Land Cover Change Detection Based on Vector Polygons and Deep Learning With High-Resolution Remote Sensing Images

Hui Zhang, Wei Liu, Hao Niu, Pengcheng Yin, Shiling Dong, Jialin Wu, Erzhu Li, Lianpeng Zhang, and Changming Zhu

***Abstract*— Vector polygons are valuable survey data, serving as crucial outputs of national geographical censuses and a fundamental data source for detecting changes in geographical conditions. Current remote sensing image change detection methods rely on comparing images but overlook abundant historical vector results, struggle with model generalization, and lack adequate samples. Consequently, change detection remains a manual process primarily, unable to meet the requirements for automated and efficient monitoring of standardized geographical conditions. Hence, this article proposes a change detection method for land cover vector polygons based on high-resolution remote sensing images and deep learning. Initially, the enhanced simple linear iterative clustering (SLIC) algorithm is applied to segment dual-temporal images from identical regions. Subsequently, an annotated dataset is generated using a multiscale extraction, cropping-with-inpainting approach. Next, datasets derived from pretemporal and posttemporal images are used for training and testing, respectively, and the training set is purified by using two-classifier cross-validation. Finally, an improved object-oriented convolutional neural network (CNN) model performs fine-grained scene classification. The change rules and postprocessing method are then integrated to identify changed vector polygons. To validate the effectiveness and superiority of the proposed method, we conducted experiments on land cover change detection using datasets from two study areas. The results indicate that the proposed method achieves precision and recall rates of 91.89% and 94.44% on Dataset 1, respectively. Similarly, in Dataset 2, the precision and recall rates reach 87.59% and 91.41%, respectively. These findings demonstrate the method's efficacy in detecting changed vector polygons, reducing manual intervention, and enhancing detection efficiency.**

Manuscript received 25 September 2023; revised 5 November 2023 and 3 December 2023; accepted 21 December 2023. Date of publication 25 December 2023; date of current version 23 January 2024. This work was supported in part by the Key Laboratory of Land Satellite Remote Sensing Application, Ministry of Natural Resources of the People's Republic of China, under Grant KLSMNR-K202206; in part by the National Key Research and Development Program of China under Grant 2023YFE0103800; and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Normal University under Grant 2022XKT0061 and Grant 2022XKT0060. *(Corresponding author: Wei Liu.)*

Hui Zhang, Wei Liu, Hao Niu, Jialin Wu, Erzhu Li, Lianpeng Zhang, and Changming Zhu are with the School of Geography, Geomatics, and Planning, Jiangsu Normal University, Xuzhou 221116, China (e-mail: zh694522220@126.com; liuw@jsnu.edu.cn; haoniu@jsnu.edu.cn; q395384541@163.com; lierzhu2008@126.com; zhanglp2000@126.com; zhuchangming@jsnu.edu.cn).

Pengcheng Yin and Shiling Dong are with the Bureau of Natural Resources and Planning, Xuzhou, Jiangsu 221006, China (e-mail: cumtyingpc@163.com; 391785773@qq.com).

Digital Object Identifier 10.1109/TGRS.2023.3346968

***Index Terms*— Change detection, deep learning, remote sensing, vector polygons.**

I. Introduction

With the steady increase in the global population and the rapid progress of the social economy, there is a persistent alteration in land surface types and land use patterns. Land cover change detection through remote sensing imagery involves comparing images of a specific region over different temporal periods to identify alterations in land cover [1]. High-resolution remote sensing imagery enables the monitoring of large-scale land changes over time [2]. Consequently, land cover change detection has emerged as a fundamental task within the remote sensing discipline, finding extensive applications in land surveying [3], [4], urban research [5], [6], [7], ecosystem monitoring [8], [9], [10], disaster detection [11], [12], and others.

Since the development of change detection methods using remote sensing images, there has been significant research and application in this area. As a result, various change detection methods have been derived. From an analytical-unit perspective, change detection can be broadly categorized into pixel-based and object-oriented methods [13]. The traditional pixel-based methods usually analyze the spectral differences of pixels directly through arithmetical operations, such as subtraction or division on a pixel-by-pixel basis, and then apply threshold method and cluster analysis to extract the variable pixels. However, this approach often overlooks contextual information and is susceptible to "salt and pepper" noise [14]. On the other hand, object-oriented methods integrate spectral, texture, and structural information of pixels in the vicinity, offering noise-resistant solid capabilities. Consequently, object-oriented methods exhibit certain advantages when dealing with high-resolution images with complex features and high redundancy [15].

Feature extraction plays a crucial role in object-oriented change detection. Traditional methods for feature extraction encounter challenges in constructing features and adapting models. In recent years, deep learning approaches have significantly enhanced the accuracy of interpreting remote sensing images by automatically extracting ground features and modeling. This advancement opens more possibilities for object-level

change detection methods [16]. For instance, Zhang et al. [17] proposed a deep learning-based framework for object-level change detection using a detector guided by dual correlation attention. Shu et al. [18] introduced a dual-perspective change contextual network that improves the extraction process of changing features by fusing dual-temporal features and employing context modeling to enhance the integrity of changing objects. Although deep learning has great potential in image analysis, its application in change detection faces several challenges due to the inherent complexity of high-resolution remote sensing images, such as the diversity of ground objects, image noise, and seasonal variations, and the dependence of deep learning models on large numbers of high-quality training samples. Specifically, these challenges include, but are not limited to, accurate boundary detection, data imbalances, high labeling costs, and significant demands on computing resources. Therefore, most of the current change detection methods based on deep learning are difficult to apply to practical tasks directly.

The selection of a data source plays a crucial role in change detection. While most existing research focuses on using image data, there is a scarcity of studies that combine vector and image data for change detection methods. Vector data, which contain boundary and category information of ground objects, are a valuable data source in various applications such as land surveys. Unlike image data, it provides essential assistance for tasks such as image segmentation and ground object recognition [19]. Feng et al. [20] employed a category vector map to segment images at multiple scales. They analyzed the change possibilities using the results of coarse-to-fine segmentation and pixel preclassification. Finally, they utilized the rotation forest classifier for classification and determined the change area by applying the majority voting rule. However, this method failed to identify the changed direction or determine the type of altered ground objects. Zhang et al. [21] proposed a change detection method based on vector data and the isolation forest algorithm. This method avoids errors caused by the mean value introduced in traditional index methods and provides a more detailed description of local changes. However, its effectiveness relies on the assumption that the proportion of various ground object change maps is small. Wei et al. [22] replaced historical remote sensing images with vector data. They introduced a spatial outlier index of texture elements based on locally accessible density to detect abnormal samples and changed objects in recent images automatically. However, the efficiency and accuracy of this approach may be compromised when detecting alterations in high-rise buildings within urban areas. In conclusion, the theory and technology behind change detection methods integrating vector and image data still need to be fully mature, demanding further research.

In order to address the issues above, this study proposes a novel method for detecting changed vector polygons using high-resolution remote sensing imagery and deep learning. The key contributions of this research are outlined as follows.

1) A framework for detecting changes in vector polygons based on high-resolution remote sensing imagery and deep learning is proposed. The framework enables an end-to-end application from preprocessed imagery to change detection, requiring only dual-temporal remote sensing images and the land cover vector data corresponding to the former temporal image. This method offers a comprehensive bottom-up solution.
2) In dataset construction, a strategy is introduced for building a sample library based on superpixel segmentation and vector polygons. Initially, an enhanced SLIC algorithm, which integrates both vector polygon constraints and texture features, is employed for improved segmentation. Capitalizing on these superpixels and corresponding land cover vector data, a multiscale sample cropping and patching scheme is designed. Supported by the DeepFill v2 model and an automatic purifier, a large quantity of high-quality samples can be harvested.
3) To enhance the accuracy of remote sensing scene classification, we employ an improved object-oriented CNN (IOCNN) model that classifies the superpixel cropping units. By integrating both first- and second-order features, a more discriminative feature representation is achieved, leading to enhanced model efficiency and classification accuracy. Furthermore, a change decision-maker is introduced, incorporating various change rules. This decision-maker compares and postprocesses the sample prediction results with land cover vector data, enabling the effective extraction of changed vector polygons.

II. Related Work

The change detection framework presented in this article is established on the foundations of deep learning technologies, superpixel segmentation, and prior-assisted vector polygons. In this section, we will explore recent works that relate to these three aspects.

A. Deep Learning-Based Change Detection

In recent years, deep learning technology has demonstrated significant proficiency in feature extraction and representation, thereby revolutionizing the field of remote sensing image analysis. This technological advancement has found extensive applications in areas such as land cover extraction, target detection, data fusion, and change detection. Specifically, deep learning-based multiclass change detection methods can be categorized into two primary types: direct classification and postclassification comparison. Unlike binary change detection, which solely determines the presence or absence of changes, multiclass change detection offers granular insights into specific "from-to" change types [23].

The direct classification approach aims to identify both the region and category of changes using an end-to-end neural network architecture. Two principal types of such end-to-end change detection networks exist: early fusion and late fusion [24]. Early fusion architectures integrate bitemporal images as multichannel inputs, tailored to fit semantic segmentation networks. Prevalent architectures such as fully convolutional networks (FCNs) [25] and U-Net [26] are not only expedient for semantic segmentation tasks but also

well-suited for change detection due to their encoder–decoder configurations. Numerous enhanced U-Net variants, such as MSCDUNet [27], M-UNet [28], LWCDNet [29], and CLNet [30], have been developed for change detection and have demonstrated commendable performance. In contrast, late fusion approaches utilize separate bitemporal images as inputs and commonly employ a Siamese network as the backbone architecture comprising identical subnetworks with shared weights. For instance, Zhang et al. [31] introduced a global-aware Siamese network (GAS-Net) that leverages a global attention module and a foreground awareness module to achieve effective change detection. Similarly, Zhu et al. [32] proposed a twin global learning framework (Siam-GL), incorporating twin networks, G-H sampling mechanisms, and change mask constraints to achieve high-accuracy, robust semantic change detection. Notably, the efficacy of these direct classification methods is contingent upon the availability and quality of training samples, which are particularly challenging to obtain for "from-to" variation types, especially in the context of data-intensive convolutional neural networks (CNNs) [33].

