

## Time series remote sensing image classification framework using combination of deep learning and multiple classifiers system

Peng Dou <sup>a,b</sup>, Huanfeng Shen <sup>b</sup>, Zhiwei Li <sup>b,\*</sup>, Xiaobin Guan <sup>b</sup>

<sup>a</sup> Key Laboratory of Remote Sensing of Gansu Province, Heihe Remote Sensing Experimental Research Station, Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou, China

<sup>b</sup> School of Resource and Environmental Sciences, Wuhan University, China



### ARTICLE INFO

#### Keywords:

Time series image classification  
Remote sensing image classification  
Ensemble learning  
Deep learning  
Normalised differential index

### ABSTRACT

Recently, time series image (TSI) has been reported to be an effective resource to mapping fine land use/land cover (LULC), and deep learning, in particular, has been gaining growing attention in this field. However, deep learning methods using single classifier need further improvement for accurate TSI classification owing to the 1D temporal properties and insufficient dense time series of the remote sensing images. To overcome such disadvantages, we proposed an innovative approach involving construction of TSI and combination of deep learning and multiple classifiers system (MCS). Firstly, we used a normalised difference index (NDI) to establish an NDIs-based TSI and then designed a framework consisting of a deep learning-based feature extractor and multiple classifiers system (MCS) based classification model to classify the TSI. With the new approach, our experiments were conducted on Landsat images located in two counties, Sutter and Kings in California, United States. The experimental results indicate that our proposed method achieves great progress on accuracy improvement and LULC mapping, outperforming classifications using comparative deep learning and non-deep learning methods.

### 1. Introduction

With the continuous expansion of human intervention on Earth, land use/land cover (LULC) has become the main shaper of the ecological environment. Using remote sensing (RS) technology to dynamically monitor LULC, quantitatively extracted information regarding LULC change has become important research content on climate change and maintenance of energy balance of Earth system (Liu et al., 2018b; Yu et al., 2018). Therefore, high accuracy in LULC extraction becomes the key in achieving related goals.

With single-temporal RS images as object, assigning a class label to each pixel with machine learning has been the main approach to produce LULC for the past decades. Many classification algorithms, including support vector machine (SVM), maximum likelihood (ML), naive Bayesian (NB), decision tree (DT), K-nearest neighbour (KNN), multi-layer perceptron (MLP) and deep belief network (DBN), have been developed and applied to LULC production (Dou and Chen, 2017a; Maxwell et al., 2018; Du et al., 2020). However, single-temporal RS image only provides the instantaneous spectrum of the land surface, and the features that can be used for classification is infrequent, resulting in very few recognitions of LULC categories (Gómez, et al., 2016;

Hakkenberg et al., 2018). Achieving high classification accuracy using single-temporal RS image is difficult especially for the categories of different crops and other vegetations (Zhong et al., 2019; Tsai et al., 2018). In addition, the classification results of single-temporal image are affected by factors such as seasons and weather, and thus cannot accurately reflect the steady state of the LULC. Yearly analysis is difficult to adapt in complex studies on LULC changes (Flamary et al., 2015; Lyu et al., 2018; Xie et al., 2018).

Using multi-temporal RS images solves the problems that cannot be handled with single-temporal images. Multi-temporal RS images possess richer spatial features as well as different temporal profiles and can meet requirements of more complex tasks and obtain more stable LULC than single temporal images (Lyu et al., 2018; Xie et al., 2018; Fang et al., 2020; Ienco et al., 2017; Pelletier et al., 2019). In recent years, multi-temporal images have been extensively used to construct time series image (TSI) for LULC mapping and have achieved great success (Fang et al., 2020; Whelen and Siqueira, 2018). Classification methods can be roughly divided into two types according to the structure of TSI. The first is to stack multi-temporal images by time sequence and classify them with SVM, random forest (RF), and other classifiers (Hakkenberg et al., 2018; Whelen and Siqueira, 2018; Gong et al., 2019). This method

\* Corresponding author.

E-mail addresses: [00032042@whu.edu.cn](mailto:00032042@whu.edu.cn) (P. Dou), [shenhf@whu.edu.cn](mailto:shenhf@whu.edu.cn) (H. Shen), [lizw@whu.edu.cn](mailto:lizw@whu.edu.cn) (Z. Li), [guanxb@whu.edu.cn](mailto:guanxb@whu.edu.cn) (X. Guan).

performs well in producing annual LULC, such as finer resolution observation and monitoring of global land cover (Gong et al., 2019). However, this approach does not model temporal correlations, and it utilises features independently from one another, ignoring temporal dependency (Ienco et al., 2017; Pelletier et al., 2019; Belgiu and Csillik, 2018). In addition, the stacked images tend to increase redundancy and curse of dimensionality as the density of time series increases, negatively affecting improvement of classification accuracy (Belgiu and Csillik, 2018). The second type is to derive new images from original RS images with specific indexes, such as normalised difference vegetation index (NDVI) and soil-adjusted vegetation index (SAVI), to construct TSI (Zhong et al., 2019; Shao et al., 2016; Diek et al., 2017). With this method, time sequential curve of different objects can be formed, and crops and other vegetations are classified with high accuracy. However, the number of time series strictly affects this approach. If the number is too small, then the temporal features will exert little effect on classification performance. In addition, manual feature engineering based on human experience and prior knowledge is needed in this method, which then increases the complexity of processing and computation (Zhong et al., 2019; Pelletier et al., 2019).

Deep learning provides sufficient model complexity to learn feature representations from data in an end-to-end regime without human intervention (Zhu et al., 2017; Yuan et al., 2020). Deep learning is considered a breakthrough technology in machine learning, and it performs extremely well in object detection, semantic segmentation, image recognition, etc. (Zhu et al., 2017; Liu and Shi, 2020; Wang et al., 2020; Dou et al. 2021). Owing to its flexibility in feature representation and automation by expert-free learning, deep learning has been successfully applied in the research field of RS. In particular, deep learning has gained interest for handling temporal dimensions in time series classification. Some methods, including convolutional neural network (CNN)-based models, such as temporal CNN (TCNN), sequential data-based models, such as deep recurrent neural networks and bidirectional long short-term memory (Bi-LSTM) have achieved great success in improving LULC mapping (Zhong et al., 2019; Pelletier et al., 2019; Wang et al., 2019; Ho Tong Minh et al., 2018; Ienco et al., 2017). As the temporal dependencies of TSI are long and complex, how to model the deep learning architecture to analyse RS time series remains an open challenge (Wang et al., 2019). CNN-based methods are reported to be the leading models to recognise patterns in the 2D domain, and they also achieve high classification accuracy in the 1D domain, including spectral and time series (Zhong et al., 2019; Liu and Shi, 2020; Kussul et al., 2017; Cheng et al., 2018). To achieve better performance, the architecture of the CNN models need to be well designed through many tests, which then make the models bloated and complex. To classify temporal dependencies using deep learning architecture, the lower layers usually capture small-scale temporal variations, whereas the upper layers focus on overall patterns (Zhong et al., 2019; Cheng et al., 2018). However, when the time series is short or the temporal density is sparse, deep architecture exerts little effect on classification improvements.

Generally, a deep learning architecture consists of feature learning and classification (Li et al., 2017). At present, TSI classification with deep learning is focused on extracting representative features from temporal profiles (Zhong et al., 2019; Lyu et al., 2018). Considering the existing cases of RS classification, designing a reasonable classifier also plays an important role in improving accuracy (Xiao et al., 2018; Ebrahim et al., 2020). Thus, based on the features learned by deep learning architecture, some efficient approaches are applied for classification improvements, including using advanced neural networks and improving classifiers (Yokoya et al., 2018; Dong et al., 2020). However, these methods learn via a stochastic training algorithm and are sensitive to the specifics of the training data, and thus easily result in high variance and over-fitting when trying to develop a final model to use for classifications (Brownlee, 2018; Tao, 2018; Chen et al., 2017). A successful approach to reducing the variance is to train multiple classifiers instead of a single classifier, and this is called multiple classifiers system

(MCS). MCS not only reduces the variance of classification but also can result in better performance than any single classifiers, and, at the same time, it shows an amazing effectiveness and robustness to over-fitting (Brownlee, 2018). With these benefits, MCS has achieved great success in the field of RS classification including TSI classification (Chen et al., 2017; Man et al., 2017; Dai et al., 2019). Yet, to the best of our knowledge, research combining deep learning and MCS to improve performance of TSI classification is worth studying thoroughly.

Temporal property construction and high-efficiency classification method are two key factors to improve accuracy of TSI classification. In this study, we started from these two aspects and propose a TSI classification framework by combining deep learning and MCS to further improve the extraction of LULC from RS images. First, we employed a normalised index to establish time-series features from the original RS image. Then, we used deep learning architecture to build a feature extractor to get different features from the temporal properties. Based on the features, multiple classifiers were trained and their outputs were integrated as a more accurate output. Finally, with the proposed new approach, finer LULC was extracted from the multi-temporal RS images. Following the introduction in Section 1, Section 2 describes the study area and materials. Section 3 presents the TSI construction, TSI classification framework, and diversity measurement. Section 4 then exhibits the experimental results. Section 5 discusses the temporal features, classification accuracy improvements, and efficiency. Finally, Section 6 concludes the paper.

## 2. Study area and materials

### 2.1. Study area

The study area includes two counties, Sutter and Kings in California, United States. Sutter is located along the Sacramento River in Sacramento Valley with a total area of 1570 km<sup>2</sup>. It is home to Sutter Buttes known as the world's smallest mountain range. Eighty-eight percent of its area is prime farmland and grazing land. Kings is located in San Joaquin Valley, a rich agricultural region in south of California, and it has a total area of 3610 km<sup>2</sup>. We chose these two counties as study area mainly because most of these areas are covered by different types of crops, which are suitable for LULC classification using time series RS images.

### 2.2. Data description

#### 2.2.1. Land use survey data

Since 1950, California Department of Water Resources (CDWR) have conducted more than 250 land use surveys on all or parts of California's 58 counties. The department surveyed land use frequently and efficiently using satellite imagery, high-elevation digital imagery, local sources of data and RS in conjunction with field surveys and provided maps with high-quality land use attributes. In this study, the survey for Sutter and Kings in 2014 was download from the website (<https://gis.water.ca.gov/app/CADWRLandUseViewer>), and a label was created for each observed combination of attributes. A total of 267 unique labels are present in the shapefile of the survey. Some rare labels are only applicable to a few polygons, so detailed labels must be merged into classes of interest. In our experiments, the shapefile of the survey was transformed into raster with a resolution of 30 m, which matched the resolution of Landsat image. Then, the classes in the raster file whose pixel count was less than 1000 were aggregated into a general class called other. After such processing, the new survey data had 20 classes (Table 1). Most of them were crop land use types, and the rest were no-crop land use such as urban, managed wetland and other vegetations (see Fig. 1). Under this classification scheme, we randomly divided the polygons of the ground truth in two parts, in which the first 50% of the polygons were used for training and 50% were used for testing. The number of pixels for training and testing are shown in Table 1.

