remote sensing

MDPI

Article

A Fast Three-Dimensional Convolutional Neural Network-Based Spatiotemporal Fusion Method (STF3DCNN) Using a Spatial-Temporal-Spectral Dataset

Mingyuan Peng [1,2], Lifu Zhang [1,*], Xuejian Sun [1], Yi Cen [1] and Xiaoyang Zhao [1,2]

[1] The State Key Laboratory of Remote Sensing Science, Aerospace Information Institute, Chinese Academy of Sciences, Beijing 100101, China; pengmy@radi.ac.cn (M.P.); sunxj@radi.ac.cn (X.S.); cenyi@radi.ac.cn (Y.C.); zhaoxy@radi.ac.cn (X.Z.)

[2] University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: zhanglf@radi.ac.cn; Tel.: +86-13716974736

Received: 1 November 2020; Accepted: 26 November 2020; Published: 27 November 2020

Abstract: With the growing development of remote sensors, huge volumes of remote sensing data are being utilized in related applications, bringing new challenges to the efficiency and capability of processing huge datasets. Spatiotemporal remote sensing data fusion can restore high spatial and high temporal resolution remote sensing data from multiple remote sensing datasets. However, the current methods require long computing times and are of low efficiency, especially the newly proposed deep learning-based methods. Here, we propose a fast three-dimensional convolutional neural network-based spatiotemporal fusion method (STF3DCNN) using a spatial-temporal-spectral dataset. This method is able to fuse low-spatial high-temporal resolution data (HTLS) and high-spatial low-temporal resolution data (HSLT) in a four-dimensional spatial-temporal-spectral dataset with increasing efficiency, while simultaneously ensuring accuracy. The method was tested using three datasets, and discussions of the network parameters were conducted. In addition, this method was compared with commonly used spatiotemporal fusion methods to verify our conclusion.

Keywords: data fusion; spatiotemporal fusion; spatial-temporal-spectral dataset; 3DCNN

1. Introduction

With the rapid development of remote sensing applications and sensors, huge amounts of data have accumulated, making long-term monitoring-related applications possible. Various satellites have acquired massive datasets with different spatial and temporal resolutions. Due to the limitations of satellite sensors, remote sensing datasets cannot have both high spatial and high temporal resolutions. Combining the advantages of different remote sensing products to obtain datasets with both high spatial and high temporal resolutions has been a growing research division. Spatiotemporal data fusion is an effective choice to accomplish this goal. However, due to the design of the algorithms, traditional methods and deep learning methods cannot fuse long time-series datasets with a high speed, making them difficult to apply to large, long time-series datasets. They are insufficient because the number of remote sensing datasets continues to grow, and the application requirements have also brought challenges to the processing of long time-series datasets.

To date, researchers have proposed many spatiotemporal fusion methods, which can be divided into three categories [1]: weight function methods, linear optimization decomposition methods, and nonlinear optimization methods (as shown in Table 1). The weight function method assumes a linear relationship between high-spatial low-temporal (HSLT) resolution images and low-spatial high-temporal (HTLS) resolution images and factors temporal, spatial, and spectral weights into the model. The pixel center of the sliding window is used to determine the center pixel value. This method originated from the spatial and temporal adaptive reflectance fusion model (STARFM) [2], and scholars have since made improvements, such as the spatial temporal adaptive algorithm for mapping reflectance change model (STAARCH) [3], an enhanced STARFM (ESTARFM) algorithm [4], a modified version of the ESTARFM (mESTARFM) [5], and a rigorously weighted spatiotemporal fusion model (RWSTFM) [6]. The advantage of the weight function method is that the model is intuitive and widely used, but it has the following problems: (1) objects with changing shapes and short-term transient changes cannot be well predicted, (2) the correlation between sensors has not been strictly proven, and (3) the size of the sliding window and other parameters must be set manually and cannot be obtained automatically. The linear optimization decomposition method is also based on a linear assumption similar in principle to the weight function method. The restored image is obtained by adding constraints to obtain the optimal solution. Based on the optimization criteria, the linear optimization decomposition method can be divided into three sub-categories: the spectral unmixing method, the Bayesian method, and the sparse representation method. Commonly used algorithms include the multisensor multiresolution technique (MMT) spectral unmixing algorithm [7], flexible spatiotemporal data fusion (FSDAF) [8], the spatial and temporal data fusion model (STDFM) algorithm [9], the enhanced STDFM [10], the modified spatial and temporal data fusion approach [11], soft clustering [12], and the object-based spatial and temporal vegetation index unmixing model algorithm (Object Based Image Analysis, OBIA) [13]. Although the weight-based fusion algorithm using linear optimization decomposition can achieve better spatiotemporal fusion reconstruction, it is difficult to satisfy the actual imaging scenario due to the assumption of the linear model. The nonlinear mapping method is based on the deep learning method, which can describe the nonlinear relationship well. Scholars have also explored the use of deep learning methods for spatiotemporal fusion. Moosavi combined a wavelet algorithm, artificial neural network, adaptive neuro-fuzzy inference system, and supported vector machine methods to propose the prediction-phase wavelet–artificial intelligence fusion approach algorithm for Landsat thermal infrared data [14]. In recent years, the use of convolutional neural networks (CNNs or ConvNets) for spatiotemporal fusion algorithms, such as the spatiotemporal fusion using deep convolutional neural networks (STFDCNN) [15] and the deep convolutional spatiotemporal fusion network (DCSTFN) [16], have also been proposed. The nonlinear optimization method can learn and accurately describe the nonlinear relationship between the known and missing time-phase images and has more mobility and accuracy than those of the linear optimization decomposition method. However, its ability is related to the network architectures and parameters. Poor network structures cannot produce good results, and they often require a large number of training samples and a long training time.

Table 1. Spatiotemporal fusion methods mentioned in the article.

Categories	Sub-Categories	Representative Methods
Weight function methods		STARFM, ESTARFM, mESTARFM, RWSTFM
Linear optimization decomposition methods	Spectral-unmixing method	MMT, STDFM, enhanced STDFM, soft clustering, OBIA
	Bayesian method	
	Sparse representation method	
Nonlinear optimization methods		ANN, DCSTFN, STFDCNN

All the traditional datasets arrange remote sensing data into three dimensions, space (height and width) and spectrum, rendering the data difficult to rearrange and process when it comes to long time-series. Zhang et al. proposed the idea of a new multi-dimensional dataset (MDD) in 2017 [17]

that can build a spatial-temporal-spectral dataset integrating the information in four dimensions. The MDD can satisfy the multi-feature correlation analysis of different dimensions and thus is a good choice for long time-series dataset fusion. This raises the question of how to extract the spatiotemporal information from four-dimensional (4D) datasets. Recently, three-dimensional convolutional neural networks (3D CNNs) have become widely used in video and computer vision fields. Compared with two-dimensional convolutional neural networks (2D CNNs), in which operations are only performed spatially, 3D CNNs perform convolution and pooling operations spatiotemporally, making them well-suited for spatiotemporal feature learning [18].

Based on the above-mentioned reasons, we propose a fast spatiotemporal remote sensing fusion method using the tools of 3D CNNs. Differently from current spatiotemporal fusion methods, our method arranges datasets based on the idea of the MDD, and the calculations are performed four-dimensionally. The method introduces the concept of residual series of the 4D long time-series dataset, to subtract the images of the former dates from latter dates (for example, the dates are arranged and indexed, and we subtract the images on the even dates from those of the odd dates) and pile them in the temporal order. The architecture learns the mappings between 4D residual series of HTLS and HSLT datasets using the 3D CNNs model. By adding predicted residual series of HSHT to the original HSLT dataset, the method can restore a compact 4D long time-series dataset in high spatial and high temporal (HSHT) resolution. The method is easy to implement and of a high efficiency, and it is able to guarantee the accuracy at the same time. The potential contribution of our method may lie in the following: (1) This is, according to the investigation of the authors, the first spatiotemporal remote sensing data fusion method performed four-dimensionally. This casts new light on remote sensing fusion methods. (2) This method has a fast speed compared to other state-of-the-art spatiotemporal fusion methods. It can cut down the processing time hugely, thus shrinking the overall time of long time-series-related remote sensing projects.

This paper is structured as follows. Section 2 presents the overall idea and framework of the proposed method. Section 3 introduces the datasets used to verify the method, the experimental settings, and the ablation study of the method. Section 4 presents a comparison of the results with those of existing spatiotemporal fusion methods. Section 5 summarizes this work.

2. Materials and Methods

Our method is based on the idea of MDD, and the data are arranged and operated 4-dimensionally. Then, a 4D residual series of the dataset is generated and input into the three-dimensional convolutional neural networks (3D CNNs) model to learn the features and the mappings between HTLS and HSLT. The ideas and methods are explained as follows.

2.1. Four-Dimensional (4D) Residual Series

The existing remote sensing datasets are based mostly on 3D datasets, in which temporal arrangements are not included [19]. The fast 3-dimensional convolutional neural network-based spatiotemporal fusion method (STF3DCNN) draws on the idea of multidimensional datasets in the four-dimensional mode [17]. The MDD arranges data in four dimensions: time, space (height and width), and spectrum. The storage structure of MDD is called SPATS (SPAtial-Temporal-Spectral). It mainly consists of five data formats, and each has different forms of data arrangements. They are: Temporal Sequential in Band (TSB), Temporal Sequential in Pixel (TSP), Temporal Interleaved by Band (TIB), Temporal Interleaved by Pixel (TIP), and Temporal Interleaved by Spectrum (TIS). Similar to the TIP structure shown in Figure 1, our method first organizes data in chronological order. Then, the data are stored in the order of "row first, column second". Finally, the data are arranged in a spectral order.