Typically, the postclassification comparison approach encompasses two fundamental stages: initial classification and subsequent postprocessing. Initially, bitemporal images undergo classification, the results of which serve as the basis for change detection through further comparative postprocessing. This renders the outcome highly contingent upon the accuracy of the initial classification. Despite its reliance on classification accuracy, the method's intuitive simplicity and reduced classification space have led to its widespread adoption [34]. For instance, Wan et al. [35] proposed a multisource change detection technique that integrates multitemporal segmentation with composite classification, viewing change detection as a specialized form of classification. Building on this foundation, they later introduced an enhanced version employing collaborative multitemporal segmentation and hierarchical compound classification to bolster classification accuracy [36]. Similarly, Dahiya et al. [37] presented a postclassification comparison method predicated on artificial neural networks, demonstrating high detection accuracy on the Hyperion EO-1 dataset.

The methodology presented in this article aims to detect changed vector polygons and fundamentally operates as a binary change detection framework guided by multiclass change detection. Initially, samples from two epochs of imagery are generated through superpixel segmentation and multiscale cropping, serving, respectively, as the training and prediction sets for the classification network. Specifically, the first epoch of imagery, along with vector data from the same epoch, forms the training set, while the second epoch of imagery, combined with vector data from the first epoch, makes up the prediction set. By examining the classification prediction results for the samples from the second epoch, it is possible to identify regions that have experienced land cover changes and determine the specific types of changes. Ultimately, a change decision-maker is employed during postprocessing to detect vector polygons that meet the predefined change criteria.

B. Superpixel Segmentation-Based Change Detection

The term "superpixel" describes a cluster of adjacent pixels in an image with similar attributes, such as color and texture. Utilizing a reduced set of superpixels as the focal point for image processing can substantially enhance computational efficiency in subsequent tasks [38]. Current superpixel segmentation techniques fall into two principal categories: graph theory- and clustering-based methods.

In graph theory-based approaches, each pixel is treated as a node in an undirected graph, with the similarity between adjacent pixels represented as edge weights. Superpixels are formed by minimizing a predefined cost function [39]. Notable examples include the normalized cut (N-Cut) [40] and lazy random walk (LRW) [41] algorithms. The N-Cut algorithm aims for balanced graph segmentation through global optimization and normalized cutting costs, while the LRW algorithm emphasizes local structure and employs iterative optimization to produce superpixels of variable size and number. However, these graph theory-based methods generally suffer from high computational complexity and lack the ability to regulate superpixel compactness.

Conversely, clustering-based methods conceptualize superpixel segmentation as a clustering issue, offering advantages in terms of speed, compactness, and controllability. Notable clustering-based algorithms include simple linear iterative clustering (SLIC) [42], linear spectral clustering (LSC) [43], and simple noniterative clustering (SNIC) [44]. SLIC leverages k-means optimization for localized searching in color and spatial dimensions, LSC utilizes a kernel similarity metric and a weighted K-means objective function in a high-dimensional feature space to achieve normalized graph partitioning, and SNIC uses noniterative methods to directly classify pixels based on color and spatial coordinates. Among these, SLIC is most commonly employed due to its simplicity, efficacy, and computational efficiency [45].

Currently, superpixel segmentation serves as a prevalent preprocessing technique in object-level change detection. Zhan et al. [46] introduced a bilinear CNN (BCNN) integrated with the SLIC superpixel algorithm, aiming to reduce annotation requirements and enhance detection performance. Li et al. [47] developed a superpixel-by-superpixel clustering framework (SSCF) that employs both SLIC and the Gaussian mixture model (GMM) to refine the detection of subtle changes in hyperspectral image change detection (HSI-CD) and mitigate confusion surrounding varying degrees of change along decision boundaries. Furthermore, Zhang et al. [48] formulated an end-to-end superpixel enhanced change detection network (ESCNet), which amalgamates differentiable superpixel segmentation with deep CNNs, allowing for more precise localization of change regions in very-high-resolution (VHR) images. The aforementioned studies collectively underscore the significant progress made in superpixel segmentation technology for object-level change detection.

C. Prior Knowledge-Based Change Detection

The intrinsically data-driven nature of deep learning often constrains model performance when training data are

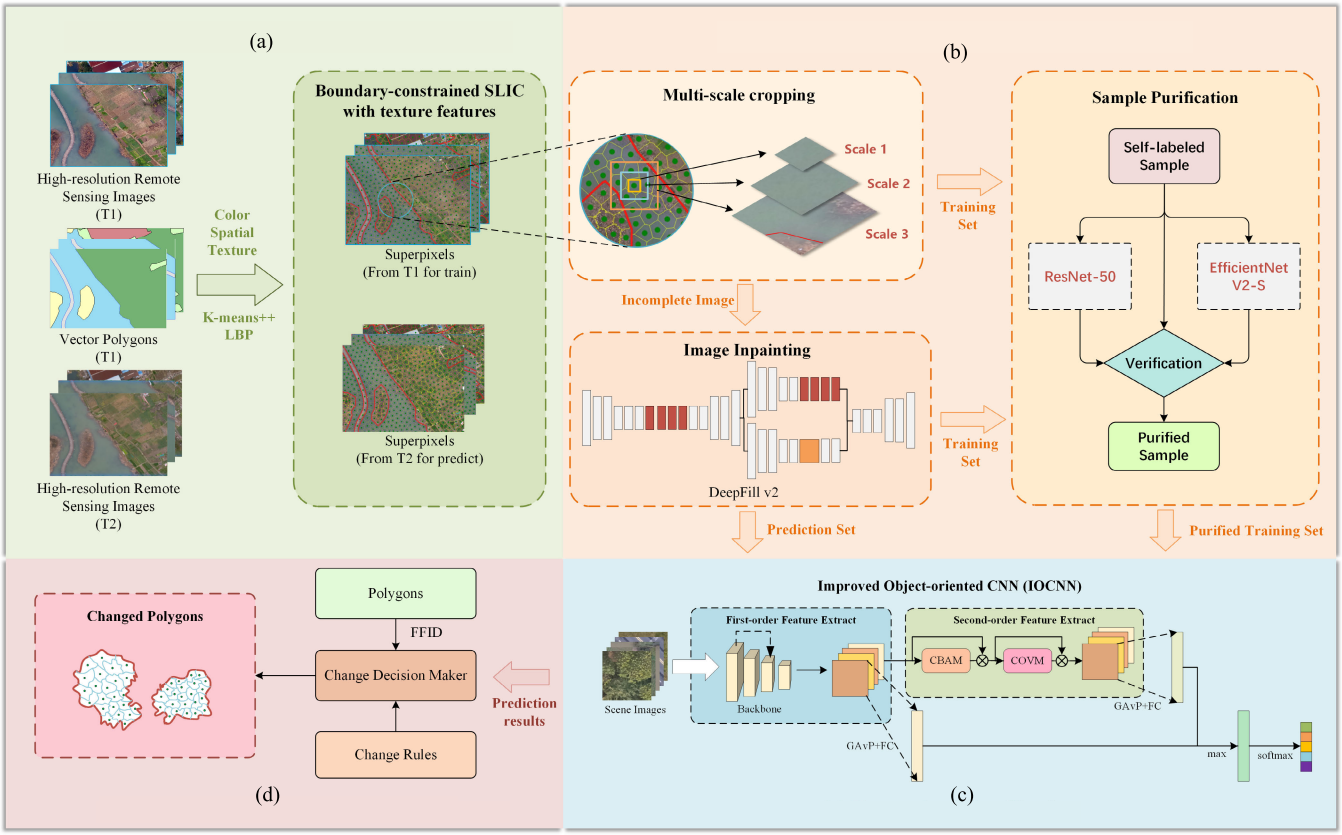


Fig. 1. Workflow of the proposed change detection approach. (a) Image Segmentation aided by vector polygons. (b) Automatic generation and purification of samples. (c) Fine-grained classification. (d) Change detection and postprocessing.

inadequate. However, the incorporation of prior knowledge can serve as an auxiliary mechanism to enhance both model performance and generalizability, particularly in scenarios with limited data or incomplete labels [49]. For instance, Zhu et al. [50] developed a knowledge-guided land pattern depicting (KGLPD) framework that leverages OpenStreetMap (OSM) data and prior knowledge to enhance land-use classification accuracy significantly. Similarly, Lv et al. [51] introduced a multiscale information attention neural network guided by the change gradient image (CGI), which markedly improved the model's change detection capabilities. Moreover, Bai et al. [52] formulated an edge-guided recurrent CNN (EGRCNN) that capitalizes on prior boundary information to refine the accuracy of building change detection. These studies illustrate that the integration of prior knowledge, such as geographic or spatial structure information, allows models to more precisely identify both land cover types and their changes, thereby advancing the accuracy of remote sensing image analysis.

III. Methods

The study establishes a comprehensive framework for detecting changes in vector polygons within high-resolution remote sensing imagery, as illustrated in Fig. 1. The methodology is built as an end-to-end processing pipeline involving four key phases. Initially, we use dual-temporal high-resolution remote sensing imagery and land cover vector data as inputs. These are subjected to superpixel segmentation using a modified SLIC algorithm. Next, high-quality samples are generated through multiscale cropping based on each superpixel unit, along with further refinement via image inpainting and sample purification. These samples are then used for fine-grained image classification via an IOCNN. Finally, a change decision-maker applies various rules to identify and locate changes within the vector polygons. Further details for each module are elaborated on in Sections III-A and III-D.