**Table 1**  
Classes used for classification.

Class label	Description	Training set		Testing	
		Sutter	Kings	Sutter	Kings
RI	Rice	203,081	0	217,524	0
SU	Sunflowers	24,530	0	18,024	0
CSS	Corn, Sorghum and Sudan	9486	162,568	11,581	164,171
AAM	Alfalfa and Alfalfa Mixtures	14,122	64,688	12,994	65,037
PN	Peaches/Nectarines	16,422	134,131	16,812	131,530
TO	Tomatoes	29,669	80,860	18,563	103,920
AL	Almonds	13,979	10,766	18,208	10,996
SA	Safflower	3633	72,278	1459	68,458
BD	Beans (Dry)	16,030	47,337	19,100	54,745
ID	Idle	95,403	39,124	101,186	47,086
MP	Mixed Pasture	7311	1430	6838	2450
PPA	Plums, Prunes and Apricots	40,721	286,506	35,236	309,840
WA	Walnuts	70,948	3343	77,280	3554
WH	Wheat	9698	7945	7800	6396
YP	Young Perennials	7936	40,237	8814	38,783
MW	Managed Wetland	26,341	66,741	7608	66,576
UR	Urban	28,353	1330	23,265	1043
CO	Cotton	0	28,152	0	20,345
PI	Pistachios	0	60,729	0	14,713
OT	Other	19,205	80,272	18,319	77,118

### 2.2.2. Landsat imagery

Images of Landsat 7 Enhanced Thematic Mapper (ETM+) and Landsat 8 Operational Land Imager (OLI) were used as the main input data of the experiment. For the time series RS image classification, data in the same domain play an important role to establish a temporal curve, so radiometric calibration on the RS image should be performed to achieve surface reflectance. In the study, we used Google Earth Engine to produce multi-temporal surface reflectance data derived from ETM+ and OLI. To maintain data consistency, only spectral bands 1–5 and 7 of ETM+ and bands 2–7 of OLI were pertinent to the process. The study areas were covered in the footprints of Path 44 and Row 33 for Sutter and Path 42 and Row 35 for Kings. For 2014, 19 ETM+ and 17 OLI

images were acquired, and a total of 36 temporal reflectance images were prepared.

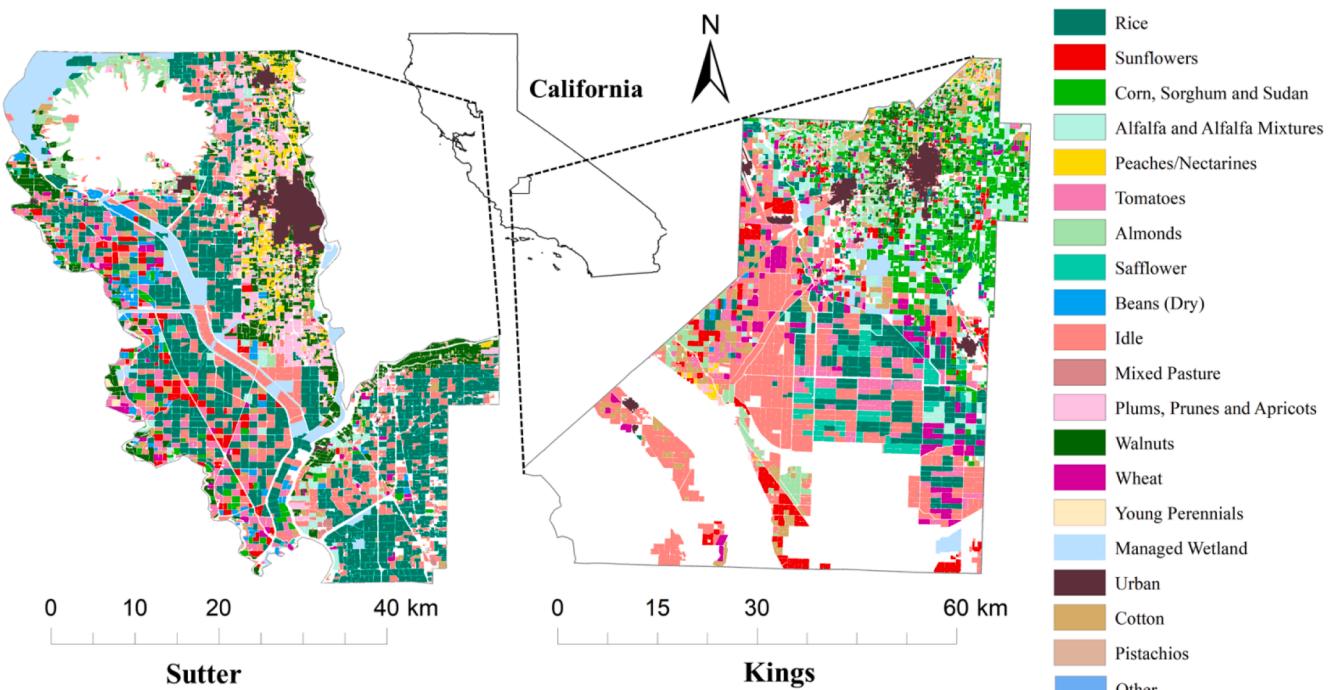
## 3. Methods

### 3.1. Deep learning architectures

In this research, one-dimensional deep learning architectures, namely, Zhong's 1D CNN (Z-1DCNN), TCNN, LSTM and 1D simple CNN (1DSCNN) are used as important base of our method for feature learning and classification.

Z-1DCNN is based on 1D convolutional layers and utilises 1D filters to capture the temporal pattern or shape of inputs. It is a multi-layer stacked model, with the lower layers focusing on local features and the upper layers extensively summarising the general features (Zhong et al., 2019). TCNN is another 1D CNN-based architecture applied in the temporal domain for TSI classification. In TCNN, three convolutional filters are consecutively applied, followed by one fully connected layer and finally connected to the Softmax layer that provides the predicting class distribution (Pelletier et al., 2019). LSTM is a network in which each LSTM unit remembers values over arbitrary time intervals, long or short. Using LSTM can improve the efficiency of depicting temporal patterns at various frequencies and learn desirable features with different lengths. The three architectures above are effective strategies to represent sequential data, and they are reported to achieve high accuracy in TSI classification (Zhong et al., 2019; Pelletier et al., 2019). According to the studies by Zhong et al. (2019) and Pelletier et al. (2019), the optimal architecture of the three models are illustrated in Fig. 2.

The 1D SCNN is our self-designed architecture which consists of a single convolution layer paired with a pooling layer (as shown in Fig. 2). The convolution layer accepts the 1D temporal profile of the input data and is fed by kernels that have 1D array of weights and a bias and scan across a temporal profile curve to capture different features. The pooling layer can be thought of as a temporal down-sampling of the convolutional feature map. It reduces the size of the feature map and preserves the main features captured by the convolution layer. Here, a max pooling operation is utilised (Dou and Zeng, 2020). Compared with the



**Fig. 1.** Study areas and land use survey.

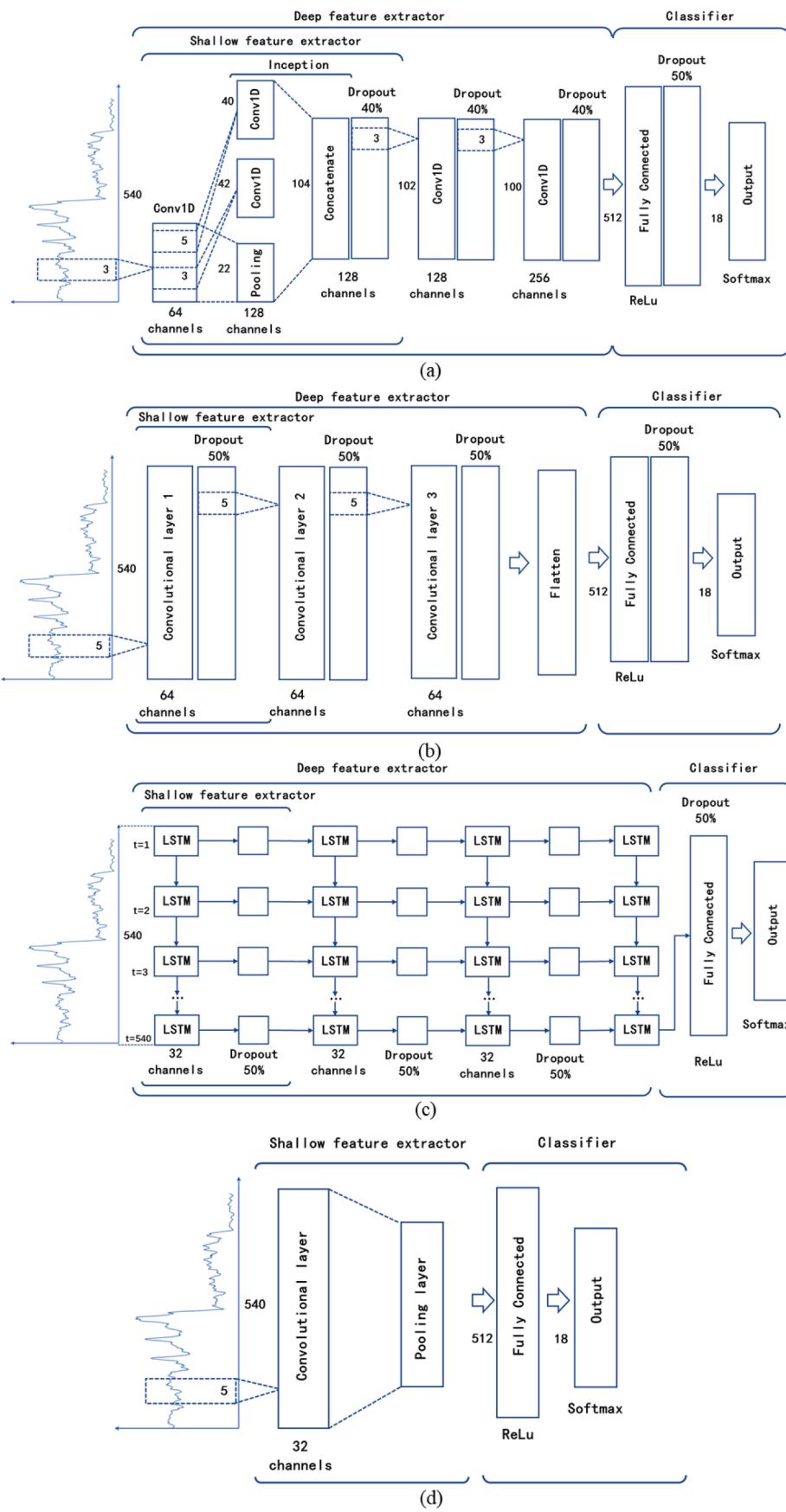


Fig. 2. Architecture of (a) Z-1DCNN, (b) TCNN, (c) LSTM and (d) 1D SCNN.

other three architectures, the 1D SCNN focuses more on learning local features at shallower level.

Given the versatility of specialized architectures, no standard is universal to search for the optimal combination of hyper-parameters and various types of layers. Thus, we followed Zhong et al. (2019) and Pelletier et al. (2019) to optimise the models of TCNN, LSTM and Z-1DCNN. The optimal convolutional filter widths and number of channels for each convolutional layer can be seen in Fig. 2 (a), (b) and (c). For the 1D-SCNN, convolutional filter widths of 2, 4, 6 and 8 were tested with 32 and 64 channels in the convolutional layer. Finally, the 1D-SCNN model was optimised, as shown in Fig. 4(d).

### 3.2. Time series imagery construction

A TSI construction strategy is a prerequisite to improve the classification. Many study cases use vegetation index (VI) time series for LULC classification (Zhong et al., 2019; Shao et al., 2016; Diek et al., 2017). However, due to the strict effect of the number of temporal features, a higher time resolution of image is sometimes required to achieve better results. In addition, the VIs are usually generated by feature enhancement with band calculation, and except for some vegetations, the feature enhancement of other objects are insignificant.

