Multitemporal Relearning With Convolutional LSTM Models for Land Use Classification

Yue Zhu, Christian Geiß, *Member, IEEE*, Emily So, and Ying Jin

***Abstract*—In this article, we present a novel hybrid framework, which integrates spatial–temporal semantic segmentation with postclassification relearning, for multitemporal land use and land cover (LULC) classification based on very high resolution (VHR) satellite imagery. To efficiently obtain optimal multitemporal LULC classification maps, the hybrid framework utilizes a spatial–temporal semantic segmentation model to harness temporal dependency for extracting high-level spatial–temporal features. In addition, the principle of postclassification relearning is adopted to efficiently optimize model output. Thereby, the initial outcome of a semantic segmentation model is provided to a subsequent model via an extended input space to guide the learning of discriminative feature representations in an end-to-end fashion. Last, object-based voting is coupled with postclassification relearning for coping with the high intraclass and low interclass variances. The framework was tested with two different postclassification relearning strategies (i.e., pixel-based relearning and object-based relearning) and three convolutional neural network models, i.e., UNet, a simple Convolutional LSTM, and a UNet Convolutional-LSTM. The experiments were conducted on two datasets with LULC labels that contain rich semantic information and variant building morphologic features (e.g., informal settlements). Each dataset contains four time steps from WorldView-2 and Quickbird imagery. The experimental results unambiguously underline that the proposed framework is efficient in terms of classifying complex LULC maps with multitemporal VHR images.**

***Index Terms*—Classification postprocessing (CPP), convolutional neural networks (CNNs), deep learning (DL), multitemporal land use classification, relearning.**

I. Introduction

OVER recent years, the number of very high resolution (VHR) multitemporal satellite imagery has significantly increased and become commercially available [1]. VHR imagery provides opportunities for extracting many details including various land use types and building morphologies. One potential application domain is land use analysis [2]–[4]. The World Bank [5] estimates that three billion people will live in substandard housing by 2030. By 2050, the UN [6] projects that two thirds of the world's population, around 7 billion people, will live in urban areas. The increasingly available multitemporal satellite imagery can be beneficial for examining urban development over time and help set policies to limit urban sprawls, among others. Efficient methods for classifying land use and land cover (LULC) from multitemporal VHR imagery is therefore timely and worthy of further exploration.

Finding suitable and effective approaches for multitemporal LULC classification based on VHR remote sensing imagery remains challenging. First, the significantly improved spatial resolution of VHR imagery leads to high intravariation and low intervariation between each LULC class [7], [8]. This issue inevitably decreases the separability between different LULC classes, especially for land use categories that contain much semantic information. Moreover, most of the existing temporal methods have not fully exploited the temporal sequential features in multitemporal data because of their limitations regarding automation and flexibility [9]. It can be argued that the temporal dependency embedded in the consecutive time steps of time-series data contains the features of transition patterns, i.e., transitions rules of LULC changes. It has been widely recognized that deep learning (DL) methods can extract rules that represent the relationship between the distributions of input and output. Convolutional neural networks (CNN) are designated for processing spatial features, whereas recurrent neural networks (RNN) excel at analyzing temporal relationships. These advanced developments in the DL field shed light on taking advantages of the temporal dependency features for improving classification accuracy.

In the past, different strategies were followed to improve the accuracy of LULC classification for VHR remote sensing imagery.

A. Convolutional Neural Networks for LULC Classification

DL algorithms, particularly CNN, have gained great success and are deployed in the remote sensing community [10]. It is because CNNs excel at effectively encoding discriminating features based on spectral and spatial information [11]. Such abilities enable CNN models to achieve remarkable accuracy in image classification tasks [12], [13]. The applications of CNNs in LULC can be mainly grouped into two categories: scene-based classification and pixel-based classification [14].

Manuscript received October 19, 2020; revised December 29, 2020; accepted January 20, 2021. Date of publication February 1, 2021; date of current version March 26, 2021. This work was supported in part by the Cities and Transport Research Group at the Martin Centre for Architectural and Urban Studies and in part by the Cambridge Centre for Digital Built Britain. The work of Ying Jin was supported in part by the 'Urban spatial modelling and decision-support system' Project of the Beijing Innovation Centre for Future Urban Design, and the EPSRC Managing Air Green Inner Cities (MAGIC) under Grant (EP/N010221/1). *(Corresponding author: Yue Zhu.)*

Yue Zhu, Emily So, and Ying Jin are with the Department of Architecture, University of Cambridge, CB2 1TN Cambridge, U.K. (e-mail: yz591@cam.ac.uk; ekms2@cam.ac.uk; yj242@cam.ac.uk).

Christian Geiß is with the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), 82234 Weßling-Oberpfaffenhofen, Germany (e-mail: christian.geiss@dlr.de).

Digital Object Identifier 10.1109/JSTARS.2021.3055784

Scene-based classification, also termed patch-based classification, refers to the categorization of images into a set of LULC classes based on the main content of each image [15]. Many efforts of deploying CNNs for scene-based classification of remote sensing imagery have been made. A typical example of scene-based classification starts with sampling numerous patches from a relatively large image for model training, then an LULC map is generated by the trained model through classifying the scene category of every sampled patch [16]. Sharma *et al.* [16] proposed a path-based CNN framework that is designated for medium-resolution remote sensing data. Liu and Shi [17] deployed a scene-based CNN model for local climate mapping. It can be argued that scene-based CNNs have several limitations in LULC classification. First, a suitable size of input patch is difficult to define, especially when the sizes of ground targets are highly variant [18]. Moreover, the patches sampled in the same large image are processed independently, which means the context information is neglected in the process of classification. Furthermore, scene-based methods usually are adopted for recognizing large objects, whereby pixel-based methods are more suitable for detecting fine details [19].

Pixel-based classification refers to techniques which assign a class label to each pixel of an image. Semantic segmentation techniques, as one type of pixel-based classification, assign labels to each pixel according to contextual information. Over recent years, fully convolutional networks (FCNs) [20] have received much attention in terms of their outstanding performance in semantic segmentation tasks. Comparing with conventional pixel-based classifiers that partition pixels solely based on the specific spectral information, FCNs perform pixel-level labeling by using multiple fully convolutional layers to extract high-level contextual features embedded in images. Although FCNs usually do not exhibit heavy salt-and-pepper noise, one of the constraints of FCNs is the blurring class boundaries caused by multilevel abstractions [18], [21].

It has been widely recognized that the performance of a CNN model can heavily depend on the quality and amount of training data. Unlike ordinary RGB images that have open datasets containing enormous amount of data for training CNN models, large-scale annotated multispectral remote sensing data is scarce [22]. Therefore, considerable efforts have been made in developing CNN models that are suitable for remote sensing imagery. One of the strategies is to exploit existing trained CNN models, by including transfer learning strategies [23] and fine tuning mechanisms [11] for model adaption. However, this direction requires highly resembled classification tasks and model structures, for example, parameters trained on ordinary dataset are very challenging to be transferred to multispectral remote sensing dataset. Alternatively, another effective direction is to develop CNN-based frameworks that are tailored for remote sensing data, i.e., integrating with object-based segments [1], and developing effective post-classification methods [18], [24].

B. Integration of Object-Based Image Analysis and Convolutional Neural Networks

Object-based image analysis (OBIA) has been widely applied for mitigating the high intraclass and low interclass variabilities in VHR imagery [25]. Object-based segmentation refers to the process of partitioning images into small objects based on homogeneity attributes of the image. One of the primary constituting aspects of such techniques is to model meaningful real-world objects (e.g., with a segmentation algorithm) before further processing. Those allow for a diversified characterization of spectral values, consideration of geometry-related properties of objects, and also encoding of additional spatial information such as relationships of (topological) neighborhood and spatial hierarchy [26]. Given the outstanding performance of DL in tasks of scene-based classification and semantic segmentation, attempts of integrating OBIA with DL models have been explored. For example, X. Zhang *et al.* [3] extracted spectral, spatial, and texture features of image segments to train a deep neural network for land-cover classification.

In terms of the integration of OBIA with CNNs, some research focused on the direction of deploying OBIA as a method of preprocessing inputs for training scene-based DL models. However, the object-based segmentation cannot be directly used for CNN training. This is because object-based segments have irregular shapes and various sizes, whereas CNN models demand input units to have square shapes and uniformed size [18]. To deal with this issue, [1] proposed to use the minimum bounding box of object-based segments to sample training inputs for scene-based classification. However, the implementation of OBIA as a preprocessing method still could not address the aforementioned limitation of scene-based CNNs in terms of deploying contexture information for classification.

To deal with VHR imagery, besides clustering pixels into objects as processing units, another strategy is to adopt effective postclassification methods [27]. Object-based voting (OBV) has been extensively acknowledged as an effective OBIA-related classification postprocessing (CPP) method, which refers to a refinement of classified labels according to the boundaries of objects to improve the classification accuracy [28]. In general, CPP methods can achieve considerable accuracy improvement efficiently and concisely. However, comparing with preprocessing methods, much less attention has been paid on CPP methods [28]. Therefore, the potential of deploying OBV-based CPP methods for refining the output of a CNN is worthy of further exploration. In this manner, Liu *et al.* [18] proposed a simple but efficient framework, which integrates the OBV with DL. Thereby, the outputs of a CNN are aggregated with a majority voting strategy to object-based segments. The hybrid method brought significant improvements to the classification accuracy of CNN outputs.

The process of object-based segmentation is conducted in an iterative bottom-up manner that starts from merging pixels into objects [29]. The sizes and geometries of objects are determined by three parameters, which are “scale,” “shape,” and “compactness.” The “scale” parameter determines the sensitivity for the object fusion [29], greater scale value results in larger segmentation area. The “shape” parameter defines the influence of color on the segmentation process, the “compactness” parameter defines the smoothness and compactness of object boundaries. Among the three parameters of object-based segmentation, it is worth noting that the “scale” parameter has the most significant effect on the segmentation of objects, it determines whether objects are segmented into appropriate sizes.

For instance, large scales can result in under-segmentation, whereas small scales can lead to over-segmentation. Consequently, Liu *et al.* [18] also claim that the selection of different scales can have significant effects on the results of refinements.

C. Postclassification Relearning for Land Use and Land Cover Classification

Although the aforementioned OBV-based CPP methods can enhance raw classification accuracy, they do not account for improving the separability between classes [28]. Regarding this issue, relearning-based CPP methods have demonstrated immense potential in terms of improving postclassification accuracy through enhancing the separability of the original classifier [28]. The basic idea of postclassification relearning is to deploy the initial output of the model for calculation of additional features. Then with such additional features, the performance of the model can be enhanced with extended input space after training for a second time [24].

