair of tokens in the sequence. In cross-attention each modality is passed through its own Transformer encoder but the queries are exchanged between them, i.e. the queries used in the encoder of one modality are derived from the corresponding token of the other modality.

$$out_{mod1} = softmax(\frac{Q_{mod2}K_{mod1}^T}{\sqrt{d}})V_{mod1} \tag{4.3}$$

$$out_{mod2} = softmax(\frac{Q_{mod1}K_{mod2}^T}{\sqrt{d}})V_{mod2} \tag{4.4}$$

Cross-attention is not limited to two modalities. For the ViT, two possible ways for applying cross-attention for three modalities have been introduced: i) stacking the keys and queries of the two complementary modalities [53], ii) introducing one encoder for each complementary modality thereby increasing the model complexity significantly [41]. In order to keep the increase in model parameters small, we compute the cross-attention weights for all modalities separately and then take the mean as shown in Figure 4.4. In the following, our adaption of the SM TSViT to the fusion of multi-modal input using cross-attention is referred to as Cross Attention Fusion (CA Fusion).

Formally, for a set of co-registered SITS from $M$ modalities $\{X_m\}_{m=1}^M$, we tokenize each modality separately into $M$ token sequences and add the temporal position encoding. To each

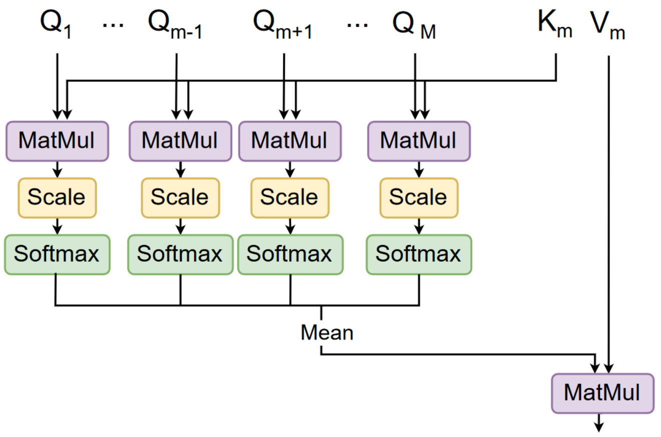


Figure 4.4: Cross-attention computation for modality-specific encoder $m \in [M]$.

resulting sequence, $\{(Z_T^0)_m\}_{m=1}^M$ with $(Z_T^0)_m \in \mathbb{R}^{N_H N_W \times N_T \times d}$ for all $m \in [M]$, $K$ class tokens $(Z_{Tcls}) \in \mathbb{R}^{K \times d}$ are prepended. Consequently, each sequence is passed through the corresponding modality-specific temporal encoder applying cross-attention. We define cross-attention between modality $m \in [M]$ and all other modalities $n \in [M]$ with $n \neq m$ in the encoder of modality $m$ as:

$$A_m(\{Q_n\}_{n=1}^M, K_m, V_m) = \frac{1}{M-1} \sum_{n \in [M], n \neq m} a_{n,m} V_m \tag{4.5}$$

$$a_{n,m} = softmax\left(\frac{Q_n K_m^T}{\sqrt{d}}\right) \tag{4.6}$$

After the final layers, the class tokens $\{(Z_{Tcls}^L)_m\}_{m=1}^M$ are extracted from the final token sequences of all modalities. We aggregate their information by taking the mean of the class tokens for each class $k \in [K]$:

$$(Z_{Tcls}^L)^k = \frac{1}{M} \sum_{m=1}^M (Z_{Tcls}^L)_m^k \tag{4.7}$$

and pass them to the spatial Transformer encoder where they continue to be processed as in the SM TSViT architecture.

This architecture requires an equal temporal resolution among all modalities. As the tokenization is done separately, the spatial resolution does not necessarily need to be the same, as long as for each patch that is mapped to a token in one modality there is also a patch that covers the same area on the same time point in all other modalities.

Introducing an individual temporal encoder for each modality increases the amount of parameters in the temporal encoder step of the architecture by factor $M$ compared to SM TSViT. Further, the computational complexity of cross-attention of each temporal encoder is bigger than with self-attention in SM TSViT. The matrix multiplication in standard self-attention $QK^T$ has quadratic complexity. In our proposed Cross Attention Fusion architecture, this is computed in the temporal encoder for modality $m \in [M]$ for each other modalities $n \in [M]$ with $n \neq m$ as $Q_n K_m^T$. This increases the complexity of the attention computation in each temporal encoder by $M-1$ compared to self-attention.

## 4.5 Synchronized Class Token Fusion

In the first part of the SM TSViT architecture information is mapped from the temporal feature tokens to the class tokens which are later mapped to class probabilities. As such they constitute the most important tokens in the temporal encoder. In the multi-modal context it would be useful if they contained the most important information from the temporal feature tokens of all modalities. In the previously introduced Cross Attention Fusion this is achieved by exchanging queries between modality-specific encoders. However, then it is possible that the feature tokens influence each other across modalities within the cross-attention mechanism. This could potentially skew the modality-specific features. In order to avoid this, we use a separate temporal encoder for all modalities and *synchronize* the class tokens after each layer as introduced in [19] for the ViT architecture. Motivated by its success in multi-modal, multi-label image classification for RS images, we study its effectiveness within the temporal encoder of the TSViT architecture.

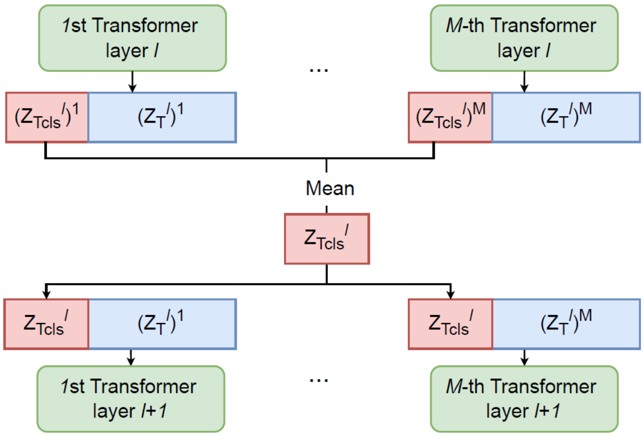


Figure 4.5: Synchronizing the class tokens between layers $l$ and $l+1$ of the $M$ modality-specific encoders by taking the mean per class.

To this end, we process the set of input SITS $\{X_m\}_{m=1}^M$ of each modality $m \in [M]$ separately until the temporal encoder. This results in $M$ token sequences $\{(Z_T^0)_m\}_{m=1}^M$ containing the modality-specific feature tokens with the temporal position encoding added and the class tokens prepended. Each is passed through a dedicated modality-specific encoder. After each layer, $l \in [L-1]$, the $K$ class tokens $\{(Z_{Tcls}^l)_m\}_{m=1}^M$ are extracted from each of the $M$ sequences. They are subsequently synchronized by taking their class-wise mean across all modalities:

$$(Z_{Tcls}^l)^k = \frac{1}{M} \sum_{m=1}^{M} (Z_{Tcls}^l)_m^k \tag{4.8}$$

The modality-agnostic, mean-aggregated class tokens are then prepended to all modality-specific feature token sequences that resulted from layer $l$ and passed to the next layer $l+1$ as shown in Figure 4.5. This synchronization is repeated after each layer until the final layer $L$. Here, the class tokens are extracted and mean-aggregated as defined in Equation 4.8 but then directly passed to the preprocessing for the spatial encoder as in the SM TSViT described in Section 2.2.

Similar to Cross Attention Fusion, the amount of parameters in the temporal encoder step is increased by $M$ compared to SM TSViT due to the utilization of one encoder per modality. The computational complexity of each encoder is not increased as the only interaction between the $M$ encoders is the exchange of class tokens which are mean averaged. The averaging itself is linear in the token dimension $d$, the amount of classes $K$ and the amount of modalities $M$. This additional complexity is added for each layer in the temporal Transformer encoders.

# 5 Data Set Description and Design of Experiments

## 5.1 Description of the Data Set

The data used for the evaluation of the presented methods stems from the EOekoLand project of which RSiM is a part of and is provided by the Thünen Institute of Farm Economics. It contains SITS from Sentinel-1, Sentinel-2 and Planet Fusion data which form the modalities. Satellite image data from all three modalities is provided for two areas of size 24km × 24km, one in Brandenburg and one in Bavaria. The data is provided for the years 2020, 2021 and 2022, respectively. This results in a total of six provided tiles for each modality.

The provided Sentinel-2 data contains 10 spectral bands, capturing different wavelengths which correspond to Red, Green, Blue, NIR, Broad NIR, Red Edge 1, Red Edge 2, Red Edge 3, SWIR 1 and SWIR 2. The temporal dimension was interpolated with an ensemble of Radial Basis Functions (RBF) using four different kernel widths to a 10-day interval [45]. The widths are ±11, ±23, ±63, and ±127 days of the target date with more emphasize being put on observations closer to the target date. Further it was radiometrically and atmospherically corrected with the FORCE-datacube [10, 11]. The final Sentinel-2 product consists of 37 equidistant timesteps per year. The Sentinel-1 data contains two bands, VH and VV, with the signal being transmitted in vertical polarization and received in both vertical and horizontal polarization. The temporal dimension has been interpolated in the same way as the Sentinel-2 data and is provided for the same 37 timesteps. The Sentinel data are provided in 10m spatial resolution. It should be noted that all preprocessing of the Sentinel data was conducted by the Thünen institute and was not part of this thesis.

The Planet Fusion data is provided in 3m spatial resolution and with the channels Red, Blue, Green and NIR. This modality contains daily images (365 images per year) but we sample to the same 10 day interval as the Sentinel data because the Planet Fusion data would be otherwise very large. Moreover, as it is already an interpolated product, changes from one day to the next are only marginal.

As each tile covers 24km x 24km, they each contain 8000x8000 Planet Fusion pixels and 2400x2400 Sentinel pixels.

### Reference Data

The labels were acquired through the Integrated Administration and Control System (IACS) of the European Union [52] which contains reports about crop cultivation from farmers who apply for EU subsidies. The crop type labels were grouped by the Thünen Institute to reduce the amount of different classes which results in 15 crop types and one Background class. However, as not all farmers apply for subsidies, pixels labeled as Background may also contain crop fields.