A. Image Segmentation

Image segmentation quality directly impacts the results of change detection in object-oriented tasks. Superpixel segmentation is a widely used method for image oversegmentation, wherein pixel regions with similar features, such as color, brightness, and texture, are merged into a smaller set of superpixels. This technique effectively reduces the complexity of subsequent image processing [53]. The SLIC algorithm [42] is a rapid method for generating uniform and compact superpixels. By incorporating color information and spatial proximity, SLIC effectively preserves image boundary features. However, SLIC does not guarantee complete consistency of land cover types within superpixels and requires improvement when dealing with complex scenes. To address this issue, we introduce a novel approach: a boundary-constrained SLIC algorithm combined with texture features. This approach leverages vector boundaries as prior knowledge and significantly improves image segmentation accuracy, as demonstrated in Fig. 2.

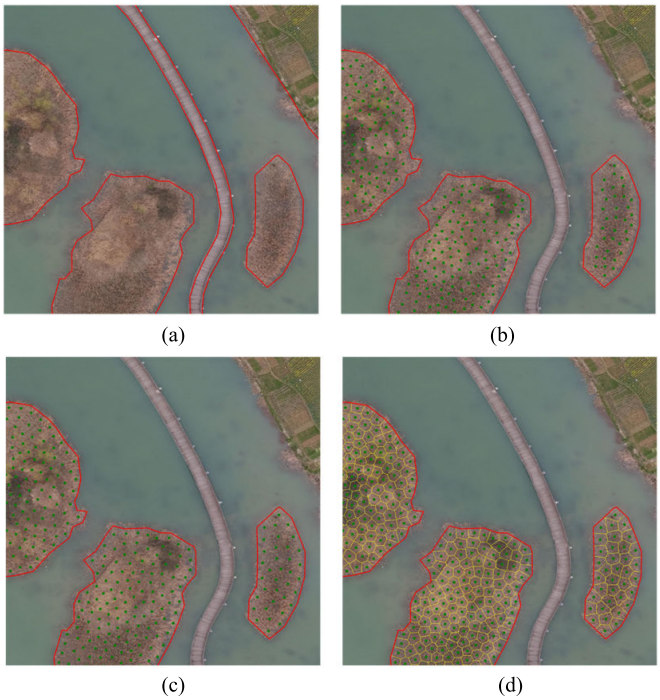


Fig. 2. Steps of vector-constrained region SLIC (as an example of grassland). (a) Original image and corresponding land-cover vector polygon. (b) Initializing seed points. (c) Fine-tuning seed points. (d) Clustering.

The specific details of the algorithm are given as follows.

1) Initialize Cluster Centers: Given the number of superpixels (K), iteratively place seed points such that the distance from each seed point to the region boundary and other seed points is maximized to achieve uniform placement of seed points in the region. The formula for calculating the seed point (P) is given as follows:

$$P = \text{argmax}_x\left(\min\left(\|x - y\|_2\right)\right) \tag{1}$$

where y represents the points on the region's boundary and the coordinates of the placed seed points, and x denotes the coordinates of the unplaced seed points. Each piece of vector polygon is regarded as an independent region. Since the shape and size of polygons are not uniform, the seed points in each polygon are calculated by the preset superpixel size to make the generated superpixel uniform and compact. As shown in (2), N is the total number of pixels in the region, and C_0 signifies the segmentation size standard. Assuming that each superpixel is a square of uniform size, its side length can be represented by S, $S = m \times C_0$, and m is a constant to measure the size of the superpixel better

$$K = N \parallel (m \times C_0)^2. \tag{2}$$

2) Iterative Update: The distance between each pixel in the region and K seed points is calculated. In the SLIC algorithm, the distance metric is represented by a 5-D feature vector, denoted as $D_k = \left[l_k, a_k, b_k, x_k, y_k\right]^T$. Here, l_k, a_k, and b_k represent the color components of the three channels in the International Commission on Illumination (CIE) color space, respectively, while x_k and y_k denote the pixel coordinates. To improve the quality and accuracy of image segmentation, we introduce a texture feature metric, LBP_k. Local binary pattern (LBP) is an algorithm describing texture features. Its fundamental principle involves comparing the central pixel's gray values with its neighborhood's surrounding pixels. The comparison result is then encoded in binary form, serving as the local texture feature of the current pixel [54]. The calculation formula for the LBP algorithm is given as follows:

$$\text{LBP}_{P,R} = \sum_{p=0}^{P-1} s\left(g_p, g_c\right) \cdot 2^p \tag{3}$$

$$s(a, b) = \begin{cases} 1, a \geqslant b \\ 0, a < b \end{cases} \tag{4}$$

where P represents the number of pixels sampled in the neighborhood, R denotes the radius of the neighborhood, g_p signifies the gray value of the neighborhood pixel, g_c means the gray value of the center pixel, and $s(\cdot)$ corresponds to the binary expression of the gray level.

Texture features can effectively describe image details, enhance boundary connectivity, and mitigate noise interference. Introducing texture feature measurement, a 6-D feature vector, $D'_k = \left[l_k, a_k, b_k, x_k, y_k, \text{LBP}_k\right]^T$, is constructed. Considering spatial, color, and texture features enhances the accuracy and robustness of image segmentation. The specific measurement formula is given as follows:

$$d_s = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} \tag{5}$$

$$d_c = \sqrt{\left(l_j - l_i\right)^2 + \left(a_j - a_i\right)^2 + \left(b_j - b_i\right)^2} \tag{6}$$

$$d_t = \sqrt{\left(\text{LBP}_j - \text{LBP}_i\right)^2} \tag{7}$$

$$D = \sqrt{d_c^2 + w d_s^2 + u d_t^2} \tag{8}$$

where d_s, d_c, and d_t are the spatial distance, color distance, and texture distance from the pixel to the seed point, respectively, while w and u are the closeness coefficients, which measure the weight of these distances in the total distance metric. To enhance clustering convergence speed and ensure segmentation quality, we define a circular search area around the seed point with a radius of $2S$ [55]. Then, we compute the shortest distance between each pixel within this area and the seed point. Furthermore, we apply an iterative clustering approach using the K-means++ algorithm [56] until reaching the maximum number of iterations or achieving convergence.

3) Connect Component: After iterative optimization, there may still be the problem of multiconnected or some superpixels being too small. To improve the connectivity, employing the region-growing method to filter out fragmented and notably small superpixels, subsequently merging them with their adjacent superpixels that possess the most similar gray value, is essential.

This method uses the vector boundary to constrain the superpixel segmentation and further considers the texture features. With the help of the vector boundary and texture features, high segmentation quality can be achieved. At the same time, by inheriting the land class attribute of the vector polygon, the unity of the land class information in the subsequent samples can be guaranteed as much as possible.

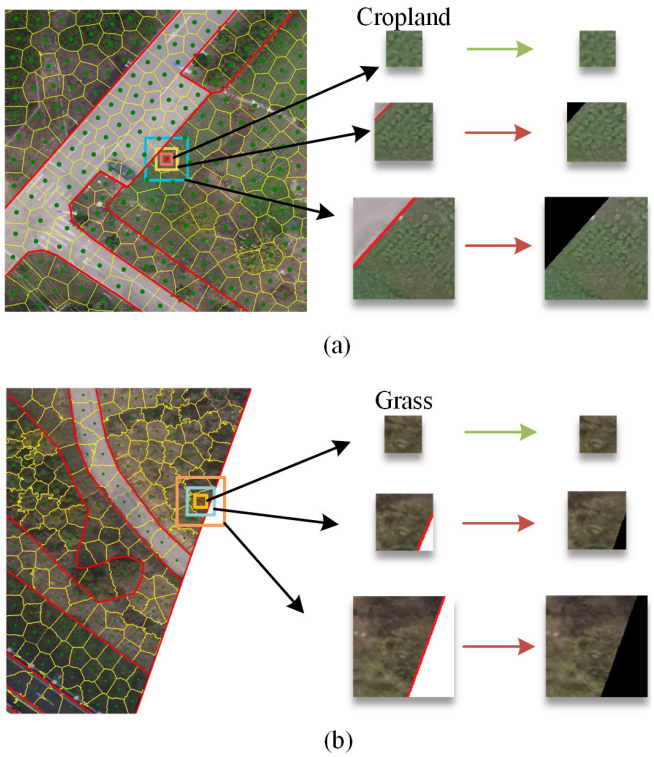


Fig. 3. Multiscale cropping of images: (a) suggests that as the cropping size expands, the likelihood of the sample incorporating additional land cover classes also rises and (b) signifies that when the boundary of the image is cropped, any areas that exceed the original image dimensions will result in null values. Only the regions within the vector boundary are preserved when cropping. Regions beyond the vector boundary are set to zero.

B. Automatic Generation, Inpainting, and Purification of Samples

For deep learning methods, in order to obtain better classification results, in addition to selecting the appropriate model, the quality and quantity of samples should also be considered. Currently, most traditional approaches for obtaining scene classification sample data involve using web crawlers or grid-cutting techniques on existing data to generate samples [57]. However, these methods often yield unlabeled and unreliable data, requiring manual verification and visual interpretation selection, resulting in inefficiency. This study proposes an efficient automatic sample generation method to address this issue. This method utilizes the centroid of a superpixel as the center for sample clipping and combines it with the corresponding geographic location and land class information from vector polygons. By doing so, the proposed method achieves the automatic generation and labeling of a large number of high-quality samples. In addition, since ground objects exhibit different characteristics at varying spatial resolutions and obtaining comprehensive ground object information at a single scale is challenging, our method predefines a segmentation size denoted as C_0. Multiple sizes are then selected based on this predefined segmentation size to crop and obtain samples of different sizes, thereby considering the diverse nature of ground objects.

