CNN, RNN, or ViT? An Evaluation of Different Deep Learning Architectures for Spatio-Temporal Representation of Sentinel Time Series

Linying Zhao and Shunping Ji, *Senior Member, IEEE*

***Abstract*—Rich information in multitemporal satellite images can facilitate pixel-level land cover classification. However, what is the most suitable deep learning architecture for high-dimension spatio-temporal representation of remote sensing time series remains unclear. In this study, we theoretically analyzed the different mechanisms of the different deep learning structures, including the commonly used convolutional neural network (CNN), the high-dimension CNN [three-dimensional (3-D) CNN], the recurrent neural network, and the newest vision transformer (ViT), with regard to learning and representing the temporal information for spatio-temporal data. The performance of the different models was comprehensively evaluated on large-scale Sentinel-1 and Sentinel-2 time-series images covering the whole of Slovenia. First, the 3-D CNN, long short-term memory (LSTM), and ViT, which all have specific structures that preserve temporal information, can effectively extract the spatio-temporal information, with the 3-D CNN and ViT showing the best performance. Second, the performance of the 2-D CNN, in which the temporal information is collapsed, is lower than that of the 3-D CNN, LSTM, and ViT but outperforms the conventional methods. Thirdly, using both optical and synthetic aperture radar (SAR) images performs almost the same as using only optical images, indicating that the information that can be extracted from optical images is sufficient for land-cover classification. However, when optical images are unavailable, SAR images can provide satisfactorily classification results. Finally, the modern deep learning methods can effectively overcome the disadvantages in imaging conditions where parts of an image or images of some periods are missing. The testing data are available at *gpcv.whu.edu.cn/data*.**

***Index Terms*—Deep learning models, land-cover classification, sentinel images, spatio-temporal remote sensing images, vision transformer.**

I. Introduction

THE land cover of forest, water, cultivated land, structures, buildings, etc., and their changes reflect the biophysical and civilization processes [1], and land-cover information can provide support for various applications, such as biodiversity change monitoring [2], disaster prevention [3], resource management [4], and urban planning [5]. Benefiting from the modern Earth observation (EO) systems, it is now possible to monitor land cover from high-revisit-rate satellite remote sensing data at a large scale. With regard to obtaining a high-accuracy and reliable land-cover classification map, the popular wisdom is to fully utilize the phenological information provided by multitemporal satellite images, e.g., the high-revisit Landsat and Sentinel image series, instead of using images of only one period. Over the past decades, computer technology and machine learning have been continuously developed, in an attempt to reduce the heavy and time-consuming human visual interpretation. In the field of remote sensing image processing and machine learning, deep learning based methods have now become the mainstream [6], [7], [8]. Among the different methods, there are several new tools that can be used for learning the spatio-temporal representation of images with the additional temporal dimension, such as convolutional neural networks (CNNs) [9] with high-dimension kernels, e.g., three-dimensional (3-D) CNN [10], recurrent neural networks (RNNs) [11], and the more recent vision transformer (ViT) models [12]. There have been some studies of applying one of these methods in land-cover classification [13], [14]. Nevertheless, which structure is most suitable one for spatio-temporal representation and land-cover classification from multi-temporal remote sensing images, including optical and SAR data, still remains an unsolved problem and requires comprehensive and in-depth investigation. In this article, we comprehensively investigate the performance of seven models based on three classic deep learning architectures that can extract the additional dimension (the temporal dimension) information when classifying land cover from Sentinel-1 and Sentinel-2 multi-temporal images. In the following section, we review the development of the related classification methods.

Before deep learning based feature extraction and representation were widely accepted, empirically handcrafted features were the usual choice for remote sensing image processing. For example, the normalized difference vegetation index [15], [16], normalized difference built-up index [17], [18], and the normalized difference water index [19], [20]. Such features are then input into a classification model (called the classifier), such as random forest (RF) [21], XGBoost [22], or support vector machine (SVM) [23], to achieve pixel-level classification. However, these features obviously are not all-cause factors for all kinds of land covers, and were originally designed for the multiband images of one period, and they lack the ability to describe the deep and complex spatio-temporal features of different land covers, other than processing temporal images with a simple linear feature concatenation [24].

Manuscript received 7 September 2022; revised 19 October 2022; accepted 26 October 2022. Date of publication 7 November 2022; date of current version 7 December 2022. This work was supported by the National Natural Science Foundation of China under Grant 42171430 and Grant 42030102. *(Corresponding author: Shunping Ji.)*

The authors are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: zhaolinying@whu.edu.cn; jishunping@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3219816

The recent deep learning methods employ multiple layers of neurons to automatically learn feature representations with multiple levels of abstraction in an end-to-end manner, with an imbedded classifier. Deep learning methods have become not only the mainstream in artificial intelligence and machine learning, but also in many other fields, such as remote sensing [25], [26], [27]. CNNs are the most widely used models in close-range and remote sensing image classification and object detection [28], [29], [30], while RNNs are mainly used to process time series. ViT is a burgeoning structure that can be used in both images and time series.

CNNs use layers of convolution kernels as units to extract and store deep features. They can be used to classify each pixel in a large-capacity remote sensing image, but this is obviously time-consuming. A CNN variant called the fully convolutional network (FCN) has now become the mainstream algorithm for semantic segmentation [31]. FCN models with 2-D convolution kernels are the most commonly used structures in remote sensing image processing, and they can also be used to process multitemporal images [32]. However, the 2-D CNN was initially designed for single images and cannot extract the temporal features well. Specifically, the 2-D features extracted from the multi-temporal input are summed along the temporal dimension, resulting in the collapse of the temporal information. Nevertheless, 3-D CNN can preserve the temporal information with the additional dimension. For example, Ji et al. [13] presented a 3-D CNN based method for crop type classification from multi-temporal satellite remote sensing images, which outperformed methods based on a 2-D CNN. Similarly, Vasit et al. [33] compared 2-D CNN and 3-D CNN models for field-scale yield prediction, and concluded that the 3-D CNN models can better extract spatio-temporal features.

RNN is another deep learning architecture that can extract multitemporal features. Variant models called the convolutional recurrent neural network (ConvRNN) models, which include the convolutional gated recurrent neural network (ConvGRU) [34] and convolutional long short-term memory neural network (ConvLSTM) [35], have been recently proposed to extract both the spatial and temporal features. For example, Rußwurm and Körner [14] utilized ConvRNN for land-cover classification from multitemporal Sentinel-2 images, and Teimouri et al. [36] combined an FCN and a ConvRNN network to classify crop types from Sentinel-1 satellite image time series.

Since their debut in 2017, transformer-based models have achieved state-of-the-art performances in the field of natural language processing (NLP) [37], [38], [39], [40]. More recently, with the emergence of ViT [12], ViT and its variants have demonstrated outstanding capabilities in the ImageNet classification challenge, due to their capability of long-range dependency modeling [41], [42]. ViT, with its ability to process time series, has also recently been used for videos [43], [44]. As a very new technology, there have been a few studies of ViT in the field of remote sensing. For example, Xu et al. [45] proposed a novel lightweight transformer model to improve the edge segmentation performance in remote sensing images, and Deng et al. [46] proposed a high-performance joint network that combines a CNN and ViT for high-resolution remote sensing image scene classification. There have been few studies that have used ViT to process temporal remote sensing images. For example, Gao et al. [47] designed a spatio-spectral vision transformer (SSViT) to extract sequential relationships from fused hyperspectral and multispectral images, and Chen et al. [48] presented a transformer-based structure for multi-temporal remote sensing interpretation. However, there are currently no studies that have comprehensively evaluated the performance of different ViT structures, as well as high-dimensional CNNs and RNNs, all of which have the ability to process time series in land-cover classification from multi-temporal remote sensing images.

In this article, we present an in-depth evaluation of the performance of the recent deep learning based methods, including high-dimensional CNN, RNN, and ViT, as well as the conventional methods, for land-cover classification from multitemporal remote sensing images, to discover which is the optimal structure. We also present an early attempt at exploring the feasibility of using various transformer-based models for multitemporal land-cover classification. We selected representative high-resolution Sentinel-2 optical and Sentinel-1 SAR time-series images as the data sources. The contributions of the article are summarized as follows.