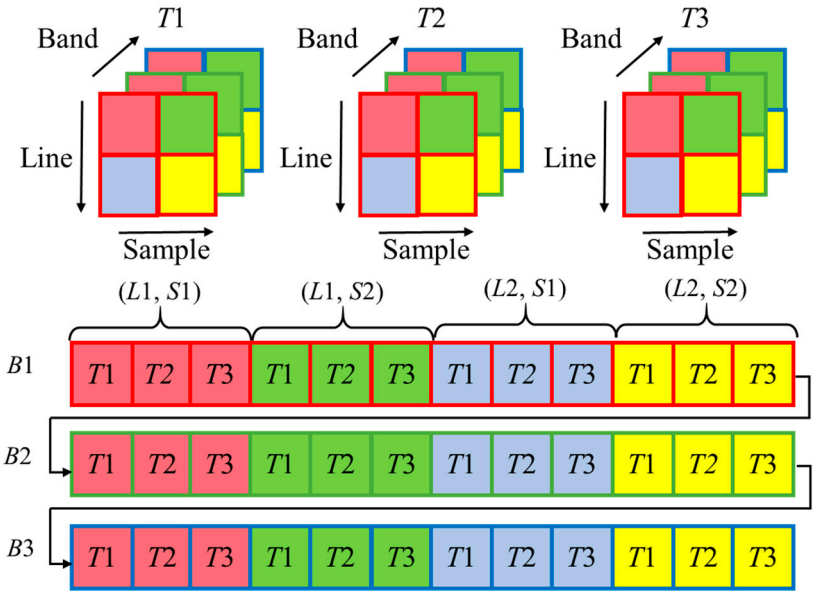


Figure 1. Temporal Interleaved by Pixel (TIP) data format of the multi-dimensional dataset.

In this method, the images in the remote sensing data series are arranged into a 4-dimensional data $X \in R^{T \times W \times H \times B}$, where T is the temporal length, W is the image width, H is the image height, and B is the number of bands. Here, we sort the dates sequentially and index them from 1 to T. The data subset X_f of former dates t_f is generated by Formula (1):

$$X_f = X(t_f, x, y, b), \quad (1)$$

where $t_f \in \{1,3,...,T\}$, $x \in \{1,2,...,W\}$, $y \in \{1,2,...,H\}$, and $b \in \{1,2,...,B\}$. Additionally, the data subset X_l of latter dates t_l can be represented as:

$$X_l = X(t_l, x, y, b), \quad (1)$$

where $t_l \in \{2,4,...,T-1\}$. Thus, the residual series of dataset X as δ_X can be calculated using Formulas (1) and (2):

$$\delta_X = X_l - X_f, \quad (2)$$

Here, we present the specific datasets used in the spatiotemporal fusion. We define the 4-dimensional dataset of HTLS as $HTLS_{4D} \in R^{T_C \times W \times H \times B}$ and the 4-dimensional dataset of HSLT as $HSLT_{4D} \in R^{T_F \times W \times H \times B}$, where $T_C = T_F \times w + 1$ and w represents the temporal resolution scale factor. The temporal resolution scale factor is calculated by dividing the temporal resolution of HSLT by that of HTLS. For example, the temporal resolution scale factor of the 8-day Moderate Resolution Imaging Spectroradiometer (MODIS) data and 16-day Landsat data is 2. As their temporal resolutions differ, we generate the temporally correlated subset $HTLS_{4D}' \in R^{T_F \times W \times H \times B}$ from $HTLS_{4D}$ according to the same/nearby dates of the $HSLT_{4D}$. The residual series of $HTLS_{4D}'$, $HTLS_{4D}$, and $HSLT_{4D}$ are calculated as $\delta_{HTLS'}$, δ_{HTLS}, and δ_{HSLT} using Formula (3).

2.2. 3D CNNs for 4D Residual Series

The residual series $\delta_{C'}$ and δ_F are used for training the spatiotemporal 3D CNNs model, the layers of which can be represented as:

$$h_i^l(x,y,z) = \sigma(b_i^l + \sum_k \sum_{u,v,w} h_k^{l-1}(x-u, y-v, z-w) W_{ki}^l(u,v,w)), \quad (3)$$

where h_i^l and h_k^{l-1} are the ith 3D feature data in the lth layer and the k th 3D feature data in the previous $l-1$th layer, respectively. W_{ki}^l is the 3D convolutional filter and the (u,v,w) represents a point in the filter. b_i^l is the 3D bias. $\sigma(\cdot)$ is the nonlinear activation function, such as the rectified linear unit function.

Here, the 3D CNNs is used to extract the features across temporal residuals and spatial dimensions. Then, datasets $HSLT_{4D}$ and δ_{HTLS} are used to predict the restored HSHT dataset $HTHS_{4D} \in R^{T_C \times W \times H \times B}$. Because some details of the extracted features are aborted when passing pooling layers [20], we choose not to use pooling layers and simply use the simple structure of fully convolutional layers.

2.3. Overall Frame

Here, we use HTLS_{4D}, HSLT_{4D}, and HTHS_{4D} to represent the HTLS, HSLT, and HSHT series with four-dimensional structures. The main idea of this method is to learn the mappings between the residual series of HTLS_{4D} and HSLT_{4D}. The architecture contains two main parts: the 4D residual series arrangements and the 4D residual feature mapping networks. First, the 4D residual series arrangements arrange the HSLT images and HTLS images into a 4D residual series dataset using Formulas (1)–(3). The 4D residual feature mapping networks are used to learn and extract the features of temporal residuals and spatial dimensions 4-dimensionally. The architecture of the mapping networks is simple and lightweight. It consists of several fully convolutional networks, after which leaky rectified linear unit layers are set to add nonlinearity and reduce the effect of overfitting.

The flow chart of the training stage is shown in Figure 2. In the training stage, the original HTLS_{4D} dataset is first used to generate a subset according to the same/nearest dates of the HSLT_{4D}. That is, it selects images of HTLS with the same/nearest dates to the HSLT and arranges them in the 4D dataset, defined as $\mathrm{HTLS'}_{4D}$. As the inputs for the training process, the $\mathrm{HTLS'}_{4D}$ and the HSLT_{4D} are first preprocessed with 4D residual spatial arrangements to obtain the 4D residual series of the $\mathrm{HTLS'}_{4D}$ and the HSLT_{4D} dataset, which we represent as $\delta_{HTLS'}$ and δ_{HSLT}. Then, the 4D residual series are input into the 4D residual mapping networks for training. The $\delta_{HTLS'}$ is set as the network input and the δ_{HSLT} is set as the network output.

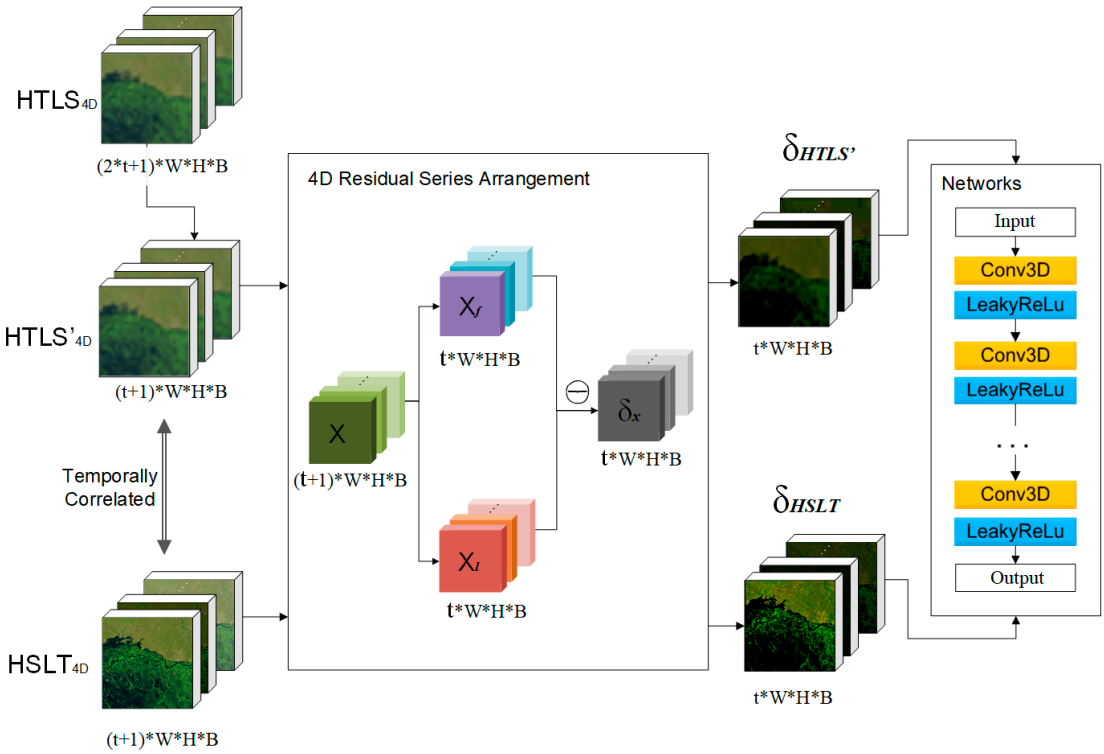


Figure 2. Training stage of the fast three-dimensional convolutional neural network-based spatiotemporal fusion method (STF3DCNN).

Figure 3 shows the predicting stage of STF3DCNN. Here, the HTLS4D is firstly preprocessed with the 4D residual spatial arrangements to obtain the 4D residual series δ_{HTLS}. Then, the δ_{HTLS} is input into the networks to predict the simulated 4D residual series of the HSHT dataset. Finally, the 4D residual series of the HSHT dataset is added to the HSLT4D, and, together with the original HSLT4D, the predicted HSTH4D is generated.