Fig. 3 illustrates the process of multiscale image cropping. As the cropping scale increases, the likelihood of generating samples that contain features from other land cover types also rises. Moreover, when cropping samples along the edges of the imagery, areas extending beyond the image boundaries may result in null values (no data). To address these issues, the study employs zero-padding and image inpainting techniques. Specifically, areas within the sample, which either belong to other land cover types or are null values, are first padded with zeros, based on the boundaries of the vector polygons. Subsequently, the repaired image of this portion is generated using the pretrained DeepFill v2 model [58]. The structure of the DeepFill v2 model can be observed in Fig. 4. As a generative adversarial network (GAN) model, it comprises two main components: the generator and the discriminator. The generator utilizes a coarse-to-fine dual encoder–decoder structure to restore image quality. At the same time, the discriminator is a convolutional network using spectral normalization [59] to assess the authenticity of the generated image. Both components continually undergo iterative training to enhance their performance, making image inpainting more realistic [60].

Initial self-labeled samples can be obtained through multiscale cropping and image inpainting. The self-labeled samples obtained from the anterior and posterior temporal images are used for training and testing remote sensing scene image classification, respectively. However, vector polygons might have inherent inaccuracies at their boundaries and in terms of land cover types. Consequently, labels inherited by samples from these vector polygons could be unreliable. While the image patching technique has its merits, it also harbors potential errors, and at times, the patching results might not meet the desired standards. Thus, further purification of the initial samples is necessary. As Fig. 5 illustrates, the purification process is composed of two pretrained classification networks, ResNet-50 [61] and EfficientNet V2-S [62], which are used to classify the self-labeled samples. Subsequently, the classification results are then cross-referenced with the sample labels. A sample is considered "pure" and retained for final training only if the predictions from both classifiers are entirely consistent with the original label. Joint purification of the samples using two classifiers can significantly enhance their accuracy and reliability. It is important to note that the purification operation is solely applied to the training data.

C. Fine-Grained Scene Classification Model

Traditional coarse-grained image classification methods struggle to capture the intricate features in images due to the complex structure, diverse scales, and irregular rotation angles found in natural resource land cover scenes. Therefore, there is a need to develop a network model capable of extracting high-order features for fine-grained scene classification. Covariance pooling is frequently used in remote sensing and geoinformatics to extract higher-order features. Compared to traditional methods such as average or max pooling, covariance pooling captures the interrelations between image features by computing their covariance matrix, which allows for a more comprehensive description of land objects and spatial structures. This article proposes using an IOCNN

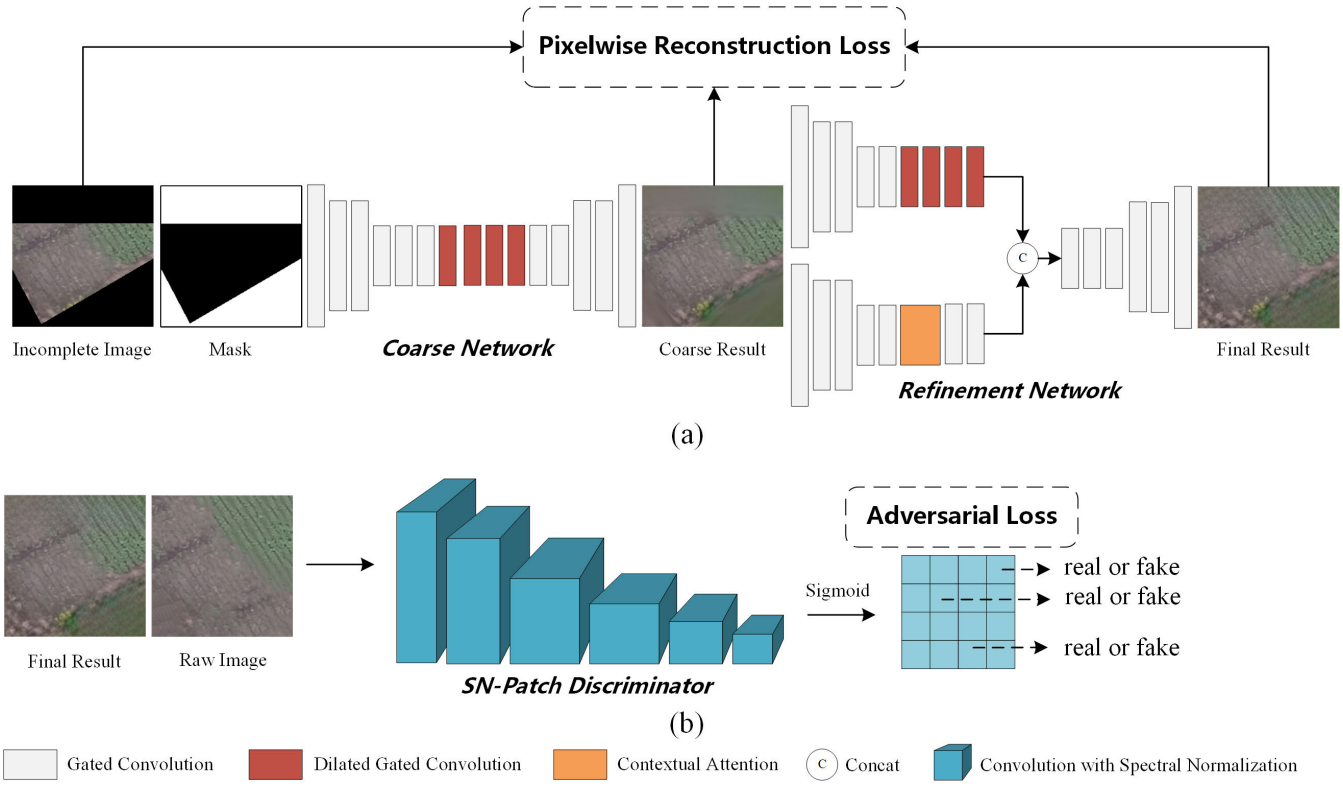


Fig. 4. Architecture of DeepFill v2. It contains two parts: (a) generator: two-stage framework with gated convolution and contextual attention and (b) discriminator: spectral-normalized markovian discriminator. Both pixelwise reconstruction loss and adversarial loss are considered in the training process.

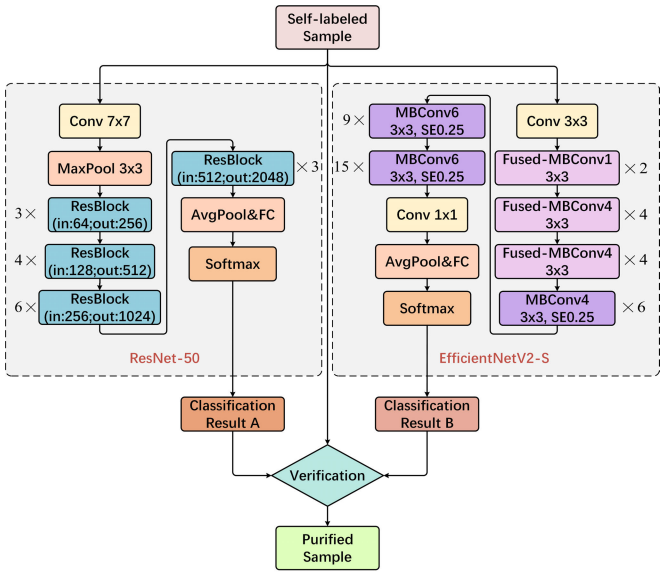


Fig. 5. Flowchart of sample purification.

model [63] to capture higher order features. Building on first-order features, the model employs attention mechanisms and covariance pooling to extract second-order features. By fusing both first- and second-order feature representations, the model achieves enhanced feature representation capabilities, leading to improved classification accuracy.

Fig. 6 shows the IOCNN network model. Initially, the model employs a conventional deep neural network to extract first-order features. Subsequently, the model utilizes the convolutional block attention module (CBAM) [64] and the covariance matrix analysis module (COVM) to acquire second-order features, which are then merged with the first-order features. Finally, a fully connected layer is employed to classify the combined features. Compared to employing BCNN for second-order feature extraction, using the covariance matrix of the final convolutional features helps reduce the number of model parameters and enhances the training efficiency. Since this article primarily focuses on the classification of RGB three-channel samples, the original model's four-channel two-input configuration is omitted. In addition, we employ the ConvNeXt [65] model as the backbone network for first-order feature extraction. Combining the advantages of CNN and transformer architectures, ConvNeXt offers higher accuracy and lower computational cost, making it one of the more advanced network structures currently available.

In recent years, attention mechanisms have been proposed to improve neural networks. The role of attention mechanisms is to enhance the semantic representation of specific input regions by assigning varying weights. Within computer vision, one commonly used attention mechanism is the CBAM, consisting of two submodules: channel attention and spatial attention.