Normalised differential index (NDI) has the advantage that it can enhance different classes by making full use of the spectral bands (Dou and Zeng, 2020). With this benefit, we firstly used NDI to analyse each temporal image in the research. The NDI is defined as (1),

$$NDI_{ij} = \frac{band_i - band_j}{band_i + band_j} \quad (1)$$

where  $i = 1, 2, 3, \dots, n$ ;  $j = 2, \dots, n$ ;  $n$  is the band count of the current temporal image. To avoid redundancy, we set up a filter condition that when  $i > j$  the NDIs become pertinent to the final results. After such processing, images with  $C_6^2 = 15$  channels were generated. For each channel, the corresponding NDIs in each image were then stacked on the temporal domain to generate a TSI. Finally, 15 TSIs were produced as shown in Fig. 3.

The cloud and gaps in the ETM+ and OLI image were not processed in our study. In the TSIs, the existing noises would easily interfere the time sequence, resulting in many uncertainties in the classification. The noises must be depressed with time sequence filtering. So we used a

moving weighted harmonic analysis method (Yang et al., 2015) to reconstruct the temporal profiles of the 15 TSIs. Finally, the 15 TSIs were stacked one by one to generate an image cube which has  $36 \times 15$  bands, and they were called NDIs-based TSI.

### 3.3. TSI classification framework

Fig. 4 illustrates the TSI classification framework proposed in our study. The framework includes two modules: feature extractor and classification model (CM). The feature extractor is a deep learning architecture, which can learn to extract more representation features from the temporal curves and outputs the results as a vector. The vector is then divided into several subsets, and each subset is called a segment. The CM consists of multiple classification networks which correspond to the segments. Each network, called as base classifier, can predict the probabilities of different classes for the input data. All predictions are then ensembled so that the output of the CM is more accurate. Finally, with use of the framework, excellent performance of LULC extraction can be expected.

In order to ensure that our experiment would be performed with high efficiency, we used TensorFlow's application programming interface (API) for programming. TensorFlow is a very famous open-source software library that was developed by the Google Brain Team for machine learning applications. It provides sophisticated DL approaches, including the necessary support for the TSI classification framework, and is compatible with graphics processing unit (GPU) for speeding up the operation.

#### 3.3.1. Feature extractor

In this study, the models of Z-1DCNN, LSTM, TCNN and 1DSCNN were trained using the training set and then the classification part of each model, which contains the latest fully connected (FC) layer and the Softmax layer, was removed and only maintained the front layers as feature extractor of the TSI classification framework. For the Z-1DCNN, LSTM and TCNN, the feature extractors are called deep feature extractors. The deep feature extractor learns features from the temporal profiles at high level (Zhong et al., 2019; Ienco et al., 2017; Pelletier et al., 2019; Ebrahim et al., 2020). Whether shallower features could be used in our proposed framework is a question worth discussing. Thus, we removed some upper layers from the deep feature extractor to get features at lower level and called the new feature extractor as shallow

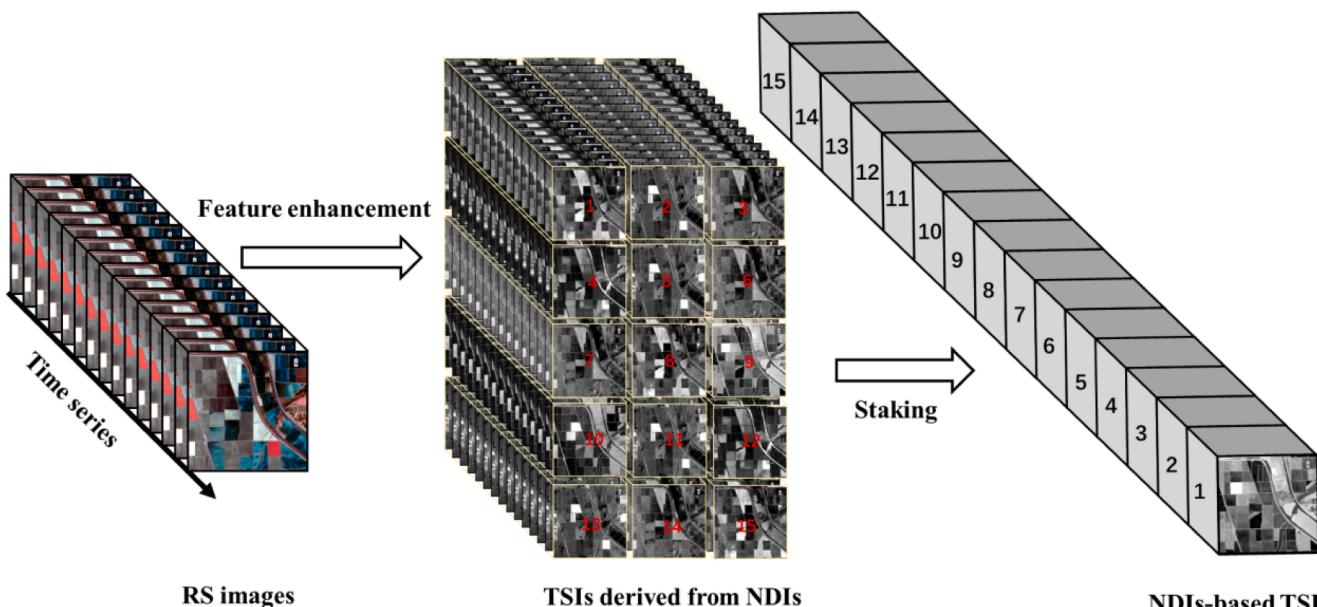
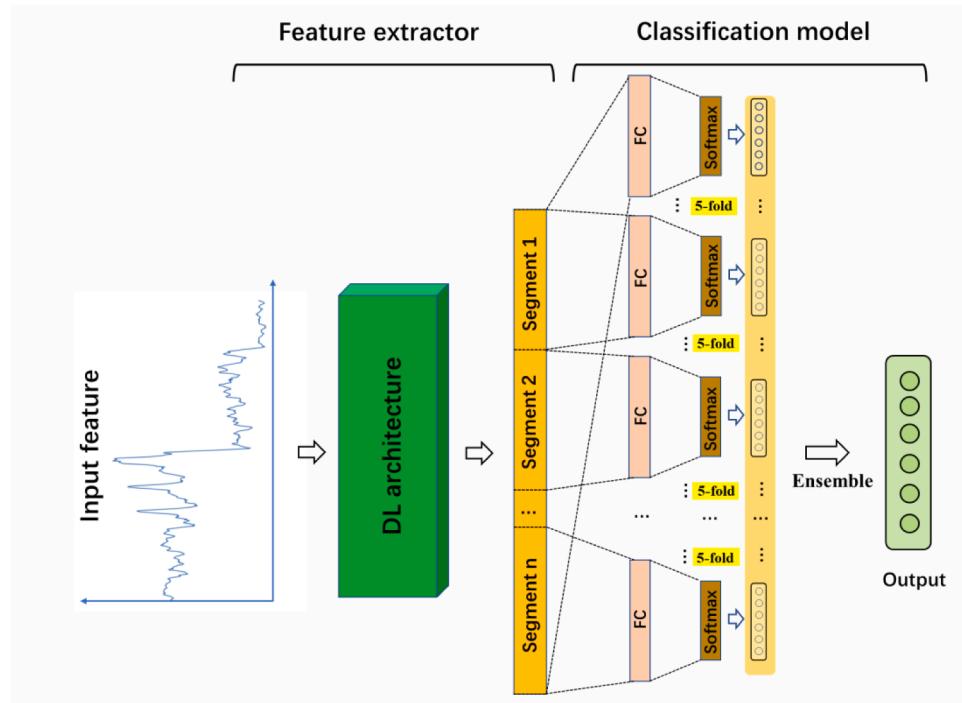


Fig. 3. Construction of NDIs-based TSI in this study.



**Fig. 4.** Construction of TSI classification framework.

feature extractor. The structure of the deep feature extractor and shallow feature extractor based on Z-1DCNN, LSTM and TCNN are illustrated in Fig. 2(a), (b) and (c), respectively. 1D SCNN has shallow architectures, so its feature extractor is a shallow feature extractor, as shown in Fig. 2 (d). Finally, we established different feature extractors, and each of them was used separately to build the TSI classification framework. The features captured by the extractor were then flattened as the output of the feature extractor module. As the feature extractors based on Z-1DCNN, LSTM and TCNN were divided into deep and shallow feature extractors, to easily distinguish the frameworks established by different extractors, we respectively used Z-1DCNN-, LSTM- and TCNN-based framework (deep) to represent the frameworks with deep feature extractors and Z-1DCNN-, LSTM- and TCNN-based framework (shallow) to represent the frameworks with shallow feature extractors.

### 3.3.2. Classification model

Without using any dense layer, the features outputted by the feature extractor module would be very high dimensional and would present a challenge for classification using a single classifier. When the dimension of the feature is much larger than the number of training samples, the classification network can very easily be over-fitting and inaccurate due to the Hughes phenomenon (Gómez, et al., 2016; Ma et al., 2013). The most common way to solve this problem is to establish fully connected layers hierarchically to extract more abstract and low-dimensional features as the input of the classifier. However, this procedure is complex and needs much testing for optimisation, and if not well handled, the advantages of the features from the feature extractor module will be lost.

To fully use the features from the feature extractor module, we designed a MCS. The construction of the system is illustrated in Fig. 4. Firstly, all the features were utilised to train a classification network, and then the feature sequence was divided into  $n$  segments. For each segment, 5-fold method was used to train classification networks as the base classifiers. Overall,  $n \times 5 + 1$  networks were obtained to form a set of base classifiers. Here, the 5-fold method is an approach which can produce various classifiers. It means to partition the sub train set into five folds and take each fold as training set to train classifiers. In this

study, MLP, which only has one hidden layer, was designed as the architecture of the base classifiers in the CM module. For each segment, the base classifiers were fully connected to the neurons which represent the features of the current segment. The MLPs were trained using the RMSprop optimizer (Kumar Reddy et al., 2018) and categorical-cross-entropy loss function. The purpose of our experiments is to evaluate how the proposed framework improves classification accuracy by using different deep learning architectures. To create a relatively fair condition for different feature extractors, we used some default values for the hyperparameters. For each MLP, the learning rate was set to 0.001, and the batch size is 32. The number of neurons in the hidden layer was set to 500, and epoch was initialised to 100.

### 3.4. Diversity measurement

In this study, the main purpose of segmenting the output features of the feature extractor module and using the 5-fold methods is to train diverse base classifiers in the CM layer. The diversity of the base classifiers is the core of MCS because only if the classification results from the base classifiers are not identical or similar can the accuracy be improved by combining (Dou and Chen, 2017b; Dou et al., 2017). Therefore, incorporating diversity of the base classifiers in the CM module is important to measure the performance of the ensemble strategy. Entropy is reported an effective measurement of base classifiers diversified in MCS (Kuncheva and Whitaker, 2003; Dou et al., 2017), so we used it as calculated by (2).

$$E = \frac{1}{N} \sum_{i=1}^N \frac{1}{L - \text{ceil}(L/2)} \min\{h(x_i), L - h(x_i)\} \quad (2)$$

where  $N$  is the total number of samples, and  $L$  is the number of classifiers.  $\text{ceil}(\cdot)$  rounds a number to the nearest integer.  $h(x_i)$  is the number of samples correctly classified by classifier  $h$ . The range of this measure is  $[0,1]$ , in which 0 represents the lowest value diversity, and 1 represents the biggest diversity value.