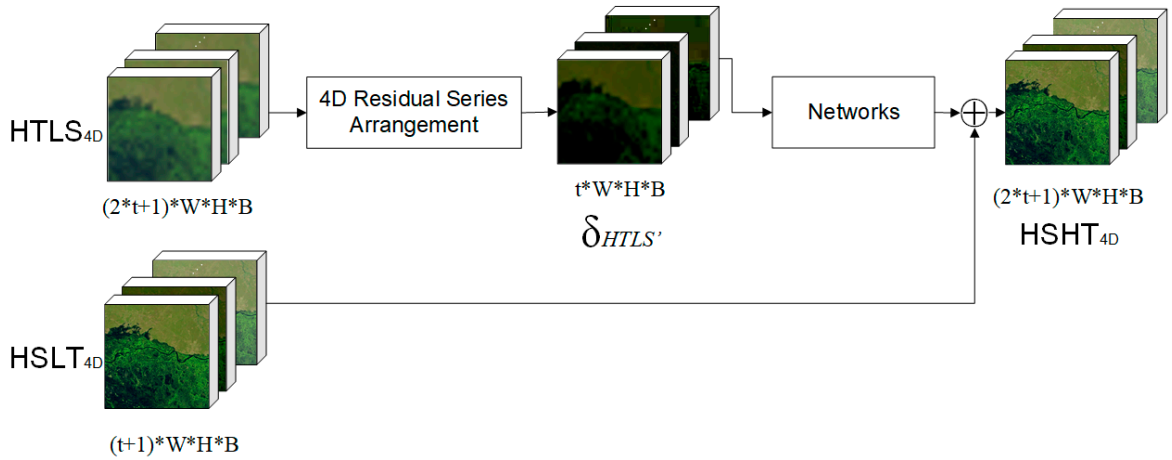


Figure 3. Predicting stage of the fast three-dimensional convolutional neural network-based spatiotemporal fusion method (STF3DCNN).

3. Experiments and Datasets

3.1. Datasets

To validate the effectiveness of the method, the following three datasets were used in the experiment.

(I) The Coleambally Irrigation Area (CIA) dataset is an open-source remote sensing dataset of a rice irrigation system located in southern New South Wales, Australia (34.0034° E, 145.0675° S), and is widely used in time-series remote sensing research [21]. It contains 17 pairs of cloud-free Landsat 7 Enhanced Thematic Mapper and Moderate Resolution Imaging Spectroradiometer (MODIS) data (MOD09GA) sensed during the 2001–2002 summer growing season. All the Landsat data underwent atmospheric correction using MODTRAN4. The CIA dataset overlaps two adjacent paths, and the total area is 2193 km ^{2} (1720 columns × 2040 lines, 25 m resolution upsampled by the data producer using nearest neighbor). For simplification, all the scenes are cropped to 1500 columns × 2000 lines. This dataset is a good sample for evaluating seasonal changes with complicated landcovers.

(II) The Lower Gwydir Catchment (LGC) dataset is another open-source remote sensing dataset sensed in the northern part of New South Wales (149.2815° E, 29.0855° S) from April 2004 to April 2005 [21]. It consists of 14 pairs of cloudless Landsat 5 Thematic Mapper and MODIS data (MOD09GA) with 3200 columns × 2720 lines at a 25 m spatial resolution. All the Landsat data underwent atmospheric correction using MODTRAN4. MODIS data contain irregular salt-and-pepper noise that is difficult to remove and may cause more distortion. Simulation is often used in spatiotemporal experiments, thus here we used a similar method to that used in Reference [8] which arranges MODIS-like data using Landsat data, and added 40 dBW of Gaussian white noise to simulate approximately the signal-to-noise ratio (SNR) of MODIS data, as shown in Reference [22]. For simplification, all the scenes were cropped to 3000 columns × 2500 lines. On the 10th date of the time-series data, a flood occurred in this field, making this dataset a good sample of a long time-series with abrupt changes, and it is more unpredictable and irregular than the CIA dataset.

(III) The Real Dataset (RDT) was sensed in Williams, CA, USA (122°15'34.33" W, 39°10'4.17" N), from June to October 2013. It consists of nine pairs of Landsat 8 Operational Land Imager and MODIS data (MOD09A1) with the spatial resolution of 30 m. Each scene was cropped to 800 columns × 800 lines. A subset of four bands was used. All the data were preprocessed using geographic registration and atmospheric correction. The MODIS data were resampled to the same resolution as that of the Landsat data using the bilinear resampling algorithm. This dataset is a good sample for evaluating seasonal changes in crop fields and vegetated mountain areas.

To download the Landsat–MODIS imagery of CIA and LGC datasets, see http://dx.doi.org/10.4225/08/5111AC0BF1229 for the CIA dataset and http://dx.doi.org/10.4225/08/5111AD2B7FEE6 for the LGC dataset. The acquisition dates and band information used in the experiments are listed in Tables 2 and 3.

Table 2. Acquisition dates of datasets of CIA, LGC, and RDT.

CIA			LGC			RDT		
Image #	**Date**	**Intervals**	**Image #**	**Date**	**Intervals**	**Image #**	**Date**	**Intervals**
1	8-Oct-2001		1	16-Apr-2004		1	3-Jun-2013	
		9			16			16
2	17-Oct-2001		2	2-May-2004		2	19-Jun-2013	
		16			64			16
3	2-Nov-2001		3	5-Jul-2004		3	5-Jul-2013	
		7			32			16
4	9-Nov-2001		4	6-Aug-2004		4	21-Jul-2013	
		16			16			32
5	25-Nov-2001		5	22-Aug-2004		5	22-Aug-2013	
		9			64			16
6	4-Dec-2001		6	25-Oct-2004		6	7-Sep-2013	
		32			32			16
7	5-Jan-2002		7	26-Nov-2004		7	23-Sep-2013	
		7			16			16

8	12-Jan-2002		8	12-Dec-2004		8	9-Oct-2013	
		32			16			16
9	13-Feb-2002		9	28-Dec-2004		9	25-Oct-2013	
		9			16			
10	22-Feb-2002		10	13-Jan-2005				
		16			16			
11	10-Mar-2002		11	29-Jan-2005				
		7			16			
12	17-Mar-2002		12	14-Feb-2005				
		16			16			
13	2-Apr-2002		13	2-Mar-2005				
		9			32			
14	11-Apr-2002		14	3-Apr-2005				
		7						
15	18-Apr-2002							
		9						
16	27-Apr-2002							
		7						
17	4-May-2002							

Image # represents the sequence number of the images in the dataset

Table 3. Band information of datasets CIA, LGC, and RDT.

CIA		LGC		RDT	
Band Names	**Wavelengths**	**Band Names**	**Wavelengths**	**Band Names**	**Wavelengths**
Band 1 Visible	0.45–0.52 μm	Band 1 Visible	0.45–0.52 μm	Band 2 Visible	0.450–0.51 μm
Band 2 Visible	0.52–0.60 μm	Band 2 Visible	0.52–0.60 μm	Band 3 Visible	0.53–0.59 μm
Band 3 Visible	0.63–0.69 μm	Band 3 Visible	0.63–0.69 μm	Band 4 Red	0.64–0.67 μm
Band 4 Near-Infrared	0.77–0.90 μm	Band 4 Near-Infrared	0.76–0.90 μm	Band 5 Near-Infrared	0.85–0.88 μm
Band 5 Short-wave Infrared	1.55–1.75 μm	Band 5 Near-Infrared	1.55–1.75 μm		
Band 7 Mid-Infrared	2.08–2.35 μm	Band 7 Mid-Infrared	2.08–2.35 μm		

3.2. Data Preprocessing and Experimental Settings

In order to assess our method, we reduced the actual acquisition rate of the Landsat images to two intervals. We used all the MODIS data series and Landsat data on the odd dates to predict the missing scenes on the even dates, and the reference images were the Landsat images on the even dates. The MODIS data in the CIA and LGC datasets were upsampled to 25 m using nearest neighbor interpolation, which is the preprocess conducted by the data producer and has a negative effect on our method. Thus, we retrieved the original 500 m MODIS data using the same method. Then, all the MODIS data in the three datasets were upsampled to the same spatial resolution using the bilinear resampling algorithm according to the need of our method.

The experiments were conducted using Python with Keras 2.0. The graphics processing unit used for acceleration was the NVIDIA Quadro P6000, and the central processing unit used was the Intel Xeon Silver 4110. For the STF3DCNN, the Adam optimization algorithm was used: the learning rate was set to 0.001 and the batch size was 32. The loss function was mean least square. For better optimization, reducing on the plateau and early stopping strategies were used. The learning rate was

reduced by a factor of 0.2 when the loss did not change within 5 epochs, and the training was stopped if the loss did not change within 15 epochs. The original number of epochs for training was set to 1000. To test the accuracy and efficiency of the different experiments, fusion quality indices and running time were utilized. The indices used were:

(1) Correlation coefficient (CC) [23]: CC is an important indicator for measuring the relevance between two images. The CC, CC_k, for band k can be represented as:

$$CC_k = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(R_k(i,j)-\overline{M(R)}_k\right)\left(F_k(i,j)-\overline{M(F)}_k\right)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(R_k(i,j)-\overline{M(R)}_k\right)^2\sum_{i=1}^{m}\sum_{j=1}^{n}\left(F_k(i,j)-\overline{M(F)}_k\right)^2}}, \quad (4)$$

where $R_k(i, j)$ represents the pixel value at the *k*-th band at the (*i*, *j*) position of the real reference image, $F_k(i, j)$ represents the pixel value at the *k*-th band at the (*i*, *j*) position of the fused image and $\overline{M(R)}_k$ and $\overline{M(F)}_k$ respectively, represent the average gray values of the two images in the band *k*. The closer the CC is to 1, the better relevance between the two images.

(2) Spectral angle mapper (SAM) [24]: SAM is a common index used to measure the spectral similarity between two spectral vectors or remote sensing images. The SAM of the (*i*, *j*)th pixel between the two images is as follows:

$$\theta(i,j) = \arccos\left(\frac{\left\langle \vec{R}(i,j), \vec{F}(i,j)\right\rangle}{\left\|\vec{R}(i,j)\right\| \left\|\vec{F}(i,j)\right\|}\right), \quad (5)$$

where $\vec{R}(i,j)$ and $\vec{F}(i,j)$ represent the two spectral vectors of the (*i*, *j*)th pixel of the two images, respectively. The smaller the spectral angle, the greater the similarity between the spectra. The perfectly matched spectral angle is 0. For the evaluation of the fusion image, the average spectral angle between the fusion image and the real reference image can be obtained by averaging the spectral angle of every pixel to evaluate the preservation of the spectral characteristics of the fusion image.