1) A comprehensive comparison of the different methods for land-cover classification with Sentinel time series is presented. Two conventional machine learning methods, i.e., RF and XGBoost, three CNN-based models, i.e., U-Net (2-D U-Net) [49], DeepLabv3 [50] and three-dimensional U-Net (3-D U-Net) [51], an RNN variant called ConvLSTM, and three transformer-based models, i.e., TransUNet [41], TransBTS [40], and U-NEt TRansformers (UNETR) [52], are applied in a country-scale test area.
2) The specific spectral-temporal-spatial data structure for high-dimension CNN, RNN, and ViT is respectively designed, the different mechanisms of the different deep learning structures are theoretically analyzed.
3) The performance when using different data sources, i.e., SAR, optical, and their combination, is explored.
4) The practical reliability of different models when images of some periods are missing, which can be caused by satellite orbits and weather conditions, is also explored.

II. Data

After searching for high-quality labeled land-cover samples with Sentinel optical and SAR time-series images in a large and representative region, we finally selected the Republic of Slovenia (Slovenia) as the test area. Slovenia locates in central Europe, and lies between 45°30'18" N to 46°50'13" N and 13°31'35" E and 16°27'3" E, covering an area of 20 271 square kilometers. The country is situated at the intersection of four major European geographic regions (the Alps, the Dinarides, the Pannonian Plain, and the Mediterranean) and has very versatile terrain and climate [53]. Thanks to the diverse topography and climate, Slovenia features various land-cover types, including cultivated land, forest, grassland, water, wetland, and so on, with an adequately large area, making it suitable for the general land-cover classification purpose, as shown in Fig. 1.

The Slovenia dataset we used was made up of the available Sentinel-1 (S1) SAR images and Sentinel-2 (S2) optical images

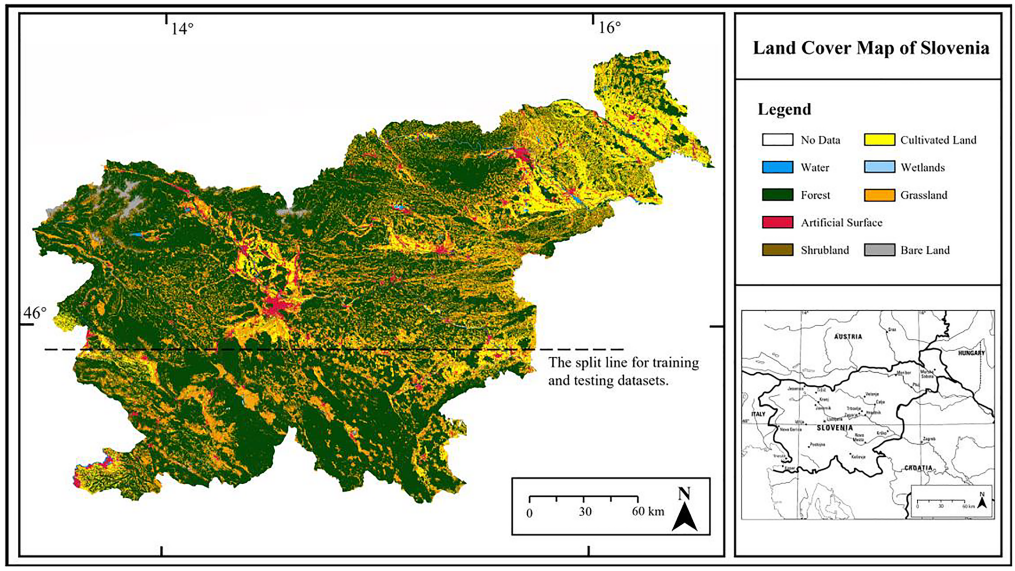


Fig. 1. Land-cover map of the study area. The test area is below the dotted line and the training area is above the line.

from January to December 2019. The S2 images were obtained from the Slovenia 2019 Land Cover Classification Dataset [54], which are level L1C images, with a total of six surface reflectance bands (2, 3, 4, 8, 11, and 12), which have been resampled at 10 m. We manually filtered out the images with cloud contamination, resulting in 12 clean (one per month) S2 images. Then, according to the date of the selected S2 images, 12 corresponding S1 images were further chosen.

All the S1 images were in Interferometric Wide Ground Range Detected mode with dual polarization (VV+VH) at a 10-m resolution, which we downloaded from the Sentinel Hub [55]. These images were preprocessed with calibrating to the beta coefficient, removing thermal noise, and orthorectification by the Copernicus digital elevation model. All the acquired S1 images were converted to the logarithmic dB scale and normalized to values between 0 and 255. We also applied nonlocal means filtering [56] to the S1 images to remove speckle noise, and then the Z-score normalization method to normalize the S1 and S2 data.

The corresponding ground truth for the land cover was also obtained from the Slovenia 2019 Land Cover Classification Dataset, at a 10-m spatial resolution. The labels included 936 patches, each with the size of 500×500 pixels, consisting of eight land-cover classes, which are cultivated land, forest, grassland, shrubland, water, wetland, artificial surface, and bare land.

The dataset, i.e., the images and the geo-aligned labels, was split into three parts according to the spatial distribution. In total, 297 patches from the south of the country were used for the testing, 130 patches are randomly selected from the north of the country as the validation set, and the remaining 509 patches from the north of the country were used for the training. For the conventional machine learning methods, which can only process images pixel by pixel, we randomly selected 2 299 549 pixels in the training dataset for all the types of land cover as their training data. Except for the wetland class, which had only 59 549 samples, each of the other classes had 320 000 sample points. The 297 patches mentioned above were used as the test data.

III. Methods

A. Processing Spectral-Temporal-Spatial Data for Modern Deep Learning Models

In this work, our input was Sentinel-1 and Sentinel-2 multitemporal remote sensing imagery, which is in a 4-D dimension spectral(1-D)-temporal(1-D)-spatial(2-D) format, with a range of $C\times T\times H\times W$, where C is the number of spectra, T is the length of the time series, and H and W are the height and width of the images, respectively. The goal was to predict the corresponding land-cover classification map with a size of $H\times W$.

1) Spectral-Temporal-Spatial Data for CNNs: A modern deep CNN architecture includes building blocks such as convolutional layers, pooling layers, and fully connected layers. The multilayer representation ability of the input is the most powerful part of the CNN, with each layer usually consisting of a linear operation and a nonlinear activation function, denoted as follows:

$$y^l = \sigma(\omega^l x^{l-1} + b^l) \tag{1}$$

where σ is the activation function; ω and b are the convolution kernel and bias of the current lth layer, respectively; and x is the input from the previous layer $l-1$. In 2-D CNNs, the convolution operations in a layer are usually computed along the spatial dimension, and multiple images (temporal images or spectral images) or feature maps are treated as the different channels of the next layer input, to output one feature map. Formally, the output at position (c, d) in the jth feature map of the ith layer, denoted as y_{ij}^{cd}, is given by the following:

$$\mathrm{y}_{ij}^{cd} = \sigma\left(\sum_n \sum_{p=1}^{M} \sum_{q=1}^{M} \omega_{ijn}^{pq} x_{(i-1)n}^{(c+p)(d+q)} + b_{ij}\right) \tag{2}$$

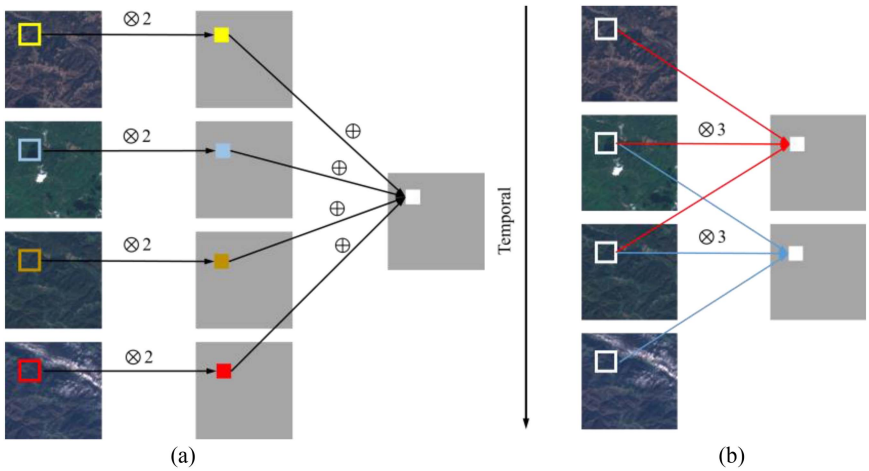


Fig. 2. Comparison of 2-D and 3-D convolution, from Ji et al. [13]. (a) 2-D convolution, where ⊗ 2 indicates the 2-D convolution operator and ⊕ indicates the sum operator, with the temporal information collapsed. (b) 3-D convolution, where ⊗3 indicates the 3-D convolution operator with length 3 in the temporal direction.

where b_{ij} is the bias for this feature map, and ω_{ijn}^{pq} is the nth shared $M\times M$ weighted kernel. To simplify this, we omit the notation in (1). Hence, for the 4-D spectral-temporal-spatial input ($C\times T\times H\times W$), the temporal and spectral dimensions should be combined into $CT\times H\times W$ before being fed into the network (the CT dimension is analogous to the red, green, and blue channels of a color image). It can be seen that the temporal information has been collapsed after the first convolutional layer, due to the addition operation.