Over recent years, the advantages of using relearning-based CPP has raised attention. Geiß and Taubenböck [24] proposed an object-based relearning (OBR) framework, which improved the classification accuracy by retraining a model with a triplet of hierarchical OBV features generated from its preliminary outcomes. Also, Han *et al.* [30] proposed an edge-preservation multiclassifier relearning framework, which includes iterative relearning procedures based on landscape metrics to enhance the separability of LULC classes. Furthermore, Shi *et al.* [31] developed an active relearning framework that can improve the classification results with less labeling costs. More recently, Lei *et al.* [32] developed an object-oriented classification method which iteratively integrates classification results. The experimental results achieved promising accuracy even with a limited number of samples. The above experiments all conducted the relearning processes in an iterative manner, in order to harness the additional feature that can be updated after each iteration. Such process can be carried on until the optimization hits a plateau.

D. Multitemporal Land Use and Land Cover Classification

Multitemporal LULC classification is an active field in remote sensing community. Over recent years, since the access to multitemporal remote sensing imagery became increasingly available, more opportunities emerged for the utilization of temporary dependency to improve LULC classification [33]. Temporal sequential LULC data can significantly facilitate change detection and growth prediction. Moreover, the temporal dependency embedded in multitemporal data can be utilized to enhance the classifier performance. Vuolo *et al.* [34] tested the effect of adding temporal information as additional features for crop classification. The classification accuracy showed considerable improvement after the utilization of temporal information. However, in practice, it is still challenging to find an effective method for multitemporal LULC classification. Conventional methods of temporal feature extraction have many limitations, including time-consuming manual feature engineering and predefined rules that lack of flexibility [9].

In the domain of DL, models of RNNs, including long short-term memory (LSTM) and gated recurrent units, are designated for processing sequential temporal data. Subsequently, to harness the power of RNNs for processing spatial–temporal sequential data, efforts of integrating CNNs and RNNs have been made. A Convolutional LSTM (ConvLSTM) network is proposed for anticipating future precipitation [31]. Given multitemporal land use data is inherently spatial–temporal sequential data, recurrent convolutional structures have been applied for LULC classification and prediction. In this manner, the framework of ConvLSTM has been deployed in the field of remote sensing. Mou *et al.* [35] integrated CNN and LSTM layers to form a recurrent CNN for detecting land cover changes. Moreover, Rußwurm and Körner [36] proposed an encoder structure with recurrent convolutional layers for land cover classification. This structure can utilize the temporal interdependency embedded in the input data. As a result, they improved classification accuracy as well as alleviated the pre-processing work regarding minor missing information (e.g., clouding filling).

Although the model of ConvLSTM is designed for processing spatial–temporal data, the structure of a simple ConvLSTM model does not have advantages for the task of pixel-based classification. To cope with this issue, attempts of combining ConvLSTM with FCNs have been made. Milletari *et al.* [37] proposed a coarse-to-fine context memory framework for medical image segmentation, which uses ResNet as encoder and ConvLSTM layers as decoder. Each ResNet block in the encoder is timely distributed with its counterpart ConvLSTM layer in the decoder. Such structure enables feature interpretation based on coarse-to-fine context information, and allowed higher classification accuracy of image data than a simple UNet. Moreover, Azad *et al.* [38] developed a ConvLSTM U-Net model for medical image segmentation, which employed bidirectional ConvLSTM layers in the decoder part of a U-Net. This framework presented better segmentation performance due to its ability of abstracting more discriminative features. As for the applications of semantic segmentation tasks for remote sensing imagery, Gallego *et al.* [39] used ConvLSTM unites as the first layer in autoencoders for segmenting oil spills from side-looking airborne radar imagery. Similarly, Teimouri *et al.* [40] adopted ConvLSTM layers at the last stage of an FCN for classifying various crop types from SAR data.

From a unifying perspective, it can be argued that coupling OBV with relearning methods can reasonably optimize classification boundaries and improve class separability. However, to the best of our knowledge, approaches which internalize those processing principles in a beneficial way for multitemporal models are currently absent. Consequently, in this article, we aim to uniquely examine the benefits of temporal dependency in a deep relearning context. For this purpose, we propose a hybrid framework for efficient multitemporal LULC classification of VHR remote sensing imagery. The framework adopts a recurrent convolutional structure as LULC classifier, which is integrated with postclassification relearning for model improvement. At

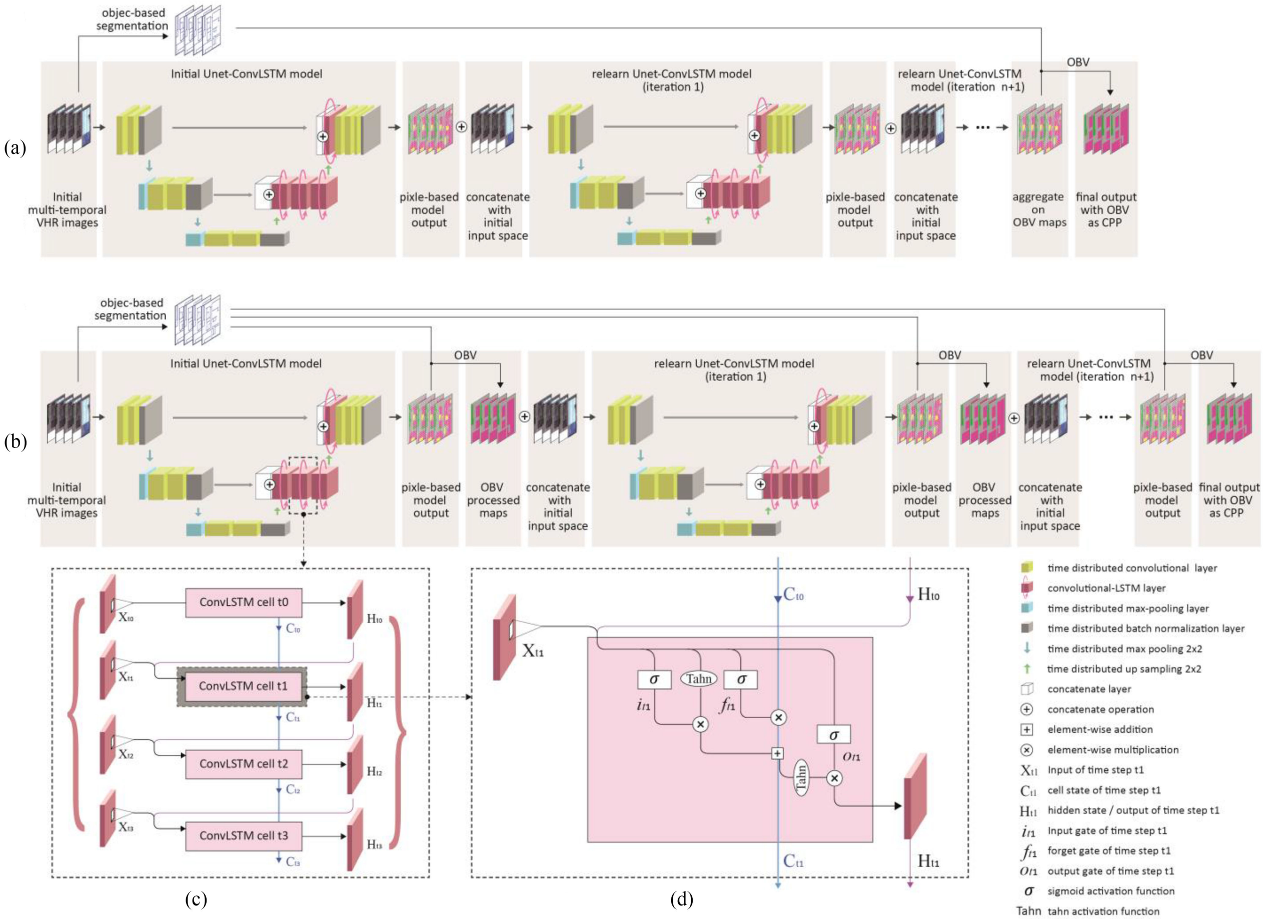

(a)
objec-based segmentation
Initial Unet-ConvLSTM model
relearn Unet-ConvLSTM model (iteration 1)
relearn Unet-ConvLSTM model (iteration n+1)
OBV
Initial multi-temporal VHR images
pixle-based model output
concatenate with initial input space
aggregate on OBV maps
final output with OBV as CPP
(b)
objec-based segmentation
Initial Unet-ConvLSTM model
relearn Unet-ConvLSTM model (iteration 1)
relearn Unet-ConvLSTM model (iteration n+1)
OBV
Initial multi-temporal VHR images
pixle-based model output
OBV processed maps
concatenate with initial input space
final output with OBV as CPP
X$_{t0}$
X$_{t1}$
X$_{t2}$
X$_{t3}$
ConvLSTM cell t0
ConvLSTM cell t1
ConvLSTM cell t2
ConvLSTM cell t3
C$_{t0}$
C$_{t1}$
C$_{t2}$
C$_{t3}$
H$_{t0}$
H$_{t1}$
H$_{t2}$
H$_{t3}$
(c)
σ
Tahn
σ
i_{t1}
f_{t1}
σ
o_{t1}
Tahn
(d)
time distributed convolutional layer
convolutional-LSTM layer
time distributed max-pooling layer
time distributed batch normalization layer
time distributed max pooling 2x2
time distributed up sampling 2x2
concatenate layer
concatenate operation
element-wise addition
element-wise multiplication
X$_{t1}$ Input of time step t1
C$_{t1}$ cell state of time step t1
H$_{t1}$ hidden state / output of time step t1
i_{t1} Input gate of time step t1
f_{t1} forget gate of time step t1
o_{t1} output gate of time step t1
σ sigmoid activation function
Tahn tahn activation function

Fig. 1. Overview of the two proposed multitemporal relearning frameworks. (a) Pixel-based relearning with OBV as CPP. (b) OBV-based relearning with OBV as CPP. (c) Structure of a ConvLSTM layer. (d) Structure of a ConvLSTM cell inside the ConvLSTM layer.

the relearning stage, the consecutive temporal outcomes of a ConvLSTM-based model were utilized to extend the original input space in the temporal dimension. Then the classifier was retrained using the extended input space to improve its class separability. The relearning process was iteratively conducted to achieve an optimal result. The various combinations of relearning strategies, including OBV-based relearning and pixel-based relearning, were tested and evaluated in an exhaustive manner. The proposed framework was examined with complex classification categories with relatively few labeled pixels in order to show that this framework has potential to be applied for a wide range of multitemporal LULC tasks.

The rest of the article is organized as follows: Section II introduces each component of the proposed framework. The experiment datasets and setup are described in Section III. Then we report the results of experiments in Section IV and finally Section V concludes this article.

II. Proposed Methodology

An overview of the proposed methods is provided in Fig. 1. They build upon an advanced UNet-ConvLSTM model for multitemporal LULC classification (Section II-A). Subsequent to that, the outputs are processed with an OBV method (Section II-B) to both eventually enhance the classification output and establish a further input for an iterative relearning strategy (Section II-C).