## 4. Experimental results

### 4.1. Comparative methods

To evaluate the impact of TSI construction method proposed in this study on the improvement of classification accuracy, a popular TSI construction method was used for comparison. It's called NDVI-based TSI which is to produce a TSI formed by NDVIs derived from multi-temporal RS images. We also tried three non-deep learning classifiers for a comparative classification: gradient boosting (GB), random forest (RF) and generative extreme learning machines (GenELM). GenELM is a classifier based on the extreme learning machine (ELM). An ELM is a single layer feed-forward network with random hidden layer components and ordinary linear least squares fitting of the hidden (Cao et al., 2019). GB is an additive model in a forward stage-wise fashion, and it allows for the optimisation of arbitrary differentiable loss functions. In each stage, the number of tree based classifiers fits the negative gradient of the multinomial deviance loss function (Liu et al., 2018a). A RF is a classifier that fits a number of DT classifiers on various sub-samples of the dataset and uses averaging to improve predictive accuracy and control over-fitting (Dou and Zeng, 2020; Chen et al., 2017). Except the GenELM, the GB and RF are families of MCS which have been extensively used in RS applications. All of them achieve great success in more complex RS classification tasks, and their results can be used as a reference which represents the performance of the current popular non-deep learning algorithms. The non-deep learning classifiers were realized by using scikit-learn library and their optimal parameters are described in Table 2.

### 4.2. Classification results

The accuracy improvement of TSI classification mainly depends on three aspects: TSI construction, feature extraction and classification methods. To this end, the design of our experiments considered verifying the contribution of the three aspects. In the experiment, NDVI-based TSI and NDI-based TSI were respectively used as input of TSI classification framework and the comparative classifiers. As mentioned in Section III, the TSI classification frameworks are modelled by feature extractors with deep and shallow strategies. To evaluate how the number of segments in feature extractor module affects the overall accuracy (OA) of the TSI classification framework, values of 1, 2, 4, 6, 8 and 10 were tested. Finally, based on this design, our experiment was conducted on the dataset of Sutter and Kings, and the accuracy of different classification methods are shown in Tables 3 and 4. The error distribution of TCNN-based framework (deep) which has highest OA is shown in Fig. 5.

When using NDVI-based TSI as input, the classification of the two datasets is a difficult task, because regardless of the classification method used, the classification accuracies are less than 76% and 73% respectively. Using NDVI-based TSI as the data input has not achieved higher classification accuracy in the experiment. This may be because the number of input features is small, and the features provided by the NDVIs are inefficient for improving classification accuracy. When using NDIs-based TSI as input data, no matter what classification method is used, the accuracy is significantly improved compared with the NDVI-

**Table 2**  
Descriptions of the main optimal parameters of the non-deep learning classifiers.

Classifier	Description of parameters
GenELM	The number of hidden layers: 5000.
RF	The number of trees in the forest: 100.
	The minimum number of samples required to split an internal node: 2.
	The minimum number of samples required to be at a leaf node: 1.
GB	The minimum number of samples required to split an internal node: 2.
	The minimum number of samples required to be at a leaf node: 1.
	Loss: exponential.
	Learning rate: 0.001.

**Table 3**

Classification accuracy achieved by various classifier methods and inputs on the test of Sutter dataset.

Classification methods	NDIs-based TSI		NDVI-based TSI	
	OA (%)	Kappa	OA (%)	Kappa
1D SCNN-based framework	83.01	0.800	73.19	0.686
TCNN-based framework (deep)	<b>83.51</b>	<b>0.805</b>	<b>75.69</b>	<b>0.715</b>
TCNN-based framework (shallow)	82.48	0.793	75.55	0.713
Z-1DCNN-based framework (deep)	79.65	0.761	73.44	0.690
Z-1DCNN-based framework (shallow)	81.33	0.776	74.45	0.701
LSTM-based framework (deep)	78.76	0.750	70.03	0.650
LSTM-based framework (shallow)	79.70	0.761	72.12	0.674
TCNN	78.81	0.752	69.04	0.641
Z-1DCNN	<b>76.22</b>	<b>0.721</b>	<b>70.03</b>	<b>0.650</b>
LSTM	75.16	0.709	68.94	0.638
GenELM	75.64	0.720	71.21	0.664
GB	74.28	0.699	66.93	0.615
RF	75.46	0.712	69.44	0.645

**Table 4**

Classification accuracy achieved by various classifier methods and inputs on the test of Kings dataset.

Classification methods	NDIs-based TSI		NDVI-based TSI	
	OA	Kappa	OA	Kappa
1D SCNN-based framework	77.18	0.742	68.61	0.647
TCNN-based framework (deep)	<b>79.97</b>	<b>0.772</b>	68.06	0.642
TCNN-based framework (shallow)	79.77	0.771	70.07	0.666
Z-1DCNN-based framework (deep)	75.09	0.718	70.39	0.666
Z-1DCNN-based framework (shallow)	77.82	0.748	<b>72.34</b>	<b>0.688</b>
LSTM-based framework (deep)	72.48	0.691	68.33	0.643
LSTM-based framework (shallow)	76.39	0.733	71.55	0.679
TCNN	75.00	0.718	63.95	0.598
Z-1DCNN	74.25	0.708	66.19	0.604
LSTM	73.06	0.697	62.82	0.571
GenELM	71.13	0.677	62.41	0.563
GB	71.95	0.685	60.12	0.538
RF	72.30	0.689	61.44	0.550

based TSI, and the OAs for Sutter and Kings datasets are above 78% and 76%, respectively. Using NDIs to build TSI has obvious advantages of using NDVI, because the NDIs-based TSI provides more abundant information for classification than NDVI-based TSI.

Taking the NDIs-based TSI as the input data, our proposed classification framework achieved obviously higher accuracy than the comparative methods. For the Sutter and Kings datasets, their highest accuracy can reach 83.51% and 79.97%. Compared with the TCNN method, which has the highest classification accuracy in the comparative methods, the OA was improved by 4.7% and 4.97%, respectively. This indicates that the theory of combining deep learning feature extractor and multi-classification network methods is very effective. Of course, the differences of the feature extractor affect the classification improvement. For example, as far as the deep structure based on deep learning feature extraction is concerned, some feature extractor with shallow structure designed in this study can obtain higher classification accuracy than deep structure. In the Sutter dataset, the classification accuracy of Z-1DCNN-based framework (deep) is only 79.65%, but after using the shallow structure, the classification accuracy of Z-1DCNN-based framework (shallow) reaches 81.33%. The TCNN-based framework has good performance on Sutter and Kings datasets, but its feature extractor with deep structure has higher accuracy (83.51% and 79.97%) than shallow structure (82.48% and 79.77%).

As using the NDVI-based TSI has obviously lower classification accuracy than using the NDI-based TSI, we only talk about the LULC mapping results using NDI-based TSI. To facilitate comparison in a limited paper space, the LULC maps produced by the TSI classification framework which has the highest accuracy in the experiment were compared with those by the original deep learning methods, including

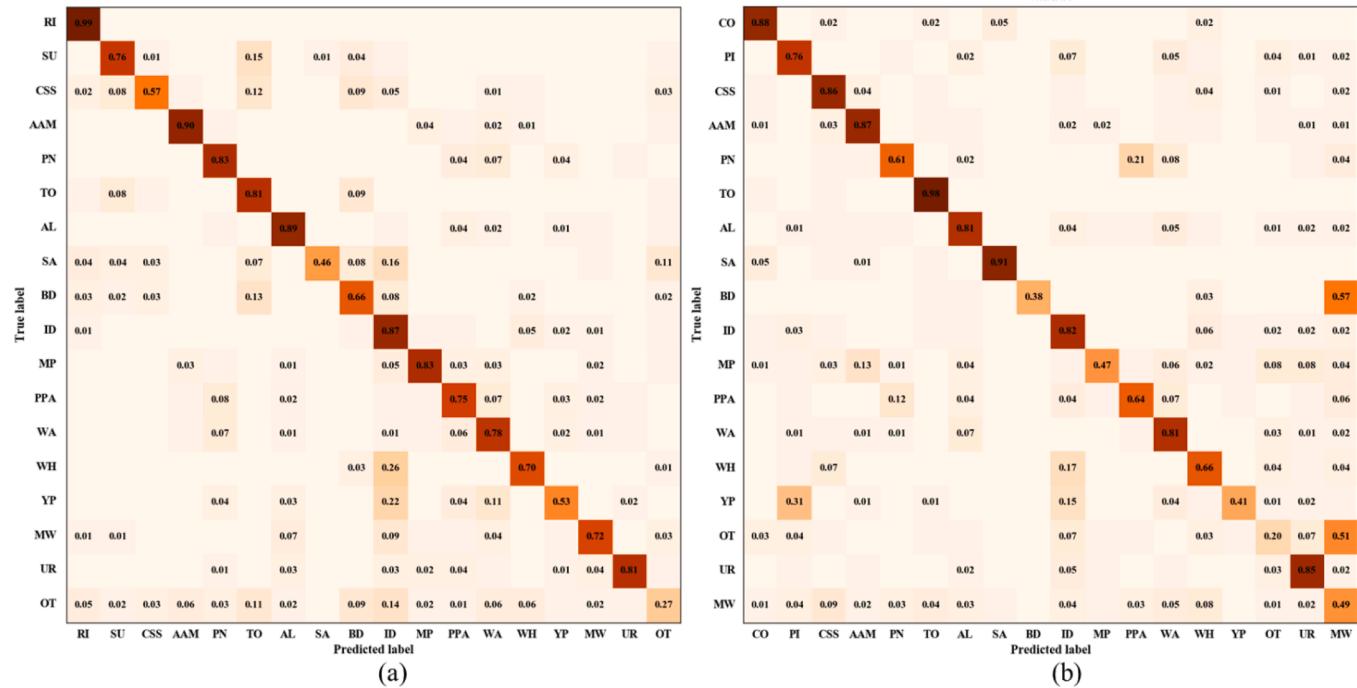


Fig. 5. Error distribution of the TCNN-based framework (deep) on (a) Sutter dataset and (b) Kings dataset.