(3) Peak signal-to-noise ratio (PSNR) [25]: PSNR is one of the most commonly used evaluation indices for image fusion evaluation. It evaluates whether the effective information of the fused image is enhanced compared to the original image, and it characterizes the quality of the spatial information reconstruction of the image. The PSNR of band k can be represented as:

$$PSNR_k = 10\log_{10}\left(mn(\max(k))^2 \Big/ \sum_{i=1}^{m}\sum_{j=1}^{n}\left[R_k(i,j)-F_k(i,j)\right]^2\right), \quad (6)$$

where $\max(k)$ represents the maximum possible value of the k-th band of the image. The larger the PSNR, the better the quality of the spatial information of the image.

(4) Universal image quality index (UIQI, Q) [26]: The UIQI is a fusion evaluation index related to the correlation, brightness difference, and contrast difference between images. The formula for calculating the UIQI index of the k-th band of the image before and after fusion is:

$$UIQI_k = \frac{SD(RF)_k}{SD(R)_k\, SD(K)_k} \cdot \frac{2\overline{M(R)}_k\,\overline{M(F)}_k}{[\overline{M(R)}_k]^2+[\overline{M(F)}_k]^2} \cdot \frac{2SD(R)_k\, SD(K)_k}{SD(R)_k{}^2 + SD(K)_k{}^2}, \quad (7)$$

where $SD(RF)_k$ represents the co-standard deviation between the corresponding bands of the two images. The ideal value of UIQI is 1. The closer the index is to 1, the better the fusion effect is.

Among the indices mentioned above, CC and PSNR measure the global fusion accuracies, SAM measures the fusion accuracies in the spectral domain, and UIQI measures the fusion accuracies in the spatial domain. Thus, using the four fusion indices is sufficient for measuring the fusion qualities.

4. Discussion

To test the theory of our method, different parameters and data arrangements were discussed in the ablation study, including the effects of utilizing (1) former prediction and latter prediction with temporal weights, (2) different convolutional layers, and (3) residual blocks (short connection).

As mentioned in Section 2.1, the residual series of MODIS δ_{C1} can be arranged by subtracting former subset series from latter subset series, which can be represented as follows:

$$\delta_{C1} = C(t_{l1}, x, y, b) - C(t_{f1}, x, y, b), \tag{8}$$

where C represents the 4D HTLS data, $t_{l1} \in \{2, 4, ..., T-1\}$, $t_{f1} \in \{1, 3, ..., T-2\}$, $x \in \{1, 2, ..., W\}$, $y \in \{1, 2, ..., H\}$, and $b \in \{1, 2, ..., B\}$. After mapping to the Landsat residual series and adding the original Landsat series, the reconstructed prediction dataset can be represented as follows:

$$X_1 = f_1(\delta_{C1}) + F(t_{f1}, x, y, b), \tag{9}$$

where f_1 represents the trained residual feature mappings and F represents the 4D HSLT data. The formulas above are for using the predicted images and the images before them to fuse the missing images. Another way is to fuse using the predicted images and the latter images. The residual series δ_{C2} is shown as Formula (11) below:

$$\delta_{C2} = C(t_{l2}, x, y, b) - C(t_{f2}, x, y, b), \tag{10}$$

where $t_{f2} \in \{2, 4, ..., T-1\}$ and $t_{l2} \in \{3, 5, ..., T\}$. The final reconstructed prediction dataset is

$$X_2 = F(t_{l2}, x, y, b) - f_2(\delta_{C2}), \tag{11}$$

where f_2 represents the trained residual feature mappings. Thus, we obtain two results, X_1 and X_2, based on the different residual series.

Here, we introduce the temporal weight to combine these two results, and the weights are larger when the dates are close together. The temporal weight calculation can be represented as follows:

$$X = \frac{\frac{X_1}{d_1} + \frac{X_2}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}}, \tag{12}$$

where d_1 and d_2 represent the vectors of time intervals for the left and right dates of the prediction dates respectively, and can be calculated as

$$d_1 = \mathrm{t}_{l1} - \mathrm{t}_{f1},\ d_2 = \mathrm{t}_{f2} - \mathrm{t}_{l2}. \tag{13}$$

For three datasets, we set the depth of the network as 3 to 9, and each experiment yields two results using the single-date prediction (Formulas (9) and (10)) and the temporal weighted prediction (Formulas (9)–(14)). We also set experiments utilizing two residual blocks on three datasets. The number of experiments is 45 in total.

4.1. Effects of Utilizing Temporal Weights

In order to show the overall effect of utilizing temporal weights, we used the results of two temporal strategies, which contain the results of models with all depths, to construct a boxplot to show the effect of utilizing temporal weights. The results are produced by grouping the single-date prediction results together regardless of the depths or dates of images. The results for three different datasets are shown in Figures 4–6. In each subplot, the x-axis represents the weight strategy and the y-axis represents the different fusion indices. As shown in Figures 4 and 5, the results of the CIA and LGC datasets have the same tendency: the temporal weighted results are worse than the single-date prediction results across all the fusion quality indices. Additionally, the ranges of the temporal weighted results are larger than those of the single-date results. The more abrupt or unpredictable the change in the dataset, the more obvious the tendency (judging by boxplot ranges and averages). However, for the RDT, the temporal weighted prediction yielded better results, with smaller ranges, compared with the single-date prediction. The reason for this may be that, when the landcover undergoes abrupt or irregular changes, the temporal weight strategy averages the abrupt changes, decreasing the accuracy, but when the landcovers change regularly, the temporal weights combine the changes to obtain the results. Thus, this method may depend highly on landcover changes. However, for the three datasets, the characteristics of the outliers are the same. The outliers of the single-date results are close to the maximum/minimum of the boxplot, yet the outliers of the temporal weighted results are far away from the maximum/minimum of the boxplot. This may indicate that the temporal weighted method may cause some uncertainties.

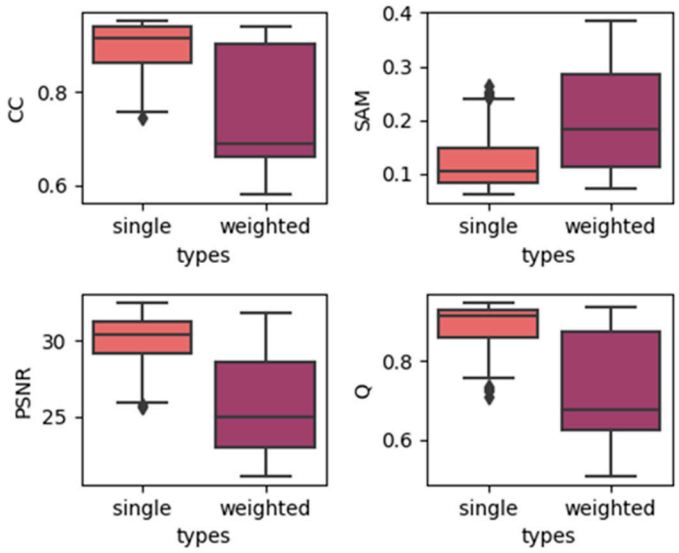


Figure 4. Results of using single/weighted double images in the CIA dataset.

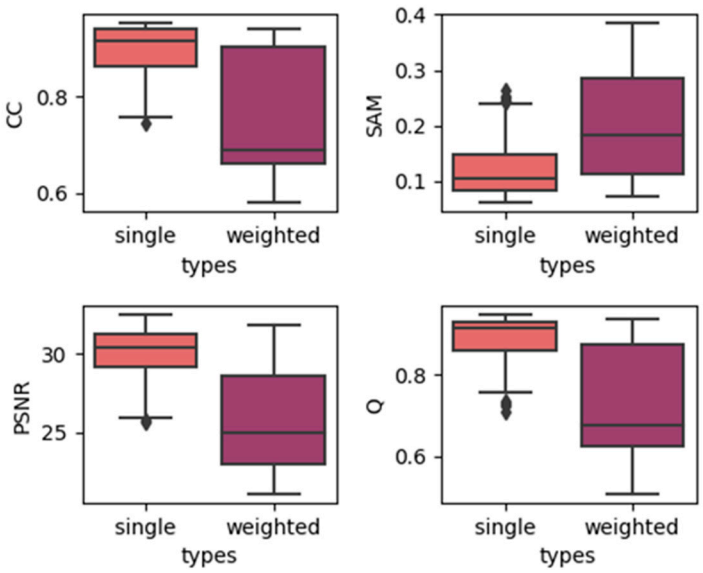


Figure 5. Results of using single/weighted double images in the LGC dataset.

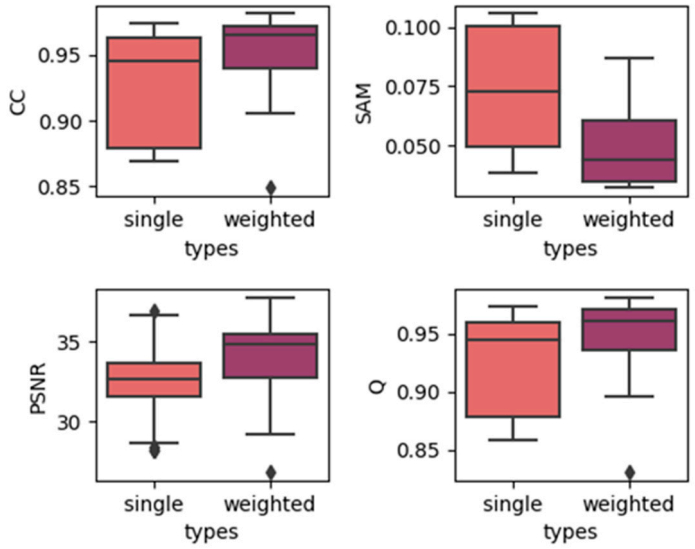


Figure 6. Results of using single/weighted double images in the RDT dataset.