3-D CNNs with 3-D convolution kernels can preserve the temporal information of the input, as shown in Fig. 2. The 3-D kernel can move along the spatial or temporal dimension and output cube-form data. Formally, the value at position (c, d, e) in the j-th feature map in the ith layer is given by the following:

$$y_{ij}^{cde} = \sigma\left(\sum_{n}\sum_{k=1}^{K}\sum_{p=1}^{M}\sum_{q=1}^{M}\omega_{ijn}^{kpq}x_{(i-1)n}^{(c+p)(d+q)(e+k)} + b_{ij}\right) \quad (3)$$

where K is the size of the temporal dimension, and ω_{ijn}^{kpq} is a 3-D tensor kernel. Clearly, the outputs of each layer are still in 4-D format. In this case, we can keep the 4-D spectral-temporal-spatial ($C\times T\times H\times W$) form of the input, and output a feature map with an additional temporal dimension, compared to 2-D convolution.

2) Spectral-Temporal-Spatial Data for ConvLSTM: In an RNN, the current output is determined by the previous state and the current input. Formally, the hidden layer output H_t at time t is given by the following:

$$H_t = \sigma(H_{t-1}, X_t; \Theta) \quad (4)$$

where σ is the nonlinear activation, Θ is the set of parameters, H_{t-1} is the previous output, and X_t is the current input. Therefore, the RNN can process the temporal development procedures by iteratively updating H_t [14]. However, while this simple version works fairly well for small sequence lengths, it cannot avoid the gradient vanishing problem during backpropagation in the training stage when dealing with long-term dependencies [51].

This issue has been addressed by ConvLSTM [29] by introducing a memory cell C_t at time step t. Each ConvLSTM unit has three gates: the input gate i, which decides whether the new input will be accumulated to the memory cell; the forget gate f, which determines whether the previous information will be forgotten; and the output gate o, which controls whether the latest memory cell will be transmitted to the output. The memory cell and gates can control the information flow and trap the gradient in the cell, to avoid gradient vanishing. ConvLSTM also introduces a convolution operation to encode the spatial information in input-to-state and state-to-state transitions, and can thus deal with long time series of spatio-temporal data.

A ConvLSTM operation at time t can be expressed as shown in the following equations:

$$i_t = \text{sigmoid}(W_{Xi} * X_t + W_{Hi} * H_{t-1} + b_i) \quad (5)$$

$$f_t = \text{sigmoid}(W_{Xf} * X_t + W_{Hf} * H_{t-1} + b_f) \quad (6)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{XC} * X_t + W_{HC} * H_{t-1} + b_C) \quad (7)$$

$$o_t = \text{sigmoid}(W_{Xo} * X_t + W_{Ho} * H_{t-1} + b_o) \quad (8)$$

$$H_t = o_t \circ \tanh(C_t) \quad (9)$$

where X_t is the input of the current cell, which in this work is images of the t-th month with the size of $C\times H\times W$. H_{t-1} and C_{t-1} are the output and state of the previous cell in ConvLSTM, respectively. The initial output and state, H_0 and C_0, are initialized with zeros. The symbol "$*$" is the convolution operation, and "$\circ$" denotes the Hadamard product. Sigmoid and tanh denote sigmoid and hyperbolic tangent activation functions, respectively. Since the input is an image of one period, W denotes the 2-D convolution filter. For the 4-D spectral-temporal-spatial input ($C\times T\times H\times W$), the ConvLSTM unit accepts the tth month's image ($C\times H\times W$) together with the output and state of the previous unit as the input each time, and iterates T times in total, as shown in Fig. 3.

3) Spectral-Temporal-Spatial Data for ViT: The original transformer for NLP accepts the input in 1-D sequence format,

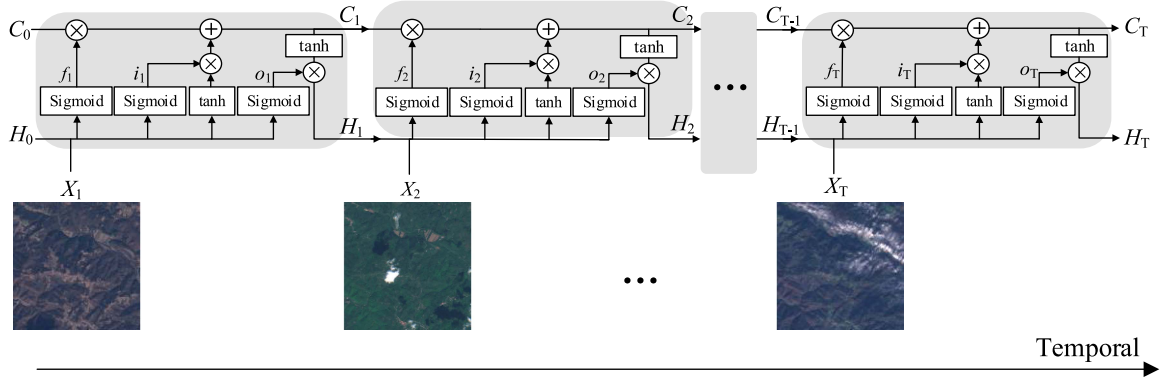


Fig. 3. Schematic diagram of ConvLSTM. Images are sent to the ConvLSTM unit according to the time sequence.

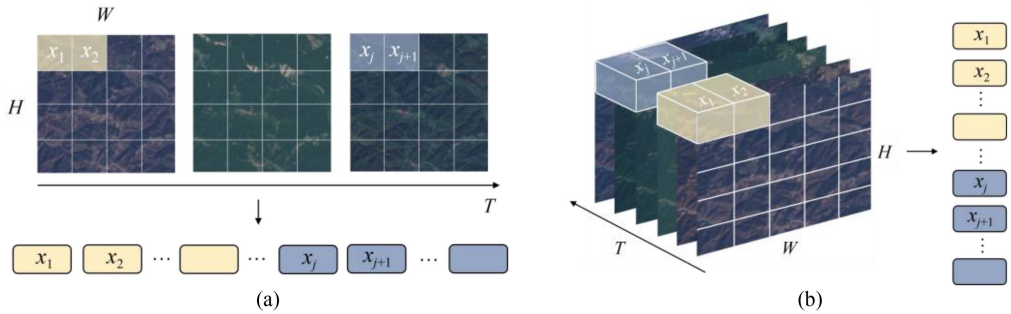


Fig. 4. Schematic diagram of the two different token approaches. (a) Uniform frame sampling. (b) Tubelet embedding.

and ViT [12] reshapes the 2-D images $x \in \mathbf{R}^{(C\times H\times W)}$ into a sequence of 2-D patches $\{x_p^i \in \mathbf{R}^{P^2\times C} | i = 1, \ldots, N\}$, where C is the channel size, $P\times P$ is the size of each image patch, and $N = HW/P^2$ is the number of patches. The patches, which are often called tokens, are flattened into a latent D-dimensional embedding space by a trainable linear projection, and are then added with a positional embedding that retains the position information, as denoted in the following:

$$z_0 = [Ex_p^1; Ex_p^2; \cdots ; Ex_p^N] + E_{\text{pos}} \tag{10}$$

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1} (l = 1, \ldots, L) \tag{11}$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l (l = 1, \ldots, L) \tag{12}$$

where $\boldsymbol{E}$ is the patch projection operation, and E_{pos} is the positional embedding. The transformer encoder consists of L layers of multiheaded self-attention (MSA) and multilayer perceptron (MLP) blocks, as shown in (11) and (12), where LN is the layer normalization operator, and z_l is the output of the lth transformer layer.