The channel attention module (CAM) compresses the global spatial information, learns features in the channel dimension, and adaptively adjusts the feature importance of each channel. The whole process can be expressed as

$$M_{\text{cam}}(\text{F}) = \sigma\left(\text{MLP}(\text{AvgPool}(\text{F})) + \text{MLP}(\text{MaxPool}(\text{F}))\right) \quad (9)$$

where $\sigma(\cdot)$ represents the activation function, MLP$(\cdot)$ denotes the shared multilayer perceptron, and AvgPool$(\cdot)$ and

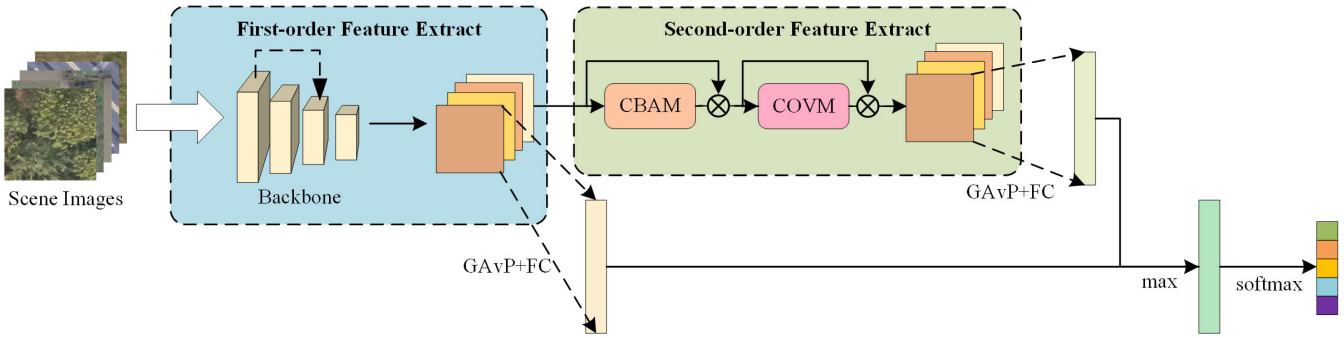


Fig. 6. Example architecture of the IOCNN.

MaxPool(·) signify the results of global average pooling and max pooling in the spatial dimension, respectively.

The spatial attention module (SAM) compresses and pools the features of each location, performs feature learning in the spatial dimension, and focuses on key information in different spatial locations. The whole process can be expressed as

$$M_{\text{sam}}(\text{F}) = \sigma\left(f^{7\times7}\left(\left\{\text{AvgPool}'(\text{F}), \text{MaxPool}'(\text{F})\right\}\right)\right) \quad (10)$$

where $f^{7\times7}(\cdot)$ represents the 7×7 convolutional layer, $\{\cdot\}$ denotes concatenation in the channel dimension, $\text{AvgPool}'(\cdot)$ and $\text{MaxPool}'(\cdot)$ signify the average pooling and max pooling in the channel dimension, respectively.

Despite the weight enhancement of the region of interest by the CBAM module, the features obtained are limited to first-order statistics. To improve the features' expressive capability, acquiring second-order statistics is necessary. The network utilizes a module for covariance matrix analysis to extract second-order statistics from the feature maps, comprising covariance pooling and matrix power normalization as two sequential steps. Covariance pooling converts the feature map into a covariance matrix primarily. The process involves centering the matrix around the mean of each random variable, followed by multiplying the centered matrix with itself to derive the covariance matrix. For a given feature map $R^{h\times w\times d}$, obtained from the CBAM module, transform it to a matrix $X^{n\times d}$, with $n = h \times w$. The covariance matrix is computed using the following procedure:

$$\text{Cov} = X\hat{I}X^T \quad (11)$$

$$\hat{I} = \frac{1}{n}\left(I - \frac{1}{n}ii^T\right) \quad (12)$$

where $\hat{I}$ represents the center matrix, I denotes the identity matrix, and i means the column vector with all values 1.

Based on the covariance matrix, a metalayer is employed to compute approximately the matrix square root to fulfill the eigenvectors' normalization requirement. The metalayer comprises three components: prenormalization, Newton–Schultz iteration, and postcompensation. Prenormalization ensures the convergence of the iteration by dividing the covariance by its trace. The Newton–Schultz iteration is a cyclic approximation procedure used to compute the covariance's square root. Postcompensation restores the covariance matrix's original scale by multiplying the covariance matrix's square root with its own trace [66]. Specifically, given initial values, $Y_0 = A$ and $Z_0 = I$. For $k = 1, 2, \ldots, N$, the formula for calculating the matrix power normalization is given as follows:

$$\begin{cases} A = \dfrac{1}{\text{tr}(\text{Cov})}\text{Cov} \\ Y_k = \dfrac{1}{2}Y_{k-1}(3I - Z_{k-1}Y_{k-1}) \\ Z_k = \dfrac{1}{2}(3I - Z_{k-1}Y_{k-1})Z_{k-1} \\ C = \sqrt{\text{tr}(\text{Cov})}Y_N \end{cases} \quad (13)$$

where $\text{tr}(\cdot)$ represents the trace of the matrix, $\text{tr}(\text{Cov}) = \sum_i \lambda_i$, λ_i denotes the eigenvalue of the covariance matrix, and the final calculated second-order pooling feature is $C \in R^{d\times d}$, which represents the statistical correlation between each channel.

While second-order features can capture higher-level image information, first-order features also contain essential basic information from the original image. In some cases, this basic information is indispensable. Considering both the first- and second-order features can yield more comprehensive and accurate image features, thereby improving the model's generalization ability. Hence, this model utilizes the maximum value of the first- and second-order features as the final feature and employs the cross-entropy function as the loss function. The specific calculation formula is given as follows:

$$\text{loss}(x_1, x_2, y) = -\log\left(\frac{\exp\left(\max(x_1, x_2)[y]\right)}{\sum_j \exp\left(\max(x_1, x_2)[j]\right)}\right) \quad (14)$$

where x_1 and x_2 are the first- and second-order features, y is the true class of the input image, and j is the sample index in each batch.

D. Change Detection and Postprocessing

This article presents a change detection method that utilizes two remote sensing images and a single vector dataset. Foremost, we generate the sample and prediction datasets that correspond to the two images. These datasets are used as the training and prediction sets for scene classification, respectively. Subsequently, the vector polygons associated with the changed dataset are utilized to locate and identify the changed areas, thereby accomplishing the change detection task.

In practical applications, various tasks may have distinct criteria for changes in different ground classes. For instance,

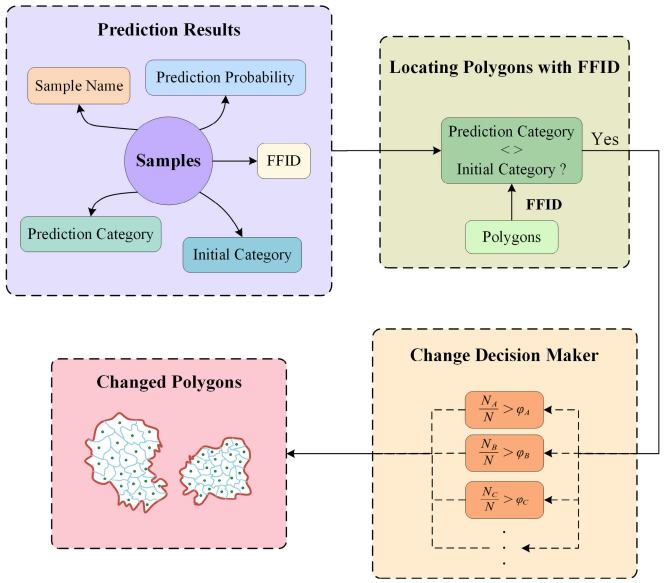


Fig. 7. Flowchart of change detection and postprocessing. (FFID is the unique identification field of polygons.)

specific applications necessitate the absence of buildings on cultivated land but permit the presence of some grass. To accommodate diverse requirements, we perform postprocessing using a decision-maker. The specific rules are outlined in the following:

$$\frac{N_A}{N} > \varphi \tag{15}$$

where N_A represents the number of samples predicted as class A in this vector polygon, N denotes the total number of samples in this vector polygon, and φ signifies the threshold. After applying various postprocessing rules, the vector polygons that satisfy the change rules are selected. This process allows for the completion of change detection between the two images. Fig. 7 illustrates the entire process. We store the sample prediction results in a database format, including the name, predicted category, predicted probability, initial category, and other attributes. Subsequently, using the unique identifier FFID of the vector polygon, the samples within the same vector polygon are grouped as a single unit, and samples with inconsistent initial and predicted categories are filtered out. Finally, the changed vector polygons are located according to the decision-maker.

IV. Experiments

To evaluate the effectiveness of the proposed method, we selected two pairs of dual-temporal remote sensing images with varying resolutions as the research dataset. The experiment was performed on a server running Windows 10 operating system. The server was equipped with an Intel Xeon CPU E52640 v4 @ 2.4-GHz processor, 48 GB of memory, and an NVIDIA Quadro P4000 GPU. The network model was developed using the PyTorch architecture and Python version 3.7. The experimental process consists of the following steps: dual-temporal image superpixel segmentation, automatic generation and purification of samples, scene classification, change detection, and postprocessing.

A. Study Area and Data Sources

Dataset 1 was sourced from the Yangxi River area in Huishan district, Wuxi city, Jiangsu province, China. It comprises two phases of RGB unmanned aerial vehicle (UAV) images and the corresponding land-cover vector data of the initial phase image. The images have a resolution of 0.05 m and cover an area of 6.5 km^2, as depicted in Fig. 8. Fig. 8(a) and (c) displays the UAV images and their corresponding vector data captured in this area in June 2019. Moreover, Fig. 8(b) presents a drone image of the same area in June 2020. The vector data are manually drawn from the original time period and exist in the form of polygons. It contains unique identification fields, such as FFID, and attributes, such as land cover categories. The land cover types in this region primarily encompass six categories: building, forest, water, cropland, grass, and road. Notably, cropland and grass exhibit the most significant changes. Superpixel segmentation employs the segmentation size criterion to regulate the number of seed points. An optimal number of seed points exists to avoid oversegmentation or undersegmentation, yielding unsatisfactory results. For this particular study area, we established the segmentation size standard (denoted as C_0) as 128, taking into account the scale of objects given the resolution of the UAV images. Subsequently, we conducted sample cropping using three different sizes: 64, 128, and 224.