4.2. Effects of Network Depth

To show the effects of network depth, we generated boxplots of experiments with different network depths. The results are produced by combining the results of the same depth together with the best temporal strategy, regardless of the predicted dates. The results are shown in Figures 7–9. In each subplot, the *x*-axis indicates the depth (layers) of the utilized model, and the *y*-axis represents the values of the different fusion indices. In the three datasets, increasing the model's depth did not increase the accuracy. In the CIA and LGC datasets, the accuracy decreases when the depth increases. As the depth increases, the outliers become more distant from the other values. In the RDT datasets, although the tendency was not very clear, as the depth increases, the accuracies do not significantly increase. The outliers do not show the same tendency with CIA and LGC, and this may be due to the reason mentioned in Section 4.1: the temporal weighted strategy may cause some uncertainties in certain points. Since the computational cost and running time increase when the depth of the model increases, three-layer fully convolutional networks are the best choice for our method. It is not worth the cost to use more layers, particularly for the architecture with residual series. This may be due to

the reason that the features of the residual series are not obvious and it is easy to cause overfitting. As the depth increases, it becomes easier to learn the mappings too delicately.

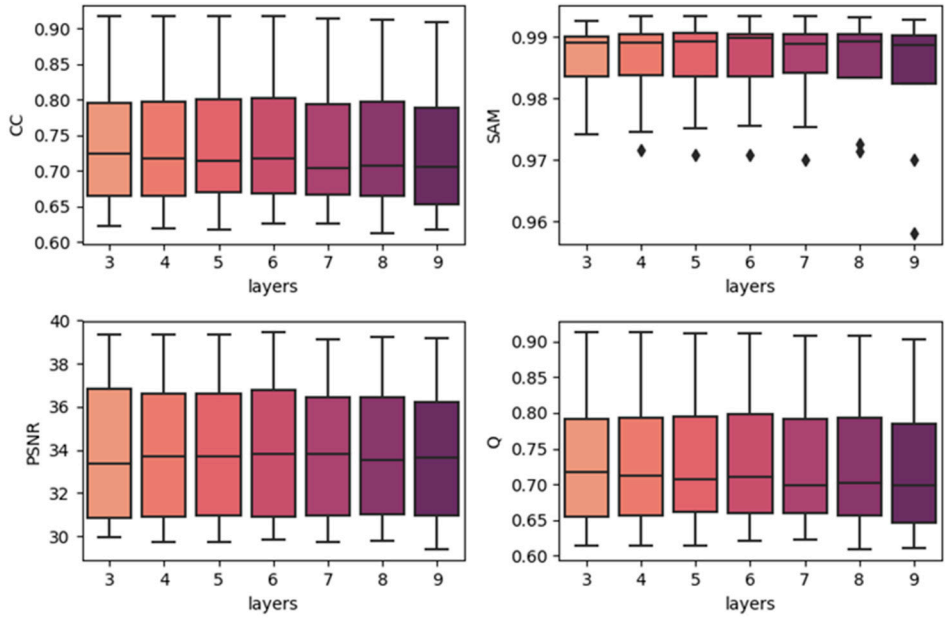


Figure 7. Results of using different network depths in the CIA dataset (single-date prediction).

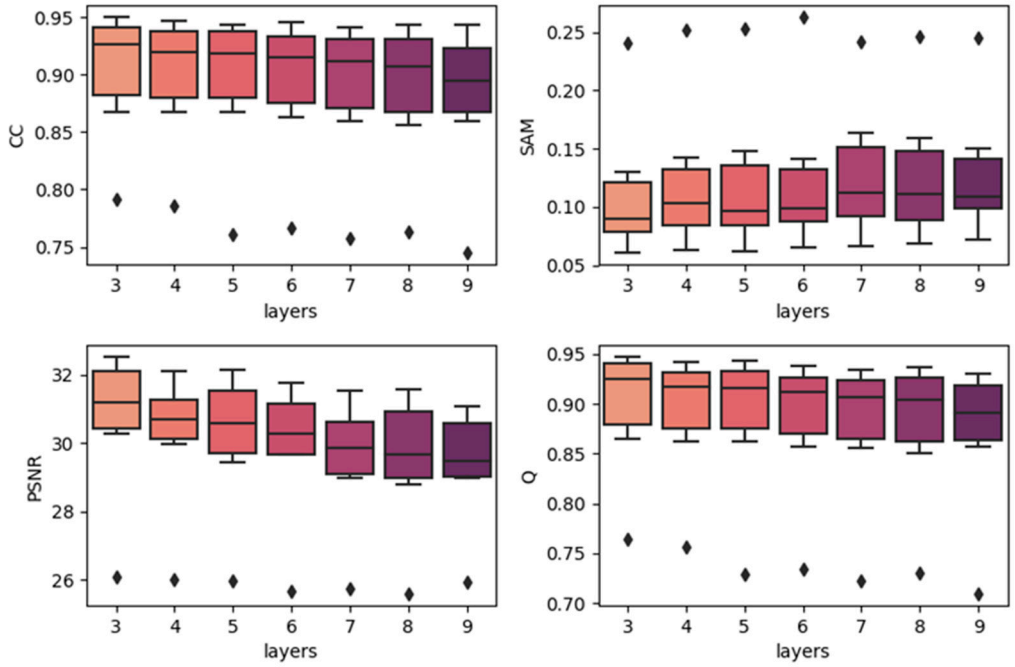


Figure 8. Results of using different network depths in the LGC dataset (single-date prediction).

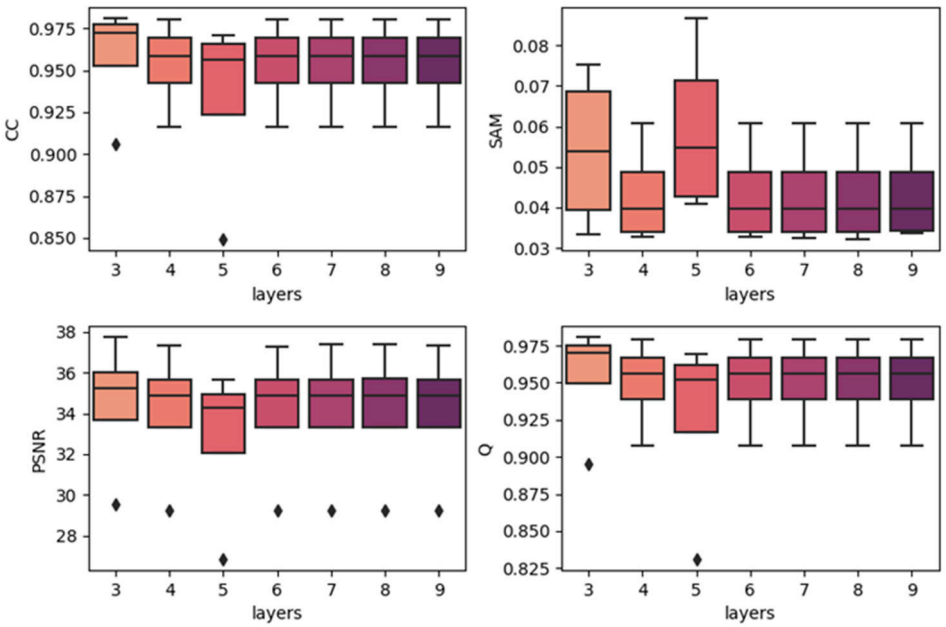


Figure 9. Results of using different network depths in the RDT (weighted date prediction).

4.3. Effects of Utilizing Residual Blocks

Piling a greater number of convolutional network layers makes it easy for the model to encounter a vanishing gradient. The idea of residual networks is proposed to deal with the difficulty of training deeper networks. The purpose is to learn the residuals of two CNN layers to overcome redundancy and a vanishing gradient. As the accuracy of our model decreases when the layer count increases, we attempted to utilize residual blocks to determine whether the accuracy could be increased. The architecture is shown in the graph in Figure 10.

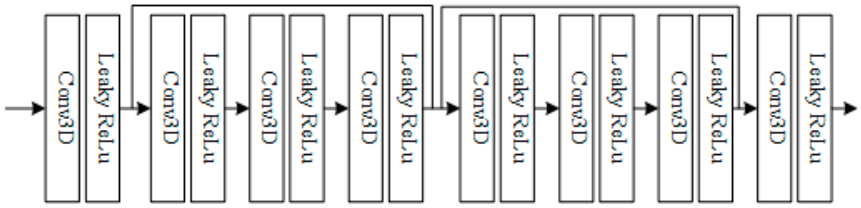


Figure 10. Residual blocks implemented in the mapping model.

Here, we generated boxplots of experiments on the three datasets with residual blocks and three-layer networks (regardless of the prediction dates), which are presented in the Figures 11–13. In each subplot, the x-axis represents the utilized model with the residual block (noted as resblk) and the model with three convolutional neural network layers (3lrs). The y-axis indicates the values for different validation indices. With all the datasets, utilizing residual blocks did not increase the accuracy and instead decreased the average accuracy as well as further increasing the divergence in the results. In addition, as the residual blocks were utilized, the processing time was increased. This may be due to the fact that, in a more sophisticated model, it is easier to cause overfitting when the feature is not obvious.

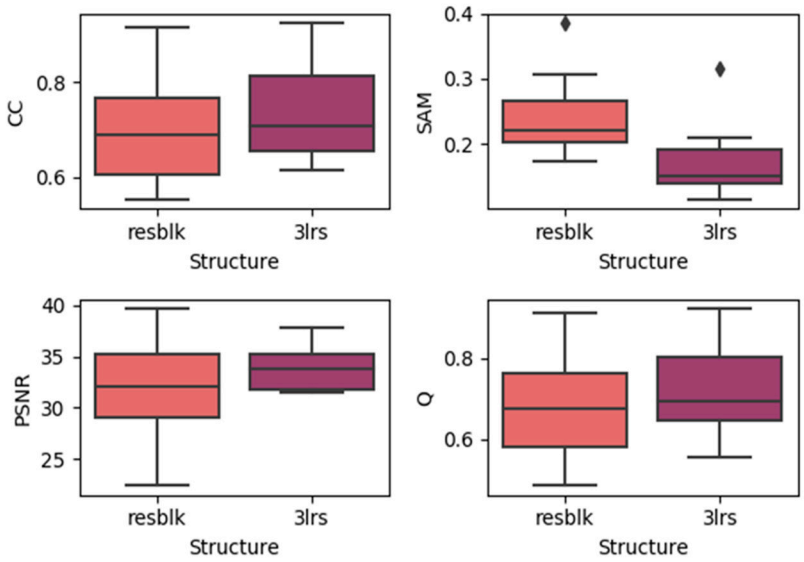


Figure 11. Results of using residual blocks in the CIA dataset.