The input is 4-D spectral-temporal-spatial data here. Following the idea of ViT, we can decompose the input into N nonoverlapping patches as the first step, and then perform linear projection and rasterize the patches into 1-D tokens. There are two different token sampling approaches for 4-D data, as shown in Fig. 4. One is called uniform frame sampling [43], which involves uniformly reshaping each month's image into nonoverlapping 2-D tokens, each of the size of $k\times k$. In this case, the projection operation is a 2-D convolution with kernel size $k\times k$, and we obtain $N = T \times \frac{H}{k} \times \frac{W}{k}$ tokens in total. The other method is called tubelet embedding [43], which involves directly embedding the input into spatio-temporal "tubes" using 3-D convolution. In this case, the projection operation is a 3-D convolution with kernel size $k_T\times k\times k$, and we obtain $N = \frac{T}{k_T} \times \frac{H}{k} \times \frac{W}{k}$ tokens that are extracted from the temporal, height, and width dimensions, respectively.

The tokens then pass through the transformer encoder. In addition to directly forwarding all the tokens extracted from the 4-D input through the transformer encoder by default (i.e., space–time attention), there are two other alternative self-attention structures for uniform frame sampling (where uniform frame sampling is analogous to monthly sampling). One is called space attention, where the spatial attention is only calculated within each frame. The other is called divided space–time attention [42], where temporal attention and spatial attention are separately applied one after the other. Fig. 5 presents a visualization of the three attention models for uniform frame sampling.

B. Implementation of the Deep Learning Based Methods

Seven neural network architectures, with four different strategies in processing spatio-temporal features, were evaluated in land-cover classification using Sentinel-1 and Sentinel-2 time series in this study.

1) U-Net (2-D U-Net) [49]. 2-D U-Net processes multitemporal images as different input channels and transforms them into 2-D feature representations, exploiting

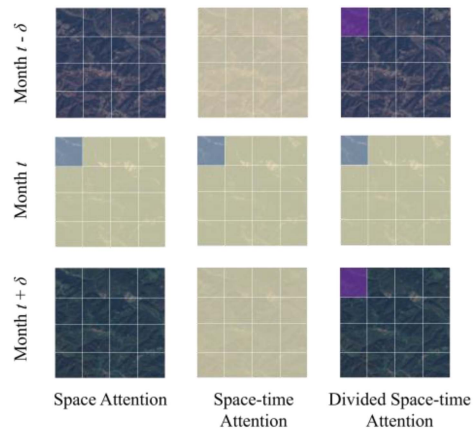


Fig. 5. Visualization of three self-attention structures. The current patches are in blue, and their self-attention neighborhoods are in light yellow. The space attention does not affect the adjacent time series, the space–time attention process all the adjacent patches simultaneously, while the divided space–time attention first processes the attention between the current patch (blue) and the patches (purple) at the same spatial location, and then the rest of the patches in the current image.

the spatial correlations of neighboring pixels of the time series [58]. Here, we used the Visual Geometry Group16 (VGG16) structure [59] as the encoder. There were four scales of convolutional layers, with each convolution operation followed by instance normalization and a rectified linear unit (ReLU) activation function. The convolution kernel size was 3×3, and the number of output channels of each scale was 32, 64, 128, and 256, respectively.

2) DeepLabv3 [50]. DeepLabv3 is a commonly used 2-D CNN model, it processes multitemporal images in the same way as 2-D U-Net. Here, we used ResNet34 [60] as the backbone.
3) Three-dimensional U-Net (3-D U-Net) [51]. 3-D U-Net processes the temporal and spatial dimensions jointly using 3-D convolution and 3-D pooling operations, thus exploiting the spatio-temporal correlations in the data. Here, the encoder of 3-D U-Net also followed the VGG16 structure, but with 3×3×3 convolution kernels, with each convolution operation followed by instance normalization and a ReLU activation function. There were four scales of convolutional layers, and the number of output channels of each scale was 32, 64, 128, and 256, respectively.
4) ConvLSTM [35]. ConvLSTM works on sequences of multitemporal data and simultaneously exploits the spatio-temporal correlations with both recurrent and convolution operations. Here, for each monthly image, it was passed through two 2-D convolutional layers and a ConvLSTM cell with 32, 32, and 128 filters, respectively.
5) TransUNet [41]. TransUNet combines a 3-D U-Net model and a transformer to access both the spatio-temporal details and the global context. Here, for the CNN encoder branch, four 3-D convolutional layers consisting of 32, 64, 128, and 256 filters and three 3-D max pooling layers were used. For the transformer encoder branch, ViT-B16 [12] with tubelet embedding and space–time attention was used, with 12 transformer layers used with an embedding size of 768 and an embedding 3-D convolution kernel size of 3×32×32. The 3-D convolution kernel size was determined by the GPU limitation.
6) TransBTS [40]. The TransBTS network encoder utilizes a 3-D CNN to capture local spatio-temporal features and then feeds the reshaped feature map into the transformer for global feature modeling [40]. Here, the CNN encoder branch used four 3-D convolutional layers consisting of 32, 64, 128, and 256 filters, respectively. The bottom transformer adopted tubelet embedding and space–time attention. The number of transformer layers was 12, with the embedding size and embedding 3-D convolution kernel size being 768 and 3×4×4.
7) U-NEt TRansformers (UNETR) [52]. In UNETR, the transformer is used as the encoder and its multiple-resolution outputs are merged with a CNN-based decoder via skip connections [52]. Here, tubelet embedding and space–time attention were used. For the transformer-based encoder, 12 transformer layers were used with an embedding size of 768 and a 3-D convolution kernel size of 4×16×16.

C. Implementation of the Conventional Machine Learning Models

In the experiments, we selected RF and XGBoost as contrastive models for the deep learning approaches.

1) RF [21]. RF needs predefined features. Here, we used the original spectral bands during different periods of the year as the input. We experimentally set the hyperparameters by trial and error. The number of decision trees was 400, the maximum depth of trees was 30, and the other hyperparameters followed the default values.
2) XGBoost [22]. XGBoost also requires predefined features, so the input is the same as RF. Here, for the hyperparameters, we set the iterative step size to 0.1, the number of trees to 1000, the depth of trees to 17, the minimum weight of leaf node to 1, and the random sampling rate of the training samples to 0.9. The other settings followed the default values.

D. Implementation Details

Most of the experiments were implemented on a single GeForce RTX 3090 GPU with 24 GB RAM, while the experiments with UNETR, which requires a larger GPU capacity, were implemented on a single RTX A6000 GPU with 48 GB RAM.

All the deep learning models were implemented in PyTorch, while RF and XGBoost were implemented using the Python scikit-learn library and XGBoost library, respectively. For all the deep learning models, except for ConvLSTM (for which the input remained the original 500×500 pixels), we padded the edges of the input images to 512×512 with zeros. All the deep learning models are optimized using the adaptive moment estimation (Adam) optimizer with a learning rate of 10e-4, with the learning rate attenuated according to a polynomial learning rate policy with the power of 0.9. The optimized loss was a cross-entropy function. During the training phase, the batch size was set to 1. Training of each model is stopped according to the early stopping criterion on the validation dataset.

TABLE I
RESULTS FOR THE DIFFERENT SAMPLING METHODS AND SELF-ATTENTION SCHEMES IN TRANSUNET

	Uniform frame sampling & space attention	Uniform frame sampling & S-T attention	Uniform frame sampling & divided S-T attention	Tubelet embedding & S-T attention
OA	0.879	0.894	0.894	0.894
mF1	0.525	0.565	0.575	0.557
Weighted F1	0.868	0.886	0.886	0.885
		F1-score		
Cultivated land	0.666	0.724	0.724	0.719
Forest	0.956	0.964	0.963	0.964
Grassland	0.800	0.823	0.826	0.824
Shrubland	0.223	0.297	0.288	0.281
Water	0.711	0.740	0.775	0.756
Wetland	0.034	0.079	0.043	0.051
Artificial surface	0.727	0.777	0.769	0.775
Bare land	0.082	0.118	0.207	0.085

Note: S-T Attention is space--time attention, Divided S-T Attention is divided space--time attention.