TCNN, Z-1DCNN and LSTM. As shown in Figs. 6 and 7, the mapping result of our proposed TSI classification framework is better than that of the comparative deep learning methods. For the Sutter dataset, the maps

produced by the comparative methods have mutual mixing of different classes, and the mapping results are obviously affected by salt-and-pepper noises. When 1D SCNN-, TCNN-, Z-1DCNN-, and LSTM-based

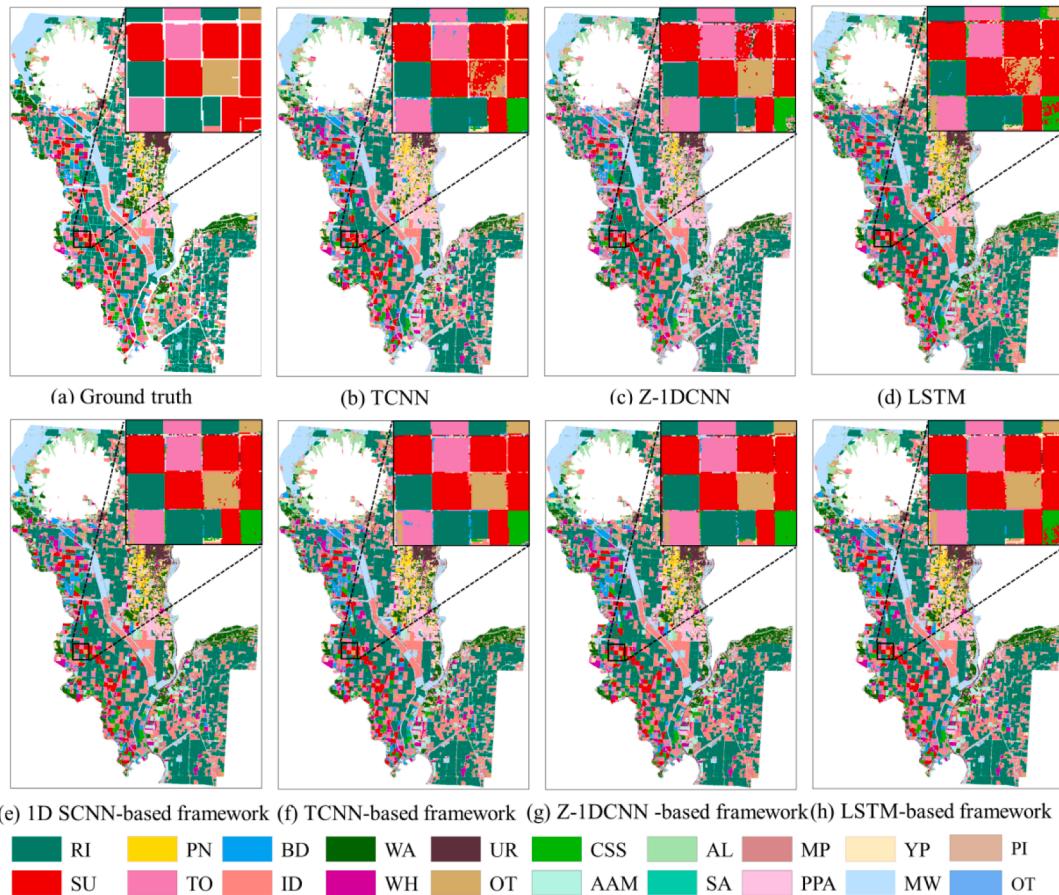


Fig. 6. LULC mapping of Sutter dataset with use of different classification methods.

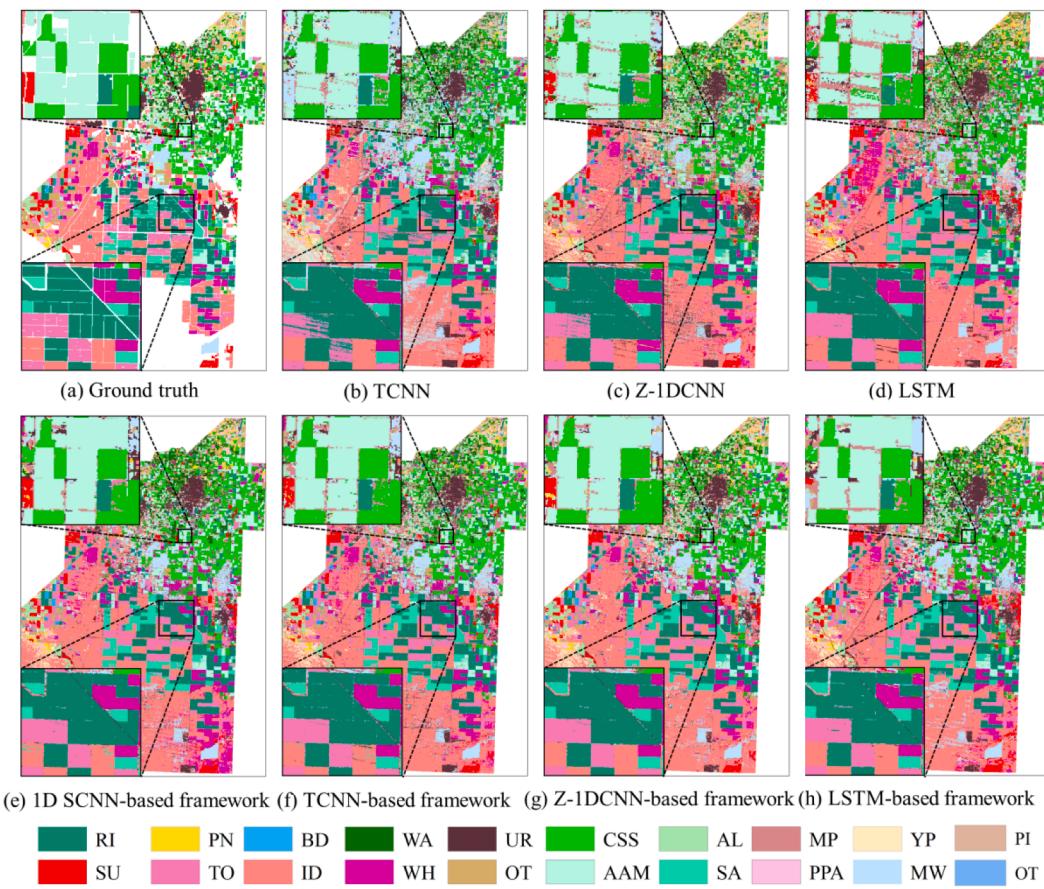


Fig. 7. LULC mapping of Kings dataset with use of different classification methods.

frameworks were utilised, the wrong classification and the effects of salt-and-pepper noises were controlled effectively, and the mapping results exhibit a higher degree of coincidence with the ground truth. This

feature also appeared on the Kings dataset. Owing to the effect of gaps in the ETM+ image, obvious banding traces are noted in the maps produced by the comparative methods. When the frameworks proposed in

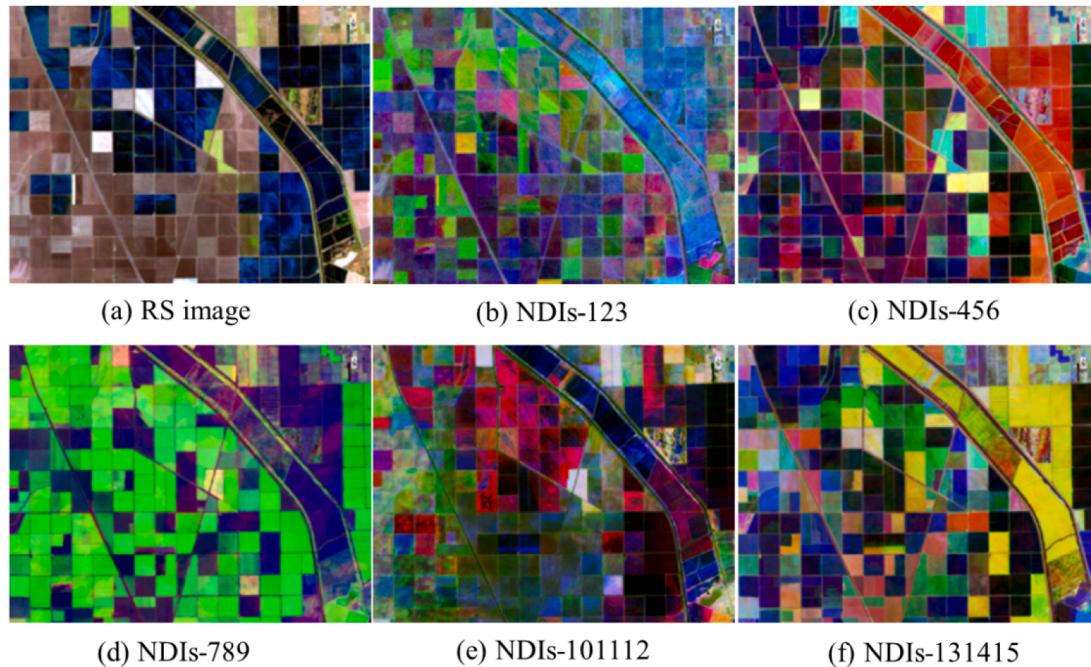


Fig. 8. Composite images assigned as RGB ordered by three different NDIs (NDIs-123 in Fig. 8 (b) means the composite image with the use of NDI channels 1, 2 and 3).

our research were utilised, these banding traces were eliminated effectively. Combined with the previous analysis of classification accuracy, the main reason the proposed frameworks achieve higher quality LULC mapping results than the comparative deep learning methods is that our frameworks fully utilised the advantages of deep learning feature extractor and multiple classifiers to improve the classifications through integration.

## 5. Discussion

### 5.1. Time series image analysis

NDIs have enhanced features using different combinations of spectral bands. Fig. 8 shows the composite image, which is assigned as RGB ordered by three different NDIs derived from the same RS image. To facilitate the illustration despite limitations in space, only five combinations of NDIs are shown in this figure (NDIs-123 in Fig. 8 (b) means the composite image with use of NDI channel 1, 2 and 3). Evidently, different objects have different colours under different combinations, thereby showing clear differences from other objects.

Fig. 9 shows the temporal feature curves of four different classes in the NDVI- and NDIs-based TSIs. The NDVI is a special NDI, it has some negative values might be due to the existing of cloud, gaps and irrigation water. This negative values might have little impact on the classification problem, so in our analysis, we are more concerned about the feature curves of the NDVI-based TSI. In Fig. 9 (b) and (d), the feature curves in the two filtered TSIs show continuous temporal properties with high separability and few redundancies. Compared with those of the NDVI-based TSI, the feature curves of the different classes in the NDIs-based TSI are longer and with more abundant features to separate classes from one another. These advantages describe the temporal properties under feature enhancement of the different NDIs.

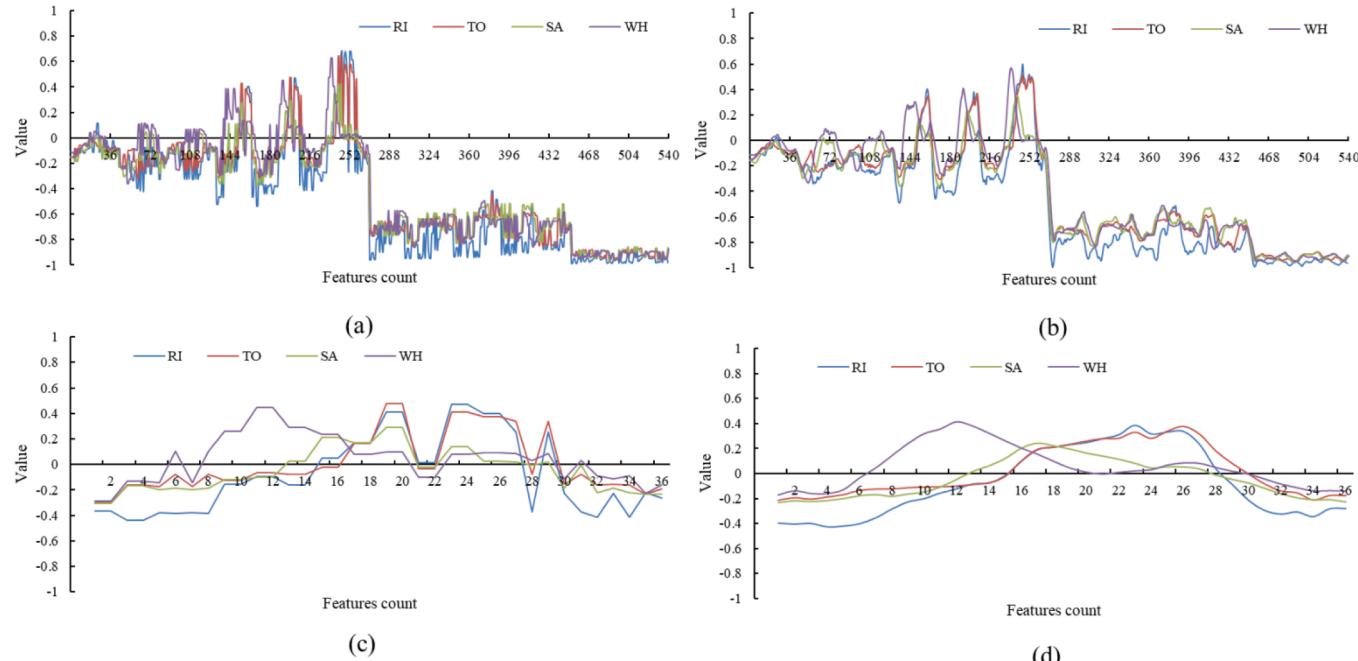
In summary, the NDIs-based TSI contains substantial enhanced information for various classes and includes temporal behaviour under the feature enhancement of NDIs. For classification, this information has become an important guarantee for improving accuracy. Thus, the NDIs-based TSI has superior classification improvement in our experiments.