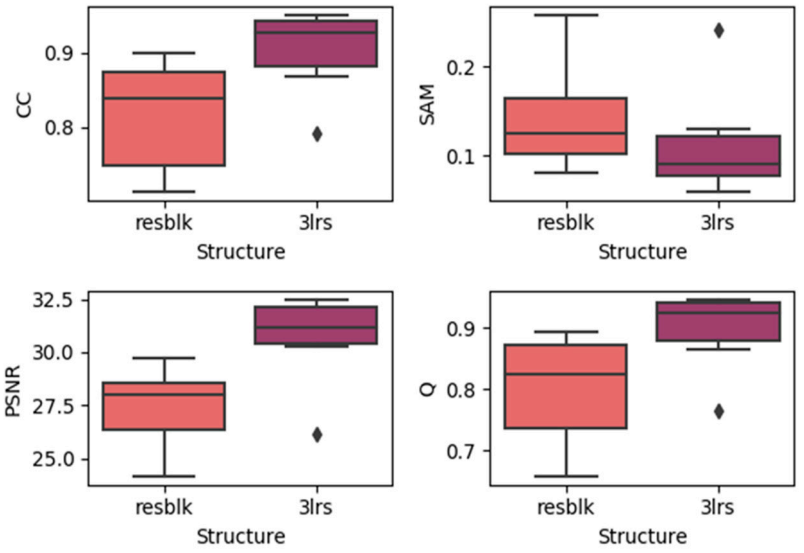


Figure 12. Results of using residual blocks in the LGC dataset.

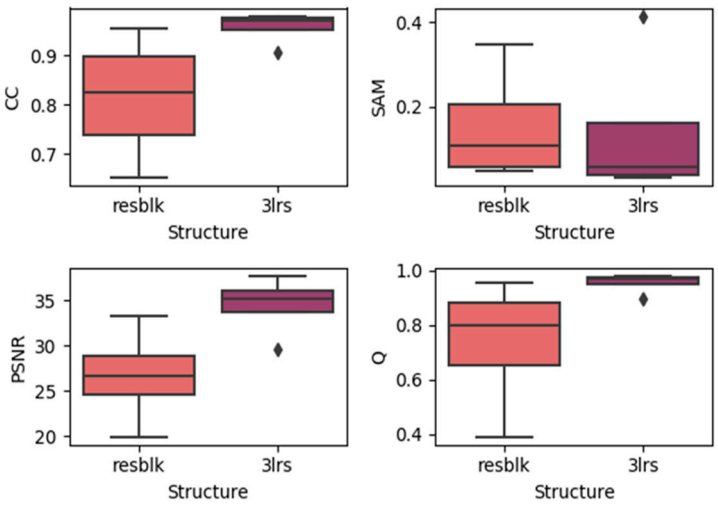


Figure 13. Results of using residual blocks in the RDT dataset.

4.4. Comparison with State-of-the-Art Spatiotemporal Fusion Methods

The goal of our method is to increase the efficiency while ensuring that the accuracy is maintained in comparison with existing spatiotemporal fusion methods. To verify whether our goals were achieved, we chose the best result from our experiments and compared it with the existing spatiotemporal fusion methods of ESTARFM, FSDAF, and DCSTFN. ESTARFM and FSDAF are traditional spatiotemporal fusion methods, while DCSTFN is a deep learning-based spatiotemporal method using 2D CNNs. For ESTARFM, we used the fast version, with the window size set to 25, the estimated number of classes set to 4, and the patch size set to 500. For FSDAF, we also used the fast version, with the window size set to 25, the estimated number of classes set to 4, and the patch size set to 50. For DCSTFN, we trained the model using Landsat and MODIS on the same date and predicted the missing Landsat images: the number of epochs was set to 100, the optimizer was Adam, the learning rate was set to the default as 0.001, and the patch size was set to 64. We measured the accuracy of each method using the nine indices and recorded the running time for the entire long time-series dataset.

Figures 14–16 show bar charts of the four indices for each method. For each subplot, the x-axis represents the dates indexed in chronological order and the y-axis represents the different fusion indices. Experiments on the CIA dataset show that the CC and Q were slightly lower for our method than for the other methods from a general perspective, but were not the lowest in all experiments; however, our method performed the best in terms of the other indices among all spatiotemporal fusion methods used in the experiments from a general perspective. For the SAM and PSNR, our method performed the best in seven predictions. Experiments on the LGC dataset show that for the CC, PSNR, and Q, our method outperforms other methods in five predictions. For SAM, the results of our methods are slightly poor yet still not the worst among all the methods in the experiment. Experiments on the RDT dataset indicate that our method yields the best results.

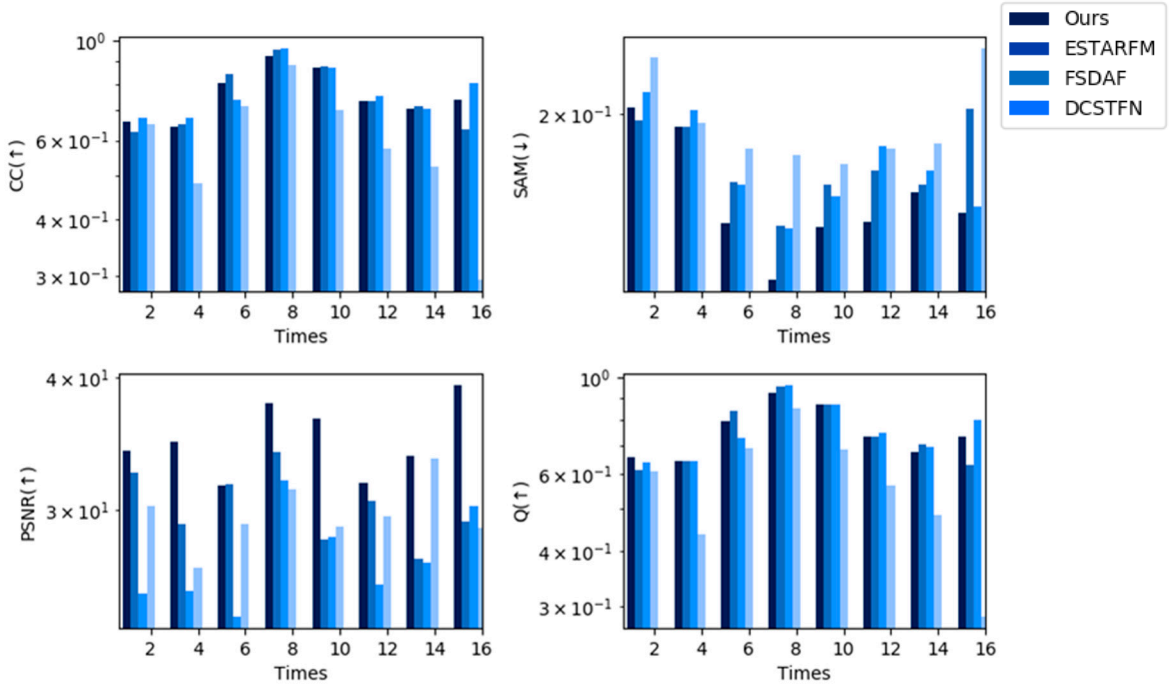


Figure 14. Indices using the CIA dataset (STF3DCNN, ESTARFM, FSDAF, DCSTFN).

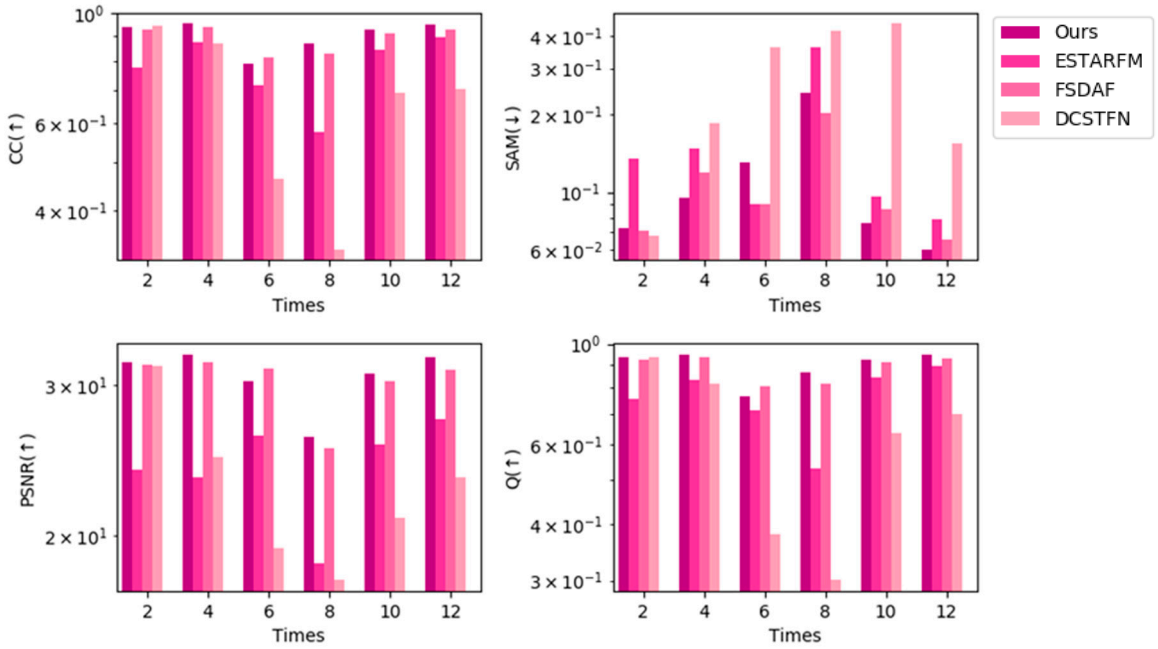


Figure 15. Indices using the LGC dataset (STF3DCNN, ESTARFM, FSDAF, DCSTFN).