TABLE II
INFLUENCE OF DIFFERENT LENGTHS OF TEMPORAL DIMENSION OF KERNELS FOR TRANSUNET AND TRANSBTS

Model	Kernel size of time dimension	OA	Weighted f1	mF1
TransUNet [41]	3	0.894	0.885	0.557
	6	0.893	0.885	0.553
	12	0.893	0.884	0.556
TransBTS [40]	3	0.893	0.884	0.568
	6	0.894	0.884	0.545
	12	0.894	0.884	0.546

E. Performance Evaluation

We evaluate the performance of the different models through the overall accuracy (OA), class-wise F1-score (F1), mean of F1-score (mF1), and weighted F1-score. The F1-score is the harmonic mean of the precision (P) and recall (R). The mF1 is the algebraic average of all the F1-scores of the different classes, which is thus affected by the performance for the rare classes, which may not be well trained by a few samples. The weighted F1-score is the weighted sum of the F1-score, where the weight is determined by the proportion of pixels of this class.

IV. RESULT AND ANALYSIS

In the following, we discuss the effects of token sampling, the self-attention strategies, and the temporal kernel size for the ViT-based methods in Section IV-A. We then present the results of the different models for the inputs of optical, SAR, and the combination of optical and SAR images, respectively, in Section IV-B. A further analysis of each land-cover map obtained under the nine models is presented in Section IV-C. The impact of local image problems (e.g., partial cloud coverage) on the different methods is analyzed in Section IV-D. We then further compare the robustness of the different models when some months of images are lacking due to weather conditions in Section IV-E.

A. Token Sampling, Self-Attention, and Kernel Size in the ViT-Based Methods

When introducing how the transformer-based models deal with spectral-temporal-spatial data, we noted that there are usually two token sampling approaches (uniform frame sampling and tubelet embedding) and three self-attention schemes (space-time attention, space attention, and divided space–time attention). We applied the different token sampling approaches and self-attention schemes in TransUNet. The results are shown in Table I. The embedding 2-D convolution kernel size for the uniform frame sampling was 32×32, and the embedding 3-D convolution kernel size for the tubelet embedding was $3\times32\times32$. According to Table I, apart from the relatively lower results for the spatial attention setting, the results for the other three variants are very close. Therefore, to simplify the transformer-based models, we selected tubelet embedding and space–time attention by default.

The influence of different lengths of temporal dimension for the kernels using tubelet embedding is shown in Table II. The results are almost the same for TransUNet and TransBTS, except for the mF1 for a dimension of 3 in TransBTS being slightly better. In the following experiments, we set the temporal dimension in the 3-D convolution of tubelet embedding as 3 for efficiency.

B. Comparative Evaluation

Table III lists the results for the nine methods with inputs of SAR images (S1), optical images (S2), and SAR+optical images (S1+S2), respectively.

First, we compare the impact of different inputs. When using the combination of S1 and S2 or only using S2 as input, all of the methods perform almost the same. Generally speaking, stacking S1 and S2 images provides more information. However, combining S1 and S2 images does not improve the result significantly. This can be explained by the fact that the multitemporal S2 images provide enough information for land-cover classification, and the additional information from the S1 images is redundant. Meanwhile, when using only S1 images, the OAs are lower than using S2, by 5%–22%. However, the S1 images can also provide important information, and there is only a 5% OA (or weighted F1-score) gap between utilizing S1 and S2 images in most of the modern deep learning based methods. This observation is very important because optical imaging suffers from bad weather, and SAR images are very convenient to access.

Second, the between-method comparison shows that the modern deep learning models obtain a much better performance than

TABLE III
RESULTS OF THE NINE MODELS OBTAINED USING THE S1, S2, AND S1+S2 DATASETS

Model	Dataset	OA	Weighted F1	mF1
RF [21]	S1	0.638	0.690	0.286
	S2	0.800	0.818	0.485
	S1+S2	0.802	0.823	0.491
XGBoost [22]	S1	0.602	0.667	0.275
	S2	0.821	0.836	0.515
	S1+S2	0.826	0.842	0.515
2D U-Net [49]	S1(July)	0.761	0.738	0.316
	S2(July)	0.862	0.850	0.483
	S1+S2(July)	0.861	0.850	0.499
2D U-Net	S1	0.840	0.821	0.414
	S2	0.880	0.869	0.514
	S1+S2	0.885	0.875	0.525
DeepLabv3 [50]	S1	0.809	0.788	0.371
	S2	0.828	0.807	0.405
	S1+S2	0.840	0.823	0.427
ConvLSTM [35]	S1	0.838	0.826	0.431
	S2	0.890	0.882	0.558
	S1+S2	0.891	0.884	0.549
3D U-Net [51]	S1	0.850	0.834	0.442
	S2	0.892	0.884	0.549
	S1+S2	0.895	0.886	0.559
TransUNet [41]	S1	0.848	0.831	0.431
	S2	0.893	0.884	0.563
	S1+S2	0.894	0.885	0.557
TransBTS [40]	S1	0.846	0.828	0.433
	S2	0.892	0.882	0.550
	S1+S2	0.893	0.884	0.568
UNETR [52]	S1	0.836	0.819	0.420
	S2	0.892	0.881	0.552
	S1+S2	0.892	0.884	0.558

RF and XGBoost. The methods that consider temporal information, i.e., ConvLSTM, 3-D U-Net, and the three transformer-based methods, perform the best. The 2-D U-Net lies between the conventional methods and the deep learning based time-series methods. Although 2-D U-Net cannot directly process the temporal information, it is apparent that using the time series as input is better than using only the image captured in July. This can be explained by the fact that although the summation operation along the temporal dimension flattens the temporal information, the network does learn how to optimally weight the features from different times. DeepLabv3 obtains worse results compared to 2-D U-Net, indicating the several new tricks in the former does not work in this case. The time-series methods perform almost the same. 3-D U-Net achieves the best OA, weighted F1-score, and mF1 for the S1+S2 and S1 images. Meanwhile, TransUNet yields the best OA, weighted F1, and mF1 for the S2 images. It can be concluded that 3-D convolution, LSTM, and transformers are all effective in extracting spatio-temporal information, and the newest transformer models have no significant advantage. Specifically, TransUNet is a combination of a 3-D U-Net model and a transformer, where the latter is designed to model the global context, as analyzed before. However, this revision does not surpass the vanilla 3-D U-Net in this case.

The other detail that should be noted is that the modern deep learning based methods show a much better performance than the conventional methods in processing SAR images for land-cover classification. When using RF or XGBoost, the OA gap between using the S1 and S2 images as input is more than 15%. However, all of the deep learning based methods (except 2-D U-Net performed on the July image) shrink this gap by 4%–5%.

TABLE IV
PARAMETER NUMBERS AND TRAINING TIME OF NINE MODELS IN THE S1+S2 DATASET. "M": MILLION, "H": HOUR, "S": SECOND

Model	Parameter (M)	Training time (H)	Testing time per image (S)
RF [21]	/	2.0	40.5
XGBoost [22]	/	50.4	82.5
2D U-Net [49]	10.9	1.6	0.5
DeepLabv3 [50]	25.7	0.6	0.5
ConvLSTM [35]	0.7	71.3	0.8
3D U-Net [51]	31.2	28.9	1.0
TransUNet [41]	83.9	19.8	1.0
TransBTS [40]	52.0	17.5	0.9
UNETR [52]	114.6	68.9	2.0

In Table IV, we list the parameters and training times of nine models in the S1+S2 dataset to analyze the efficiency. The ConvLSTM has the fewest parameter, as it processes multitemporal data sequentially in time order, but has the longest training time. The transformer-based models have more parameters than the CNN-based models. Nevertheless, TransUNet and TransBTS that combines both CNN and transformer convergent more quickly than 3-D CNN. In contrast, UNETR has the heaviest parameter, requires the most training time, and costs the most testing time among deep learning based methods. However, all of the testing time of different methods are on the same level.