A. *Convolutional Neural Network Models*

1) UNet Model for Semantic Segmentation: It has been widely recognized that FCNs can achieve robust performance in the tasks of LULC classification. UNet, a semantic segmentation model built upon conventional FCNs, was first introduced by Ronneberger *et al.* [41] for biomedical image segmentation. The name of “UNet” comes from its distinctive u-shaped encoder–decoder architecture, which presents the down-sampling and up-sampling process. To be more specific, the down-sampling part of the network acts as the encoder, which extracts features through convolutional layers and downscales data by max pooling layers at multiple scales. Then the up-sampling part of the network functions as the decoder, which expands the down-sampled data at each scale to match its counterpart layer of the encoder. The encoder and the decoder form a more or less symmetric structure. This structure of UNet facilitates feature extraction at multiple scales, which enables the classifier to consider both global context and local high-level features.

Another distinctive feature of UNet comprises the skip connections between encoder and decoder, which concatenate the low level, coarse feature maps in the encoder with the high level, fine feature maps in the corresponding scale of decoder [42]. Such skip connections between the encoder and decoders can effectively restore the fine-grained features in the model output.

UNet-based models exhibited promising performance when being applied on tasks of pixelwise image segmentation, including high spatial resolution aerial photo and remote sensing images [43]. Comparing with conventional CNNs that require a large amount of training data, UNet excels at achieving precise segmentations with much fewer training images [41] and less training time. However, in the process of fusion features extracted by encoder and decoder, the low-level features generate noise in high-level features, therefore result in blurring segment boundaries [44]. Furthermore, although UNet has robust performance in semantic segmentation tasks, a two-dimenional (2-D) UNet does not consider temporal dependency in the data.

In this article, we deploy UNet as a benchmark model to compare the effects of relearning strategies on a single-temporal model with multitemporal models.

2) ConvLSTM Models for Spatial–Temporal Data: ConvLSTM, first proposed by X. Shi *et al.* [45], is developed based on the structure of fully connected LSTM (FC-LSTM). Similar with FC-LSTMs, ConvLSTM structures contain hidden states $\mathrm{H}_1, \ldots, \mathrm{H}_t$, and cell states $\mathrm{C}_1, \ldots, \mathrm{C}_t$. The hidden states can be regarded as cell outputs. The cell states function as the memory of layer, the information in which can be selectively updated or discarded. In a ConvLSTM model, convolutional structures are deployed to replace the full connected layers for input-to-state and state-to-state transition. In this manner, both temporal consistency and spatial correlation are taken into consideration in a ConvLSTM model.

Multiple ConvLSTM layers can be stacked together to form more complex structures. The input space of a ConvLSTM layer $\mathrm{X} \in \mathrm{R}^{t\times h\times w\times c}$, in which t, h, w, and c, respectively, refer to time steps, height, width, and channels. A simple ConvLSTM model consists of multiple ConvLSTM layers, and a ConvLSTM layer contains several ConvLSTM cells. The number of ConvLSTM cells in a ConvLSTM layer corresponds to the number of time steps of the dataset. Each cell takes $\mathrm{X}_t \in \mathrm{R}^{h\times w\times c}$ as input, and generates its hidden state $\mathrm{H}_t \in \mathrm{R}^{h\times w\times c}$ and cell state $\mathrm{C}_t \in \mathrm{R}^{h\times w\times c}$ at time step *t*. The cell output H_t and state C_t are controlled by three gates (i.e., forget gate f_t, input gate i_t, and output gate o_t), which have same spatial dimensions (i.e., $\mathrm{R}^{h\times w\times c}$). The three gates are computed by corresponding learnable weights and biases with activation functions [Fig. 1(d)]. The key equations of a typical ConvLSTM layer can be described as follows [45]:

$$\mathrm{i}_t = \sigma\left(W_{xi} * \mathrm{X}_t + W_{hi} * \mathrm{H}_{t-1} + W_{ci} \circ \mathrm{C}_{t-1} + \mathrm{b}_i\right) \quad (1)$$

$$\mathrm{f}_t = \sigma\left(W_{xf} * \mathrm{X}_t + W_{hf} * \mathrm{H}_{t-1} + W_{cf} \circ \mathrm{C}_{t-1} + \mathrm{b}_f\right) \quad (2)$$

$$\mathrm{o}_t = \sigma\left(W_{xo} * \mathrm{X}_t + W_{ho} * \mathrm{H}_{t-1} + W_{co} \circ \mathrm{C}_t + \mathrm{b}_o\right) \quad (3)$$

$$\mathrm{C}_t = f_t \circ \mathrm{C}_{t-1} + \mathrm{i}_t \circ \tanh\left(W_{xc} * \mathrm{X}_t + W_{hc} * \mathrm{H}_{t-1} + \mathrm{b}_c\right) \quad (4)$$

$$\mathrm{H}_t = \mathrm{o}_t \circ \tanh\left(\mathrm{C}_t\right) \quad (5)$$

where W and b refer to weight and bias, σ and $\tanh$ refer to sigmoid activation function and tanh activation function. "$*$" denotes the convolution operator and "$\circ$" represents the Hadamard product.

3) Integrating UNet and ConvLSTM for Multitemporal LULC Classification: In essence, consecutive time steps of LULC maps are spatial–temporal sequences. As such, multitemporal LULC classification is not only an image segmentation problem, but also contains issues related to the encoding of spatial-temporal relationships (i.e., urban transition rules).

As discussed in Section I-D, the integration of ConvLSTM units with FCNs can facilitate the extracted spatial–temporal features for image segmentation. Therefore, to develop efficient methods for multitemporal LULC classification, this article incorporates ConvLSTM layers into a UNet-like encoder–decoder structure to form an architecture named as UNet-ConvLSTM.

The encoder part of the UNet-ConvLSTM follows a contracting path, which adopts convolutional layers for retrieving spatial features and max pooling layers for down-sampling and getting global context information. The decoder part of the model follows an expanding path, which stacks multiple ConvLSTM layers for extracting temporal relationships. On top of the encoder–decoder structure, the corresponding encoder and decoder blocks are concatenated by the skip connections (Fig. 1), which aims to integrate low-level spatial features with high-level spatial–temporal features. Furthermore, considering the datasets only contain four time steps, which can be regarded as short-term prediction problems for LSTM models, therefore it is not necessary to implement a complex ConvLSTM structure. To reduce the complexity of the integration of UNet and ConvLSTM, the depth of the encoder–decoder structure of UNet-ConvLSTM was decreased to two down-sampling scales.

B. Object-Based Voting (OBV)

OBV has been tested and proven to be an effective CPP method for CNNs [18]. Therefore, coupling OBV with relearning is very likely to bring improvements to initial model outputs. In this article, a combination of OBV and relearning is adopted as one of the main relearning strategy categories (Fig. 1). The process of conducting OBV-based relearning is described in Section II-C. Furthermore, OBV is also used as a simple CPP method at the last step for benchmarking. The detailed process of conducting OBV can be described as follows:

For pixels x in the image $\boldsymbol{I}$, they are partitioned into objects according to a scale parameter s. As shown in Fig. 2, an optimal value of s can be selected through exhaustive tests of a set of values $\mathrm{S} \in \{\ldots,\ s-1,\ s,\ s+1,\ \ldots\}$. In general, multiscale segmentation can be conducted following the constraint:

$$\bigcup_{O_i^{s-1} \subseteq O_j^s} O_i^{s-1} = O_j^s. \quad (6)$$

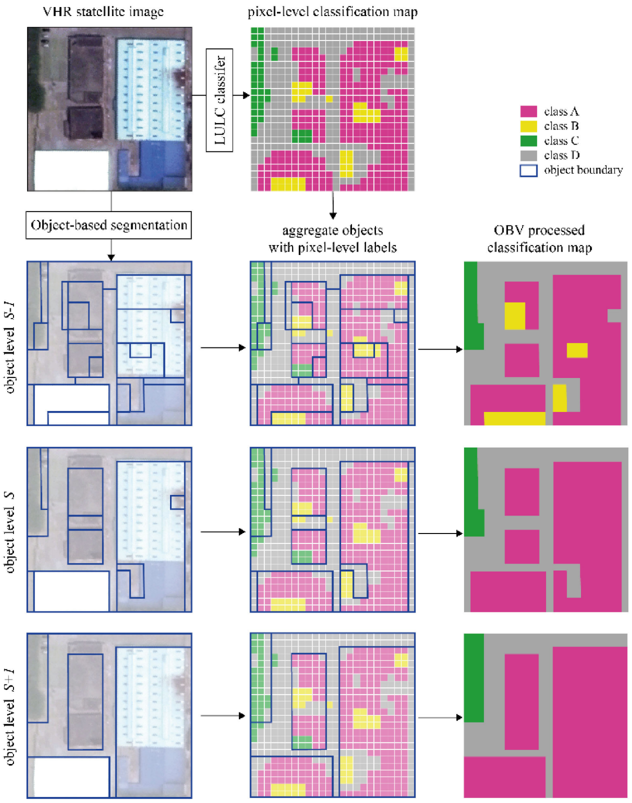

VHR statellite image
pixel-level classification map
LULC classifier
class A
class B
class C
class D
object boundary
Object-based segmentation
aggregate objects with pixel-level labels
OBV processed classification map
object level S-1
object level S
object level S+1

Fig. 2. Effects of applying object-based voting (OBV) with different scale parameters on a pixel-level classification map.

Segments O_j^s with larger scale s are generated based on the segments O_i^{s-1} of smaller scale $s-1$. Subsequently, the generated object-based segments are aggregated with pixel-based labels of the classification maps produced by LULC classifiers. Then pixels inside the object-based segments of each scale are processed by majority voting to update the values of labels:

$$F_j^s = \operatorname{argmax}\left(f\left(O_j^s\right)\right) \tag{7}$$

where $f(O_j^s)$ refers to the frequency of each label inside an object O_j^s. After the OBV postprocessing, the most frequent label is assigned to all the pixels inside the object O_j^s. F_j^s denotes the updated pixel labels at level s. After comparisons of F_j^s with ground truth F_j^{gt} an optimal scale can be determined.

C. Postclassification Relearning

A simple postclassification relearning has three main consecutive steps: 1) a pixel-based supervised model is trained with multispectral features F_s as initial input; 2) the overlap-tile strategy segmentation [41] is deployed to obtain a seamless classification map by the trained model, whereas the resulting two dimensional classification map can be regarded as an additional feature F_p; 3) then F_s and F_p are stacked together at the beginning of each relearning phase to form an extended input space F'_s for the next phase of relearning. The purpose of this concatenation operation is to generate new features that are likely to be beneficial for improving the discriminative capability of the model. As such, the process from step 1 to step 3 can be carried out iteratively to seek an optimal result. Two main types of relearning strategies were tested in this article, pixel-based relearning [Fig. 1(a)] and OBV-based relearning [Fig. 1(b)]. The former refers to using the output of the initial trained model, a pixel-based classification map, as a relearned feature F_p. Then F_p is concatenated with the initial multispectral input features F_s to form a new input for the next iteration of relearning. The latter adds an extra step of applying OBV on F_p to generate a postprocessed output $\mathrm{F}_{\mathrm{obv}}$, which is then concatenated with F_s for OBV-based relearning. The reason of adding the extra step of OBV is because the OBV-processed results can yield improvement regarding classification accuracy, which means that $\mathrm{F}_{\mathrm{obv}}$ tends to have higher classification accuracy than F_p. As such, the extend input space F'_s generated with $\mathrm{F}_{\mathrm{obv}}$ is likely to provide the next relearning phase with better guided information for LULC classification.