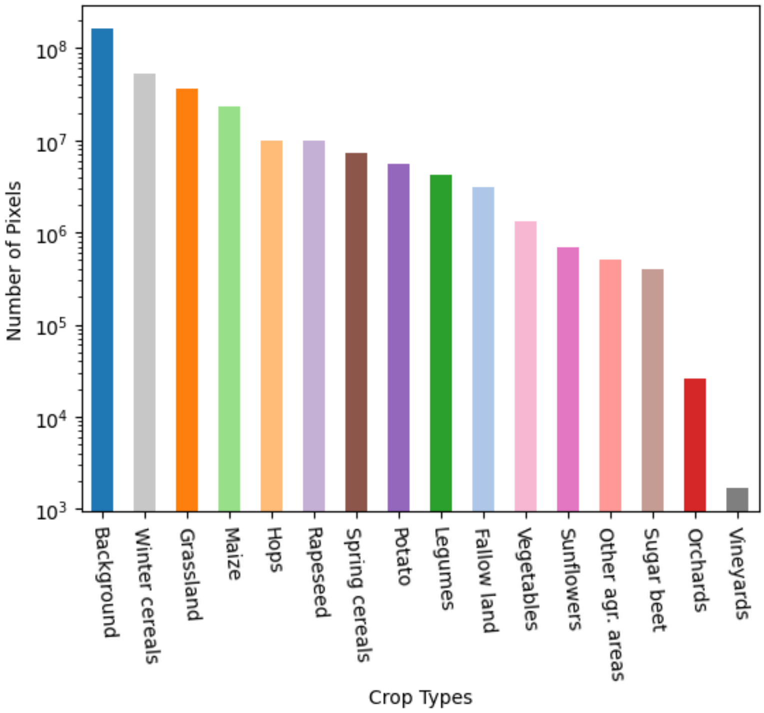


Figure 5.1: Number of pixels per class.

The reference data is available for the Brandenburg area for the years 2020, 2021 and 2022 and for Bavaria for the years 2020 and 2021. The number of pixels per class is shown in Figure 5.1.

## Data Preprocessing

The input sample size for the TSViT model is chosen to be 240m x 240m as in the original paper [51] which results in patches of size 24x24 pixels for the Sentinel images and 80x80 pixels for the Planet Fusion images. An example for a patch of each modality for the same area is shown in Figure 5.2. By cutting to 240m × 240m patches, we get 100 × 100 patches for each geographic area and year which results in a total amount of 50.000 patches. Because the class ”Vineyard” is very rare with only 1670 pixel samples, we exclude it from the data and instead assign the Background label to the pixels.

Due to overlaps in the provided polygons defining the field outline, we filter for patches that contain only overlaps smaller than $4m^2$. The number was chosen as it is smaller than half of the largest area per pixel which is $9m^2$. The reflectance measured for this pixel is influenced by both fields anyways, so the error that is introduced with an overlap of at most $4m^2$ is not very large. Moreover, when using $4m^2$ as a threshold, only few patches are lost. The amount of patches per tile and year that are obtained with these preprocessing steps are shown in Table 5.1.

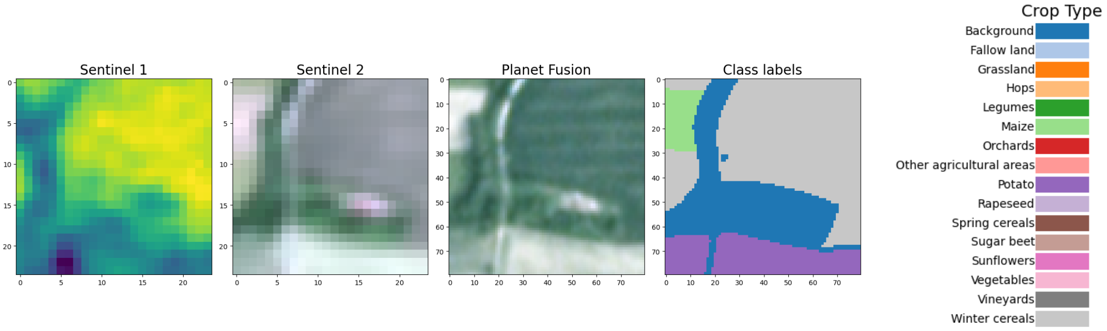
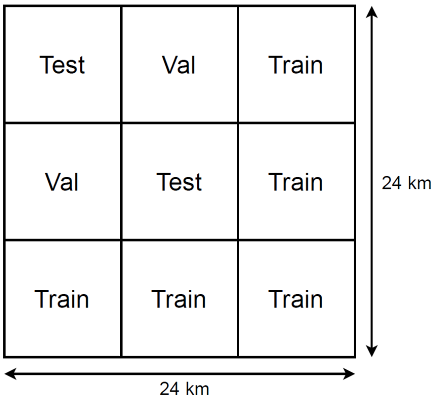


Figure 5.2: Example 240m x 240m image patch for each modality. The Sentinel-2 and Planet Fusion patches are shown as a true color images. The Sentinel-1 patch shows the radar vegetation index (RVI) which is the ratio of cross polarized backscatter to total backscatter: $RVI = \frac{VH}{VH+VV}$.

Table 5.1: Number of patches per tile after removing those that contain fields with overlaps larger than $4m^2$. Every tile contained 10.000 patches before applying this filter. BB=Brandenburg, Bav=Bavaria.

| Tile | BB 2020 | BB 2021 | BB 2022 | Bav 2020 | Bav 2021 | Total |
|---|---|---|---|---|---|---|
| Number of patches | 9993 | 9854 | 9865 | 9582 | 9915 | 49209 |

Each tile is split into 3 by 3 areas which are divided into training, validation and test area as shown in Figure 5.3. The patches within those areas are then assigned to training, validation and test set, respectively. The split is chosen so that samples of all classes are as evenly distributed to the three sets as possible. All five tiles are split in the same way. The class distribution in our data set is uneven with approximately 50% of all labeled pixels belonging to the Background



| | | |
|---|---|---|
| Test | Val | Train |
| Val | Test | Train |
| Train | Train | Train |

Figure 5.3: Division of each tile into 3 × 3 areas and the assignment of the patches in the respective areas to training, validation and test set.

class as shown in Figure 5.1. In order to achieve a more even class distribution in the training set and to avoid extreme under-representation of smaller classes, we use only patches with less than 80% of pixels belonging to the Background class from the training and validation set. Finally, we get 17.981 training, 5275 validation and 11.090 test patches. The final class distribution in each set can be seen in Figure 5.4.

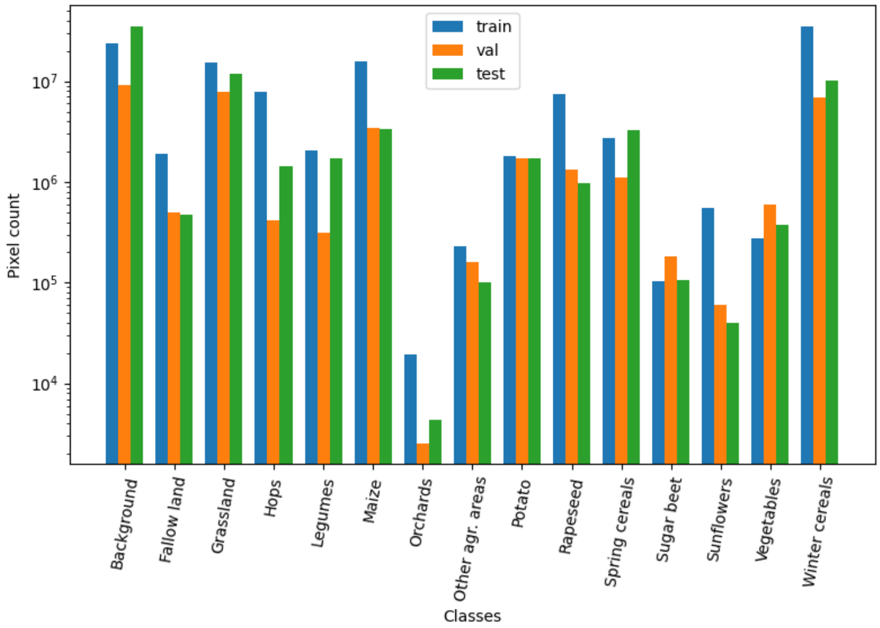


Figure 5.4: Pixel class distribution in training, validation and test set.

## 5.2 Experimental Design

For training, we use an Adam optimizer [22] with a learning rate of $10^{-4}$. Further, we use a Reduce on Plateau learning rate scheduler that reduces the learning rate by factor 10 if the validation loss does not decrease for five consecutive epochs. We use unweighted Cross-Entropy loss and a batch size of 8. Due to the large size of a single SITS, we use gradient accumulation which processes instances within a batch sequentially, accumulates the resulting gradients and updates the model parameters only after the whole batch has passed through the model [47]. Moreover, we use 16-bit mixed precision in order to train faster and use less memory [34]. All code is implemented in Pytorch [37]. For implementing the training and evaluation framework we use Pytorch Lightning. With this, gradient accumulation and mixed precision training can easily be enabled by a setting a flag in the training setup. The patches are z-score normalized by channel with the mean and standard deviation being caluclated on the training set. In order to increase the variety of patches in our training set, we apply data augmentations to their spatial dimensions. Each patch is randomly flipped along its vertical or horizontal or both axes. While this does not change which pixels are mapped to a token, it does affect the order of the tokens in the spatial encoder and the order of the pixels which are mapped to a single token. Thereby, we emulate a larger training data set.

All models are trained on a NVIDIA A100 Graphic Processing Unit with 80GB of Random Access Memory.

### Metrics

For the evaluation we compute the commonly used metrics Overall Accuracy (OA), Mean Accuracy (MA) and Mean Intersection over Union (mIoU). Moreover, we consider class-wise confusion matrices to get an insight about the classification performance of specific classes.

For a multi-class classification problem, Overall Accuracy refers to the number of samples for which the predicted label $\hat{y}$ coincides with the true label $y$ divided by the total number of samples $N$:

$$OA = \frac{1}{N}\sum_{i}^{N}\mathbf{1}(y_i = \hat{y}_i) \tag{5.1}$$

where $\mathbf{1}$ is a function that is 1 if its input is true and 0 if not. Generally, over-represented classes such as the Background class in our data set, are learned better than smaller classes as the model has more samples to learn from. This leads to many samples of this class being classified correctly during testing and thus a high Overall Accuracy. However, as we are more interested in identifying crop types correctly, Overall Accuracy might not give the desired information. To mitigate this, we also report Overall Accuracy without considering samples that belong to the Background class in its computation. In the following, we denote Overall Accuracy computed from all sample as Overall Accuracy A, and metrics computed without considering samples belonging to the Background class as Overall Accuracy B.

To better assess the performance of a model on all classes regardless of their size, we also compute Mean Accuracy. For this, we first compute class-wise accuracies by dividing the number of correctly classified samples of a class by the total number of samples within this class. When considering only one class within a multi-class classification problem, it can be modeled as a binary classification problem. For this, we define for a class $c$ true positives ($TP_c$) as samples that belong to this class and were classified as such and false negatives ($FN_c$) as samples that belong to $c$ but were not classified as such. Note that the sum of these is the total set of samples within class $c$. Then, Mean Accuracy is the average class-wise accuracy over all classes:

$$MA = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c} \tag{5.2}$$

Thereby, each class independent of its size contributes equally to the metric which thus assesses the ability of a model to correctly identify all crops.

Furthermore, we report the mIoU, also known as the Jaccard Index. For this, we denote samples not belonging to class $c$ but being identified as such as false positives ($FP_c$). The mIoU is then the fraction of correctly classified samples and the size of the union of samples classified as this class and samples belonging to the class.