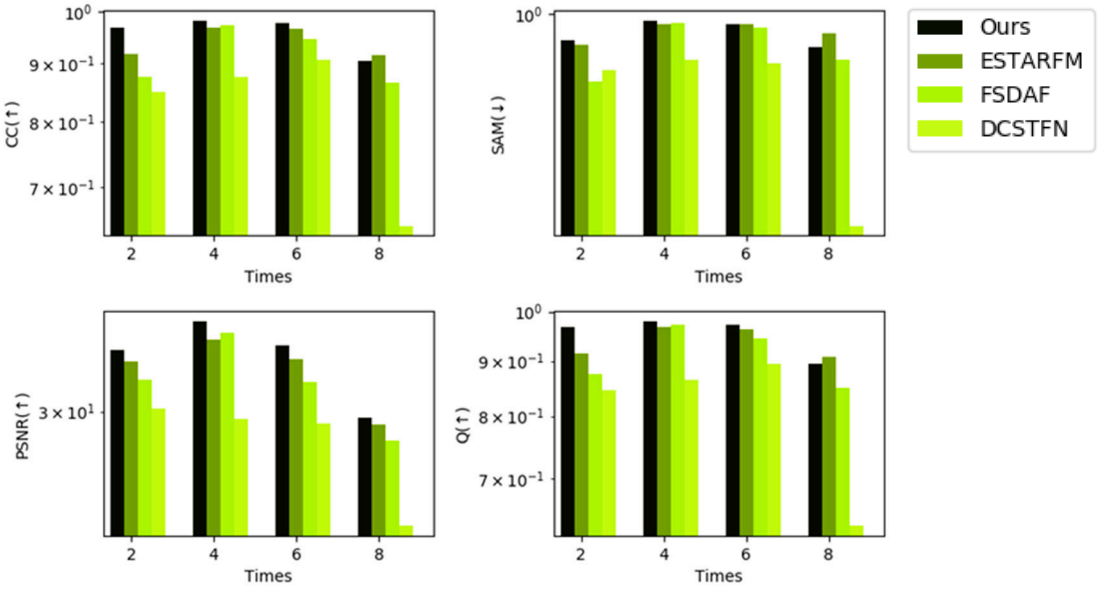
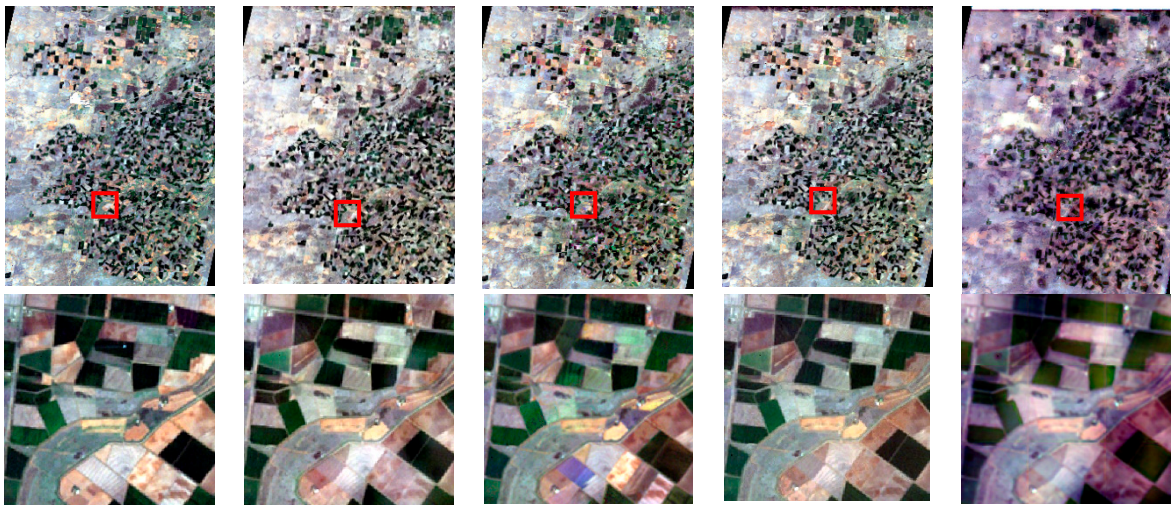


Figure 16. Indices using the RDT (STF3DCNN, ESTARFM, FSDAF, DCSTFN).

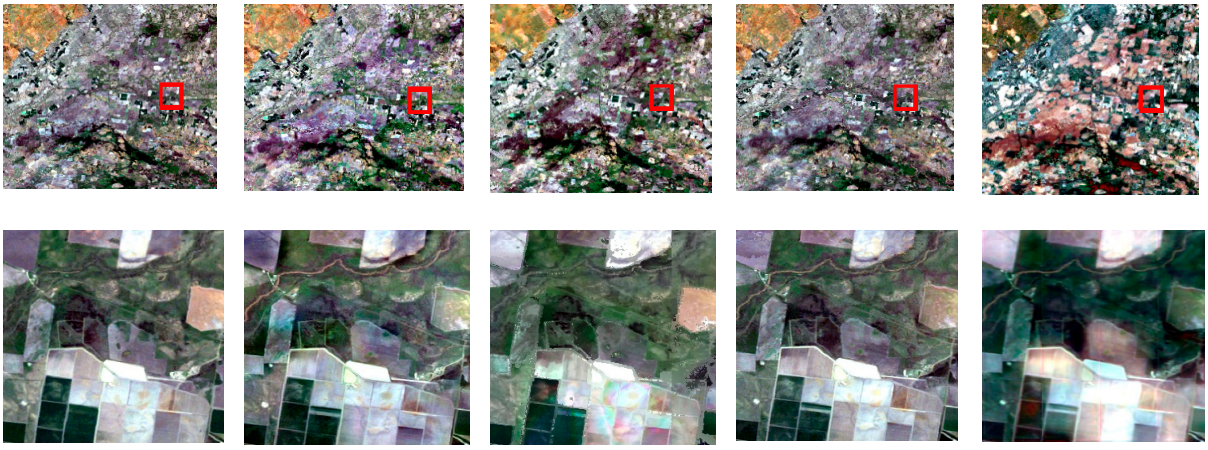
To compare the texture and tones of the reconstructed dataset, we chose one date of each dataset to display the fusion results of the whole scene in detail. For the CIA dataset, the 10th date was selected, and the RGB composites are shown in Figure 17. Our method showed a good recovery of the texture and tone similar to that of the FSDAF. As seen in the results of the ESTARFM, some landcovers were mis-predicted. The DCSTFN recovered the data but suffered from blur. Thus, for the CIA dataset, our method and the FSDAF predicted the missing image best.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 17. RGB composites (Red-Band 3, Green-Band 2, Blue- Band 1) of the results of the CIA dataset on date 10 (overall and in detail): (**a**) reference image, (**b**) STF3DCNN, (**c**) ESTARFM, (**d**) FSDAF, and (**e**) DCSTFN.

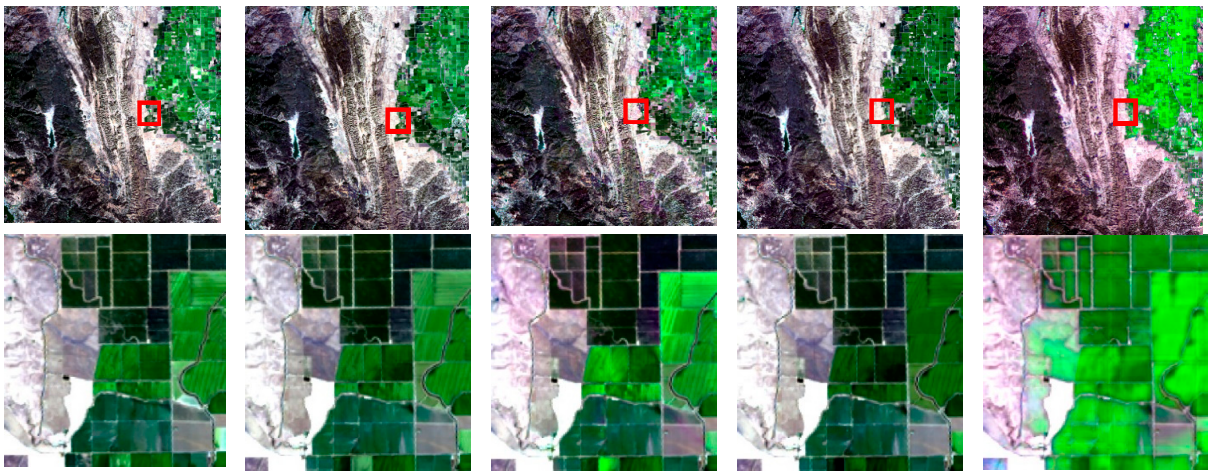
In the LGC dataset, the 10th date was selected, which was the date that the flood occurred, causing drastic change. The RGB composites are shown in Figure 18. Our method was able to recover the flooded area with an accurate texture and tone. The FSDAF could also well-predict the flooded area and seemed to outperform the ESTARFM. With the DCSTFN, the tones were slightly different from the reference image and the texture suffered from blur.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 18. RGB (Red-Band 3, Green-Band 2, Blue- Band 1) composites of the results of the LGC dataset on date 10 (overall and in detail): (**a**) reference image, (**b**) STF3DCNN, (**c**) ESTARFM, (**d**) FSDAF, and (**e**) DCSTFN.

In the RDT dataset, the 6th date was selected, and the RGB composites are shown in Figure 19. In the overall and detailed images (with multiple landcover types), our method performed well, as did the FSDAF. The tone was not completely true to the original when using the DCSTFN and ESTARFM.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 19. RGB (Red-Band 4, Green-Band 3, Blue- Band 2) composites of the results of the LGC dataset on date 10 (overall and in detail): (**a**) reference image, (**b**) STF3DCNN, (**c**) ESTARFM, (**d**) FSDAF, and (**e**) DCSTFN.