### 5.2. Classification accuracy improvements

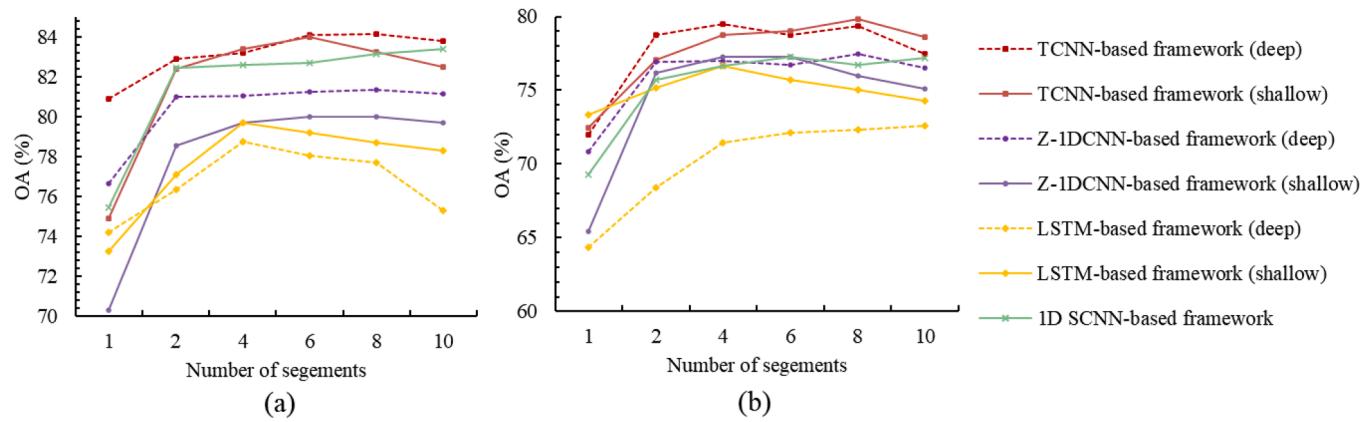
#### 5.2.1. Effects of feature segment count

In the TSI classification framework, the number of feature segments for the output of the feature extractor module has a certain impact on classification accuracy. Specifically, under different numbers of the feature segments, the accuracy of each classification method is shown in Fig. 10. When the number of feature segments is 1, the OAs are basically low for all the classification methods. This is mainly because in our TSI classification framework, the base classifiers in the CM module is very simple with low accuracy, and the count of base classifiers are not enough to be diverse for the ensemble to improve classification accuracy. The strategy of dividing features of the feature extractor module into several segments is an important approach to train various base classifiers. When the number of the segments increases, the OA of the entire framework increases as well. However, this phenomenon does not mean that the more feature segments, the higher the accuracy. The 1D SCNN-based framework is an exception with the highest OA in the case of 10 segments. The highest OA of other methods are distributed in the case of 4, 6 and 8 segments though, and in the case of 10 segments, their OA shows a downward trend. This situation is attributed to the fact that under a fixed dimension of the output feature in the feature extractor module of the framework, if the number of segments increases to an extent, the number of features in each segment will decrease (as shown in Table 5), resulting in low accuracy of the base classifiers in the CM module. That is, when the number of features in each segment reaches a certain degree, the improvement of OA for the final classification model will be slightly affected. So, in the TSI classification framework, the number of feature segments in the feature extractor module should be set with a modest value.

In our proposed framework, different feature extractors provided features with different dimensions and qualities. For a given feature extractor, the number of trainable parameters of base classifiers in the CM module would be different from other feature extractors. This aspect may be a potential reason that affects the improvement of the proposed framework's accuracy. However, it is difficult to say that the number of trainable parameters of the base classifiers completely determined the accuracy improvement. For the TCNN-based frameworks (deep and



**Fig. 9.** Temporal feature curves of RI, TO, SA and WH in the (a) original NDIs-based TSI, (b) filtered NDIs-based TSI, (c) original NDVI-based TSI, and (d) filtered NDVI-based TSI.



**Fig. 10.** Accuracy improvements of TSI classification framework under different numbers of the feature segments: (a) Sutter dataset and (b) Kings dataset.

**Table 5**  
The number of features in each segment.

Method	Feature segments count					
	1	2	4	6	8	10
1D SCNN-base framework	8640	4320	2160	1440	1080	864
TCNN-based framework (deep)	34,560	17,280	8640	5760	4320	3456
TCNN-based framework (shallow)	34,560	17,280	8640	5760	4320	3456
Z-1DCNN-based framework (deep)	13,312	6656	3328	2219	1664	1331
Z-1DCNN-based framework (shallow)	25,600	12,800	6400	4267	3200	2560
LSTM-based framework (deep)	17,280	8640	4320	2880	2160	1728
LSTM-based framework (shallow)	17,280	8640	4320	2880	2160	1728

shallow), their base classifiers have the same number of trainable parameters (Table 5) but with different OAs (Fig. 10). Evidently, the structure of the feature extractor plays an more important role in the accuracy improvement of the TSI classification framework.

### 5.2.2. Effects of base classifier count

The number of base classifiers in our proposed framework is increased depending on the increase in the number of segments. For each segment increase, five base classifiers are added into the ensemble classification. Fig. 11 reflects the accuracy of the base classifiers, and the ensemble classification accuracy changes as the number of base classifiers increase in the TSI classification framework. To facilitate analysis, the results of the framework under the segment case which has highest accuracy were selected for comparison. Given the performance of frameworks based on different methods on different datasets, the OA gradually increases with the increase of base classifiers, and then when the number of classifiers reaches a certain level, the OA will gradually show a saturation status. The classification framework does not suffer from any accuracy degradation due to the increase in the number of base classifiers, indicating that the system has strong robustness. For the 1D SCNN-based frameworks, the base classifiers at the later stage have lower accuracy than the classifiers in the early stage, and some of them even have an OA of less than 65%. However, regarding the ensemble classification, the OA basically is unaffected by the low-accuracy base classifiers.

Certainly, the classifiers have restrictions on the ensemble classification accuracy improvements, and they are mainly reflected in two aspects. The first one is that in the entire system, if all base classifiers have high accuracy, the ensemble classification accuracy must be higher. Another is that if some lower-accuracy base classifiers

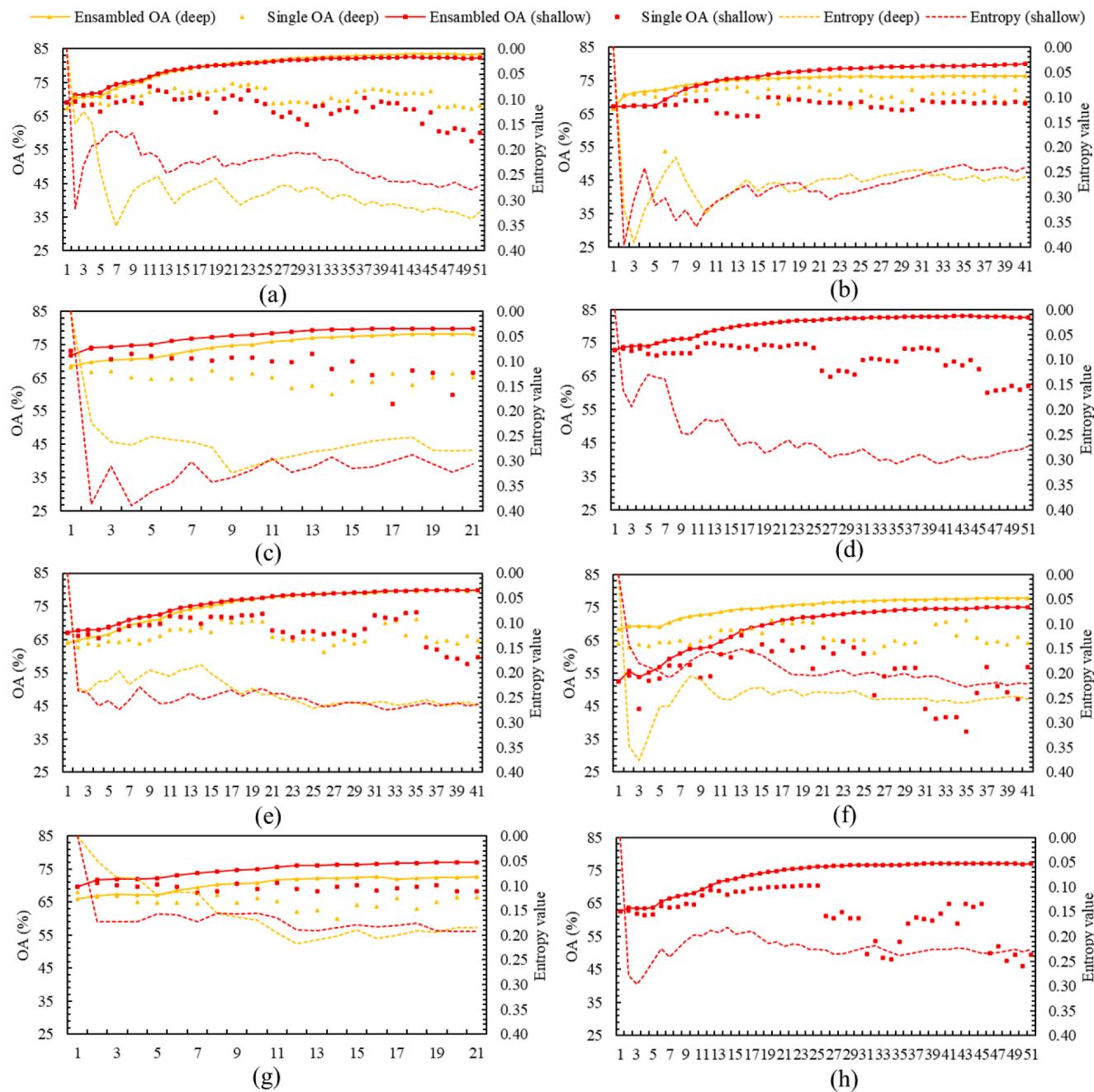
continuously appear in the system, the trend of ensemble OA increase will slow down and be inconspicuous. For example, on the Sutter dataset, after the base classifiers of the Z-1DCNN-based framework reaches 40, the accuracy of later base classifiers dropped to a large extent. The OA of ensemble classification stopped increasing after the 40th base classifier. The LSTM-based framework shows a very different situation from Z-1DCNN-based framework. The very low-accuracy base classifiers continuously appear in the framework, thus the ensemble OA gradually increases with the increase of the base classifiers, as shown in Fig. 10 (c) and (g).