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{FP_c+FN_c+TP_c} \tag{5.3}$$

By penalizing samples that were wrongly classified ($FP_c$) as well as samples that belong to this class and were not identified ($FN_c$), the mIoU gives information about how well the model captures the samples of each class.

Finally, we also consider confusion matrices computed from predictions on the test set. For this, we compute for each class the fraction of samples classified as each other class. Plotting these values as a matrix gives the class-wise accuracies on the diagonal and helps to understand which classes are most often confused with which other classes.

# 6 Experimental Results

In this chapter we present the result of the analysis of the multi-modal fusion architectures proposed in Chapter 4 with the metrics and data set introduced in Chapter 5.

For all TSViT-based architectures, we use mostly the standard hyperparameters of the SM TSViT. This constitutes a temporal encoder depth of 6, a spatial encoder depth of 2, a token dimension $d$ of 128, 4 attention heads with key, value and query dimensions of 64 and a temporal patch size of 1. As the size of the images is 80×80 pixels, the standard SM TSViT spatial patch size of 3×3 could not be applied. Instead the images were divided into $2 \times 2$ pixel patches.

Moreover, we conduct an ablation study to assess how the performance of our architecture changes when varying certain hyperparameters. Finally, we compare the performance of our architectures to two other crop mapping architectures from the literature.

## 6.1 Baselines

To obtain a baseline to assess the potential enhancement in crop mapping performance through utilizing multi-modal data, we report the results of the SM TSViT applied to each modality individually. Moreover, the results indicate how much each modality contributes to the classification. For these experiments, we use the initial spatial dimension of the modalities, i.e. $80 \times 80$ pixels for Planet Fusion and $24 \times 24$ for Sentinel-1 and Sentinel-2.

Table 6.1: Metric results (in %) for the SM TSViT on single-modal data.

| Modality | OA | MA | mIoU |
|---|---|---|---|
| Sentinel-1 | 85.07 | 65.45 | 52.43 |
| Sentinel-2 | 88.72 | **75.12** | **61.08** |
| Planet Fusion | **88.79** | 72.57 | 60.30 |

From Table 6.1 it can be seen that the metric results obtained with optical modalities are better than that of the SAR modality. This indicates that optical data is better suited for distinguishing crop types than SAR data. SAR reflectance values are related to roughness and relief of the surface. The lower metric results for SAR data potentially indicate that the crop types in the data set cannot be differentiated well in these aspects.

Among the optical modalities, training with Sentinel-2 data yields the best results in the mean aggregated metrics, whereas with Planet Fusion data the highest Overall Accuracy is achieved. This indicates that the SM TSViT is more capable of identifying small classes from Sentinel-2 data. For the class with the least samples in the training set, Orchards, no sample is correctly classified with any of the modalities. Instead, 92-100% of samples are assigned to the Background class, likely because of similarities to forests or other trees. However, the SM TSViT correctly identifies 88% of test samples of the second smallest class, Sugar Beets, when trained on Sentinel-2 data and only 75% from Planet Fusion data. Similar differences are observed for the class Sunflowers where 87% of samples are recognized with Sentinel-2 data and 66% with Planet Fusion data. Further, though less prevalent, for Vegetables with 56% and 53% correctly classified samples from Sentinel-2 and Planet Fusion, respectively. This can be explained by the additional bands provided by Sentinel-2 which give additional information in the infrared spectrum that seems to be especially valuable for under represented classes.

The classes with the best classification performance are Hops, Maize, Rapeseed and Winter Cereals. With Sentinel-2 data an accuracy larger than 90% is achieved for these classes and even with Sentinel-1 data, their accuracy is around 90%. These are also the classes with the most samples in the training set.

Interestingly, the classes Other Agricultural Areas and Fallow Land are not well recognized and often confused with Background or Grassland despite being as under represented as other, better recognized classes. Fallow Lands are crop fields which are temporarily uncultivated. It is very well possible that other fallow areas that do not belong to farmers, and thus fall in the Background class, show similar features. Moreover, fallow, uncultivated areas often grow weeds that likely display similar properties as cultivated grassland. The class Other Agricultural Areas contains various other crop types. One can assume that samples of this class also show a wide range of spectral signatures which are likely overlapping with those of other crops in the nomenclature. It is thus not surprising that only at most 46% of samples in this class are recognized.

Similarly, for the Background class, despite being the second largest class in the training set, the accuracy is up to 10% lower than for some crop classes with less samples. This is likely because the large diversity in this class leads to very different spectral signatures of samples. Moreover, not all farmers report their crops so there are likely some crop fields present. As the Background class is very large but not particularly informative for the crop mapping task, we also report the metrics without considering the Background class in their computation. These results are shown in Table 6.2.

Table 6.2: Metric results (in %) for the SM TSViT on single-modal data where samples belonging to the Background class are excluded from metric calculation.

| Modality | OA | MA | mIoU |
|---|---|---|---|
| Sentinel-1 | 84.31 | 64.0 | 55.27 |
| Sentinel-2 | **90.77** | **74.30** | **64.31** |
| Planet Fusion | 89.03 | 71.42 | 62.91 |

From Table 6.2, one can see that the Mean Accuracy is decreased for all modalities compared to the metrics considering all classes. This is expected as the accuracy of the Background class is 86-88% which is more than the reported Mean Accuracy in Table 6.1. The Overall Accuracy increases by 1% for Planet Fusion and 2% for Sentinel-2 data as large crop classes are identified with more than 90% accuracy and thus better than Background samples. More importantly, the Overall Accuracy is now highest with 90.77% for Sentinel-2 data. This further supports the hypothesis that with Sentinel-2 data crop classes are generally better identified than with Planet

Fusion data.

Interestingly, the mIoU increases when excluding Background samples from its computation. The highest fraction of misclassified samples are classified as Background for most classes. As mentioned before, this is especially the case for Fallow Land and Other Agricultural Areas. For the smallest class Orchards this constitutes even 92%-100% of all samples. This means that the number of False Positives for the Background class is relatively high and consequently its IoU small. When it is excluded from the average computation for the mIoU, the latter becomes larger.

From our experiments, we conclude that among the single modalities, the best results are achieved with Sentinel-2, followed by Planet Fusion and finally Sentinel-1. This is likely due to the type of information, i.e. optical data being more useful to distinguish crop types than SAR data. However, the metric results also correlate with the amount of bands and thus information per pixel in the modalities. For the largest classes, Sentinel-1 achieves a similar accuracy as the other two modalities. With Sentinel-2 and Planet Fusion, the lack of samples in smaller classes could potentially be more easily compensated by more information per sample. To further evaluate this hypothesis, one could train a model with only two bands from Sentinel-2.

## 6.2 Multi-modal Fusion Analysis

In the following, we analyze the results of our MM TSViT architectures obtained with the same hyperparameters as the SM TSViT in the previous section.

Table 6.3: Metric results (in %) for our proposed fusion architectures with the standard hyperparameters.

| Architecture | OA |  | MA | mIoU |
|---|---|---|---|---|
|  | A | B |  |  |
| Early Fusion | 89.79 | **92.66** | **80.34** | 66.96 |
| MTC Fusion | **90.90** | 91.66 | 77.92 | 67.80 |
| CA Fusion | 90.39 | 92.52 | 79.38 | 66.66 |
| SCT Fusion | 90.50 | 92.57 | 79.72 | **68.39** |
| Channel Encoding | 89.94 | 92.00 | 77.51 | 64.24 |

The metric results are shown in Table 6.3. When comparing to the single-modal results in Table 6.1, one can see that all multi-modal fusion architectures perform better than the single-modal baselines. For Overall Accuracy, the highest value on single-modal data is achieved with Planet Fusion with 88.7%. Without considering the Background class, the highest OA of 90.77% is reached with Sentinel-2 data. These are increased by up to 2% with all proposed multi-modal architectures. Mean Accuracy is increased by 2-5 % compared to the best single-modal configuration and mIoU by 3-7%. This clearly shows an improved performance when utilizing SAR data as well as optical data for pixel-based crop mapping.

When considering the class-wise accuracies shown in Table 6.4, one can see improvements especially in small and medium sized classes such as Spring Cereals, Sunflowers and Sugar

Table 6.4: Accuracy per class (in %) for each of the fusion architectures and the best baseline SM TSViT with Sentinel-2 data. The classes are sorted by the amount of pixels in the training set in descending order.

| Class | SM TSViT Sentinel-2 | Early Fusion | MTC Fusion | CA Fusion | SCT Fusion | Channel Encoding |
|---|---|---|---|---|---|---|
| Winter Cereals | 96.43 | **97.02** | 96.67 | 96.17 | 96.86 | 96.24 |
| Background | 86.64 | 86.87 | **90.13** | 88.21 | 88.40 | 87.85 |
| Grassland | 89.05 | 90.90 | 89.22 | 90.69 | **91.68** | 91.22 |
| Maize | 94.54 | 95.69 | 95.68 | 95.23 | **96.43** | 95.46 |
| Rapeseed | 96.64 | 97.36 | 97.85 | **98.17** | 97.86 | 97.07 |
| Hops | 97.34 | 97.16 | 96.81 | **98.17** | 97.71 | 97.56 |
| Spring Cereals | 86.22 | **91.91** | 90.14 | 91.39 | 90.60 | 88.23 |
| Potato | 94.20 | 95.61 | 93.65 | **96.33** | 95.37 | 95.11 |
| Fallow Land | 46.68 | 55.54 | 56.89 | 57.19 | **60.95** | 51.85 |
| Legumes | 81.90 | 84.31 | 83.82 | **89.09** | 77.31 | 84.82 |
| Sunflowers | 86.80 | 93.81 | **95.30** | 85.39 | 95.14 | 93.83 |
| Other Agr. Areas | 26.29 | **66.09** | 30.37 | 56.94 | 49.98 | 36.30 |
| Vegetables | 56.25 | 58.32 | 60.96 | 53.78 | **61.00** | 52.11 |
| Sugar Beet | 87.81 | 94.51 | 91.27 | 93.87 | **96.49** | 95.09 |
| Orchards | 0 | 0 | 0 | 0 | 0 | 0 |