To present the average accuracies of our methods, we calculated the average indices of each method on all dates and recorded the overall running times of all datasets. The average accuracies and total running times are shown in Tables 4 and 5. For all the fusion indices, the performance of the STF3DCNN was the best among the four methods. Most importantly, the total running time was significantly decreased. Compared with traditional methods using a central processing unit, our method was able to decrease the running time by 10^9 times (ESTARFM) and 10^4 times (FSDAF). Compared with the DCSTFN, a deep learning spatiotemporal fusion method, the average running time decreased 12-fold among three datasets. We also observed the tendency of a higher efficiency with larger datasets. The experiments demonstrated that the STF3DCNN successfully maintains the average accuracy of the existing methods, while, at the same time, significantly increasing the efficiency.

Table 4. Average fusion indices of the three datasets.

	CC(↑)	SAM(↓)	PSNR(↑)	Q(↑)
STF3DCNN	**0.8740**	**0.1201**	**33.3709**	**0.8684**
ESTARFM	0.8255	0.1471	29.2558	0.8158
FSDAF	0.8595	0.1266	29.9007	0.8525
DCSTFN	0.6969	0.2227	26.8396	0.6715

[1] Bold values represent the best performance.

Table 5. Running times of the entire time series using different methods.

	CIA	LGC	RDT
STF3DCNN	**552**	**987**	**77**
ESTARFM	6.40×10^{11}	9.63×10^{15}	14,435.744
FSDAF	2.94×10^{6}	6.40×10^{9}	7595.211
DCSTFN	6910	12,278	489.4740

[1] Bold values represent the best performance. [2] Times are expressed in seconds.

5. Conclusions

We proposed a fast spatiotemporal fusion method using 3D fully convolutional networks (STF3DCNN) on a spatial-temporal-spectral dataset. The long time-series data were arranged and operated four-dimensionally in a spatial-temporal-spectral dataset based on the idea of an MDD. By learning the mappings between residual series of HTLS and HSLT datasets using 3D fully

convolutional networks, the model can restore the 4D HSHT dataset with a high efficiency compared with existing methods, while maintaining the accuracy.

We used three datasets to verify the efficiency and accuracy of our method. An ablation study was performed to discuss the effects of parameters on long time-series spatiotemporal fusion on different occasions, including network depth, the use of temporal weights, and the use of residual blocks. Fewer layers resulted in a better accuracy and increased efficiency. Temporal weights improved the model performance for landcover with regular seasonal changes, but not for landcover with sudden irregular changes, and utilizing residual blocks did not increase the accuracy. Finally, our method was compared with existing methods in terms of accuracy and running time. Our method can highly improve the efficiency while maintaining its overall accuracy, especially in the case of very large datasets covering long time periods.

Author Contributions: Conceptualization, L.Z., M.P. and X.S.; methodology, M.P.; software, M.P.; validation, M.P.; formal analysis, M.P.; investigation, M.P.; resources, L.Z., X.S. and Y.C.; data curation, M.P.; writing—original draft preparation, M.P.; writing—review and editing, L.Z. and M.P.; visualization, M.P. and X.Z.; supervision, L.Z.; project administration, L.Z. and X.S.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 41830108,41701404, the National Key R&D Program of China, grant number 2017YFC1500900, XPCC major science and technology projects, grant number 2018AA004, and XPCC innovation team in key areas, grant number 2018CB004.

Acknowledgments: The authors would like to thank anonymous reviewers for their great comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, L.; Peng, M.; Sun, X.; Cen, Y.; Tong, Q. Progress and bibliometric analysis of remote sensing data fusion methods (1992–2018). *J. Remote Sens.* **2019**, *23*, 1993–2002, doi:10.11834/jrs.20199073.
2. Gao, F.; Masek, J.; Schwaller, M.; Hall, F. On the blending of the Landsat and MODIS surface reflectance: Predicting daily Landsat surface reflectance. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2207–2218.
3. Hilker, T.; Wulder, M.A.; Coops, N.C.; Linke, J.; Mcdermid, G.; Masek, J.G.; Gao, F.; White, J.C. A new data fusion model for high spatial- and temporal-resolution mapping of forest disturbance based on Landsat and MODIS. *Remote Sens. Environ.* **2009**, *113*, 1613–1627.
4. Zhu, X.; Jin, C.; Feng, G.; Chen, X.; Masek, J.G. An enhanced spatial and temporal adaptive reflectance fusion model for complex heterogeneous regions. *Remote Sens. Environ.* **2010**, *114*, 2610–2623.
5. Fu, D.; Chen, B.; Wang, J.; Zhu, X.; Hilker, T. An Improved Image Fusion Approach Based on Enhanced Spatial and Temporal the Adaptive Reflectance Fusion Model. *Remote Sens.* **2013**, *5*, 6346–6360, doi:10.3390/rs5126346.
6. Wang, J.; Huang, B. A Rigorously-Weighted Spatiotemporal Fusion Model with Uncertainty Analysis. *Remote Sens.* **2017**, *9*, 990, doi:10.3390/rs9100990.
7. Zhukov, B.; Oertel, D.; Lanzl, F.; Reinhackel, G. Unmixing-based multisensor multiresolution image fusion. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 1212–1226.
8. Zhu, X.; Helmer, E.H.; Gao, F.; Liu, D.; Chen, J.; Lefsky, M.A. A flexible spatiotemporal method for fusing satellite images with different resolutions. *Remote Sens. Environ.* **2016**, *172*, 165–177.
9. Mingquan, W.; Zheng, N.; Changyao, W.; Chaoyang, W.; Li, W. Use of MODIS and Landsat time series data to generate high-resolution temporal synthetic Landsat data using a spatial and temporal reflectance fusion model. *J. Appl. Remote Sens.* **2012**, *6*, 063507, doi:10.1117/1.JRS.6.063507.
10. Zhang, W.; Li, A.; Jin, H.; Bian, J.; Zhengjian, Z.; Lei, G.; Qin, Z.; Huang, C. An Enhanced Spatial and Temporal Data Fusion Model for Fusing Landsat and MODIS Surface Reflectance to Generate High Temporal Landsat-Like Data. *Remote Sens.* **2013**, *5*, 5346–5368, doi:10.3390/rs5105346.
11. Wu, P.; Shen, H.; Zhang, L.; Gottsche, F.M. Integrated fusion of multi-scale polar-orbiting and geostationary satellite observations for the mapping of high spatial and temporal resolution land surface temperature. *Remote Sens. Environ.* **2015**, *156*, 169–181.

12. Amorós-López, J.; Gómez-Chova, L.; Alonso, L.; Guanter, L.; Zurita-Milla, R.; Moreno, J.; Camps-Valls, G. Multitemporal fusion of Landsat/TM and ENVISAT/MERIS for crop monitoring. *Int. J. Appl. Earth Obs. Geoinf.* **2013**, *23*, 132–141.
13. Lu, M.; Chen, J.; Tang, H.; Rao, Y.; Yang, P.; Wu, W. Land cover change detection by integrating object-based data blending model of Landsat and MODIS. *Remote Sens. Environ.* **2016**, *184*, 374–386.
14. Moosavi, V.; Talebi, A.; Mokhtari, M.H.; Shamsi, S.R.F.; Niazi, Y. A wavelet-artificial intelligence fusion approach (WAIFA) for blending Landsat and MODIS surface temperature. *Remote Sens. Environ.* **2015**, *169*, 243–254.
15. Song, H.; Liu, Q.; Wang, G.; Hang, R.; Huang, B. Spatiotemporal Satellite Image Fusion Using Deep Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 821–829.
16. Tan, Z.; Peng, Y.; Di, L.; Tang, J. Deriving High Spatiotemporal Remote Sensing Images Using Deep Convolutional Network. *Remote Sens.* **2018**, *10*, 1066.
17. Zhang, L.; Chen, H.; Sun, X.; Fu, D.; Tong, Q. Designing spatial-temporal-spectral integrated storage structure of multi-dimensional remote sensing images. *Yaogan Xuebao J. Remote Sens.* **2017**, *21*, 62–73, doi:10.11834/jrs.20176091.
18. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
19. Sebastianelli, A.; Del Rosso, M.P.; Ullo, S. *Automatic Dataset Builder for Machine Learning Applications to Satellite Imagery*; 2020. Available online: https://arxiv.org/abs/2008.01578 (accessed on 18 September 2020)
20. Sun, Z.; Wang, J.; Lei, P.; Qin, Z. Multiple Walking People Classification with Convolutional Neural Networks Based on Micro-Doppler. In Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; pp. 1–4.
21. Emelyanova, I.V.; McVicar, T.R.; Van Niel, T.G.; Li, L.T.; van Dijk, A.I.J.M. Assessing the accuracy of blending Landsat–MODIS surface reflectances in two landscapes with contrasting spatial and temporal dynamics: A framework for algorithm selection. *Remote Sens. Environ.* **2013**, *133*, 193–209, doi:10.1016/j.rse.2013.02.007.
22. Salomonson, V.V.; Guenther, B.; Masuoka, E. A summary of the status of the EOS Terra mission Moderate Resolution Imaging Spectroradiometer (MODIS) and attendant data product development after one year of on-orbit performance. In Proceedings of the Scanning the Present and Resolving the Future, Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217), Sydney, Australia, 9–13 July 2001; Volume 1193, pp. 1197–1199.
23. Alparone, L.; Wald, L.; Chanussot, J.; Thomas, C.; Gamba, P.; Bruce, L. Comparison of Pansharpening Algorithms: Outcome of the 2006 GRS-S Data Fusion Contest. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3012–3021, doi:10.1109/TGRS.2007.904923.
24. Zhang, Y.; De Backer, S.; Scheunders, P. Noise-Resistant Wavelet-Based Bayesian Fusion of Multispectral and Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 3834–3843.
25. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801.
26. Dan, L.; Hao, M.; Zhang, J.Q.; Bo, H.; Lu, Q. A universal hypercomplex color image quality index. In Proceedings of the IEEE Instrumentation & Measurement Technology Conference, Graz, Austria, 13–16 May 2012.