Dataset 2 was obtained from Guangling, Yangzhou city, Jiangsu, China. It comprises two phases of RGB Gaofen-2 satellite images along with the corresponding land-cover vector data of the first phase image. The images have a resolution of 1 m and cover an area of 265.36 km^2, as shown in Fig. 9. Specifically, Fig. 9(a) and (c) displays the region's GF-2 satellite imagery and accompanying vector data in 2021, respectively. In addition, Fig. 9(b) presents a GF-2 satellite image of the same region in 2022. The vector data were created through manual delineation during the original period, similar to Dataset 1. The land cover types within this region primarily consist of building, forest, water, agricultural, and road. For this study area, we set the segmentation size standard C_0 to 64 and performed sample cropping at three sizes of 48, 64, and 96.

B. Superpixel Segmentation and Sample Purification

Considering the performance of the experimental equipment, we divided the original dataset into subsets of 5000 $\times$ 5000 pixels. Furthermore, set the segmentation overlap rate to 10% to ensure the effective detection of boundary polygons. Each subset underwent superpixel segmentation using the improved SLIC (ISLIC) algorithm. Then, we generated self-labeled samples through multiscale cropping and image inpainting. In the experiments, the maximum number of iterations for the ISLIC algorithm is set to 10, with a compactness value of 10. The parameters P and R for the LBP features are set to 16 and 3, respectively. The purpose of the samples varied depending on the period when they were generated. For samples generated from the pretemporal image, a purification process was conducted before utilizing them for model training and validation with a ratio of 6:4. Conversely, samples

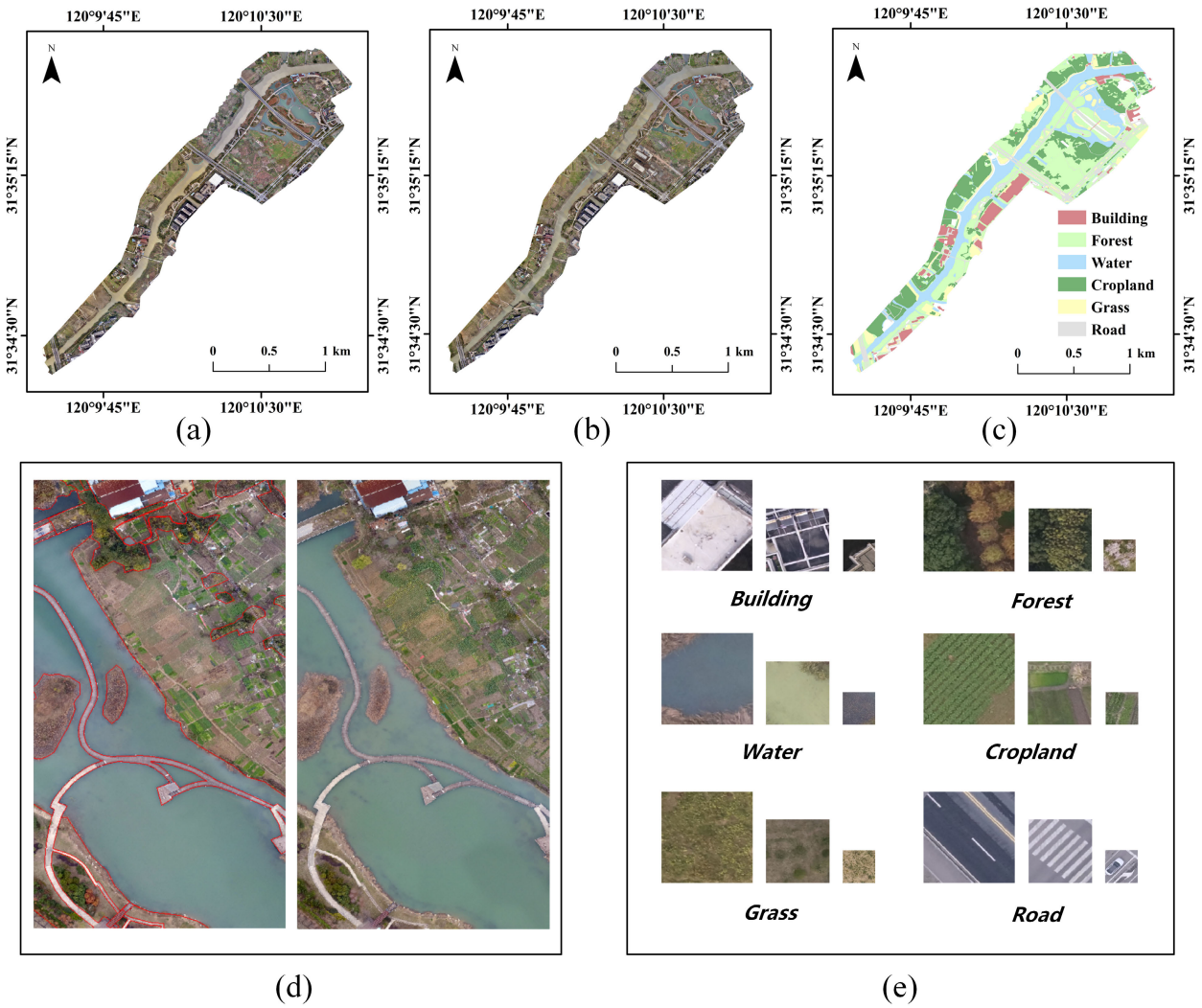


Fig. 8. Dataset 1 collected from the vicinity of Yangxi River, Huishan, Wuxi, Jiangsu, China. (a) UAV image in June 2019. (b) UAV image in June 2020. (c) Vector data corresponding to the 2019 image. (d) Detail presentation of two phases of images. (e) Examples of land cover samples in Dataset 1.

generated from the posttemporal image were directly used as the test set. In this study, cross-validation was performed using two classifiers, ResNet-50 and EfficientNet V2-S, to purify the samples. Purified samples are defined as those with classification results consistent with the original label, thus effectively improving the model's classification performance. For the test set, the sample name stored the label, center coordinate, and subset number, allowing easy localization of the vector polygon in which each sample resides based on the center coordinate. As depicted in Fig. 10, our proposed segmentation method accurately captures the boundaries of vector polygons, producing uniform and compact superpixels. This feature contributes to the generation of high-quality samples.

C. Scene Classification

The accuracy of subsequent change detection is directly influenced by the classification accuracy of the proposed method [67], which serves as a postclassification comparison approach. To validate the classification performance of the IOCNN model, we compared it with several other state-of-the-art models, including ResNet-50, EfficientNet V2-S, BCNN [68], PatchConvnet [69], PVTv2 [70], WaveMLP [71], and ConvNeXt. Specifically, the BCNN model employed a dual-stream ResNet implementation, PatchConvnet used its S60 version, PVTv2 utilized its B2 version, WaveMLP experiments were based on its S version, and ConvNeXt adopted its T version. We selected samples from two periods in two datasets for training and testing, using overall accuracy (OA) and kappa coefficient as evaluation metrics. All models were evaluated under the same baseline conditions, and the classification results are presented in Table I.

As we can see from Table I, the following holds.

1) In terms of both OA and kappa coefficient, the IOCNN model outperforms other architectures. Specifically, on Dataset 1, IOCNN registers an OA of 91.20% and a kappa coefficient of 0.894, representing a 0.33% increase in OA compared to the ConvNeXt model (90.87% OA). On Dataset 2, the respective values for IOCNN stand at 88.92% for OA and 0.862 for the kappa coefficient. This underscores IOCNN's advantage in classification accuracy.