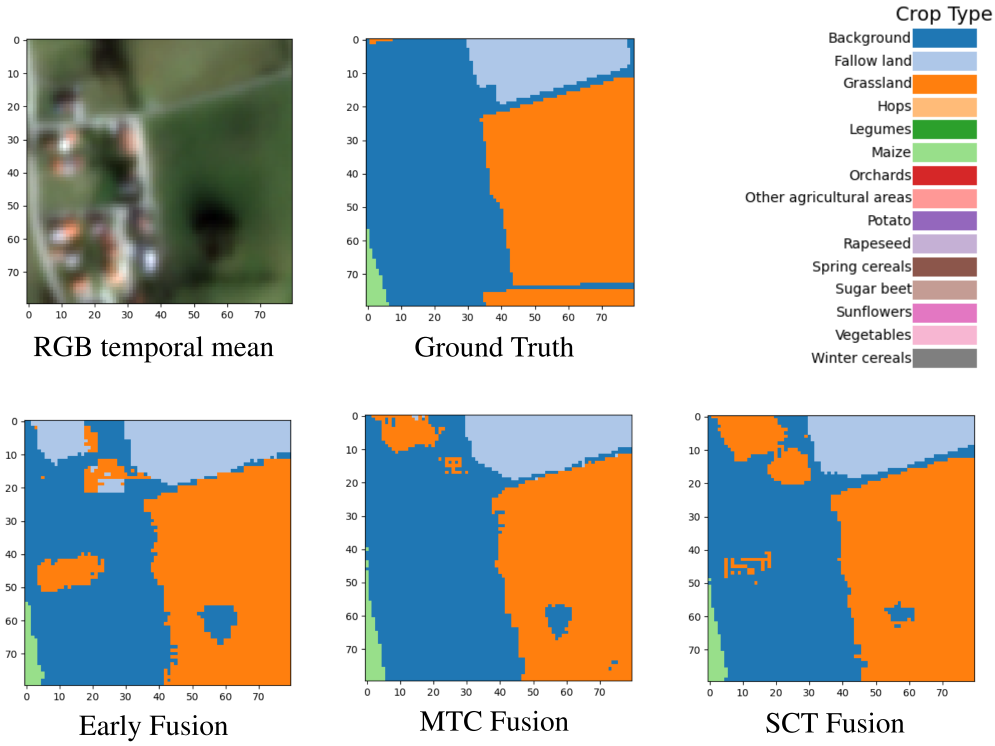
Beet. With Sentinel-2 data, the SM TSViT achieves an accuracy of about 87% for these classes, whereas with most of our fusion architectures this value is increased to well over 90%. Classes that are already well classified with $>$ 90% on Sentinel-2 data such as Winter Cereals, Rapeseed and Hops on the other hand are not much improved with multi-modal data, independent of the utilized fusion method.

Especially notable are the most ambiguous classes Fallow Land and Other Agricultural Areas. The accuracy for Fallow Land is increased by 5-14%, with SCT Fusion achieving the best result of 60.95%.

For samples belonging to the class Other Agricultural Areas, the performance differs vastly among our architectures. While the accuracy for this class with Sentinel-2 data is only about 26%, this value is 40% higher when combining all modalities with Early Fusion. An improvement of 23-30% is achieved with Cross Attention Fusion and SCT Fusion. MTC Fusion and Channel Encoding on the other hand are only able to add 4-10% to the classification accuracy on this class compared to the baseline with Sentinel-2 data. For MTC Fusion this could be rooted in a general tendency to classify ambiguous samples as Background, as it has the highest Background classification accuracy and more than 50% of Other Agricultural Area samples are misclassified as such.

Figure 6.1 shows the predictions for a sample patch from the test set with three of our proposed architectures. The predictions illustrate common problems in crop mapping, while also showing differences in the performances of our architectures. The sample patch contains a Background area, two fields of Grassland, one Fallow Land and a small part of a Maize field. On the temporally averaged RGB image of the area, one can see that the Background contains houses as well as roads and grassy areas. Moreover, there is a tree growing in the Grassland field.

As can be seen from the predictions, all three architectures are able to correctly identify large areas of the four fields and the Background area. A common problem in all architectures seems to be the classification of pixels at field borders which can be seen for example at the path between the Fallow Land and Grassland areas. Further, one can see that several Background samples are misclassified as Grassland or Fallow Land. From the RGB image, it can be inferred that the misclassified Background areas are grassy, which is why their spectral pattern likely resembles that of cultivated grassland or fallow areas. MTC Fusion, as the architecture with the highest Background accuracy, is less affected by this. All models classify the tree in the Grassland field as Background. This represents an example of the reported reference data not always being accurate on pixel level.



RGB temporal mean

Ground Truth

Early Fusion

MTC Fusion

SCT Fusion

Figure 6.1: Predicted classifications for a sample patch from the test set with three fusion architectures.

When comparing the results in Table 6.3 with each other, one can see that the differences between our proposed fusion architectures are only marginal, indicating that the way the modalities are fused does not greatly affect the performance. Moreover, due to random fluctuations in the training procedure based on random initialization of model weights, the reported results might not accurately reflect superiority of one architecture over the other. In order to reduce the influence of randomness, it would be necessary to evaluate each model with multiple seeds and provide the mean and standard deviation of the metric results. Due to long model training times, this is out of the scope of this thesis and left for future work. Nevertheless, there are still some differences that show advantages of some architectures over others.

Early Fusion outperforms the other architectures in terms of Mean Accuracy by 1-3%. Moreover, it achieves the highest Overall Accuracy B among our fusion methods, proving that it achieves the best overall classification performance for crop types. Interestingly, its Overall Accuracy A on the other hand is the lowest which results from a comparatively low classification performance on the Background class. As shown in Table 6.4, the Background class accuracy is with 86.87% on the same level as the single-modal baseline with Sentinel-2 data. The other architectures are able to increase this value by 1-4% when using all modalities. Being able to correctly identify crop types is vital for pixel-based crop mapping. However, due to the lack of information about field borders, it is equally important to distinguish non-crop areas from crops.

From Table 6.3 one can see that CA Fusion and SCT Fusion have consistently high values for all metrics. Moreover, they achieve the best class-wise accuracy for 4-5 classes each as shown in Table 6.4. Most notably, SCT Fusion achieves the highest accuracy for two of the smallest classes, Vegetables and Sugar Beets, indicating that it is able to learn representative features even from a small amount of samples. Moreover, it achieves the highest mIoU among all architectures. A significant drawback of both architectures lies in their computational complexity as they rely on one temporal encoder for each modality. Compared to the smallest architecture, Early Fusion, their amount of parameters increases from 2.4 million parameters to 5.5 million. Depending on the use case, this increase in complexity might not justify the improved classification performance in a few crop classes.

The fifth architecture, Channel Encoding, while achieving better results than the baselines, performs consistently a little worse than the other fusion architectures. This could be because only a very small amount of information is mapped to a token in the tokenization step. With Early Fusion, the amount of values mapped to a single token of dimension $d = 128$ is $h \cdot w \cdot \sum_j^M C_j = 2 \cdot 2 \cdot 16 = 64$. With Channel Encoding, this is reduced by factor $C = 16$ to only 4 values, as each channel is mapped to a separate token. Not a lot of information can be derived from these four values in the tokenization step. Thus, when it comes to the fusion in the subsequent channel encoder, it fails to combine the tokens in a meaningful way. Although this could potentially be mitigated by using a smaller token dimension or more layers in the channel encoder, our results indicate that Channel Encoding is not as effective as our other proposed fusion architectures. Another drawback is the significant increase in storage requirements on the GPU as discussed in Chapter 4. Based on our experiments, the poor performance of this architecture does not justify the additional computational overhead.

From our conducted experiments, we therefore conclude that Early Fusion is, both in terms of metric results and computing complexity, the best among our proposed architectures. It is followed by SCT Fusion and Cross Attention Fusion which provide only slightly lower metric results but are accompanied by a larger computational complexity. MTC Fusion and Channel Encoding outperform single-modal baselines but fall behind the other fusion architectures.

## 6.3 Ablation Studies

In order to assess the influence of individual hyperparameters on the performance of our architectures, we conduct further ablation studies. As the training on the whole data set for 50 epochs is time consuming, we choose a random subset of one third of training and validation patches and train for 30 epochs, while continuing to test on the whole test set. Samples of all classes remain present in the subsets. For a fair comparison we also evaluate the architectures with the previously investigated hyperparameter setting on the reduced training and validation sets. For the sake of brevity, we evaluate here only the three best performing architectures from the previous section, namely Early Fusion, Cross Attention Fusion and SCT Fusion. The results for these architectures are shown in Table 6.5. For the remaining architectures the results can be found in the appendix.

Table 6.5: Metric results (in %) for training with 1/3 of the training and validation data with standard hyperparameters.

| Architecture | OA |  | MA | mIoU |
|---|---|---|---|---|
|  | A | B |  |  |
| Early Fusion | **90.06** | 91.07 | **79.12** | **66.60** |
| CA Fusion | 89.67 | **91.14** | 77.48 | 66.32 |
| SCT Fusion | 89.79 | 90.35 | 78.12 | 64.02 |

### Learning Rate

First, we investigate the learning rate. Our very first experiments on the whole data set were conducted with a learning rate of $10^{-3}$ which led to an error in the training procedure after around 20 epochs. Particularly, the training loss becomes ”not a number” potentially due to a division by zero in the backpropagation and too large a gradient. This was eventually mitigated by reducing the learning rate to $10^{-4}$ for which the results are reported in the previous section. However, the models achieved the lowest validation loss of about 0.3 already after 8-17 epochs, while the training loss was further reduced and converged to 0.15 at epoch 20-24.

Moreover, the validation loss curve fluctuates a lot as shown in Figure 6.2. Depending on the random initialization, the lowest validation loss is reached at different epochs for training and validation on the 1/3 subset. The plots show the validation loss curve for two different seeds until the lowest loss is reached. The right curve was obtained with our Early Fusion architecture whose results are reported in Table 6.5. It reaches the lowest loss at epoch 15. The left plot shows the same architecture and training procedure with a different seed, which results in the lowest loss being reached at epoch 9 already. The value of the lowest loss itself is very similar with 0.38 and 0.40.

In Figure 6.3, the left image shows the validation loss curve beyond its minimum at epoch 9 in red. One can see that the validation loss continues to go up and down without dropping below the lowest loss reached at epoch 9. This is because the training loss converges at epoch 20 but while the training set performance is continuously increased until then, the performance on the

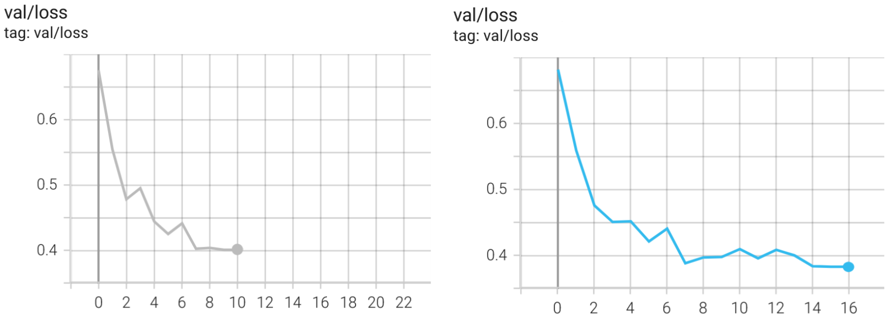


Figure 6.2: Validation loss per epoch of TSViT with Early Fusion until lowest loss value for two different seeds.