### 5.2.3. Effects of base classifier diversity

Fig. 11 also illustrates the influence of the diversity of the base classifier on the accuracy of ensemble classification under features extracted by deep and shallow architectures. For the same deep learning approach, feature extractor which has higher accuracy during the ensemble classification of the TSI classification basically exhibits higher entropy values. That is to say, for the design of the feature extractor of the proposed framework, base classifier diversity should be carefully considered to improve classification accuracy. In this paper, diversity of the base classifier is affected by the depths of the feature extractor. For the Z-1DCNN- and LSTM-based frameworks the diversity of the base classifier generated by the classification using the shallow architecture is obviously greater than that using the deep architecture, but for the TCNN- based frameworks, it's the other way round. Under the same deep learning approach, features extracted by deep or shallow extractor are very different in qualities and dimensions, so for the base classifiers which have the same structures, these differences make them exhibits diverse at different levels. Diversity of base classifiers is a vital requirement for ensemble classification (Dou and Chen, 2017b), as well as for our proposed framework. Thus in this study, encouraging the use of the feature extractor whose output could train diverse base classifiers for improve accuracy of the TSI classification framework.

### 5.2.4. Classification improvements at per-class level

The analysis above reveals that for the same deep learning method, the accuracy of the TSI framework greater than the accuracy of original method. The classification accuracy of the three different architectures at per-class level is shown in Fig. 12. The classification accuracy of individual class basically conforms to the law above. This phenomenon shows that the TSI classification frameworks proposed in this paper improve the accuracy at overall level based on the improvements at per-class level in most cases, and there is no case where the high-accuracy classification results are concentrated in a certain class. However, there may be some exceptions for a few categories. For example, when the TCNN-based methods worked on Kintgs dataset (as shown in Fig. 12 (d)), the accuracy of MP obtained by original TCNN is greater than that of the two TCNN-based TSI classification frameworks (deep and



**Fig. 11.** Change of the classification accuracy and diversity as the base classifiers increase: (a), (b), (c) and (d) are results of TCNN-, Z-1DCNN-, LSTM- and 1D SCNN-based frameworks, respectively, on the Sutter dataset; (e), (f), (g) and (h) are results of TCNN-, Z-1DCNN-, LSTM- and 1D SCNN-based frameworks, respectively, on the Kings dataset.

shallow). This may be because the classification accuracy of base classifiers in the two frameworks is too low, and the complementarity between them is poor, making it difficult to obtain more accurate results than the original TCNN with simple voting method to ensemble.

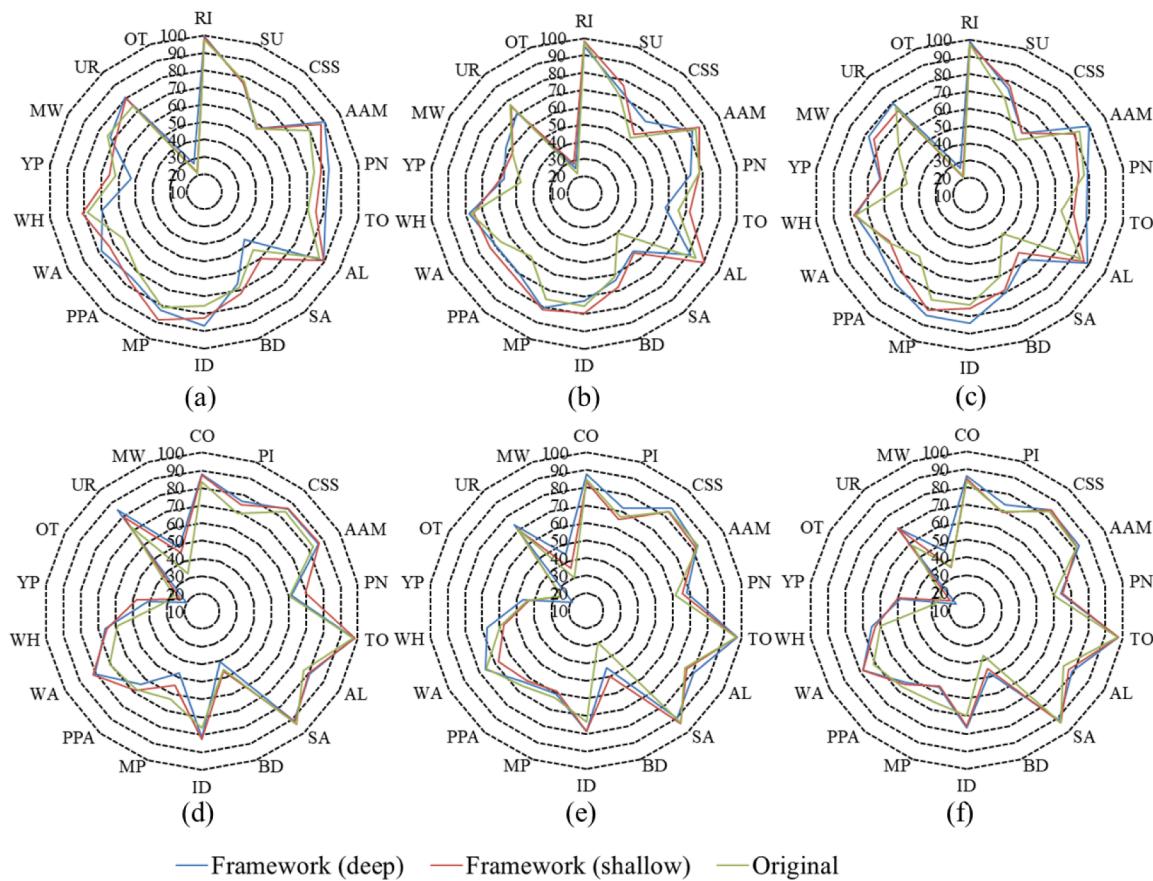
#### 5.2.5. Time costs

Our experiments were implemented using an i9-10900 K 3.7 GHz processor with 64 GB RAM and NVIDIA TITAN RTX 24 GB graphic card. Python was used in all programming. Table 6 lists the time consumption of model training and classification using NDIs-based TSIs of Sutter and Kings. Deep learning is time consuming due to its complex computation. Our proposed framework used deep learning-based feature extractor, and its multiple classifiers require more time to train. Unlike the comparative methods, the proposed framework is time consuming, but it

improves the accuracy and LULC mapping as analysed in the above sections. The time cost of the proposed frameworks also can be affected by programming, library, and computing platform, thus the program must be optimised with parallel approach to speed up the running rate in the future work. In this research, accuracy should be the most important factor in performance evaluation.

## 6. Conclusion

By establishing TSI using NDIs, the NDI-based TSI has feature curves with more abundant information to separate classes from one another. It also describes the temporal properties under feature enhancement of different NDIs effectively. Based on these findings, we proposed a TSI classification framework to classify the NDIs-based TSI. The framework



**Fig. 12.** Classification accuracy at per-class level with use of TSI classification frameworks and original deep learning methods: (a), (b) and (c) are TCNN-, Z-1DCNN- and LSTM-based methods on Sutter dataset, respectively; (d), (e) and (f) are TCNN-, Z-1DCNN- and LSTM-based methods on Kings dataset, respectively.

**Table 6**  
Time consumption of different methods (Hour).

Method	Sutter dataset		Kings dataset	
	Training	Classification	Training	Classification
1D SCNN-base framework	2.532	0.788	3.04	1.302
TCNN-based framework (deep)	2.671	0.742	2.95	1.029
Z-1DCNN-based framework (shallow)	2.869	0.761	3.08	0.953
LSTM-based framework (shallow)	2.988	0.69	3.11	0.942
TCNN	0.143	0.284	0.206	0.594
Z-1DCNN	0.147	0.263	0.265	0.558
LSTM	0.162	0.250	0.277	0.502
GenELM	0.183	0.258	0.292	0.583
GB	0.213	0.292	0.302	0.653
RF	0.162	0.281	0.101	0.318

consists of a feature extractor and a CM which contains multiple classifiers. The benefits of the feature extractor is that it can learn more discriminative features from the time series. Combined with the ensemble learning using multiple classification networks, the proposed framework can achieve higher accuracy than comparative methods, including single deep learning and non-deep learning classifiers. We also explored the deep and shallow features captured by feature extractor of the framework, and the results indicate that using different feature extractors, the accuracy of TSI classification framework can be improved to different levels. In the meanwhile, with the segment strategy of the framework, the output of the feature extractor can be used to generate more diverse base classifiers for ensemble classification.

Our new approach fused TSI construction, feature extraction and classification method to improve TSI classification. By combining deep learning and MCS, the new method has great progress on accuracy and LULC mapping. However, some of its aspects need improvement. For example, in our experiments, only four deep learning architectures were used to establish feature extractor. To fully explore our proposed framework with high accuracy, more deep learning architectures should be used. To improve the generalisation of the proposed method, more datasets of TSI should be used. Last but not least, the TSI classification framework suffers from a disadvantage of time cost. Thus, the procedure code needs more optimisation to accelerate operation speed in future studies.

To summarise, our proposed new method successfully uses a combination of deep learning and MCS to improve TSI classification. The framework is flexible and advanced and expected to be suitable for more complex RS image classifications.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

The authors wish to thank Dr. C. PelletierGeoffrey, Dr. I. WebbGeoffrey and Dr. F. Petitjean for their code of TCNN, and Dr. L. Zhong, Dr. L. Hu, and Dr. H. Zhou for the idea of their deep learning architecture. We also would like acknowledge China National Postdoctoral Council (CNPC) for providing funding for this research.

## Funding

This work was supported by National Natural Science Foundation of China (grant number 42001370) and China Postdoctoral Science Foundation (grant numbers 2019TQ0233 and 2019M662698).