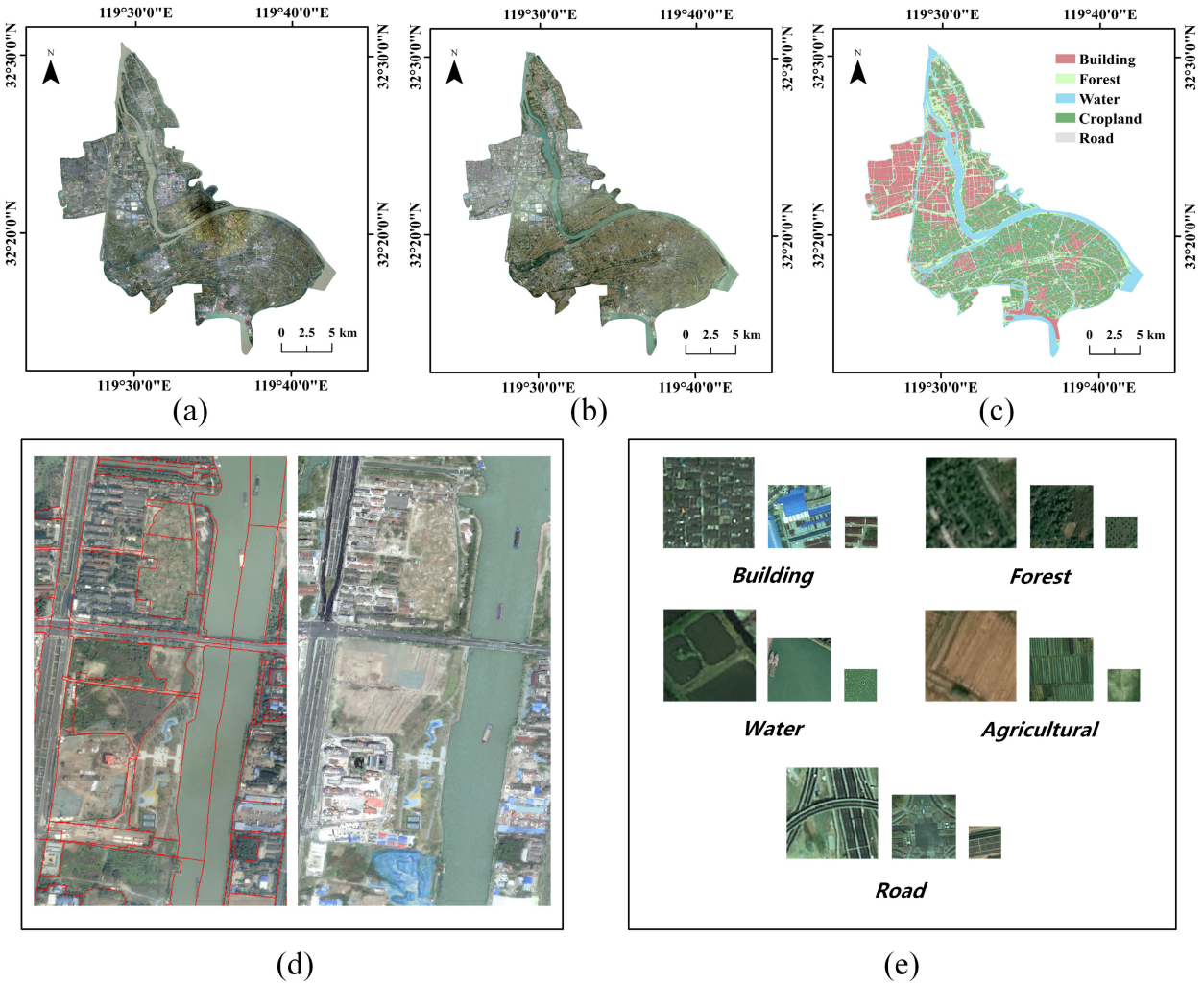


Fig. 9. Dataset 2 collected from Guangling, Yangzhou, Jiangsu, China. (a) Gaofen-2 satellite image in 2021. (b) Gaofen-2 satellite image in 2022. (c) Vector data corresponding to the 2021 image. (d) Detail presentation of two phases of images. (e) Examples of land cover samples in Dataset 2.

TABLE I
COMPARISON OF THE CLASSIFICATION METRIC OF DIFFERENT MODELS FOR TWO DATASETS

Model	Params(M)	FLOPs(G)	Dataset 1		Dataset 2	
			OA(%)	Kappa	OA(%)	Kappa
ResNet-50	25.56	4.12	87.26	0.847	87.44	0.843
EfficeintNet V2-S	**21.46**	**2.87**	89.27	0.871	87.56	0.844
BCNN **[68]**	285.65	4.38	89.93	0.879	87.76	0.847
PatchConvnet **[69]**	25.23	4.00	90.83	0.890	88.40	0.855
PVTv2 **[70]**	25.36	3.90	89.73	0.877	87.24	0.840
WaveMLP **[71]**	30.71	4.55	89.17	0.870	88.04	0.850
ConvNeXt **[65]**	28.59	4.46	90.87	0.890	88.60	0.857
IOCNN	30.56	4.48	**91.20**	**0.894**	**88.92**	**0.862**

2) Despite the BCNN possessing a high parameter count of approximately 285.65M, its performance metrics do not surpass those of IOCNN. This suggests that IOCNN offers more efficient parameter utilization while retaining high accuracy levels, thereby making it a viable option for fine-grained classification tasks within complex scenes.

3) It is noteworthy that model performance varies between the two datasets. For instance, the Pyramid Vision Transformer v2 (PVTv2) exhibits superior performance on drone datasets (Dataset 1) compared to satellite imagery (Dataset 2). This discrepancy is likely attributable to PVTv2's transformer architecture, which excels in handling global information and long-range dependencies—features that may be particularly beneficial for high-resolution, complex drone datasets. Nevertheless, IOCNN demonstrates consistent performance across both datasets, highlighting its adaptability to various dataset characteristics.

D. Postprocessing and Change Detection Results

Because most of the vector polygons are manually delineated, the distribution of land classes differs from that of remote sensing images. For instance, vector polygons representing buildings may include small forests, roads, and other land classes. To avoid false changes caused by nonsubject land class predictions, it is essential to develop a change decision-maker that establishes distinct thresholds based on different change criteria, catering to diverse requirements

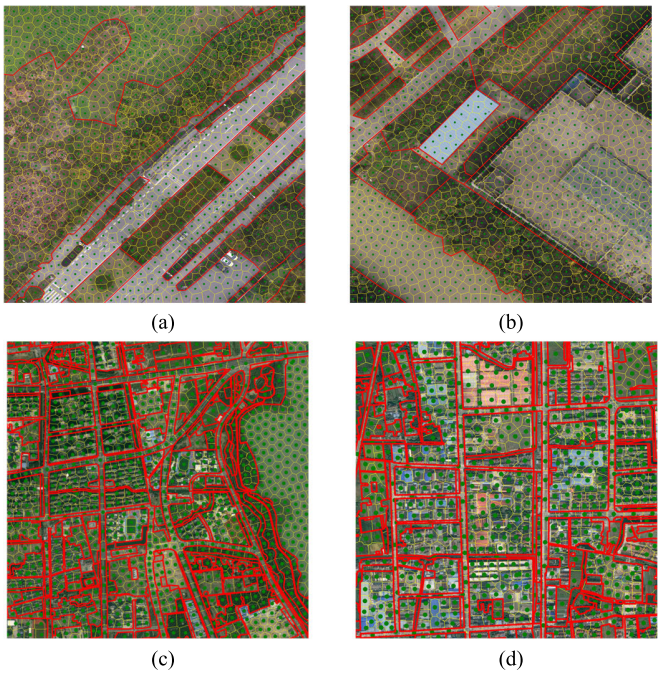


Fig. 10. Examples of superpixel segmentation results. (a) and (b) Outcomes of Dataset 1. (c) and (d) Results of Dataset 2.

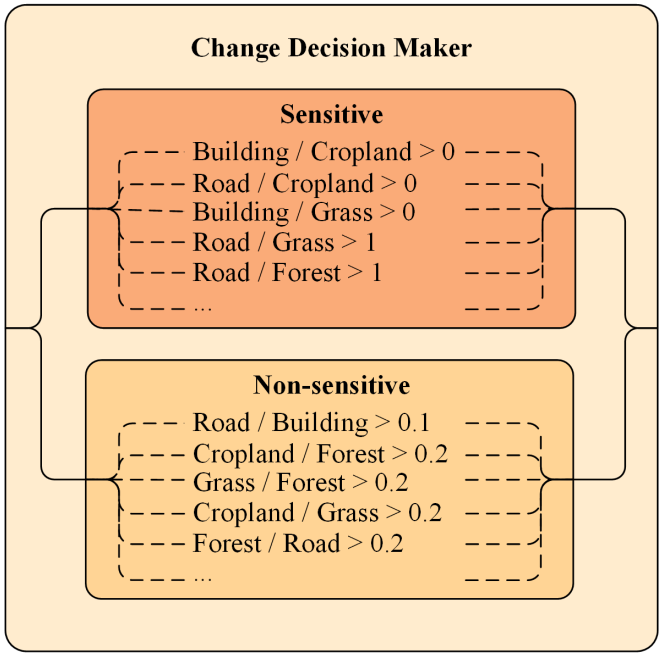


Fig. 11. Structure of change decision-maker.

in change detection tasks. Fig. 11 illustrates the change decision-maker employed in this research, which categorizes change rules into two classes: "sensitive" and "nonsensitive." In the "sensitive" class, the threshold is defined by the number of instances that comply with the rule, catering to scenarios requiring stringent land cover change management. For instance, if construction is strictly prohibited in cropland areas, the rule "building/cropland > 0" can be set. In contrast, for the "nonsensitive" class, the threshold is determined by the proportion of rule-compliant instances within the entire land polygon, accommodating more lenient land cover modifications. For example, if up to 20% grass coverage is permissible within forest areas, the rule "grass/forest > 0.2" could be implemented. By integrating multiple change rules, well-defined change criteria can be established to meet diverse detection requirements.

The changed vector polygons can be obtained through the postprocessing of the decision-maker, enabling the detection of changes between two images. Fig. 12(d) and (h) displays the change detection results for both datasets. Visual interpretation is utilized to verify the results and calculate the precision and recall rates by tallying the changes, false detections, and leak detections in the vector polygons. The experimental results demonstrate precision and recall rates of 91.89% and 94.44% for Dataset 1 and 87.59% and 91.41% for Dataset 2. Fig. 13 reveals part of the change detection results: Fig. 13(a) depicts the transformation of cultivated land into forest land, Fig. 13(b) illustrates a substantial conversion of cultivated land to buildings, Fig. 13(c) demonstrates the appearance of sheds potentially serving as shelters for building materials in the grassland, and Fig. 13(d) demonstrates the emergence of extensive building and road networks in the woodland. The obtained results illustrate the effectiveness of the proposed change detection method in accurately identifying and quantifying ground object changes within the vector polygons during the given time intervals, resulting in high precision and recall rates across both UAV and satellite datasets. Notably, this method excels in areas characterized by complex topography and occlusion.