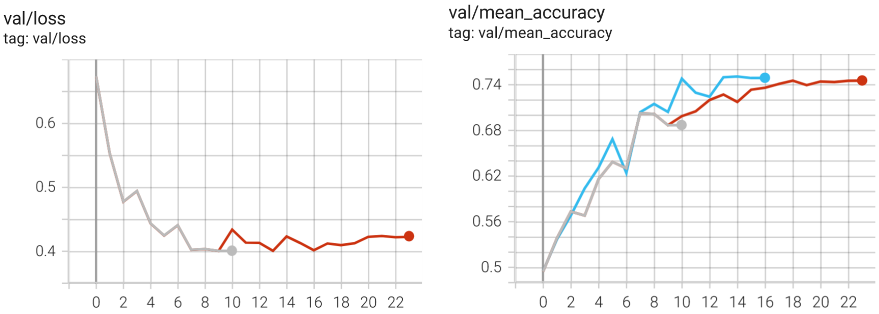


Figure 6.3: Left: Validation loss per epoch for one seed. Right: Validation Mean Accuracy per epoch for both seeds.

validation set stagnates earlier. A reason for this could be overfitting or a validation set that is too different from the training set.

The right plot in Figure 6.3 shows the corresponding Mean Accuracy on the validation set. The blue curve corresponds to the blue validation loss curve in Figure 6.2. The red curve shows that despite the validation loss having not decreased since epoch 9, the Mean Accuracy is still increasing. This is likely because the larger classes dominate the Cross-Entropy loss while smaller classes have less influence. In order to avoid large steps along the gradient and therefore large jumps in the validation loss curve, we reduce the learning rate. To this end, we train our models on the reduced sets with a learning rate of $10^{-5}$. The results are shown in Table 6.6.

While we indeed observe much smoother validation loss curves and a later convergence compared to a learning rate of $10^{-4}$, the smallest value is with 0.4-0.5 much higher. This is reflected in the metric results where we observe a decrease in Mean Accuracy by 9-17 % and in mIoU by 7-17%. The Overall Accuracy A is only 1% smaller, whereas Overall Accuracy B is reduced by 2-6% compared to the results for the larger learning rate in Table 6.5. As mean aggregated metrics drop, while Overall Accuracy stays high, especially when including the Background class which constitutes the second largest class in the training set, it seems that the classification performance is negatively affected mostly for classes with less samples.

Table 6.6: Metric results (in %) for training with 1/3 of the training and validation data with a learning rate of $10^{-5}$ and otherwise standard hyperparameters.

| Architecture | OA | | MA | mIoU |
|---|---|---|---|---|
| | A | B | | |
| Early Fusion | **88.68** | 85.63 | 64.56 | 55.34 |
| CA Fusion | 88.17 | 85.22 | 60.58 | 49.91 |
| SCT Fusion | 88.01 | **88.89** | **69.81** | **57.07** |

Indeed, when considering the confusion matrices for the test set classification, we find that especially the accuracies of the smallest classes, such as Other Agricultural Areas, Sugar Beet and Vegetables, are reduced. For all three architectures the accuracy of Other Agricultural Areas drops to zero. With Early Fusion and Cross Attention Fusion this is also the case for the class Sugar Beet, while SCT Fusion identifies a solid 77% of all samples in this class correctly.

When comparing the results of our three proposed architectures in Table 6.6, one can see that SCT Fusion indeed outperforms Early Fusion and Cross Attention Fusion in all metrics except Overall Accuracy A. Generally, the lower learning rate leads the models to converge to a suboptimal solution where mostly large classes are well identified as they affect the loss the most. SCT Fusion however, seems to be able to handle this better.

Converging to a suboptimal solution could potentially be mitigated by using a weighted Cross-Entropy loss in which samples affect the loss inversely proportional to the class size or Focal Loss which takes class imbalances into account as well [30].

### Temporal Encoder Depth

Since two of the best of our proposed fusion architectures, Cross Attention Fusion and SCT Fusion, fuse the modality specific features in the temporal encoder, we evaluate how its encoder depth, i.e. the amount of layers, affects their performance. For this, we consider a depth of 4 and 8 in addition to the previously evaluated depth of 6. The results are shown in Table 6.7.

From the results one can see that a temporal encoder depth of 6 is overall the best choice for all architectures. The differences between encoder depths are most distinguished for Early Fusion where the Overall Accuracy A is 1-2 % higher and Mean Accuracy and mIoU up to 3% compared to using 4 or 8 layers. This is an interesting development as more layers and thus a higher number of parameter usually goes along with an increase in classification performance [48]. Tarasiou et al. for example observed an increase in mIoU when using more layers in the temporal encoder for the SM TSViT [50].

The decreasing performance when increasing the number of layers from 6 to 8 with Early Fusion could be related to overfitting. However, the training and validation loss curves are similar for both configurations. They both reach the lowest validation loss at epoch 16 with values that are very close together with 0.3826 for 6 layers and 0.3843 for 8 layers. The unweighted Cross-Entropy loss is closely related to Overall Accuracy which is also within a small range of 88.38-88.75 for both configurations. The Mean Accuracy on the other hand shows larger differences. It is with 6 layers 2% higher than for 8 layers. This indicates that Early Fusion with 6

Table 6.7: Metric results (in %) for different temporal encoder depths on a 1/3 subset of training and validation set.

| Architecture | Temporal Depth | OA A | MA | mIoU |
|---|---|---|---|---|
| Early Fusion | 4 | 89.06 | 77.32 | 63.34 |
| | 6 | **90.06** | **79.12** | **66.60** |
| | 8 | 88.34 | 76.84 | 63.61 |
| CA Fusion | 4 | 89.56 | 77.37 | 63.87 |
| | 6 | **89.67** | 77.48 | **66.32** |
| | 8 | 89.26 | **77.82** | 63.41 |
| SCT Fusion | 4 | 89.36 | 75.65 | **64.32** |
| | 6 | **89.79** | **78.12** | 64.02 |
| | 8 | 89.58 | 77.35 | 64.11 |

layers is much better at identifying smaller classes. Moreover, it is more capable to generalize to the test set as shown in Table 6.7. It is possible that when using too many layers, their attention maps become similar which can decrease performance for deeper architectures as shown in [62] for ViTs. To investigate whether this is the case for our architectures as well, further analysis of the attention maps is required.

For Cross Attention and SCT Fusion this effect can also be observed, though it is less pronounced. While Early Fusion achieves the best results with a depth of 6 layers, its performance drops below that of the other two architectures when 4 or 8 layers are used. With 4 layers this could be rooted in the lower number of parameters that comes along with removing encoders. Additionally, architectures with one encoder per modality might be more stable to varying the encoder's depth due to their parallel processing of the modalities. This could lead to an effect similar to what is achieved with ensemble learning; Each encoder extracts different features and by combining them, the overall performance is increased.

### Spatial Encoder Depth

In the beginning of this thesis, the use of the SM TSViT was motivated by the fact that it purely uses self-attention to extract both spatial and temporal features. Previous state-of-the-art architectures process only the temporal dimension with self-attention but rely on convolutional modules for spatial feature extraction or do not take the spatial dimension into account at all. These were shown to perform worse on single-modal data [51].

For the SM TSViT it was shown that its performance also drops significantly when no spatial encoder is used [50]. In the following, we evaluate the influence of spatial encoder depth on the performance of our multi-modal TSViT architectures. Specifically, we compare using no spatial encoder to a spatial encoder depth of 2 and 4.

From the metric results in Table 6.8 one can see that the overall best results are achieved when using a spatial encoder of depth 2. However, when removing the spatial encoder, all three architectures are still able to give a good performance. Cross Attention and SCT Fusion

Table 6.8: Metric results (in %) for different spatial encoder depths on a 1/3 subset of training and validation set.

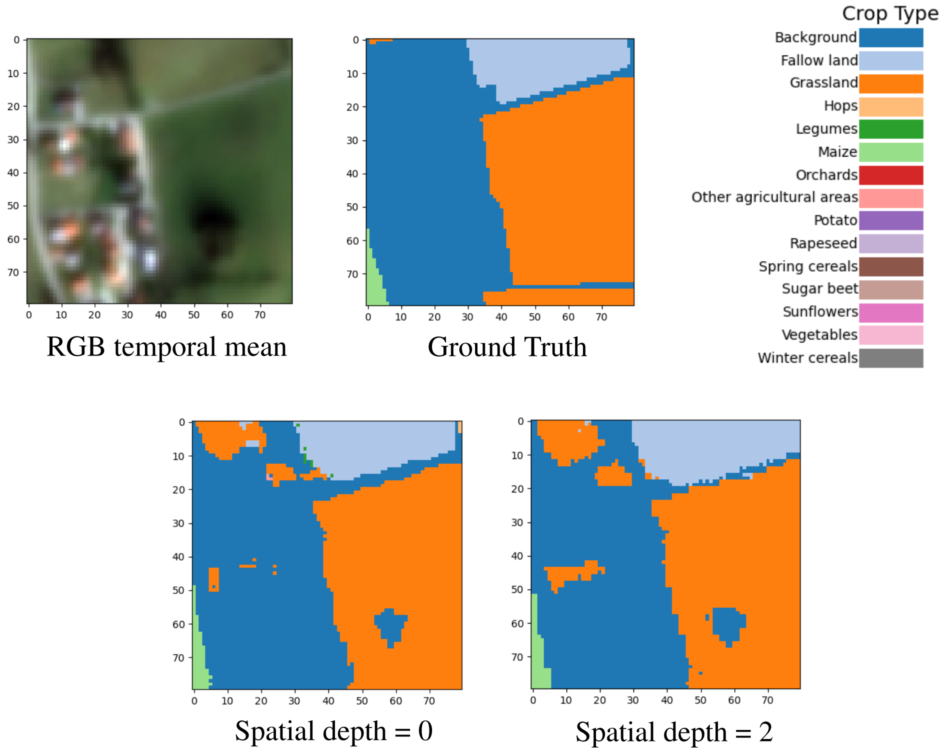
| Architecture | Spatial Depth | OA A | MA | mIoU |
|---|---|---|---|---|
| Early Fusion | 0 | 89.23 | 75.05 | 60.86 |
| | 2 | **90.06** | **79.12** | **66.60** |
| | 4 | 88.87 | 78.5 | 62.96 |
| CA Fusion | 0 | **90.26** | 75.71 | 63.01 |
| | 2 | 89.67 | 77.48 | **66.32** |
| | 4 | 90.25 | **78.26** | 65.9 |
| SCT Fusion | 0 | **89.89** | 73.31 | 60.94 |
| | 2 | 89.79 | **78.12** | **64.02** |
| | 4 | 89.13 | 76.85 | 63.99 |

even achieve their highest Overall Accuracy among all three evaluated configurations without a spatial encoder. This shows that only the temporal features extracted from a 2×2 pixel patch contain enough information to correctly classify a large fraction of samples. However, without a spatial encoder the Mean Accuracy is 2-5% decreased for all three architectures, indicating that mostly smaller classes are benefitting from a spatial encoder. With Cross Attention Fusion for example, the class-wise accuracies show that for Fallow Land 10% and for Sunflowers 22% less samples are correctly classified without a spatial encoder compared to an encoder with 4 layers.

It should be noted that this drop in performance could also be rooted in a reduced number of parameters. The number of parameters in the MM TSViT containing a spatial encoder of depth 2 is 2.4 million with Early Fusion and 5.5 million with both Cross Attention and SCT Fusion. Without a spatial encoder this number is reduced to 1.6 million for Early Fusion and 4.8 million for Cross Attention and SCT Fusion.