III. Datasets and Experimental Setup

A. Study Area

Our study area includes a suburban area situated at the city border of Dongguan and Shenzhen, Pearl River Delta, southern China. Shenzhen and Dongguan have experienced rapid urbanization over the last four decades. During such expansion, a significant number of rural villages have been merged into the urban area, resulting in the prevalence of urban villages. Due to the complex social-economic development in China, these urban villages have unique building morphologies compared with informal settlements elsewhere [46]. More specifically, the informal settlements in Pearl River Delta mostly have their original rural settlements as old cores [47]. These old cores have been gradually encompassed by highly dense mid-rise informal settlements (so-called "handshake buildings") that were redeveloped on the plots of original rural settlements. The advent of DL provides new opportunities for remote sensing community to map informal settlements [23], [48]. In this article, we adjust the LULC category according to the local context to include "informal settlements" and the related "rural settlements" in the category.

B. Datasets

Two datasets were deployed for a quantitative evaluation of the models. Both datasets have a spatial coverage of 90 km^2. Each dataset contains four time steps with an interval of approximately 5 years (Fig. 3). Due to the limited availability of temporal sequences, the datasets are from two sensor sources: WorldView-2 and QuickBird. The WorldView-2 data covers the years 2012 and 2018, and the QuickBird data covers the years 2002 and 2007. In terms of the spatial resolution, the data from both sources have a spatial resolution of 0.5 m with an image size of 4096 × 4096 pixels. Regarding the spectral resolution, QuickBird data contains four spectral bands, which are red, blue, green, and NIR. Although WorldView-2 data provides eight spectral bands, only four spectral bands (red, blue, green, and NIR1) are used to match the four corresponding bands of

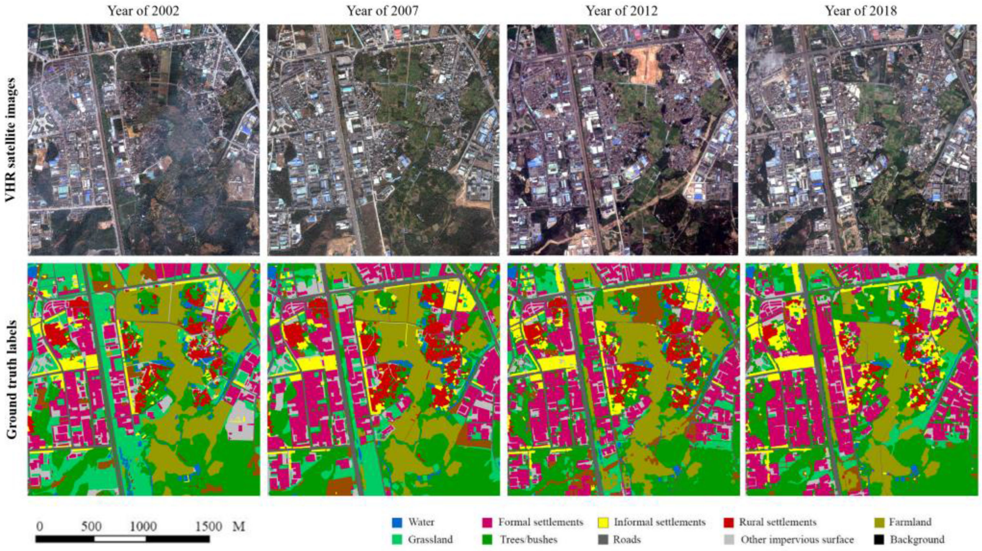


Fig. 3. Multitemporal VHR imagery and corresponding ground truth labels of dataset I.

QuickBird data. All the ground truth labels were manually made under consideration of ancillary cadastral maps.

Satellite images were cropped into image tiles of 128 × 128 pixels with an overlap of 32 pixels for the purpose of increasing the amount of training data. The total number of cropped patches was 7056, 80% of which were randomly sampled for training and 20% were used for validation. To ensure a fair evaluation of all the experimental approaches, a subset of the validation dataset is selected as testing dataset for evaluation purposes. It should be noted that the subset selectively consists of the areas of the validation dataset that are not overlapped with training dataset, i.e., areas which are strictly spatially disjoint [49]. Consequently, the testing dataset consists of data that has completely not been used for training.

The VHR satellite images enable the observation of various building morphologies and different vegetation types, especially the differentiation between original rural villages and follow-up informal settlements in the study area. In total, 10 LULC classes were defined in this article, including: "rural settlements," "informal settlements," "formal settlements," "bare soil," "grassland," "farmland," "trees and bushes," "water," "roads," and "other impervious surface" (Fig. 4). It should be noted that the accuracy of LULC classification can be largely subject to the settings of LULC classes [10], larger numbers of LULC classes or semantic categories could result in a decrease of classification accuracy. Comparing with other commonly adopted LULC categories, the inclusion of "informal settlements" and "rural settlement" significantly increases the complexity of the classification task due to their unique urban fabric and the need to incorporate large amounts of semantic information.

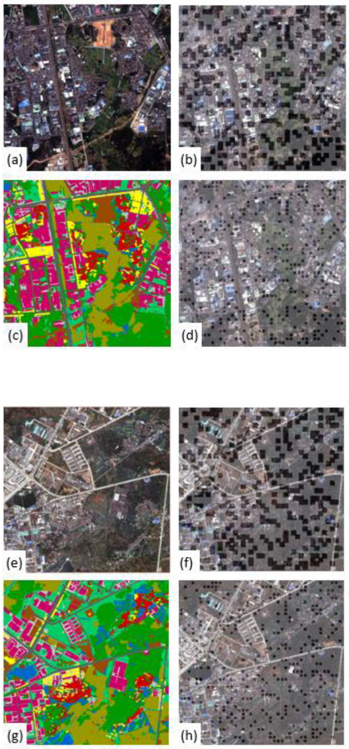
C. Experimental Setup

Three convolutional models were built in this article, including a simple UNet model, a simple ConvLSTM model and an UNet-structured ConvLSTM model (Fig. 5). A simple UNet model was built to create a baseline to compare with the performance of the other two ConvLSTM-based models. The three models were trained on a Nvidia GeForce RTX 2080 GPU using Keras framework (Tensorflow backend). The parameters for all the three models were set to be consistent; each model has a batch size of six and uses "Adam" as the optimizer. The loss function chosen was categorical cross-entropy, and the initial learning rate was set to *10^-4*, decreasing by a factor of 0.1 when validation loss stagnates for more than three epochs.

The three models were tested following two main relearning strategies: pixel-based relearning and OBV-based relearning. Experiments on each relearning strategies were set to follow three iterations to test to what extent this relearning strategy can improve the initial result. For each relearning strategy, OBV was applied to the relearned map after three iterations of relearning, in other words, OBV was deployed at the last step to provide a comparison with the iteratively relearned results.

In the process of segmenting an image into object-based segments, three parameters were determined: "scale," "shape," and "compactness." Since the scale parameter plays the most significant role regarding the effect of CPP, we kept "shape" and "compactness" constant but tested a wide range of "scale" parameters. "Shape" and "compactness" for all the approaches were set to 0.3 and 0.3, respectively. A wide range of scale parameters were tested with a linear ascending setting of S $\in \{10, 15, 20, 25, \ldots, 115,\ 120\}$. Among the scale parameters in set S, we selected a scale parameter that has the largest improvement on initial model prediction for all the OBV-related approaches of each dataset.

Regarding the evaluation of classification results, the performance of all the approaches were assessed by the overall accuracy (OA) and Kappa coefficient. Perclass accuracy was also assessed to examine the separability between classes, especially for the effects of relearning approaches on the improvement of classification accuracy in thematic classes.



Dataset 1		Number of labelled pixels farmland	formal	grassland	informal	impervious	roads	rural	soil	trees	water	Sum
2002	train	1,953,941	1,464,554	2,133,567	493,420	1,124,257	838,854	592,509	326,934	2,428,957	305,988	11,667,456
	valid	970,203	590,350	881,597	242,047	437,542	306,711	308,029	193,195	1,059,159	117,322	5,109,760
	test	281,019	164,751	252,110	74,744	116,997	78,328	85,746	55,638	300,997	30,630	1,441,792
2007	train	1,798,239	2,165,978	1,177,513	583,879	966,750	1,077,312	778,020	317,256	2,568,932	229,356	11,667,456
	valid	832,578	943,846	531,180	272,523	377,538	400,052	367,934	97,942	1,194,501	89,431	5,109,760
	test	245,015	260,470	142,189	83,143	106,122	111,832	103,382	23,121	341,215	24,906	1,441,792
2012	train	1,646,846	2,445,212	836,542	696,754	779,668	1,078,600	561,220	651,711	2,758,601	210,549	11,667,456
	valid	686,165	1,012,700	354,971	354,092	288,123	396,691	288,878	328,314	1,319,834	79,007	5,109,760
	test	195,626	291,569	95,143	104,959	80,607	101,363	85,627	92,880	371,839	21,897	1,441,792
2018	train	1,467,439	2,862,898	751,313	866,359	852,340	876,028	398,705	197,638	3,225,266	150,500	11,667,456
	valid	632,712	1,206,568	332,466	496,248	268,861	338,423	189,908	59,643	1,528,047	51,828	5,109,760
	test	177,787	344,122	93,423	146,291	70,684	89,192	47,466	10,985	446,153	15,078	1,441,792

Dataset 2		Number of labelled pixels farmland	formal	grassland	informal	impervious	roads	rural	soil	trees	water	Sum
2002	train	1,561,072	1,379,481	1,969,209	325,912	947,954	816,922	562,577	676,953	2,694,901	727,960	11,667,456
	valid	728,844	586,621	799,631	164,082	333,781	333,385	309,386	254,733	1,285,891	311,335	5,109,760
	test	206,369	167,736	225,627	44,928	88,839	96,325	97,411	75,699	356,957	81,725	1,441,792
2007	train	609,728	2,614,452	934,996	723,010	1,752,143	834,552	666,322	487,613	2,744,718	292,231	11,667,456
	valid	300,545	1,109,664	392,308	339,135	683,023	373,744	325,147	222,738	1,226,868	134,072	5,109,760
	test	87,459	312,084	117,710	98,216	193,567	99,103	95,141	63,154	333,127	41,693	1,441,792
2012	train	610,274	2,862,992	819,696	859,474	1,674,608	833,253	652,198	205,503	2,861,588	285,572	11,667,456
	valid	283,274	1,198,548	348,253	395,469	715,826	343,430	313,426	108,707	1,263,335	138,262	5,109,760
	test	81,683	338,476	93,651	114,875	208,717	87,945	90,453	29,781	351,112	44,777	1,441,792
2018	train	571,413	3,388,292	296,604	936,233	1,136,370	998,051	577,391	320,553	3,189,821	242,959	11,667,456
	valid	298,413	1,385,385	138,838	471,741	475,822	440,433	264,857	96,755	1,430,707	102,007	5,109,760
	test	83,737	400,339	35,838	142,018	129,316	112,950	74,427	25,816	405,094	30,987	1,441,792

Fig. 4. Data settings for two study areas. (a) VHR imagery of dataset I. (b) Settings of validation data (highlighted) of dataset I. (c) Ground truth labels of dataset I. (d) Settings of testing data (highlighted) of dataset I. (e) VHR imagery of dataset II. (f) Settings of validation data (highlighted) of dataset II. (g) Ground truth labels of dataset II. (h) Settings of testing data (highlighted) of dataset II.