We selected two representative sites from the S1+S2 dataset—one from a mountainous area and the other from an urban area—to show the classification details in Fig. 6. As for the reference image, we use the red (band 4), green (band 3),

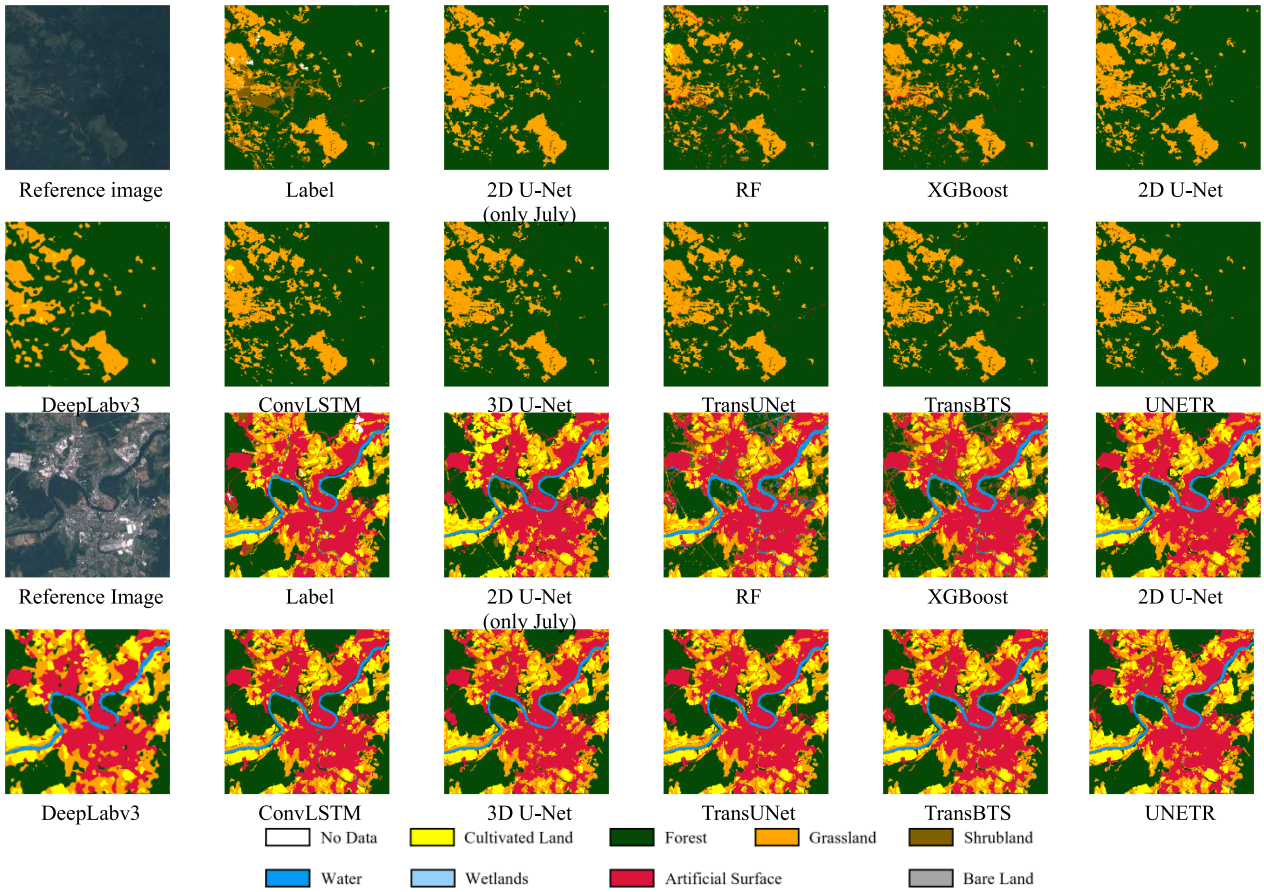


Fig. 6. Qualitative investigation of the land-cover map details produced by the nine models for a mountainous area (top) and an urban area (bottom) in the S1+S2 dataset.

and blue (band 2) of the Sentinel-2 image of July. We can see that the prediction results of all the models are not significantly different, except for the conventional machine learning methods, which have some salt-and-pepper errors. Almost every model can correctly identify the land-cover outlines, and the error mainly occurs when classifying the small land-cover areas or land-cover boundaries. An obvious error is the shrubland in the first image is mistaken as grassland by all the methods.

C. Per-Class Analysis

We provide the class-wise F1-scores of each land-cover type under the nine models in Table V and the confusion matrix for the S1+S2 dataset in Fig. 7. The highest values of F1-scores for the different land-cover types are almost always achieved by the 3-D U-Net or transformer-based models, which is consistent with Table III. By comparing the performance of 2-D U-Net with all the periods of images (M3) and only July (M3J), it can be observed that the temporal features improve the performance of 2-D U-Net on most of the land-cover types. Specifically, the land-cover types with apparent phenological phenomena, such as cultivated land, grassland, shrubland, etc., show a significant improvement in F1-score. However, for the wetland and bare land land-cover types, the time-series information does not improve the classification results. One reason for this is that there are only few samples of wetland and bare land in this area, which results in insufficient training of the model. The other reason could be that the wetland has complex land-cover components, and is easily confused with other land-cover types.

It can also be observed that, although the conventional methods perform worse than the deep learning based methods in general, the former methods perform better on the wetland and bare land. This can also be explained by the weakness of the small number of samples. In the conventional methods, the sample numbers of the two types are comparable with those of the other types, but in the deep learning based methods, the patch samples are extremely imbalanced.

According to Fig. 7, the misclassification rate for the four land-cover types of forest, water, impervious surface, and grassland are low. Shrubland tends to be confused with forest and grassland. This can be explained by the fact that the three types look similar in these high-resolution remote sensing images. Other notable confusion is between water/forest and shrubland/artificial surface. Such unexpected errors can be caused by the narrow river routes in Slovenia, many of which are shaded by densely distributed buildings and vegetation on either side of the water channels. Other confusion is between bare land and vegetation. After comparing the images of the different periods, we found that some of the bare land is covered by vegetation

TABLE V
CLASS-WISE F1-SCORE (F1) FOR THE LAND-COVER CLASSIFICATION WITH RF (M1), XGBOOST (M2), 2-D U-NET (M3), 2-D U-NET WITH IMAGES ONLY IN JULY (M3J), DEEPLABV3 (M4), CONVLSTM (M5), 3-D U-NET (M6), TRANSUNET (M7), TRANSBTS (M8), AND UNETR (M9) UNDER DIFFERENT INPUTS

Input	Model	M1	M2	M3J	M3	M4	M5	M6	M7	M8	M9
S1	1-Cultivated land	0.366	0.384	0.351	0.524	0.469	0.560	0.587	0.565	0.574	0.546
	2-Forest	0.810	0.783	0.889	0.944	0.926	0.942	0.947	0.947	0.943	0.939
	3-Grassland	0.607	0.577	0.568	0.721	0.654	0.722	0.737	0.734	0.729	0.718
	4-Shrubland	0.139	0.126	0.0	0.082	0.012	0.133	0.100	0.102	0.079	0.068
	5-Water	0.119	0.087	0.285	0.504	0.438	0.525	0.553	0.511	0.552	0.519
	6-Wetland	0.007	0.012	0.0	0.0	0.0	0.0	0.001	0.010	0.006	0.0
	7-Artificial surface	0.237	0.228	0.436	0.541	0.468	0.556	0.582	0.580	0.577	0.544
	8-Bare land	0.002	0.002	0.0	0.0	0.0	0.011	0.025	0.0	0.002	0.025
S2	1-Cultivated land	0.566	0.622	0.603	0.668	0.506	0.723	0.722	0.728	0.716	0.730
	2-Forest	0.922	0.930	0.953	0.958	0.936	0.962	0.962	0.963	0.963	0.963
	3-Grassland	0.728	0.760	0.751	0.790	0.677	0.810	0.820	0.820	0.816	0.816
	4-Shrubland	0.233	0.261	0.157	0.228	0.026	0.290	0.285	0.272	0.260	0.225
	5-Water	0.570	0.616	0.650	0.675	0.494	0.724	0.757	0.763	0.763	0.761
	6-Wetland	0.043	0.048	0.0	0.0	0.0	0.0	0.051	0.043	0.029	0.059
	7-Artificial surface	0.562	0.605	0.720	0.758	0.597	0.769	0.767	0.769	0.773	0.772
	8-Bare land	0.257	0.279	0.031	0.032	0.007	0.183	0.029	0.147	0.082	0.093
S1+S2	1-Cultivated land	0.573	0.633	0.595	0.687	0.583	0.724	0.725	0.719	0.723	0.728
	2-Forest	0.922	0.934	0.952	0.962	0.945	0.963	0.964	0.964	0.962	0.962
	3-Grassland	0.748	0.771	0.756	0.804	0.699	0.816	0.825	0.824	0.820	0.822
	4-Shrubland	0.246	0.279	0.173	0.248	0.034	0.311	0.279	0.281	0.281	0.280
	5-Water	0.553	0.601	0.682	0.712	0.549	0.722	0.764	0.756	0.760	0.769
	6-Wetland	0.053	0.060	0.0	0.0	0.0	0.0	0.049	0.051	0.035	0.031
	7-Artificial surface	0.567	0.602	0.725	0.748	0.601	0.769	0.774	0.775	0.772	0.771
	8-Bare land	0.263	0.242	0.109	0.036	0.0	0.088	0.093	0.085	0.196	0.098

Fig. 7. Confusion matrices of (a) RF, (b) XGBoost, (c) 2-D U-Net, (d) DeepLabv3, (e) ConvLSTM, (f) 3-D U-Net, (g) TransUNet, (h) TransBTS, and (i) UNETR for the S1+S2 dataset. 1 represents the cultivated land, 2 represents the forest, 3 represents the grassland, 4 represents the shrubland, 5 represents water, 6 represents the wetland, 7 represents the artificial surface, and 8 represents the bare land.