Figure 6.4 shows the predicted segmentation maps for an example patch from the test set created with Cross Attention Fusion, once with no spatial encoder and once with a spatial encoder depth of 2. What immediately stands out for the prediction made with no spatial encoder, is that there are a few pixels classified as classes that do not occur in the ground truth of the patch at all. At the left border of the Fallow field for example the model predicts Legumes (dark green) and in the top right corner it classifies some pixels as Hops (light orange). This is not the case for the other configuration that uses a spatial encoder of depth 2 which shows that considering the surrounding pixels is important to remove outliers in the predictions and make them more spatially coherent.

On the other hand, the architecture with a spatial encoder seems to be less capable of preserving pixel-specific features within a $2 \times 2$ patch that is mapped to one token. In the predicted segmentation map, one can see that the model tends to give the same label to all pixels within a $2 \times 2$ patch which results in a much rougher outline of field borders. This is especially notable at the border of the Maize field in the bottom left corner of the patch. As a result, the spatial resolution of the prediction is seemingly reduced. When no spatial encoder is used, it is more often the case that the four pixels mapped to one token are assigned different labels.



RGB temporal mean

Ground Truth

Spatial depth = 0

Spatial depth = 2

Figure 6.4: Predicted segmentation maps for a patch of the test set with Cross Attention Fusion once without a spatial encoder and once with a spatial encoder depth of 2.

This could be due to the spatial position encoding added to each token. It affects all 4 pixels in the same way and might potentially erase pixel-specific information within the token. Another possibility is that it is difficult for the spatial encoder to modify only specific pixels of a patch in the spatial attention module, as it operates on the whole token at once.

The spatial encoder operates on sequences of class tokens belonging to the same class from all spatial locations in the image. As the spatial attention mechanism consists of a weighted sum of value embeddings derived from the class tokens, it is only possible to modify the whole token and not only a portion of it. It follows that the class probabilities of either all 4 or none of the pixels can be enhanced or diminished. This can lead to all pixels within one 2×2 patch having the highest probability for the same class after the spatial encoder.

### Channel Encoder Depth

Our proposed Channel Encoding fusion architecture contains an additional encoder, for which we evaluate the effect of its depth on the performance. The results reported in Section 6.2 were obtained with a depth of 4. For the reduced training and validation set, we additionally evaluate the architecture with 2,6 and 8 layers. The results are shown in Table 6.9.

From the results, one can see that there is no clear best choice for the channel encoder depth.

Table 6.9: Metric results (in %) for different channel encoder depths from a 1/3 subset of training and validation set. Additionally, the epoch in which the lowest validation loss is reached is given.

| Channel Depth | OA |  | MA | mIoU | epoch with lowest val loss |
|---|---|---|---|---|---|
|  | A | B |  |  |  |
| 2 | 89.15 | 90.41 | **77.55** | 64.58 | 22 |
| 4 | **89.82** | 90.10 | 75.37 | **64.90** | 11 |
| 6 | 88.85 | **90.76** | 76.50 | 63.66 | 18 |
| 8 | 89.51 | 88.71 | 74.73 | 62.03 | 13 |

With 4 layers, the highest Overall Accuracy A and mIoU is obtained. The highest Mean Accuracy however, is achieved using only 2 layers. The best Overall Accuracy B is measured when using 6 layers. Despite this, its Overall Accuracy A is the lowest among all four configurations while the Mean Accuracy is fairly high with 76.5%. This indicates a good performance on the smaller classes, as Overall Accuracy B does not take samples of the Background class into account.

When considering the epoch in which the lowest validation loss is reached, we find that models for which this epoch is smaller, have a high Overall Accuracy A but a low Mean Accuracy and mIoU. With 2 layers, for example, the lowest validation loss is reached after 22 epochs and the Mean Accuracy and mIoU are high with 77.55% and 64.58%, respectively. The models that reach the lowest loss sooner, show lower values in the mean aggregated metrics. With 8 layers, for instance, the lowest validation loss is reached at epoch 13 while Mean Accuracy and mIoU are 2-3% smaller than with 2 layers. The Overall Accuracy A on the other hand is similar for both models. This further supports our previous finding that using an unweighted Cross-Entropy loss puts more emphasis on large classes. The resulting loss therefore correlates with the Overall Accuracy A. Moreover, smaller classes are learned in later epochs which results in a further increase of Mean Accuracy when the validation loss is already stagnating as shown in Figure 6.3.

For future work, the models can be trained until convergence instead of using their state at the epoch in which the lowest validation loss is reached. Moreover, as mentioned before, a loss function that takes class imbalance into account could be evaluated.

## 6.4 Comparison to State Of The Art

Finally, we also compare the performance of our proposed architectures to some architectures that represent the state of the art in crop mapping from SITS.

For one, we evaluate a Convolutional Temporal Attention Network named U-TAE, that was mentioned in Chapter 3. This architecture consists of a U-Net with a Lightweight Temporal Attention Encoder (L-TAE) at its lowest layer [13]. Initially, it was developed for single-modal crop mapping but was later evaluated for multi-modal input consisting of radar and optical data [15]. The input to the architecture is a SITS $X \in \mathbb{R}^{H \times W \times T \times C}$. In the down-sampling branch of the U-Net the images of each time point $t \in [T]$ are processed separately in parallel. Each down-sampling layer reduces the spatial resolution by factor 2. After $L$ layers the time series of each of the $H/2^L \times W/2^L$ pixels are gathered and processed by the L-TAE. This module groups the channels per pixel in the feature map and computes self-attention over the temporal dimension. The attention weights are used to collapse the temporal dimension such that the output that is up-sampled in the second branch of the U-Net has no temporal dimension anymore. Moreover, an attention map is created from the attention weights of each pixel. It is also up-sampled along the second U-Net branch and used to collapse the temporal dimension in the short-cut connections that incorporate the feature maps from the down-sampling branch into the up-sampling branch. The resulting feature map after the final layer has again the initial spatial resolution $H \times W$ and is mapped to class probabilities. We use the standard parameters for the U-TAE which consist of four encoder and decoder layers each with 64, 64, 64 and 128 kernels in the encoder and 32, 32, 64 and 128 kernels in the four layers of the decoder. Moreover, 16 attention heads are used with keys and queries of dimension 4 [14].

The second architecture that we evaluate is the temporal CNN (tempCNN) [38]. It contains convolutional layers with one dimensional kernels that convolve only the temporal dimension for each pixel of the input SITS, followed by a dense layer and a Softmax to map the output to class probabilities. We use 3 convolutional layers, each consisting of a temporal convolution with a kernel of size 7, a batch norm, a ReLU activation function and a dropout layer.

Both architectures are trained for 50 epochs on the whole dataset with Early Fusion (EF).

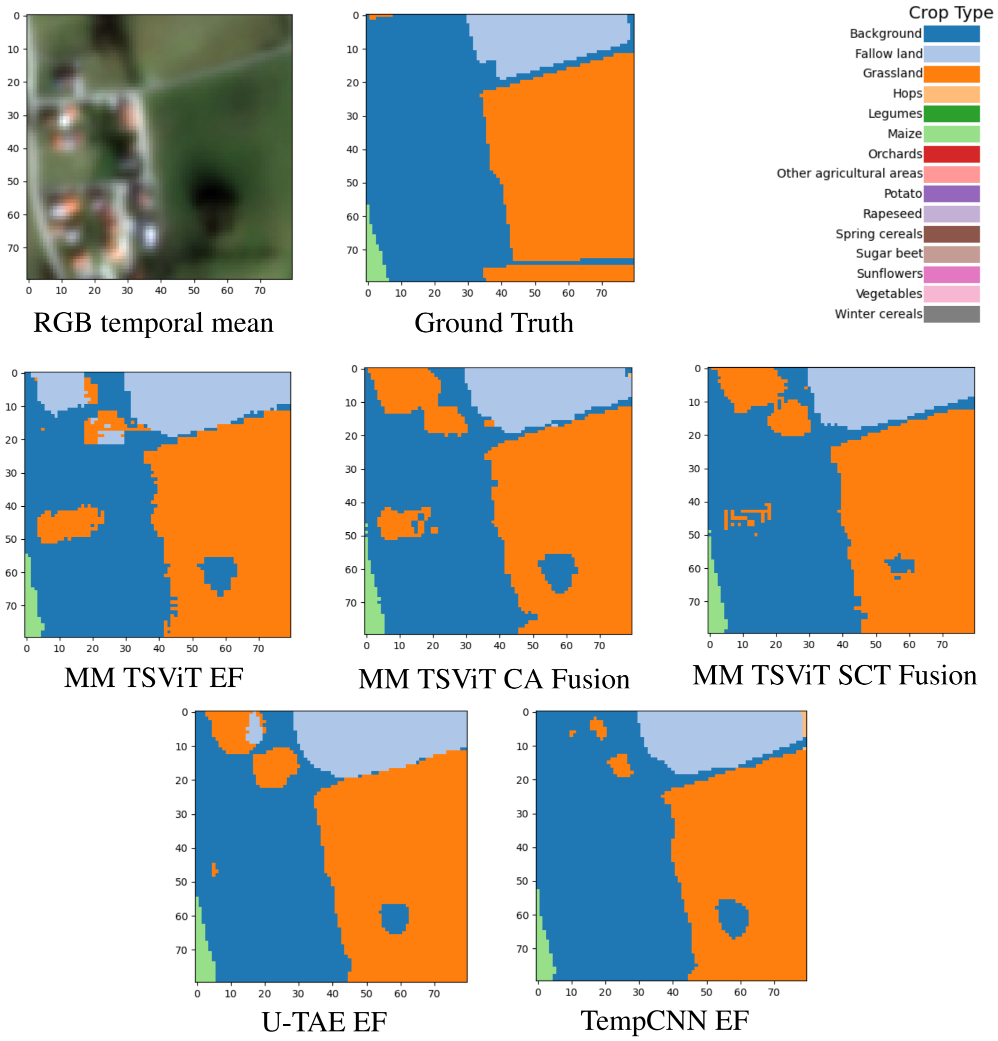
Table 6.10: Results (in %) achieved by U-TAE with EF, TempCNN with EF and our proposed MM TSViT architectures.

| Metrics | U-TAE EF [15] | TempCNN EF [38] | Proposed MM TSViT | | |
|---|---|---|---|---|---|
| | | | EF | SCT Fusion | CA Fusion |
| OA A | 89.75 | **91.31** | 89.79 | 90.50 | 90.39 |
| OA B | 90.24 | 90.17 | **92.66** | 92.52 | 92.57 |
| MA | 67.70 | 78.22 | **80.34** | 79.72 | 79.38 |
| mIoU | 55.68 | 66.66 | 66.96 | **68.39** | 66.66 |

The results are shown in Table 6.10 with the results of the best of our proposed MM TSViT fusion methods from Section 6.2, Early Fusion, Cross Attention Fusion and SCT Fusion for comparison. One can see that our proposed architectures outperform U-TAE EF in all metrics. The margin is especially large for mean aggregated metrics. The Mean Accuracy of our architectures is 22% higher, while mIoU is increased by 11%. This shows a clear superiority of our architectures over the Convolutional Temporal Attention Network.