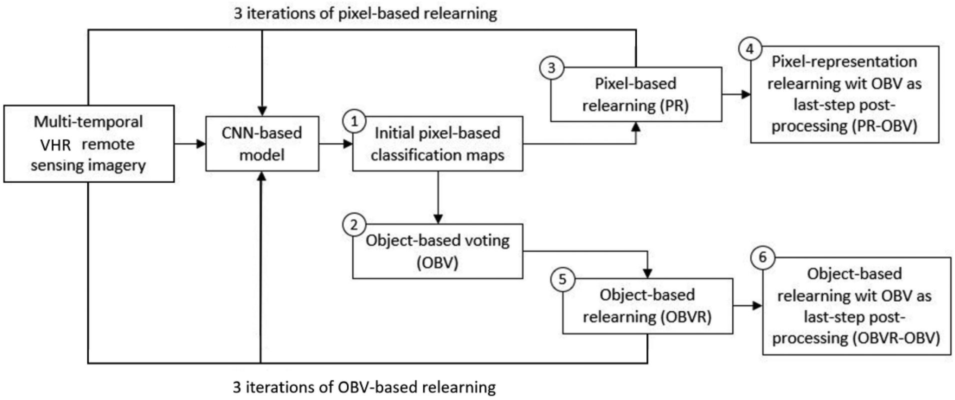

3 iterations of pixel-based relearning
Multi-temporal VHR remote sensing imagery
CNN-based model
1
Initial pixel-based classification maps
3
Pixel-based relearning (PR)
4
Pixel-representation relearning wit OBV as last-step post-processing (PR-OBV)
2
Object-based voting (OBV)
5
Object-based relearning (OBVR)
6
Object-based relearning wit OBV as last-step post-processing (OBVR-OBV)
3 iterations of OBV-based relearning

Fig. 5. Flowchart of the experiment setup. Multitemporal VHR images are first input in a CNN-based model for initial training to produce (1) initial pixel-based classification maps, which are (2) postprocessed by OBV for benchmarking with relearning approaches; alternatively, the initial pixel-based classification maps are aggregated with VHR images for (3) pixel-relearning; after three iterations of pixel relearning, the PR maps are processed by OBV to generate (4) PR-OBV; following a similar work flow, (5) OBVR maps are generated after three iterations of aggregating OBV maps with VHR images. Finally, (6) OBVR–OBV maps are produced by post-processing OBVR maps with OBV.

IV. Results and Discussion

In this section, we evaluate the overall performances of two relearning strategies applying on three different models. Moreover, we compare the pixel accuracies of each LULC class for different relearning methods. Visual observations were conducted for evaluating the temporal correlations between the classifications of each time steps.

A. *Analysis of Segmentation Scale*

The analysis of segmentation scale was conducted for each model in the two datasets. An optimal segmentation scale for each dataset varies from model to model. In dataset I, the optimal segmentation scales for UNet model, simple ConvLSTM model and UNet-ConvLSTM were 20, 85, and 70, respectively. In dataset II, the counterpart scales were 55, 100, and 60.

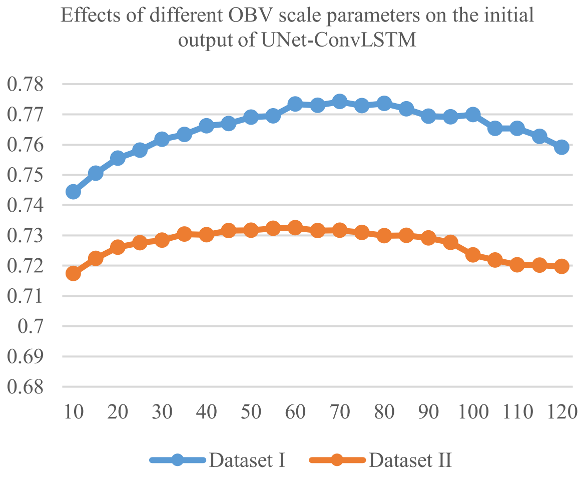


Fig. 6. Accuracy effects of OBV on the initial output of UNet-ConvLSTM according to different scale parameters.

Although the optimal segmentation scales of a model are different in the two datasets, accuracy effects with different scales of the same model share similar patterns. It can be observed that, in both datasets, UNets required relatively smaller segmentation scales to obtain an optimal improvement, whereas ConvLSTMs demand larger scales to achieve more accuracy improvement in OBV operation. As can be observed in Fig. 6, the accuracy effects were optimal when the scale was around 60 to 80, smaller or larger scale parameters receive less accuracy improvement.

B. *Overall Accuracies and* k *Statistics*

The initial training of three models followed by two relearning strategies generated results for nine approaches, then OBV was adopted as a last-step CPP method for each model to generate benchmarks for the two relearning strategies. In all, 18 different approaches for each dataset were evaluated and compared.

In general, as shown in Fig. 7, in both datasets I and II, UNet-ConvLSTM with relearning strategies achieved the highest accuracy, 79.1% in dataset I and 84.4% in dataset II. UNet with relearning strategies showed slightly less accuracy, the best accuracy of UNet achieved in two datasets were 79% and 77.4%. Whereas the classification results achieved by a simple ConvLSTM was much lower than the results achieved by the other two models.

Before using OBV as a last-step CPP, both of pixel-based relearning and OBV-relearning presented improvements on model performance. The accuracy of the initial training of all three models significantly increased after applying the two relearning strategies. In dataset I, OBV-relearning approaches showed higher kappa and OA values compared with initial training and pixel-based relearning approaches in all three models. More specifically, UNet-ConvLSTM with OBV-relearning achieved the best performance in terms of kappa and OA, which were 75.2% and 79.2%, respectively; they increased 9% and 5.8% compared with the initial training outcomes. However, for UNet-ConvLSTM in dataset II, the OBV-relearning result achieved the same accuracy with its pixel-relearning result, both were the best score of all the approaches in dataset II.

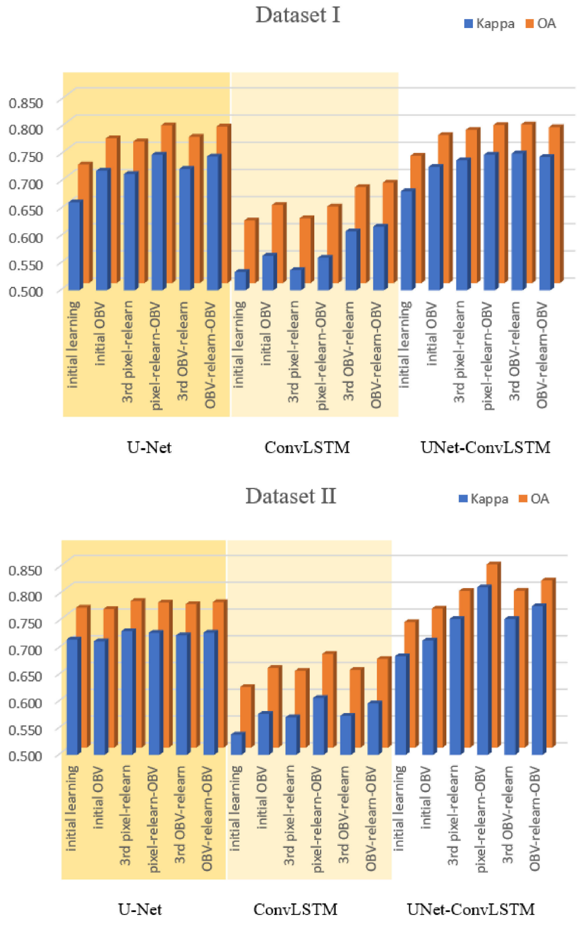


Fig. 7. Averaged Kappa and OA of 18 different approaches for the two test datasets.

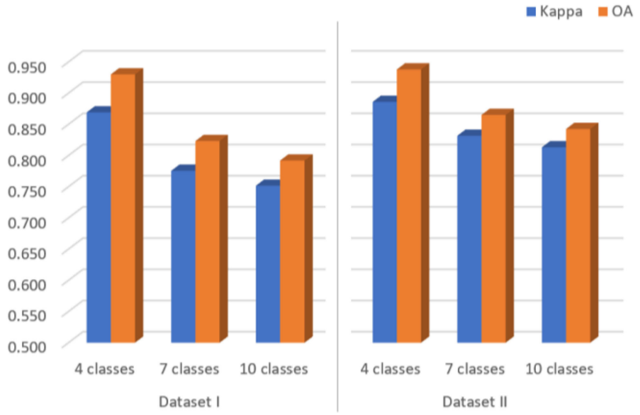


Fig. 8. Accuracy effects of UNet-ConvLSTM PR-OBV according to number of classes.

Furthermore, it was also observed that the two relearning strategies have better effects on a UNet-ConvLSTM than a UNet. In dataset I, the OA of UNet-ConvLSTM increased from 73.4% to 79.2%; the OA of UNet raised from 71.8% to

TABLE I
PRECLASS ACCURACY OF 18 DIFFERENT APPROACHES FOR DATASET I AND DATASET II

Dataset I.	Farmland	Formal settlements	Grassland	Informal settlements	Other impervious surface	Roads	Rural settlements	Soil	Trees	Water
U-Net initial learning	0.852	0.688	0.620	0.527	0.409	0.748	0.561	0.619	0.791	0.852
U-Net initial OBV	0.892	0.783	0.689	0.607	0.458	0.736	0.683	0.705	0.811	0.858
U-Net 3rd pixel-relearn	0.881	0.782	0.717	0.592	0.428	0.716	0.665	0.635	0.805	0.832
U-Net pixel-relearn-OBV	0.921	0.799	0.738	0.660	0.542	0.792	0.740	0.710	0.815	0.871
U-Net 3rd OBV-relearn	0.896	0.783	0.704	0.627	0.452	0.732	0.675	0.676	0.812	0.845
U-Net OBV-relearn-OBV	0.914	0.795	0.724	0.676	0.542	0.793	0.731	0.748	0.812	0.895
ConvLSTM initial learning	0.896	0.537	0.443	0.288	0.321	0.460	0.488	0.578	0.709	0.737
ConvLSTM initial OBV	0.928	0.496	0.589	0.125	0.437	0.587	0.648	0.653	0.713	0.828
ConvLSTM 3rd pixel-relearn	0.847	0.547	0.424	0.199	0.304	0.457	0.344	0.584	0.727	0.727
ConvLSTM pixel-relearn-OBV	0.884	0.527	0.501	0.019	0.330	0.522	0.513	0.632	0.720	0.786
ConvLSTM 3rd OBV-relearn	0.894	0.649	0.608	0.392	0.355	0.427	0.492	0.592	0.747	0.723
ConvLSTM OBV-relearn-OBV	0.896	0.630	0.673	0.405	0.393	0.438	0.605	0.622	0.735	0.810
UNet-ConvLSTM initial learning	0.900	0.718	0.565	0.558	0.485	0.731	0.665	0.653	0.815	0.886
UNet-ConvLSTM initial OBV	**0.922**	0.755	0.656	**0.696**	**0.616**	0.733	0.719	0.691	0.807	0.865
UNet-ConvLSTM 3rd pixel-relearn	0.917	**0.814**	0.692	0.614	0.498	0.783	0.742	0.692	0.816	0.897
UNet-ConvLSTM pixel-relearn-OBV	0.909	0.804	**0.745**	0.647	0.564	0.789	0.739	**0.746**	0.807	0.880
UNet-ConvLSTM 3rd OBV-relearn	0.920	0.789	0.726	0.685	0.502	**0.817**	**0.759**	0.707	**0.835**	**0.913**
UNet-ConvLSTM OBV-relearn-OBV	0.916	0.780	0.728	0.693	0.524	0.801	0.735	0.701	0.823	0.898