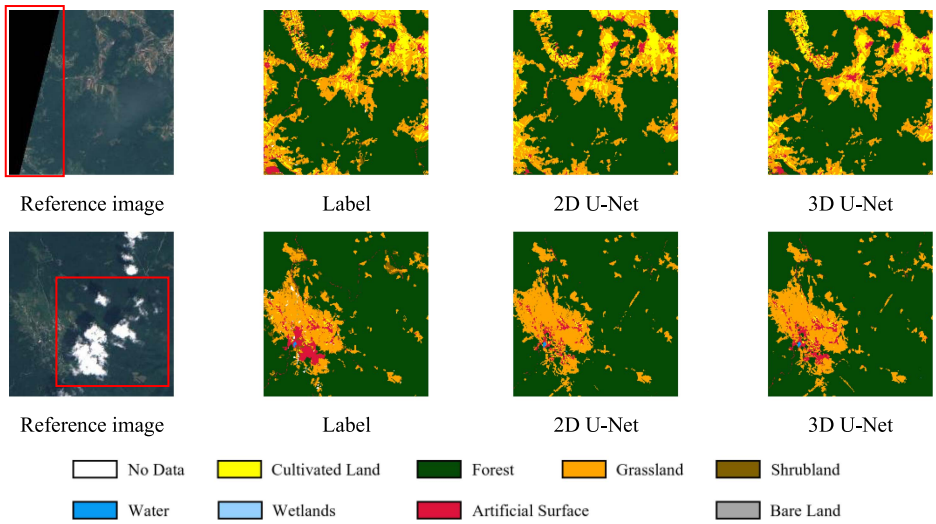


Fig. 8. Classifying images with invalid pixels (top) or cloud-covered pixels (bottom).

in spring or summer. In addition, inaccurate labeling can also influence the confusion degree of the classification.

D. Robustness Analysis to Local Image Problems

Some images are covered by a small proportion of clouds or invalid pixels in the image boundaries, as can be seen in the two reference images in Fig. 8. Fig. 8 shows the land-cover maps predicted by 2-D U-Net and 3-D U-Net with multitemporal optical and SAR (S1+S2) images. By using the multitemporal information for other periods, both 2-D U-Net and 3-D U-Net can successfully predict the land-cover of the area, including those covered by invalid pixels. It can therefore be concluded that local image problems have little effect on deep learning based land-cover classification from time-series images.

E. Robustness Analysis to a Lack of Images for Some Periods

Optical images can suffer not only from a part of invalid pixels, but it is highly possible that, in one or several months, optical images cannot be obtained, due to the weather conditions. We simulated the case of optical images of certain months being missing in the test set and kept the pretrained model with all of the training data unchanged. This is in line with an actual real-world situation, in that we can prepare a high-quality training dataset, but the actual images needing to be classified have real-world restrictions. To keep the classification model unchanged, we filled the missing images with blank images (i.e., with the image pixels all set to 0) to preserve the same temporal dimension. Table VI lists the results. It can be noted that all the pretrained deep learning models can make sound predictions when some temporal images are unavailable. The OA generally shows a downward trend as the number of unavailable optical images increases, but only very slowly. This indicates that it is possible to pre-train a classification model with high-quality samples, and then apply the model directly on new multitemporal images under imperfect conditions. The exception is the two conventional methods, where the performances can even be improved in a few cases when some images are removed. This can be explained by the fact that the input (i.e., the direct concatenation of pixel values in different times) is not optimal. For example, more sophisticated features should be applied, or there is noise that should be filtered out.

TABLE VI
OA OF THE LAND-COVER CLASSIFICATION RESULTS OF THE NINE MODELS WITH CERTAIN MONTH INPUTS EXCLUDED (DENOTED AS "−")

Model	All	−Feb	−Feb/Jul	−Feb/Jul/Nov	−Feb/May/Jul/Nov
RF [21]	0.802	0.791	0.820	0.816	0.790
XGBoost [22]	0.826	0.818	0.850	0.848	0.841
2D U-Net [49]	0.885	0.884	0.880	0.880	0.881
DeepLabv3 [50]	0.840	0.839	0.839	0.839	0.839
ConvLSTM [35]	0.891	0.885	0.882	0.881	0.873
3D U-Net [51]	0.895	0.887	0.879	0.880	0.877
TransUNet [41]	0.894	0.889	0.881	0.880	0.876
TransBTS [40]	0.893	0.878	0.870	0.873	0.868
UNETR [52]	0.892	0.887	0.878	0.878	0.872

V. DISCUSSION

This article evaluates three different deep learning architectures with their respective mechanisms for processing high-dimension spatio-temporal remote sensing data. We revealed that CNN, RNN, and ViT can effectively fulfill the task of spatio-temporal representation learning and obtain good land-cover classification results from Sentinel time series. It is not very surprising that the most recent ViT structure has not shown significant advantage. In fact, several reports on 2-D spatial land-cover classification obtained similar results. For example, Gu et al. [61] reported that ViT-based segmentation models are slightly worse than the top CNN-based models and comparable to most of them, on optical remote sensing images. Wu et al. [62] reported ViT and CNN had a similar performance on land cover classification. Wang et al. [63] indicated the best ViT model only outperformed CNN models by 0.3% in OA score. Jamali and Mahdianpari [64] used Sentinel-1, Sentinel-2, and LiDAR for Wetland classification. They compared Swin Transformer, 3-D CNN and VGG-16, where 3-D CNN showed significantly better results. However, in their work the multitemporal images

are directly averaged before being sent into the deep learning model, i.e., the temporal information is discarded.

These studies partially support the conclusion of our work, where different deep learning architectures are evaluated on high-dimensional spatio-temporal remote sensing images instead of 2-D spatial images. Our findings provide new insights in spatio-temporal representation learning, specifically, high-dimension convolution, recurrent unit, and transformer are proved almost equivalently effective and much better than conventional methods and those deep learning methods for 2-D images.

One of the limitations of this work is data selection and data distribution. Although high-accurate data samples covering the whole Slovenia are used, more data around the world would be better for diversity. There are several studies claimed to have provided large-scale land cover samples with about 10 m ground resolution; however, we empirically found the pixel-level classification accuracy of these datasets is questionable. When more open-source and high-accurate datasets are available, we will further testify the performance of modern methods on spatio-temporal representation of remote sensing image time series.

VI. Conclusion

In this study, we explored the potential of recent deep learning based methods as well as conventional methods for land-cover classification from multitemporal Sentinel-1 and Sentinel-2 images. A total of nine architectures were considered: RF, XGBoost, 2-D U-Net, DeepLabv3, 3-D U-Net, ConvLSTM, TransUNet, TransBTS, and UNETR. Several conclusions can be drawn. First, the modern deep learning models outperform the conventional methods. Second, 3-D convolution, LSTM, and transformer-based models are all effective in extracting spatio-temporal information, although they handle the spectral-temporal-spatial data in different ways. The experimental results suggested that the latest transformer-based models do not surpass the vanilla 3-D CNN model by much but with more GPU capacity requirement. Third, using SAR images as input cannot reach the high performance of using optical images, but when optical images are unavailable, SAR images can also provide satisfactory classification results, i.e., only 4%–5% OA gap between using SAR and optical, when used with the modern deep learning based models. Fourth, introducing temporal features can improve the land-cover classification results, in general, especially for those land-cover types with apparent phenological phenomena. Finally, the robustness of the models was validated in the case of invalid pixels in one image and images of certain periods being missing. We found that a model well pretrained with the complete multitemporal images can be directly applied to such images. Given these conclusions, we believe that this work will be particularly instructive for studies of large-scale land-cover classification via multitemporal optical and/or SAR remote sensing images.