The U-TAE EF's large difference between high Overall Accuracy and low mean aggregated metrics suggest a low performance on underrepresented classes. Indeed, the confusion matrix shows that none of the samples from the smallest classes Orchards, Sugar Beets and Vegetables are identified correctly. For the larger classes the performance is similar to the other architectures.

Our architectures outperform the TempCNN EF in almost all metrics by a margin of up to 2%. The only exception is Overall Accuracy A, for which the TempCNN EF achieves a 1% higher value. This indicates a better classification performance of the TempCNN EF on samples of the Background class. When considering the Background accuracy, one can indeed see a difference between the architectures. Our MM TSViT's identify 87-88% percent of samples in this class correctly. With the U-TAE EF this value is increased to 89% and with the TempCNN EF even to 93%.



RGB temporal mean

Ground Truth

MM TSViT EF

MM TSViT CA Fusion

MM TSViT SCT Fusion

U-TAE EF

TempCNN EF

Figure 6.5: Predicted classifications for a sample patch from the test set for the evaluated architectures.

TempCNN EF and our proposed fusion architectures show very similar class-wise accuracies. The most notable difference is observed for the class Other Agricultural Areas. With our TSViT with Early Fusion an accuracy of 66% is achieved. The TempCNN EF is only able to identify 43% correctly and misclassifies 39% of samples as Background. This could be related to a general tendency to classify ambiguous samples as Background, which was also observed for MTC

Fusion in Section 6.2. This hypothesis is further supported by the high Background accuracy of both architectures.

Figure 6.5 shows the classification maps obtained with each architecture for a patch from the test set. One can see that the U-TAE produces a smoothed outline of field borders which makes them appear more round. None of the architectures detects the small stripe of Background between the two Grassland areas in the bottom right corner. However, the U-TAE EF is also not able to detect large parts of the Background stripe between the Grassland and Fallow Land areas. This is likely due to the down- and subsequent up-sampling of the spatial resolution in the U-Net which leads to a loss of fine structures.

As mentioned before, the TempCNN is particularly strong when it comes to classifying Background samples. This is reflected in the predicted classification map. Only three small spots within the Background area are misclassified.

Interestingly, the TempCNN detects a small area of Hops in the top right corner of the patch. The same behavior was also observed for the TSViT with Cross Attention Fusion without the spatial encoder shown in Figure 6.4. Both architectures do not take the spatial dimension into account. The TSViT without the spatial encoder processes tokens representing $2 \times 2$ pixels, whereas the TempCNN operates on individual pixels. This further supports the hypothesis that considering spatial correlations reduces noise in the classification.

Upon closer inspection, one can also find a single pixel that is classified as Hops in the predicted classification map of Cross Attention Fusion. Moreover, the spatially neighboring patch towards the right contains a Hops field that reaches to the top left patch corner. It is very well possible that the satellite images corresponding to the patch in Figure 6.5 also contain some reflectance values of this Hops field. The reason why it does not appear in the ground truth of this patch could be small errors in the co-registration of the satellite images or in the field border coordinates of the provided IACS reference data. Nevertheless, one could still interpret these pixels as noise in the classification, as they are not labeled as Hops in the ground truth.

# 7 Conclusion and Discussion

In this thesis we investigated the potential of the TSViT architecture to be adapted to multi-modal fusion of SAR and multi-spectral data for pixel-based crop mapping. To this end, we developed several architectures based on the TSViT that enable it to extract complementing features from SITS of multiple modalities and fuse them to produce a single prediction. We evaluated our architectures on a crop mapping dataset containing SITS from Sentinel-1, Sentinel-2 and Planet Fusion and compared their performances with each other, to single-modal baselines and two existing crop mapping architectures from the literature.

The proposed architectures can be divided into three categories: The first does not modify the SM TSViT architecture but instead directly fuses the input. This category contains only Early Fusion which concatenates input SITS from all modalities in the channel dimension to produce a multi-modal SITS that can directly be processed by the SM TSViT.

The second type of architectures modifies the creation of the token time series that is passed to the temporal encoder. With Modality Token Concatenation, each modality is tokenized separately. Then, the resulting token time series are concatenated to a single token sequence and subsequently passed to the temporal encoder. Another architecture in this category is Channel Encoding. Here, not only each modality but also each channel is mapped to a separate token. The modalities are then fused in an additional channel encoder that computes self-attention over all channels per spatial and temporal location. Averaging the resulting token sequence condenses its information and allows for creating a temporal token sequence that is passed to the temporal encoder.

Finally, the third category provides a temporal encoder for each modality between which information is exchanged. With Synchronized Class Token Fusion this is achieved by extracting and averaging the class tokens from the encoders after each layer. The mean aggregated class tokens are then prepended to the modality-specific feature tokens of each encoder. In the second architecture in this category, Cross Attention Fusion, a modality-specific encoder receives queries from all other encoders, computes cross-attention and averages the resulting attention weights.

All architectures are evaluated on the EOekoLand crop mapping data set provided by the Thünen Institute of Farm Economics. The dataset contains SITS of three years and two geographic areas from Sentinel-1, Sentinel-2 and the Planet Fusion product, which constitute the three modalities for our evaluation. The provided reference data contains 15 different crop type classes and a Background class.

The results show that all proposed fusion architectures outperform the single-modal baselines. This proves that all architectures are able to extract meaningful features from individual modalities and combine them in a way that is beneficial for the task of crop mapping. Early Fusion gives the best overall performance while also having the lowest computational complexity among the proposed architectures. Synchronized Class Token Fusion achieves similar metric results and moreover the highest class-wise accuracies for the most classes. Cross Attention Fusion follows with only small reductions in metric results. However, the latter two architectures have the highest number of parameters and thus a much larger computational complexity than Early Fusion. Therefore, they are not necessarily favorable. Finally, the experimental results show that Modality Token Concatenation Fusion and Channel Encoding are not able to level with the other architectures.

In a detailed ablation study we investigated the effect of different hyperparameter settings on the classification performance of the best three architectures Early Fusion, Cross Attention Fusion and Synchronized Class Token Fusion. When varying the depth of the temporal and spatial encoders, we found that both using too little as well as too many layers reduces classification performance for all three architectures. The overall best performance was observed with 6 temporal and 2 spatial Transformer layers. Moreover, we found that removing the spatial encoder altogether has pros and cons: On the one hand, detected field borders are much smoother. On the other hand, more noise is introduced in the predictions due to ignoring spatial relationships among tokens. As the Channel Encoding architecture contains an additional encoder, we also study the changes in its performance upon varying the encoder depth. The results do not show a clear favorable depth for this encoder but instead limitations of the chosen loss function that can be further investigated in future work.

Finally, the proposed multi-modal fusion architectures are compared to two other crop mapping architectures. The first is a Convolutional Temporal Attention Network and the second a one-dimensional temporal CNN. Both were initially proposed for crop mapping from multi-spectral data and one of them has already been evaluated on multi-modal data with Early Fusion in previous work. For the comparison to our architectures we use Early Fusion to enable both architectures to fuse multi-modal input data. The results show a clear superiority of our architectures over the Convolutional Temporal Attention Network and slightly better performance than the temporal CNN. These observations further affirm the effectiveness of our proposed architectures.

For future work, one could investigate the contribution of each modality in the fusion process. It is especially interesting to study the effectiveness of the Planet Fusion modality in combination with Sentinel-2. Both contain multi-spectral data with Planet Fusion containing only a subset of the bands of Sentinel-2. Planet Fusion, however, is provided in a higher spatial and temporal resolution. The increased spatial resolution could be useful in creating finer crop maps which is especially beneficial for identifying smaller fields and field borders. To find out to which degree this modality contributes to the classification, one could compare the fusion of only Sentinel modalities resized to Planet Fusion resolution to the fusion of all three modalities. To improve the classification performance on underrepresented classes, future research could investigate the effectiveness of different loss functions such as weighted Cross Entropy loss and Focal Loss. Finally, it will be interesting to dive deeper into the contribution of each individual part of the architectures to the fusion process. This could help to understand the strengths of each of our fusion strategies and identify potential points for improvements.

# Bibliography

[1] European Space Agency. *Sentinel User Guides*. URL: https://sentinels.copernicus.eu/web/sentinel/user-guides/ (visited on 11/16/2023).

[2] Yujia Bao, Srinivasan Sivanandan, and Theofanis Karaletsos. “Channel Vision Transformers: An Image Is Worth C x 16 x 16 Words”. In: *UniReps: The First Workshop on Unifying Representations in Neural Models*. 2023.

[3] Lukas Blickensdoerfer et al. “Mapping of crop types and crop sequences with combined time series of Sentinel-1, Sentinel-2 and Landsat 8 data for Germany”. In: *Remote Sensing of Environment* 269 (2022), p. 112831. ISSN: 0034-4257.

[4] James Brinkhoff, Rasmus Houborg, and Brian W Dunn. “Rice ponding date detection in Australia using Sentinel-2 and Planet Fusion imagery”. In: *Agricultural Water Management* 273 (2022), p. 107907.

[5] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, June 2019, pp. 4171–4186.

[6] Laura Dingle Robertson et al. “C-band Synthetic Aperture Radar (SAR) Imagery for the Classification of Diverse Cropping Systems”. In: *International Journal of Remote Sensing* 41.24 (2020), pp. 9628–9649.

[7] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2020.

[8] Matthias Drusch et al. “Sentinel-2: ESA’s optical high-resolution mission for GMES operational services”. In: *Remote Sensing of Environment* 120 (2012), pp. 25–36.

[9] ESA. *User Guides - Sentinel-2 MSI*. URL: https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/overview (visited on 11/14/2023).

[10] David Frantz. “FORCE - Landsat + Sentinel-2 Analysis Ready Data and Beyond”. In: *Remote Sensing* 11.9 (2019), p. 1124.

[11] David Frantz et al. “An Operational Radiometric Landsat Preprocessing Framework for Large-area Time Series Applications”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.7 (2016), pp. 3928–3943.

[12] Krishna Karthik Gadiraju et al. “Multimodal Deep Learning based Crop Classification using Multispectral and Multitemporal Satellite Imagery”. In: *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3234–3242.

[13] Vivien Sainte Fare Garnot and Loic Landrieu. "Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks". In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4872–4881.

[14] Vivien Sainte Fare Garnot and Loic Landrieu. *Supplementary Material for: Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks*.

[15] 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.

[16] Vivien Sainte Fare Garnot et al. "Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-attention". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12325–12334.

[17] Getachew Workineh Gella, Wietske Bijker, and Mariana Belgiu. "Mapping Crop Types in Complex Farming Areas using SAR Imagery with Dynamic Time Warping". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 175 (2021), pp. 171–183.

[18] Gohar Ghazaryan et al. "Crop Yield Estimation using Multi-source Satellite Image Series and Deep Learning". In: *IEEE International Geoscience and Remote Sensing Symposium*. 2020, pp. 5163–5166.