Dataset II.	Farmland	Formal settlements	Grassland	Informal settlements	Other impervious surface	Roads	Rural settlements	Soil	Trees	Water
U-Net initial learning	0.828	0.777	0.650	0.605	0.531	0.784	0.727	0.591	0.873	0.918
U-Net initial OBV	0.842	0.766	0.661	0.595	0.540	0.776	0.707	0.649	0.858	0.893
U-Net 3rd pixel-relearn	0.855	0.810	0.665	0.603	0.577	0.750	0.707	0.572	0.862	0.893
U-Net pixel-relearn-OBV	0.863	0.793	0.679	0.578	0.597	0.760	0.696	0.548	0.863	0.881
U-Net 3rd OBV-relearn	0.850	0.791	0.677	0.613	0.561	0.763	0.698	0.550	0.859	0.877
U-Net OBV-relearn-OBV	0.872	0.800	0.656	0.631	0.574	0.777	0.709	0.579	0.858	0.874
ConvLSTM initial learning	0.847	0.648	0.379	0.351	0.385	0.477	0.538	0.442	0.790	0.802
ConvLSTM initial OBV	0.940	0.631	0.440	0.621	0.450	0.519	0.650	0.524	0.775	0.852
ConvLSTM 3rd pixel-relearn	0.899	0.676	0.440	0.373	0.372	0.622	0.685	0.464	0.777	0.882
ConvLSTM pixel-relearn-OBV	0.972	0.672	0.679	0.446	0.429	0.694	0.766	0.549	0.771	0.894
ConvLSTM 3rd OBV-relearn	0.831	0.687	0.386	0.348	0.386	0.584	0.694	0.473	0.777	0.884
ConvLSTM OBV-relearn-OBV	0.913	0.671	0.413	0.408	0.438	0.623	0.734	0.529	0.771	0.882
UNet-ConvLSTM initial learning	0.944	0.808	0.489	0.604	0.490	0.760	0.734	0.548	0.848	0.890
UNet-ConvLSTM initial OBV	**0.964**	0.828	0.574	0.699	0.519	0.775	0.742	0.599	0.842	0.886
UNet-ConvLSTM 3rd pixel-relearn	0.900	0.857	0.646	0.664	0.575	0.817	0.764	0.573	0.871	**0.931**
UNet-ConvLSTM pixel-relearn-OBV	0.922	**0.885**	**0.772**	**0.723**	**0.711**	**0.851**	**0.797**	**0.682**	**0.889**	0.921
UNet-ConvLSTM 3rd OBV-relearn	0.933	0.842	0.645	0.709	0.576	0.814	0.766	0.572	0.871	0.911
UNet-ConvLSTM OBV-relearn-OBV	0.946	0.855	0.734	0.704	0.635	0.828	0.768	0.636	0.870	0.904

76.9%. The improvements were also significant in dataset II: the OA of UNet-ConvLSTM climbed from 73.5% to 79.3%; for UNet, it increased from 76.2% to 77.4%. It is also interesting to observe that although in dataset II the UNet had better initial prediction (76.2%) than UNet-ConvLSTM (73.5%), the better effect of relearning strategies on the later allowed it to surpass the relearned prediction of the former.

When using OBV as a last-step CPP, the rates of improvement were the most significant when applying on the results of initial training and pixel-based relearning. After applying OBV as last-step CPP on UNet-ConvLSTM, the accuracy of pixel-relearned classification showed more considerable improvement than the accuracy of OBV-relearned classification. In dataset II, the OA of pixel-relearned UNet-ConvLSTM improved from 79.3% to 84.3%, whereas the OA of OBV-relearned UNet-ConvLSTM only improved from 79.3% to 81.3%. Such effect also could be seen in dataset I; this phenomenon is very likely due to the effect of OBV has already been functional during the process of OBV-relearning; therefore, its effect became less significant

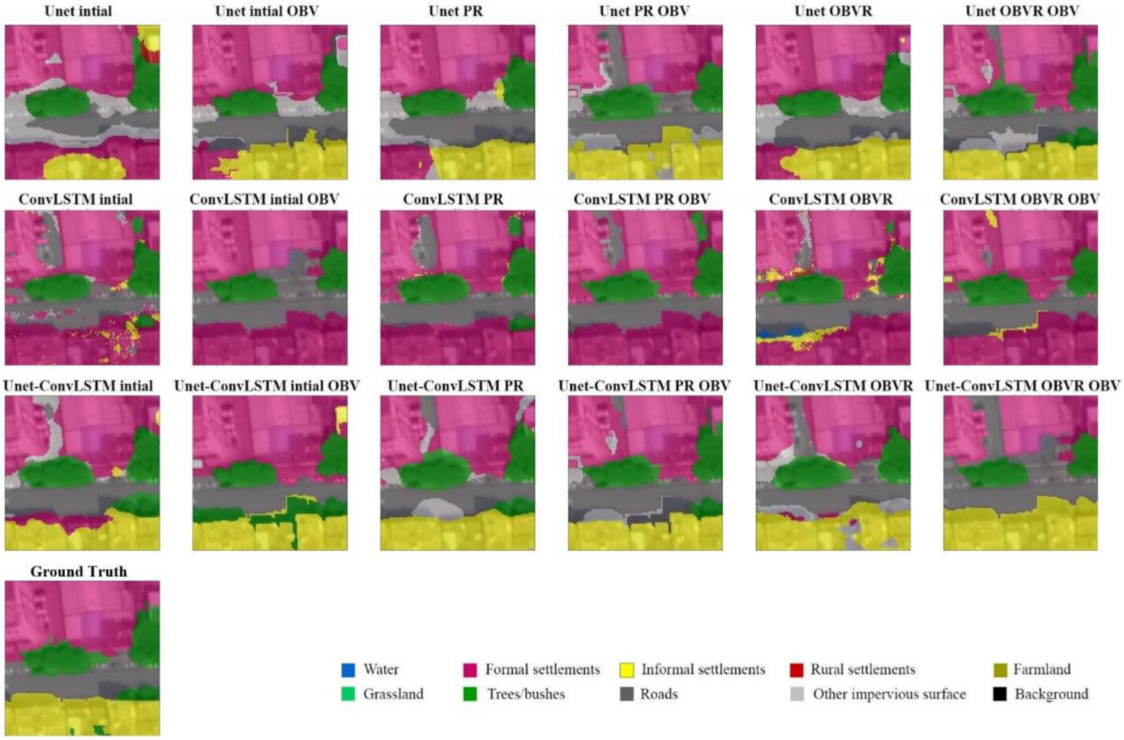


Fig. 9. Comparisons of year 2012 classification maps generated by 18 difference approaches in dataset I.

when using OBV as a last-step CPP on the OBV-relearned result.

However, it is also evident from the results of both dataset I and II, using OBV as last-step CPP did not bring improvements on the classification result of all the approaches, a few exceptional test results showed a decrease in the classification accuracy. For instance, the OA of the OBV-relearned UNet-ConvLSTM in dataset I dropped 0.2% after applying OBV CPP; also, the OA of the pixel-relearned UNet in dataset II decreased 0.3% after using OBV CPP.

In addition, the accuracy effects of UNet-ConvLSTM PR-OBV according to number of classes have been studied (Fig. 8). Besides the 10-class LULC categorization deployed in the main experiments, a 4-class LULC category and a 7-class LULC category were adopted for comparison. The 4-class categorization consists of 1) vegetation, 2) urban extent, 3) soil, and 4) water, and the 7-class categorization contains 1) farmland and grassland, 2) formal settlements, 3) informal settlements (including rural settlements), 4) other impervious surface (including roads), 5) soil, 6) trees, and 7) water. As can be seen from Fig. 8, in both dataset I and II, a smaller number of classes, that need to be distinguished, allows obtaining a higher level of accuracy.

C. Perclass Accuracies

As shown in Table I, in both dataset I and dataset II, all the top scores in each LULC class were distributed in UNet-ConvLSTM approaches. To be more specific, in dataset I, the best scores were scattered in various relearning strategies of UNet-ConvLSTM. The OBV-relearned UNet-ConvLSTM (UC-OBVR) captured more best scores than others. It can also be seen that the performance of the pixel-relearned UNet-ConvLSTM (UC-PR) was just slightly better than the performance of the one adding OBV as CPP (UC-PR-OBV). Whereas, in dataset II, most of the best scores of LULC classes were achieved by UC-PR-OBV.

The results also show that simple ConvLSTM approaches failed to recognize “informal settlements,” as well as showed the poor performance of classifying “rural settlements” and “other impervious surface.” These poor performances contributed to the lowest OA comparing with the other two models.

The class-based accuracy reflected that thematic classes have lower class separability than others. For example, “other impervious surface,” having almost same spectral value with “roads,” received the lowest accuracy in all the three models due to the separability of the class is mainly based on semantic information. Among all the LULC classes, relearning strategies had the most significant improvement of accuracy in “grassland” and “impervious surface.” The pixel accuracy of “grassland” was improved from 56.5% to 74.5% by the pixel-relearned UNet-ConvLSTM with OBV as CPP (UC-PR-OBV) in dataset I and improved from 48.9% to 77.2% by UC-PR-OBV in dataset II. Meanwhile, the accuracy of “impervious surface” was improved from 49% to 71.1% by UC-PR-OBV in dataset II.