References

[1] G. Veeraswamy, A. Nagaraju, E. Balaji, and Y. Sreedhar, "Land use land cover studies of using remote sensing and GIS: A case study in Gudur Area, Nellore District, Andhra Pradesh, India," *Int. J. Res.*, vol. 4, no. 17, pp. 3145–3154, 2017.

[2] M. P. Henrique and H. D. Cooper, "Towards the global monitoring of biodiversity change," *Trends Ecol. Evol.*, vol. 21, no. 3, pp. 123–129, 2006.

[3] T. Sritarapipat and W. Takeuchi, "Land cover change simulations in Yangon under several scenarios of flood and earthquake vulnerabilities with master plan," *J. Disaster Res.*, vol. 13, no. 1, pp. 50–61, 2018.

[4] A. Fikir et al., "The impacts of watershed management on land use and land cover dynamics in eastern Tigray (Ethiopia)," *Resour., Conservation Recycling*, vol. 53, no. 4, pp. 192–198, 2009.

[5] S. Pauleit and F. Duhme, "Assessing the environmental performance of land cover types for urban planning," *Landscape Urban Plan.*, vol. 52, no. 1, pp. 1–20, 2000.

[6] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[7] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[8] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2012.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. Cambridge, MA, USA: MIT Press, 1987, pp. 318–362.

[12] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.

[13] S. Ji, C. Zhang, A. Xu, Y. Shi, and Y. Duan, "3D convolutional neural networks for crop classification with multi-temporal remote sensing images," *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 75.

[14] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, 2018, Art. no. 129.

[15] B. D. Wardlow, S. L. Egbert, and J. H. Kastens, "Analysis of time-series MODIS 250 m vegetation index data for crop classification in the US Central Great Plains," *Remote Sens. Environ.*, vol. 108, no. 3, pp. 290–310, 2007.

[16] X. Xiao et al., "Mapping paddy rice agriculture in southern China using multi-temporal MODIS images," *Remote Sens. Environ.*, vol. 95, no. 4, pp. 480–492, 2005.

[17] P. E. Osgouei, S. Kaya, E. Sertel, and U. Alganci, "Separating built-up areas from bare land in Mediterranean cities using Sentinel-2A imagery," *Remote Sens.*, vol. 11, no. 3, 2019, Art. no. 345.

[18] A. Bhatt, S. K. Ghosh, and A. Kumar, "Automated change detection in satellite images using machine learning algorithms for Delhi, India," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 1678–1681.

[19] W. Li et al., "A comparison of land surface water mapping using the normalized difference water index from TM, ETM+ and ALI," *Remote Sens.*, vol. 5, no. 11, pp. 5530–5549, 2013.

[20] T. Bangira, S. M. Alfieri, M. Menenti, and A. Van Niekerk, "Comparing thresholding with machine learning classifiers for mapping complex water," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1351.

[21] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

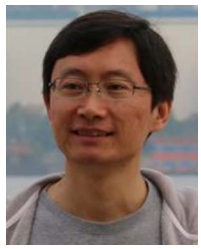
[23] G. M. Foody, N. Campbell, N. Trodd, and T. Wood, "Derivation and applications of probabilistic measures of class membership from the maximum-likelihood classification," *Photogrammetric Eng. Remote Sens.*, vol. 58, no. 9, pp. 1335–1341, 1992.

[24] L. H. Zhong, L. N. Hu, and H. Zhou, "Deep learning based multi-temporal crop classification," *Remote Sens. Environ.*, vol. 221, pp. 430–443, Feb. 2019.

[25] J. Kang et al., "DisOptNet: Distilling semantic knowledge from optical images for weather-independent building segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[26] R. Hang, X. Qian, and Q. Liu, "Cross-modality contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5532812.
[27] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, "Hyperspectral image classification with attention-aided CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2281–2293, Mar. 2021.
[28] J. Parajuli, R. Fernandez-Beltran, J. Kang, and F. Pla, "Attentional dense convolutional neural network for water body extraction from sentinel-2 images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6804–6816, 2022.
[29] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.
[30] R. Hang, P. Yang, F. Zhou, and Q. Liu, "Multiscale progressive segmentation network for high-resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.
[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[32] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.
[33] S. Vasit et al., "Field-scale crop yield prediction using multi-temporal worldview-3 and planetscope satellite data and deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 174, pp. 265–281, 2021.
[34] Y. Xu, Q. Q. Kong, Q. Huang, W. W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 3461–3466.
[35] X. Shi, Z. Chen, H. Wang, D. - Y. Yeung, W. - K. Wong, and W. - C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, Art. no. 28.
[36] N. Teimouri, M. Dyrmann, and R. N. Jørgensen, "A novel spatio-temporal FCN-LSTM network for recognizing various crop types using multi-temporal radar images," *Remote Sens.*, vol. 11, no. 8, 2019, Art. no. 990.
[37] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, Art. no. 30.
[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. ACL*, 2019, pp. 4171–4186.
[39] Y. Liu et al., "Roberta: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[40] W. Wang, C. Chen, M. Ding, H. Yu, S. Zha, and J. Li, "Transbts: Multimodal brain tumor segmentation using transformer," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2021, pp. 109–119.
[41] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.
[42] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in *Proc. ICML*, 2021, pp. 813–824.
[43] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6836–6846.
[44] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3163–3172.
[45] Z. Xu, W. Zhang, T. Zhang, Z. Yang, and J. Li, "Efficient transformer for remote sensing image segmentation," *Remote Sens.*, vol. 13, no. 18, 2021, Art. no. 3585.
[46] P. Deng, K. Xu, and H. Huang, "When CNNs meet vision transformer: A joint framework for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021.
[47] Y. Gao et al., "Fusion classification of HSI and MSI using a spatial-spectral vision transformer for wetland biodiversity estimation," *Remote Sens.*, vol. 14, no. 4, 2022, Art. no. 850.
[48] H. Chen, E. Nemni, S. Vallecorsa, X. Li, C. Wu, and L. Bromley, "Dual-Tasks siamese transformer framework for building damage assessment," 2022, *arXiv:2201.10953*.
[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.
[50] L. - C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.
[51] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2016, pp. 424–432.
[52] A. Hatamizadeh et al., "UNETR: Transformers for 3D medical image segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 574–584.
[53] K. Kozjek, M. Dolinar, and G. Skok, "Objective climate classification of Slovenia," *Int. J. Climatol.*, vol. 37, pp. 848–860, 2017.
[54] "Example dataset of EOPatches for Slovenia 2019," 2019. [Online]. Available: http://eo-learn.sentinel-hub.com/
[55] "Sentinel-hub," 2019. [Online]. Available: https://www.sentinel-hub.com/
[56] O. D'Hondt, C. López-Martínez, S. Guillaso, and O. Hellwich, "Non-local filtering applied to 3-D reconstruction of tomographic SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 272–285, Jan. 2017.
[57] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*, vol. 385. New York, NY, USA: Springer, 2012, pp. 37–45.
[58] R. N. Masolele et al., "Spatial and temporal deep learning methods for deriving land-use following deforestation: A pan-tropical case study using Landsat time series," *Remote Sens. Environ.*, vol. 264, 2021, Art. no. 112600.
[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[61] G. Xingjian, L. Sizhe, R. Shougang, Z. Hengbiao, F. Chengcheng, and X. Huanliang, "Adaptive enhanced swin transformer with U-net for remote sensing image segmentation," *Comput. Elect. Eng.*, vol. 102, 2022, Art. no. 108223.
[62] Y. Wu et al., "RA-ViT: Patch-wise radially-accumulate module for ViT in hyperspectral image classification," *J. Phys.: Conf. Ser.*, vol. 2278, no. 1, 2022, Art. no. 012009.
[63] W. Wang, C. Tang, X. Wang, and B. Zheng, "A ViT-based multiscale feature fusion approach for remote sensing image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[64] A. Jamali and M. Mahdianpari, "Swin transformer and deep convolutional neural networks for coastal wetland classification using sentinel-1, sentinel-2, and LiDAR data," *Remote Sens.*, vol. 14, no. 2, 2022, Art. no. 359.

Linying Zhao received the B.E. degree in surveying and mapping engineering from the Southwest Jiaotong University, Chengdu, China, in 2020. She is currently working toward the M.S. degree in photogrammetry and remote sensing with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China.



Shunping Ji (Senior Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2007.

He is currently a Professor with the School of Remote Sensing and Information Engineering, Wuhan University. His research interests include photogrammetry, remote sensing image processing, mobile mapping system, and machine learning.