[19] David Sebastian Hoffmann, Kai Norman Clasen, and Begüm Demir. "Transformer-Based Multi-Modal Learning for Multi-Label Remote Sensing Image Classification". In: *IEEE International Geoscience and Remote Sensing Symposium*. 2023, pp. 4891–4894.

[20] Shunping Ji et al. "3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images". In: *Remote Sensing* 10.1 (2018), p. 75.

[21] Priyabrata Karmakar et al. "Crop Monitoring by Multimodal Remote Sensing: A review". In: *Remote Sensing Applications: Society and Environment* (2023), p. 101093.

[22] Diederik Pieter Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representatives* abs/1412.6980 (2014).

[23] Lukas Kondmann et al. "DENETHOR: The DynamicEarthNET dataset for Harmonized, Inter-operable, Analysis-ready, daily Crop Monitoring from space". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[24] Daniel Kpienbaareh et al. "Crop Type and Land Cover Mapping in Northern Malawi using the Integration of Sentinel-1, Sentinel-2, and Planetscope Satellite Data". In: *Remote Sensing* 13.4 (2021), p. 700.

[25] Planet Labs. *Planet Fusion Monitoring Technical Specification*. URL: `https://assets.planet.com/docs/Planet_fusion_specification_March_2021.pdf` (visited on 11/16/2023).

[26] Planet Labs. *Planet Scope Product Specifications*. URL: `https://assets.planet.com/docs/Planet_PSScene_Imagery_Product_Spec_letter_screen.pdf` (visited on 12/16/2023).

[27] Kaiyuan Li et al. “Multi-branch Self-learning Vision Transformer (MSViT) for Crop Type Mapping with Optical-SAR Time-Series”. In: *Computers and Electronics in Agriculture* 203 (2022), p. 107497.

[28] Kaiyuan Li et al. “Predicting Crop Growth Patterns with Spatial–Temporal Deep Feature Exploration for Early Mapping”. In: *Remote Sensing* 15.13 (2023), p. 3285.

[29] Ruilong Li et al. “AI Choreographer: Music conditioned 3D Dance Generation with aist++”. In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13401–13412.

[30] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.

[31] Ze Liu et al. “Swin transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[32] Jiasen Lu et al. “Vilbert: Pretraining Task-agnostic Visiolinguistic Representations for Vision-and-language Tasks”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[33] Dong Luo et al. “Utility of daily 3 m Planet Fusion Surface Reflectance data for tillage practice mapping with Deep Learning”. In: *Science of Remote Sensing* 7 (2023), p. 100085.

[34] Paulius Micikevicius et al. “Mixed Precision Training”. In: *International Conference on Learning Representations*. 2018.

[35] Arsha Nagrani et al. “Attention Bottlenecks for Multimodal Fusion”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14200–14213.

[36] United Nations. *Sustainable Development goal 2: Zero Hunger*. URL: `https://www.un.org/sustainabledevelopment/hunger/` (visited on 08/25/2023).

[37] Adam Paszke et al. “Pytorch: An Imperative Style, High-performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[38] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. “Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series”. In: *Remote Sensing* 11.5 (2019), p. 523.

[39] Claudio Persello et al. “Deep Learning and Earth Observation to support the Sustainable Development Goals: Current approaches, open challenges, and future opportunities”. In: *IEEE Geoscience and Remote Sensing Magazine* 10.2 (2022), pp. 172–200.

[40] Yang Qu et al. “Crop Mapping from Sentinel-1 Polarimetric Time-Series with a Deep Neural Network”. In: *Remote Sensing* 12.15 (2020), p. 2493.

[41] Tanzila Rahman, Mengyu Yang, and Leonid Sigal. “TriBERT: Human-centric Audio-visual Representation Learning”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 9774–9787.

[42] Swalpa Kumar Roy et al. “Multimodal Fusion Transformer for Remote Sensing Image Classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2023).

[43] 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.

[44] Marc Rußwurm and Marco Korner. “Temporal Vegetation Modelling using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-Spectral Satellite Images”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 11–19.

[45] Marcel Schwieder et al. “Mapping Brazilian Savanna Vegetation Gradients with Landsat Time Series”. In: *International Journal of Applied Earth Observation and Geoinformation* 52 (2016), pp. 361–370. ISSN: 1569-8432.

[46] Caglar Senaras et al. “Self-Supervised Learning for Crop Classification Using Planet Fusion”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 48 (2023), pp. 309–315.

[47] Derya Soydaner. “A Comparison of Optimization Algorithms for Deep Learning”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 34.13 (2020), p. 2052013.

[48] Robin Strudel et al. “Segmenter: Transformer for Semantic Segmentation”. In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7262–7272.

[49] Chen Sun et al. “Videobert: A joint model for Video and Language Representation Learning”. In: *IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7464–7473.

[50] Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. *Supplementary material for:” ViTs for SITS: Vision Transformers for Satellite Image Time Series”*.

[51] Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. “ViTs for SITS: Vision Transformers for Satellite Image Time Series”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 10418–10428.

[52] Katalin Tóth and Andrius Kučas. “Spatial Information in European Agricultural Data Management. Requirements and Interoperability supported by a Domain Model”. In: *Land Use Policy* 57 (2016), pp. 64–79.

[53] Yao-Hung Hubert Tsai et al. “Multimodal Transformer for Unaligned Multimodal Language Sequences”. In: *Annual Meeting of the Association for Computational Linguistics*. Vol. 57. NIH Public Access. 2019, 6558—6569.

[54] Gabriel Tseng et al. “Lightweight, Pre-trained Transformers for Remote Sensing Time-series”. In: *arXiv preprint arXiv:2304.14065* (2023).

[55] Rahat Tufail et al. “A Machine Learning Approach for Accurate Crop Type Mapping using Combined SAR and Optical Time Series Data”. In: *Advances in Space Research* 69.1 (2022), pp. 331–346.

[56] Mehmet Ozgur Turkoglu et al. “Crop Mapping from Image Time Series: Deep Learning with Multi-scale Label Hierarchies”. In: *Remote Sensing of Environment* 264 (2021), p. 112603.

[57] Mehmet Ozgur Turkoglu et al. “Gating revisited: Deep multi-layer RNNs that can be trained”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.8 (2021), pp. 4081–4092.

[58] Mustafa Ustuner et al. “Crop type Classification using Vegetation Indices of Rapideye Imagery”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 40 (2014), pp. 195–198.

[59] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[60] Frank Weilandt et al. “Early Crop Classification via Multi-Modal Satellite Data Fusion and Temporal Attention”. In: *Remote Sensing* 15.3 (2023), p. 799.

[61] Yuan Yuan et al. “SITS-Former: A Pre-trained Spatio-Spectral-Temporal Representation Model for Sentinel-2 Time Series Classification”. In: *International Journal of Applied Earth Observation and Geoinformation* 106 (2022), p. 102651.

[62] Daquan Zhou et al. “Deepvit: Towards deeper Vision Transformer”. In: *arXiv preprint arXiv:2103.11886* (2021).

# Appendix

Table .1: Metric results (in %) for different learning rates on a 1/3 subset of training and validation set.

| Architecture | Learning rate | OA |  | MA | mIoU |
|---|---|---|---|---|---|
|  |  | A | B |  |  |
| Early Fusion | $10^{-4}$ | **90.06** | **91.07** | **79.12** | **66.60** |
|  | $10^{-5}$ | 88.68 | 85.63 | 64.56 | 55.34 |
| MTC | $10^{-4}$ | **89.67** | **89.70** | **77.03** | **62.68** |
|  | $10^{-5}$ | 87.31 | 85.10 | 57.81 | 49.26 |
| Cross Attention | $10^{-4}$ | **89.67** | **91.14** | **77.48** | **66.32** |
|  | $10^{-5}$ | 88.17 | 85.22 | 60.58 | 49.91 |
| SCT Fusion | $10^{-4}$ | **89.79** | **90.35** | **78.12** | **64.02** |
|  | $10^{-5}$ | 88.01 | 88.89 | 69.81 | 57.07 |
| Channel Encoder | $10^{-4}$ | **89.82** | **90.10** | **75.37** | **64.90** |
|  | $10^{-5}$ | 88.61 | 86.48 | 61.11 | 51.87 |

Table .2: Metric results (in %) for different temporal encoder depths on a 1/3 subset of training and validation set.

| Architecture | Temporal Depth | OA |  | MA | mIoU |
|---|---|---|---|---|---|
|  |  | A | B |  |  |
| Early Fusion | 4 | 89.06 | 90.59 | 77.32 | 63.34 |
|  | 6 | **90.06** | 91.07 | **79.12** | **66.60** |
|  | 8 | 88.34 | **91.19** | 76.84 | 63.61 |
| MTC Fusion | 4 | 89.00 | **90.76** | **77.20** | 62.52 |
|  | 6 | **89.67** | 89.70 | 77.03 | **62.68** |
|  | 8 | 89.21 | 89.55 | 75.98 | 60.42 |
| Cross Attention | 4 | 89.56 | 90.66 | 77.37 | 63.87 |
|  | 6 | **89.67** | **91.14** | 77.48 | **66.32** |
|  | 8 | 89.26 | 90.45 | **77.82** | 63.41 |
| SCT Fusion | 4 | 89.36 | 89.47 | 75.65 | **64.32** |
|  | 6 | **89.79** | 90.35 | **78.12** | 64.02 |
|  | 8 | 89.58 | **90.37** | 77.35 | 64.11 |
| Channel Encoding | 4 | 88.89 | 88.96 | 74.39 | 63.06 |
|  | 6 | 89.82 | 90.10 | 75.37 | **64.90** |
|  | 8 | **89.95** | **90.58** | **78.09** | 64.38 |

Table .3: Metric results (in %) for different spatial encoder depths on a 1/3 subset of training and validation set.

| Architecture | Spatial Depth | OA |  | MA | mIoU |
|---|---|---|---|---|---|
|  |  | A | B |  |  |
| Early Fusion | 0 | 89.23 | 90.31 | 75.05 | 60.86 |
|  | 2 | **90.06** | **91.07** | **79.12** | **66.60** |
|  | 4 | 88.87 | 90.02 | 78.5 | 62.96 |
| MTC Fusion | 0 | 88.96 | 89.12 | 75.63 | 59.08 |
|  | 2 | **89.67** | 89.70 | 77.03 | 62.68 |
|  | 4 | 89.17 | **89.99** | **77.24** | **62.86** |
| CA Fusion | 0 | **90.26** | 90.83 | 75.71 | 63.01 |
|  | 2 | 89.67 | **91.14** | 77.48 | **66.32** |
|  | 4 | 90.25 | **91.14** | **78.26** | 65.9 |
| SCT Fusion | 0 | **89.89** | 89.69 | 73.31 | 60.94 |
|  | 2 | 89.79 | **90.35** | **78.12** | **64.02** |
|  | 4 | 89.13 | 89.74 | 76.85 | 63.99 |
| Channel Encoding | 0 | 88.98 | 89.13 | 75.08 | 60.02 |
|  | 2 | **89.82** | **90.10** | **75.37** | **64.90** |
|  | 4 | 87.56 | 89.24 | 73.65 | 61.74 |