D. Visual Observation

In Figs. 9 and 18 approaches of the year 2012 classification maps of dataset I were compared. It is evident that all the simple ConvLSTM approaches showed poor performance in terms of recognizing “informal settlements,” whereas UNet

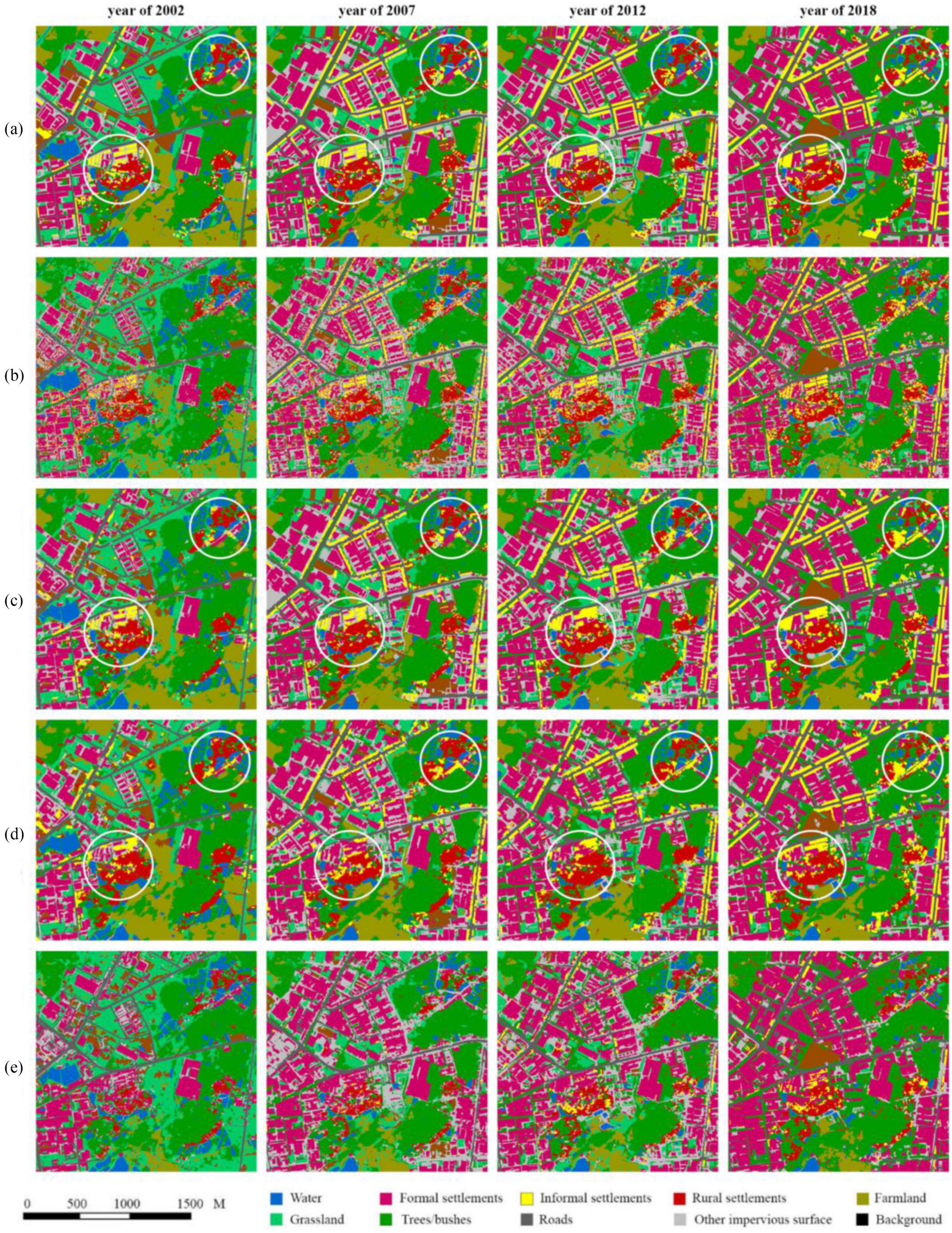


Fig. 10. Comparisons between PR-OBV-based approaches and ground truth images in dataset II. (a) Ground truth. (b) UNet-ConvLSTM initial training. (c) UNet-ConvLSTM-PR-OBV. (d) UNet-PR-OBV. (e) ConvLSTM-PR-OBV.

approaches and UNet-ConvLSTM approaches showed satisfactory results. This is very likely due to a ConvLSTM structure, which simply consist of a stack of ConvLSTM layers, does not have the strength of encoder-decoder structures in terms of extracting high-level spatial context features. Similarly, it can be argued that the encoder-decoder structure of UNet-ConvLSTM-based approaches is able to complement the disadvantage of a ConvLSTM model in terms of extracting high-level representation. As such, it is observed that the classification maps of UNet-ConvLSTM-based approaches achieved good performance in segmenting thematic classes.

Comparing UNet-ConvLSTM approaches and UNet approaches, the result of initial training of UNet-ConvLSTM presented noises in the classification labels, while the result of the initial training of UNet shows the effect of blurring boundaries. After three iterations of relearning, the OBV-relearned and pixel-relearned results showed similar visual representations. Both issues of the salt-and-pepper effects and blurring boundaries were considerably alleviated, especially in the results of UNet-ConvLSTM. However, not all the notable changes after relearning brought improvements to the accuracy of classification maps. For instance, a larger scale of OBV merged small segments into big parcels, although this process was helpful for mitigating salt-and-pepper effects, it resulted in wrongly classified pixels (e.g., the result of UNet-ConvLSTM initial OBV mistakenly classified parts of roads into trees and bushes).

As mentioned in the previous sections, the approaches of pixel relearning with OBV as CPP (PR-OBV) in dataset II achieved the best scores in both OA and preclass accuracy. In Fig. 10, obtained temporal sequence by PR-OBV-based approaches in dataset II are compared with the corresponding ground truth. First, temporal consistency regarding LULC changes are obvious in the ground truth maps, see Fig. 10(a). Furthermore, the results show that UNet-ConvLSTM-PR-OBV successfully captured such temporal consistency, see Fig. 10(c). In particular, the "informal settlements" in the classification maps of UNet-ConvLSTM grows in a consistent manner from the year 2002 to 2018. However, since UNet approaches did not take temporal relationship into account, the four time steps of classification maps generated by UNet-based approaches show inconsistencies in the changes of many LULC classes. For instance, the patterns of "informal settlements" obtained by UNet changed into rural settlements then changed back to informal settlements [Fig. 10(d)]. In contrast, the UNet-ConvLSTM approaches that designed for spatial-temporal segmentation tasks showed better performance regarding reflecting temporal consistency. Since the capability of reflecting temporal consistency in classification results is not only useful in improving classification accuracy, but can be valuable in terms of change detection and trend analysis. Such clear benefit of the UNet-ConvLSTM relearning approaches should not be ignored.

E. Time Consumption of Relearning

The time consumption of pixel-based relearning and OBV-based relearning of three models (i.e., UNet, ConvLSTM, and UNet-ConvLSTM) was compared in Fig. 11. As mentioned, all the training operations were carried on a Nvidia GeForce RTX 2080 GPU using Keras framework (Tensorflow backend). Except the initial training of three models, the relearning operations were conducted for three iterations in the experiments. Since operating OBV is a manual process that does not involve model training, the time consumption of OBV operation is not included in this comparison. In total, six relearning approaches were compared with a breakdown of time consumed in each phase.

It can be observed that UNet-based approaches tend to demand less training time to complete three iterations of relearning (less than three hours), while UNet-ConvLSTM-based approaches required almost double the training time (less than 6 h). This is very likely due to the recurrent model structure in ConvLSTM-based models containing more learnable parameters and therefore demanding more time for data processing. It should be noted that, for the UNet-ConvLSTM-based methods in both datasets, the pixel-based relearning approach is more efficient than OBV-based relearning. By and large, the extra training time of the proposed method appears to internalize a good tradeoff given the achievement of a higher OA.

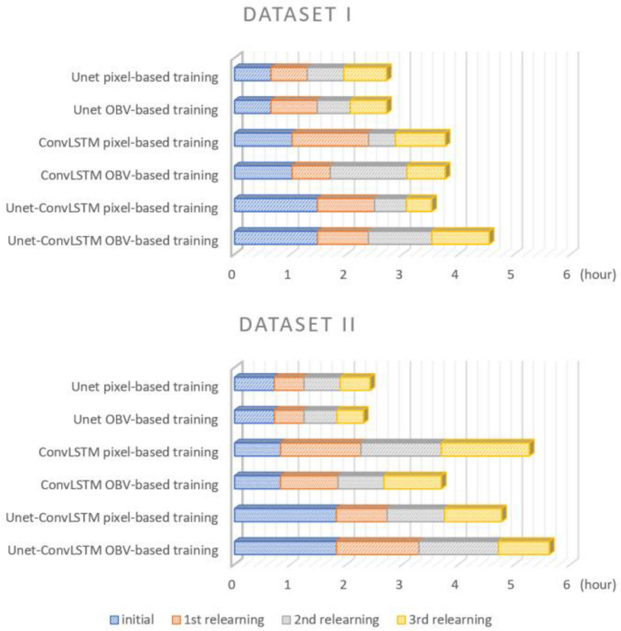


Fig. 11. Training time consumption of different relearning approaches in dataset I and dataset II.

V. Conclusion

The main purpose of this article is to explore the extent to which a hybrid multitemporal relearning method can improve the accuracy of LULC classification. In this article, three CNN models with two main relearning strategies, pixel-based relearning and OBV relearning, were tested and analyzed. OBV was also deployed after each relearning approach as a last-step CPP for benchmarking. The classification tasks are set to be complex and challenging in order to test a wide range of capabilities of different relearning approaches. The main findings are as follows:

1) UNet-ConvLSTM outperforms UNet and the simple ConvLSTM by achieving higher classification accuracy, as well as reflecting temporal consistency in multitemporal LULC classification. It is because a UNet-ConvLSTM takes advantages of encoder–decoder structures, as well as exploits the temporal dependency embedded in the multitemporal data.
2) Both pixel-representation relearning and OBR can improve classification accuracy. When conducting OBR, the process of selecting the most effective segmentation scale is crucial, too large scales can result in a failure of preserving small objects in the classification maps. It should be noted that although OBV relearning could be more effective in alleviating salt-and-pepper noise, conducting OBV scale selection is a time-consuming process. Since PR-OBV achieved similar, or even better, accuracy in this article, an integration of multitemporal segmentation model with PR-OBV proves to be an efficient method.

3) The performance of pixel-relearning and OBV-relearning can vary from model to model. Based on the test results in this article, these two learning strategies can achieve similar accuracy improvements. However, adding OBV as a last-step CPP can boost the accuracy of pixel-relearning approaches to a large extent than the OBV-relearning approaches.
4) It is true that the optimal OA of classification maps achieved in this article is not eminently high (e.g., over 90%). This is mostly due to the complex thematic LULC class category in this study. Compared with formal settlements, the difference between the building morphologies of informal settlements and rural settlement remains very subtle. Furthermore, the informal settlements in the study area have tiny plot sizes and mostly scattering inside rural settlements. These factors inevitably limited the optimization of classification accuracy; however, these features are not unique to this case study but reflect the reality of many rapidly urbanizing cities around the world.
5) During the experiments, we observed that the accuracy levels of the classification results have increased with an increasing number of labeled samples (i.e., number of the training patches). Moreover, considering that certain thematic classes generally have lower class separability than others and that training sets can be imbalanced, data augmentation techniques could be specifically applied on certain thematic classes to eventually enhance the discriminative properties of classifiers generally. Consequently, further improvements regarding the accuracy levels can be expected when integrating larger training sets and tailored augmentation procedures.