## References

- Belgiu, M., Csillik, O., 2018. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sens. Environ.* 204, 509–523. <https://doi.org/10.1016/j.rse.2017.10.005>.
- Brownlee, J., 2018. Ensemble Learning Methods for Deep Learning Neural Networks, URL <https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/> (Accessed 19 December, 2018).
- Cao, F., Yang, Z., Ren, J., Chen, W., Han, G., Shen, Y., 2019. Local block multilayer sparse extreme learning machine for effective feature extraction and classification of hyperspectral images. *IEEE Trans. Geosci. Remote Sensing* 57 (8), 5580–5594. <https://doi.org/10.1109/TGRS.3610.1109/TGRS.2019.2900509>.
- Chen, Y., Dou, P., Yang, X., 2017. Improving land use/cover classification with a multiple classifier system using adaboost integration technique. *Remote Sensing* 9, 1055. <https://doi.org/10.3390/rs9101055>.
- Cheng, G., Yang, C., Yao, X., Guo, L., Han, J., 2018. When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sensing* 56 (5), 2811–2821. <https://doi.org/10.1109/TGRS.2017.2783902>.
- Dai, X., Wu, X., Wang, B., Zhang, L., 2019. Semisupervised scene classification for remote sensing images: a method based on convolutional neural networks and ensemble learning. *IEEE Geosci. Remote Sensing Lett.* 16 (6), 869–873. <https://doi.org/10.1109/LGRS.885910.1109/LGRS.2018.2886534>.
- Diek, S., Fornallaz, F., Schaeppman, M.E., 2017. Barest pixel composite for agricultural areas using landsat time series. *Remote Sensing* 9, 1245. <https://doi.org/10.3390/rs9121245>.
- Dong, L., Du, H., Mao, F., Han, N., Li, X., Zhou, G., Zhu, Di'en, Zheng, J., Zhang, M., Xing, L., Liu, T., 2020. Very high resolution remote sensing imagery classification using a fusion of random forest and deep learning technique—subtropical area for example. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 13, 113–128. <https://doi.org/10.1109/JSTARS.460944310.1109/JSTARS.2019.2953234>.
- Dou, P., Chen, Y., 2017a. Dynamic monitoring of land-use/land-cover change and urban expansion in Shenzhen using Landsat imagery from 1988 to 2015. *Int. J. Remote Sens.* 38 (19), 5388–5407. <https://doi.org/10.1080/01431161.2017.1339926>.
- Dou, P., Chen, Y., 2017b. Remote sensing imagery classification using AdaBoost with a weight vector (WV AdaBoost). *Remote Sensing Lett.* 8 (8), 733–742. <https://doi.org/10.1080/2150704X.2017.1319987>.
- Dou, P., Zeng, C., 2020. Hyperspectral image classification using feature relations map learning. *Remote Sensing* 12, 2956. <https://doi.org/10.3390/rs12182956>.
- Dou, P., Chen, Y., Yue, H., 2017. Remote-sensing imagery classification using multiple classification algorithm-based AdaBoost. *Int. J. Remote Sens.* 39 (3), 619–639. <https://doi.org/10.1080/01431161.2017.1390276>.
- Dou, P., Shen, H., Li, Z., Guan, X., Huang, W., 2021. Remote sensing image classification using deep-shallow learning. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 14, 3070–3083. <https://doi.org/10.1109/JSTARS.460944310.1109/JSTARS.2021.3062635>.
- Du, P., Bai, X., Tan, K., Xue, Z., Samat, A., Xia, J., Li, E., Su, H., Liu, W., 2020. Advances of four machine learning methods for spatial data handling: a review. *J. Geovis. Spatial Anal.* 4 (1) <https://doi.org/10.1007/s41651-020-00048-5>.
- Ebrahim, S.A., Poshtan, J., Jamali, S.M., Ebrahim, N.A., 2020. Quantitative and qualitative analysis of time-series classification using deep learning. *IEEE Access* 8, 90202–90215. <https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.2993538>.
- Fang, F., McNeil, B.E., Warner, T.A., Maxwell, A.E., Pelletier, C., Webb, G., Petitjean, F., 2019. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing* 11, 523. <https://doi.org/10.3390/rs11050523>.
- Fang, F., McNeil, B.E., Warner, T.A., Maxwell, A.E., Dahle, G.A., Eutsler, E., Li, J., 2020. Discriminating tree species at different taxonomic levels using multi-temporal WorldView-3 imagery in Washington D.C., USA. *Remote Sens. Environ.* 246, 111811. <https://doi.org/10.1016/j.rse.2020.111811>.
- Flamary, R., Fauvel, M., Dalla Mura, M., Valero, S., 2015. Analysis of multitemporal classification techniques for forecasting image time series. *IEEE Geosci. Remote Sensing Lett.* 12 (5), 953–957. <https://doi.org/10.1109/LGRS.2014.2368988>.
- Gómez, C., White, J.C., Wulder, M.A., 2016. Optical remotely sensed time series data for land cover classification: a review. *ISPRS J. Photogramm. Remote Sens.* 116, 55–72. <https://doi.org/10.1016/j.isprsjprs.2016.03.008>.
- Gong, P., Liu, H., Zhang, M., Li, C., Wang, J., Huang, H., Clinton, N., Ji, L., Li, W., Bai, Y., Chen, B., Xu, B., Zhu, Z., Yuan, C., Ping Stuen, H., Guo, J., Xu, N., Li, W., Zhao, Y., Yang, J., Yu, C., Wang, X., Fu, H., Yu, L., Dronova, I., Hui, F., Cheng, X., Shi, X., Xiao, F., Liu, Q., Song, L., 2019. Stable classification with limited sample: transferring a 30-m resolution sample set collected in 2015 to mapping 10-m resolution global land cover in 2017. *Sci. Bull.* 64 (6), 370–373. <https://doi.org/10.1016/j.scib.2019.03.002>.
- Hakkenberg, C.R., Dannenberg, M.P., Song, C., Ensor, K.B., 2018. Characterizing multi-decadal, annual land cover change dynamics in Houston, TX based on automated classification of Landsat imagery. *Int. J. Remote Sens.* 40 (2), 693–718. <https://doi.org/10.1080/01431161.2018.1516318>.
- Ho Tong Minh, D., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., Maurel, P., 2018. Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1. *IEEE Geosci. Remote Sensing Lett.* 15 (3), 464–468. <https://doi.org/10.1109/LGRS.2018.2794581>.
- Ienco, D., Gaetano, R., Dupaquier, C., Maurel, P., 2017. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geosci. Remote Sensing Lett.* 14 (10), 1685–1689. <https://doi.org/10.1109/LGRS.2017.2728698>.
- Kumar Reddy, R.V., Srinivasa Rao, B., Raju, K.P., 2018. Handwritten hindi digits recognition using convolutional neural network with RMSprop optimization. In: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). Presented at the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE. <https://doi.org/10.1109/iciccs.2018.8662969>.
- Kuncheva, L.I., Whitaker, C.J., 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learn.* 51, 181–207. <https://doi.org/10.1023/a:1022859003006>.
- Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sensing Lett.* 14 (5), 778–782. <https://doi.org/10.1109/LGRS.2017.2681128>.
- Li, J., Zhang, Z., He, H., 2017. Hierarchical convolutional neural networks for EEG-based emotion recognition. *Cogn. Comput.* 10 (2), 368–380. <https://doi.org/10.1007/s12559-017-9533-x>.
- Liu, X., Hu, G., Chen, Y., Li, X., Xu, X., Li, S., Pei, F., Wang, S., 2018b. High-resolution multi-temporal mapping of global urban land using Landsat images based on the Google Earth Engine Platform. *Remote Sens. Environ.* 209, 227–239. <https://doi.org/10.1016/j.rse.2018.02.055>.
- Liu, T., Liu, X., Liu, M., Wu, L., 2018a. Classification of rice heavy metal stress levels based on phenological characteristics using remote sensing time-series images and data mining algorithms. *Sensors* 18, 4425. <https://doi.org/10.3390/s18124425>.
- Liu, S., Shi, Q., 2020. Local climate zone mapping as remote sensing scene classification using deep learning: a case study of metropolitan China. *ISPRS J. Photogramm. Remote Sens.* 164, 229–242. <https://doi.org/10.1016/j.isprsjprs.2020.04.008>.
- Lyu, H., Lu, H., Mou, L., Li, W., Wright, J., Li, X., Li, X., Zhu, X., Wang, J., Yu, L., Gong, P., 2018. Long-term annual mapping of four cities on different continents by applying a deep information learning method to landsat data. *Remote Sensing* 10, 471. <https://doi.org/10.3390/rs10030471>.
- Ma, W., Gong, C., Hu, Y., Meng, P., Xu, F., 2013. The Hughes phenomenon in hyperspectral classification based on the ground spectrum of grasslands in the region around Qinghai Lake. In: Zhang, L., Yang, J. (Eds.), International Symposium on Photoelectronic Detection and Imaging 2013: Imaging Spectrometer Technologies and Applications. Presented at the ISPD1 2013 - Fifth International Symposium on Photoelectronic Detection and Imaging. SPIE. <https://doi.org/10.1117/12.2034457>.
- Man, C.D., Nguyen, T.T., Bui, H.Q., Lasko, K., Nguyen, T.N.T., 2017. Improvement of land-cover classification over frequently cloud-covered areas using Landsat 8 time-series composites and an ensemble of supervised classifiers. *Int. J. Remote Sens.* 39 (4), 1243–1255. <https://doi.org/10.1080/01431161.2017.1399477>.
- Maxwell, A.E., Warner, T.A., Fang, F., 2018. Implementation of machine-learning classification in remote sensing: an applied review. *Int. J. Remote Sens.* 39 (9), 2784–2817. <https://doi.org/10.1080/01431161.2018.1433343>.
- Shao, Y., Lunetta, R.S., Wheeler, B., Iiames, J.S., Campbell, J.B., 2016. An evaluation of time-series smoothing algorithms for land-cover classifications using MODIS-NDVI multi-temporal data. *Remote Sens. Environ.* 174, 258–265. <https://doi.org/10.1016/j.rse.2015.12.023>.
- Tao, Y., Xu, M., Lu, Z., Zhong, Y., 2018. DenseNet-based depth-width double reinforced deep learning neural network for high-resolution remote sensing image per-pixel classification. *Remote Sensing* 10, 779. <https://doi.org/10.3390/rs10050779>.
- Tsai, Y., Stow, D., Chen, H., Lewison, R., An, L., Shi, L., 2018. Mapping vegetation and land use types in Fanjingshan national nature reserve using google earth engine. *Remote Sensing* 10, 927. <https://doi.org/10.3390/rs10060927>.
- Wang, Y., Li, Z., Zeng, C., Xia, G.-S., Shen, H., 2020. An urban water extraction method combining deep learning and google earth engine. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 13, 769–782. <https://doi.org/10.1109/JSTARS.460944310.1109/JSTARS.2020.2971783>.
- Wang, H., Zhao, X., Zhang, X., Wu, D., Du, X., 2019. Long time series land cover classification in China from 1982 to 2015 based on Bi-LSTM deep learning. *Remote Sensing* 11, 1639. <https://doi.org/10.3390/rs11141639>.
- Whelen, T., Siqueira, P., 2018. Time-series classification of Sentinel-1 agricultural data over North Dakota. *Remote Sensing Lett.* 9 (5), 411–420. <https://doi.org/10.1080/2150704X.2018.1430393>.
- Xiao, J., Wu, H., Wang, C., Xia, H., 2018. Land cover classification using features generated from annual time-series landsat data. *IEEE Geosci. Remote Sensing Lett.* 15 (5), 739–743. <https://doi.org/10.1109/LGRS.2018.2805715>.
- Xie, S., Liu, L., Zhang, X., Chen, X., 2018. Annual land-cover mapping based on multi-temporal cloud-contaminated landsat images. *Int. J. Remote Sens.* 40 (10), 3855–3877. <https://doi.org/10.1080/01431161.2018.1553320>.
- Yang, G., Shen, H., Zhang, L., He, Z., Li, X., 2015. A moving weighted harmonic analysis method for reconstructing high-quality SPOT VEGETATION NDVI time-series data. *IEEE Trans. Geosci. Remote Sensing* 53 (11), 6008–6021. <https://doi.org/10.1109/TGRS.2015.2431315>.
- Yokoya, N., Ghamisi, P., Xia, J., Sukhanov, S., Heremans, R., Tankoyeu, I., Bechtel, B., Le Saux, B., Moser, G., Tuia, D., 2018. Open data for global multimodal land use classification: outcome of the 2017 IEEE GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 11 (5), 1363–1377. <https://doi.org/10.1109/JSTARS.460944310.1109/JSTARS.2018.2799698>.
- Yu, L., Liu, X., Zhao, Y., Yu, C., Gong, P., 2018. Difficult to map regions in 30 m global land cover mapping determined with a common validation dataset. *Int. J. Remote Sens.* 39 (12), 4077–4087. <https://doi.org/10.1080/01431161.2018.1455238>.

- Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang, Q., Wang, J., Gao, J., Zhang, L., 2020. Deep learning in environmental remote sensing: achievements and challenges. *Remote Sens. Environ.* 241, 111716. <https://doi.org/10.1016/j.rse.2020.111716>.
- Zhong, L., Hu, L., Zhou, H., 2019. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* 221, 430–443. <https://doi.org/10.1016/j.rse.2018.11.032>.
- Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* 5 (4), 8–36. <https://doi.org/10.1109/MGRS.2017.2762307>.