In summary, this article shows that a combination of spatial–temporal models and corresponding suitable relearning strategies can produce very promising LULC classification maps from VHR satellite imagery, even for highly complex classification tasks with limited training data. Assuming a longer temporal sequence could better reflect features of temporal dependency, further research could focus on testing and developing relearning methods on very long temporal sequential data for optimizing multitemporal LULC classification.

Acknowledgment

The authors would wish to gratefully acknowledge the funding support from the EPSRC Innovation and Knowledge Centre for Smart Infrastructure and Construction - Collaborative Programme Tranche 2 (EP/L010917/1). Additionally, Jin would like to acknowledge the support of the ‘Urban spatial modelling and decision-support system’ project of the Beijing Innovation Centre for Future Urban Design, and the EPSRC Managing Air Green Inner Cities (MAGIC) Grant (EP/N010221/1).

References

[1] C. Zhang *et al.*, “An object-based convolutional neural network (OCNN) for urban land use classification,” *Remote Sens. Environ.*, vol. 216, pp. 57–70, Oct. 2018.

[2] M. Li, A. Stein, W. Bijker, and Q. Zhan, “Urban land use extraction from very high resolution remote sensing imagery using a bayesian network,” *ISPRS J. Photogramm. Remote Sens.*, vol. 122, pp. 192–205, Dec. 2016.

[3] X. Zhang, S. Du, and Q. Wang, “Hierarchical semantic cognition for urban functional zones with VHR satellite images and POI data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 132, pp. 170–184, Oct. 2017.

[4] X. Zhang, S. Du, and Z. Zheng, “Heuristic sample learning for complex urban scenes: Application to urban functional-zone mapping with VHR images and POI data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 1–12, Mar. 2020.

[5] “Housing for all by 2030,” World Bank, 2016. Accessed: Sep. 17, 2020. [Online]. Available: https://www.worldbank.org/en/news/infographic/2016/05/13/housing-for-all-by-2030

[6] World urbanization prospects: The 2018 revision, United Nations, 2019.

[7] C. Geiß, P. Aravena Pelizari, L. Blickensdörfer, and H. Taubenböck, “Virtual support vector machines with self-learning strategy for classification of multispectral remote sensing imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 42–58, May 2019.

[8] S. W. Myint, P. Gober, A. Brazel, S. Grossman-Clarke, and Q. Weng, “Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery,” *Remote Sens. Environ.*, vol. 115, no. 5, pp. 1145–1161, 2011.

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

[10] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019.

[11] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, “Towards better exploiting convolutional neural networks for remote sensing scene classification,” *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[13] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[14] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, “Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 680–688.

[15] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[16] A. Sharma, X. Liu, X. Yang, and D. Shi, “A patch-based convolutional neural network for remote sensing image classification,” *Neural Netw.*, vol. 95, pp. 19–28, Nov. 2017.

[17] S. Liu and Q. Shi, “Local climate zone mapping as remote sensing scene classification using deep learning: A case study of metropolitan china,” *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 229–242, Jun. 2020.

[18] S. Liu, Z. Qi, X. Li, and A. G.-O. Yeh, “Integration of convolutional neural networks and object-based post-classification refinement for land use and land cover mapping with optical and SAR data,” *Remote Sens.*, vol. 11, no. 6, Jan. 2019, Art. no. 6.

[19] R. Alshehhi, P. R. Marpu, W. L. Woon, and M. D. Mura, “Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 139–149, Aug. 2017.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” Nov. 2014. Accessed: Jun. 12, 2020. [Online]. Available: https://arxiv.org/abs/1411.4038v2

[21] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

[22] X. Yu, X. Wu, C. Luo, and P. Ren, “Deep learning in remote sensing scene classification: A data augmentation enhanced convolutional neural network framework,” *GISci. Remote Sens.*, vol. 54, no. 5, pp. 741–758, Sep. 2017.

[23] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, “Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 59–69, Apr. 2019.

[24] C. Geiß and H. Taubenböck, “Object-based postclassification relearning,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2336–2340, Nov. 2015.

[25] T. Blaschke, “Object based image analysis for remote sensing,” *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 2, pp. 2–16, 2010.

[26] C. Geiß, M. Klotz, A. Schmitt, and H. Taubenböck, “Object-based morphological profiles for classification of remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5952–5963, Oct. 2016.

[27] R. Kemker, C. Salvaggio, and C. Kanan, “Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 60–77, Nov. 2018.

[28] X. Huang, Q. Lu, L. Zhang, and A. Plaza, “New postprocessing methods for remote sensing image classification: A systematic study,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7140–7159, Nov. 2014.

[29] S. Martinis, A. Twele, and S. Voigt, “Unsupervised extraction of flood-induced backscatter changes in SAR data using markov image modeling on irregular graphs,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 251–263, Jan. 2011.

[30] X. Han, X. Huang, J. Li, Y. Li, M. Y. Yang, and J. Gong, “The edge-preservation multi-classifier relearning framework for the classification of high-resolution remotely sensed imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 57–73, Apr. 2018.

[31] Q. Shi, X. Liu, and X. Huang, “An active relearning framework for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3468–3486, Jun. 2018.

[32] G. Lei *et al.*, “OIC-MCE: A practical land cover mapping approach for limited samples based on multiple classifier ensemble and iterative classification,” *Remote Sens.*, vol. 12, no. 6, Jan. 2020, Art. no. 6.

[33] A. Sharma, X. Liu, and X. Yang, “Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks,” *Neural Netw.*, vol. 105, pp. 346–355, Sep. 2018.

[34] F. Vuolo, M. Neuwirth, M. Immitzer, C. Atzberger, and W.-T. Ng, “How much does multi-temporal sentinel-2 data improve crop type classification?,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 72, pp. 122–130, Oct. 2018.

[35] L. Mou, L. Bruzzone, and X. X. Zhu, “Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.

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

[37] F. Milletari, N. Rieke, M. Baust, M. Esposito, and N. Navab, “CFCM: Segmentation via coarse to fine context memory,” in *Proc. Med. Image Comput. Comput. Assist. Intervention*, 2018, pp. 667–674.

[38] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, “Bi-Directional ConvLSTM U-Net with densley connected convolutions,” Aug. 2019. Accessed: Sep. 09, 2020. [Online]. Available: http://arxiv.org/abs/1909.00166

[39] A.-J. Gallego, P. Gil, A. Pertusa, and R. B. Fisher, “Semantic segmentation of SLAR imagery with convolutional LSTM selectional autoencoders,” *Remote Sens.*, vol. 11, no. 12, Jan. 2019, Art. no. 12.

[40] N. Teimouri, M. Dyrmann, and R. N. Jørgensen, “A novel spatio-temporal FCN-LSTM network for recognizing various crop types using multi-temporal radar images,” *Remote Sens.*, vol. 11, no. 8, Jan. 2019, Art. no. 8.

[41] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” May 2015. Accessed: Mar. 26, 2019. [Online]. Available: http://arxiv.org/abs/1505.04597

[42] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: Redesigning skip connections to exploit multiscale features in image segmentation,” Jan. 2020. Accessed: Dec. 21, 2020. [Online]. Available: http://arxiv.org/abs/1912.05074

[43] B. Cui, X. Chen, and Y. Lu, “Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection,” *IEEE Access*, vol. 8, pp. 116744–116755, 2020.

[44] H. Hosseinpoor and F. Samadzadegan, “Convolutional neural network for building extraction from high-resolution remote sensing images,” in *Proc. Int. Conf. Mach. Vis. Image Process.*, Feb. 2020, pp. 1–5.

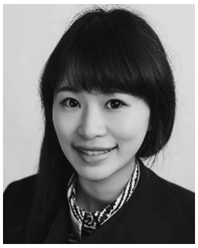
[45] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. WOO, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[46] P. Hao, P. Hooimeijer, R. Sliuzas, and S. Geertman, “What drives the spatial development of urban villages in China?,” *Urban Stud.*, vol. 50, no. 16, pp. 3394–3411, May 2013.

[47] M. van Oostrum, “Urbanizing villages: Informal morphologies in shenzhen’s urban periphery,” *J. Urban Des.*, vol. 23, no. 5, pp. 732–748, Sep. 2018.

[48] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenböck, “Satellite-Based mapping of urban poverty with transfer-learned slum morphologies,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5251–5263, 2020.

[49] C. Geiß, P. A. Pelizari, H. Schrade, A. Brenning, and H. Taubenböck, “On the effect of spatially non-disjoint training and test samples on estimated model generalization capabilities in supervised classification with spatial features,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2008–2012, Nov. 2017.



Yue Zhu received the M.Eng. degree in industrial design engineering from Tongji University, Shanghai, China, in 2014 and the M.Sc. degree in emergent technologies and design from Architectural Association, London, U.K., in 2016. She is currently working toward the Ph.D. degree in architecture with the University of Cambridge, Cambridge, U.K.

Her research interests include developing machine learning methods for the investigation of spatial temporal evolution of land use patterns.

Christian Geiß (Member, IEEE) received the M.Sc. degree in applied geoinformatics from the Paris Lodron University of Salzburg, Salzburg, Austria, in 2010 and the Ph.D. degree (Dr. rer. nat.) in geography from the Humboldt University of Berlin, Berlin, Germany, in 2014.

Since 2010, he has been with the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Cologne, Germany. In 2017, he was also at the Cambridge University Centre for Risk in the Built Environment (CURBE), University of Cambridge as a Visiting Scholar. He is currently working toward a Habilitation project in geography with the Julius-Maximilians University of Wurzburg, Wurzburg, Germany, entitled Collective Sensing Techniques and Artificial Intelligence for the Natural Hazard Risk and Impact Assessment. His research interests include the development of machine learning methods for the interpretation of earth observation data, multimodal remote sensing of the built environment, exposure and vulnerability assessment in the context of natural hazards, as well as techniques for automated damage assessment after natural disasters.

Emily So is currently a Reader of architectural engineering with the University of Cambridge, Cambridge, U.K., and the Director of the Cambridge University Centre for Risk in the Built Environment. Her research interests include estimating the dead and injured and proposes ways of improving data collection and modeling techniques.

Dr. So is a Chartered Civil Engineer with specialist experience in loss assessments earthquake engineering designs. As an expert in the field of casualty estimation in earthquakes, she is on the UK Scientific Advisory Group for Emergencies (SAGE).

Ying Jin is currently a University Reader with the Department of Architecture, University of Cambridge, Cambridge, U.K. He lectures on city planning, urban design, and urban modeling. He is particularly interested in understanding how technology, policy, and human behavior affect the development of cities and their infrastructure, and in using this knowledge to create new design solutions.

Mr. Jin is a Fellow of Robinson College, Cambridge.