E. Ablation Study

To rigorously evaluate the effectiveness and contributions of each module within our proposed change detection framework, we performed ablation studies on two distinct datasets, the outcomes of which are tabulated in Table II. We established a baseline model that employed the SLIC segmentation algorithm, without any purification mechanism, and utilized ResNet-50 for classification. The baseline model yielded precision and recall rates of 78.38% and 80.56% on Dataset 1 and 81.20% and 85.60% on Dataset 2, respectively. First, upon incorporating the ISLIC algorithm proposed in this study, the precision rate improved to 85.71% on Dataset 1 and 83.59% on Dataset 2. Correspondingly, the recall rates increased to 88.33% and 87.45%, respectively. These results substantiate the efficacy of ISLIC, which accounts for boundary constraints and texture features, thereby enhancing boundary delineation and sample homogeneity. Second, the integration of a sample purification mechanism led to further improvement in the classification performance, boosting the precision rate to 86.49% on Dataset 1 and 84.76% on Dataset 2. The recall rates attained were 88.89% and 88.90%, respectively. This uptick underscores the vital role of sample purification in elevating sample quality and classification accuracy. Finally, the employment of the IOCNN elevated the precision and recall rates to 91.89% and 94.44% for Dataset 1 and 87.59% and 91.41% for Dataset 2, respectively. These results further corroborate the robustness of the IOCNN model in handling high-resolution

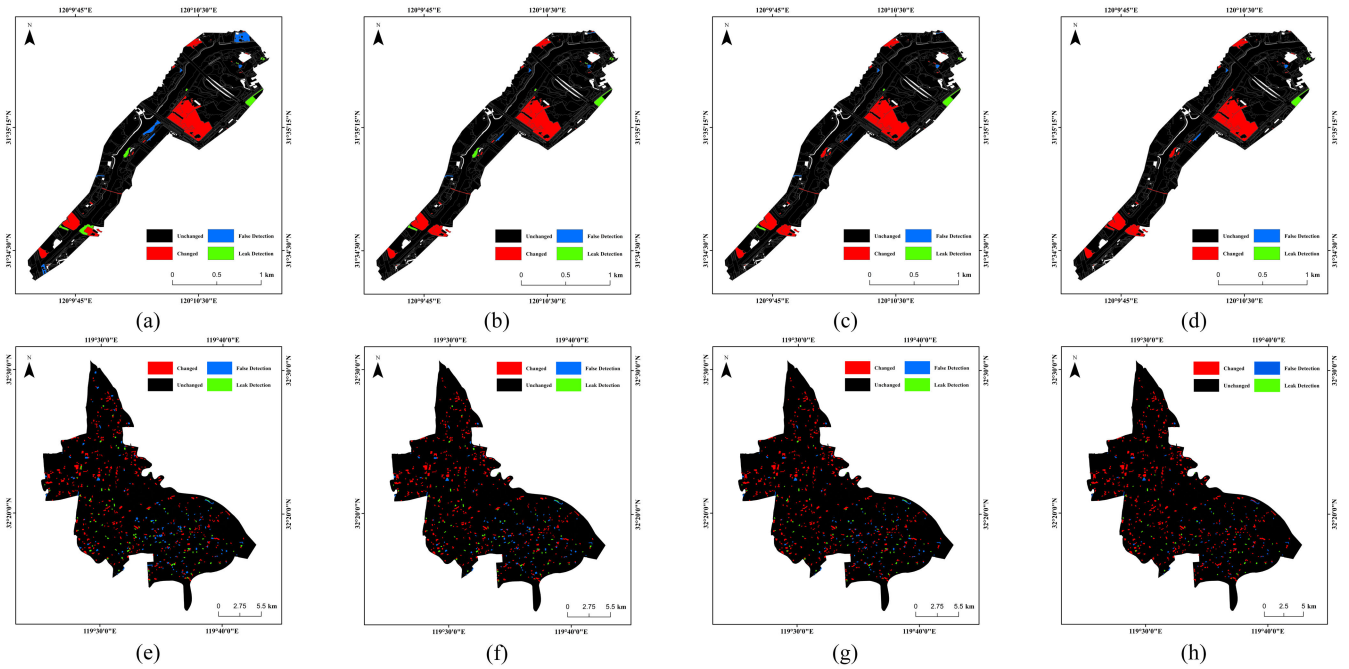


Fig. 12. Comparison of change detection results on two datasets. (a) SLIC + w/o purification + ResNet-50 (Dataset 1). (b) ISLIC + w/o purification + ResNet-50 (Dataset 1). (c) ISLIC + purification + ResNet-50 (Dataset 1). (d) ISLIC + purification + IOCNN (Dataset 1). (e) SLIC + w/o purification + ResNet-50 (Dataset 2). (f) ISLIC + w/o purification + ResNet-50 (Dataset 2). (g) ISLIC + purification + ResNet-50 (Dataset 2). (h) ISLIC + purification + IOCNN (Dataset 2).

TABLE II
ABLATION STUDY RESULTS (%) WITH DIFFERENT MODULES ON TWO DATASETS

Baseline	ISLIC	Purification	IOCNN	Dataset 1			Dataset 2		
				Precision	Recall	F1	Precision	Recall	F1
√				78.38	80.56	79.46	81.20	85.60	83.34
√	√			85.71	88.33	87.00	83.59	87.45	85.48
√	√	√		86.49	88.89	87.67	84.76	88.90	86.78
√	√	√	√	**91.89**	**94.44**	**93.15**	**87.59**	**91.41**	**89.46**

remote sensing imagery. In summary, each module contributes substantively to the overall performance enhancement of the proposed framework, unequivocally validating the utility of each component. Fig. 12 visually compares the ablation study results on both datasets, showing improved detection accuracy with the proposed method.

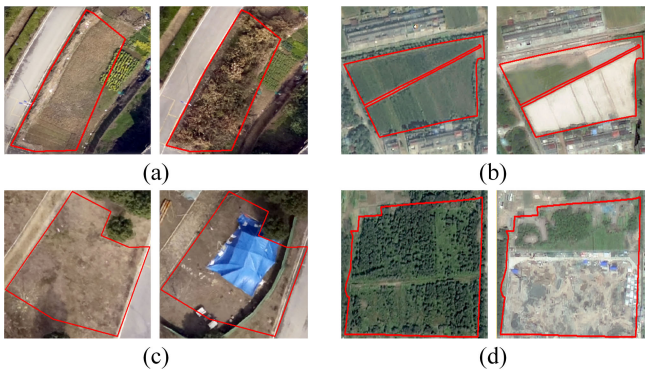


Fig. 13. Examples of change detection results. The findings presented in (a) and (c) are derived from the results of Dataset 1, whereas the findings in (b) and (d) are derived from Dataset 2. Each pair of images depicts the prechange and postchange states, respectively.

V. DISCUSSION

A. Comparison of Image Segmentation Methods

In order to verify the effectiveness of the proposed image segmentation method, the boundary constraints and texture features on the segmentation algorithm are tested by ablation experiments. A subset of the region from Dataset 1 was chosen as the experimental data, and the boundary recall rate [72] was employed as an evaluation metric for assessing the segmentation effect. The boundary recall rate serves as a metric to quantify the alignment between the segmentation boundary and the actual boundary. Its calculation formula is given in the following:

$$\mathrm{BR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{16}$$

where TP represents the number of correctly segmented boundary pixels and FN is the number of incorrectly segmented boundary pixels. The segmentation performance of each method is shown in Fig. 14.

From the graph analysis, the following can be obtained.

1) The SLIC algorithm with boundary constraints (BCSLIC) consistently achieves a higher boundary recall rate. This effect is more pronounced when the number of segmentation units is small, but as the number of segmentation units increases, the effect diminishes gradually. This reduction in effect may stem from the discrepancy between manually drawn boundary priors and the actual boundaries of ground objects. While increasing the number of segmentation units improves the accuracy of capturing detailed features

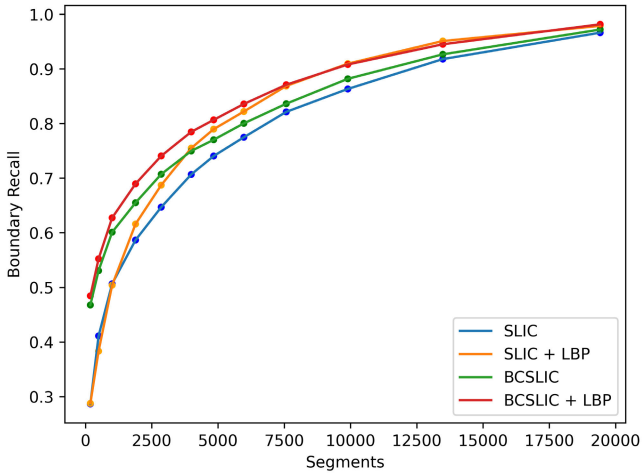


Fig. 14. Boundary performance with different methods on Dataset 1.

of ground objects, these features might go beyond the boundaries defined by the prior, exacerbating their limitations.

2) Incorporating the LBP algorithm for texture feature fusion further enhances the boundary recall rate of the segmentation algorithm. As shown in the figure, this enhancement is more prominent when the number of segmentation units is moderate, enabling the segmentation units to integrate color, spatial, and texture information effectively. However, when the number of segmentation units is too small, each unit encompasses more pixels, amplifying the influence of texture features and rendering the algorithm more vulnerable to noise and local variations. Conversely, when the number of segmentation units is too large, each unit contains fewer pixels, diluting the role of incorporating texture features and diminishing their contribution to boundary detection.
3) Generally, segmentation algorithms that incorporate boundary constraints and texture features tend to exhibit superior boundary performance, thereby validating the effectiveness of the enhanced segmentation method proposed in this study. In addition, the segmentation results depicted in Fig. 15 illustrate that the improved algorithm better approximates the actual boundary of the ground class while maintaining a more uniform and compact distribution. This characteristic is advantageous for generating subsequent samples.

B. Effectiveness of Sample Inpainting

The superpixels generated by the segmentation process are irregular and unsuitable for direct use in network model training. Therefore, it is necessary to crop them into regular samples. Considering the constraints imposed by vector polygons and the clear division of ground objects, the samples within the polygons can be directly clipped using the center point of each superpixel and a preset size.

However, when clipping near the edges of the polygons, there is a possibility of exceeding the boundary of the vector

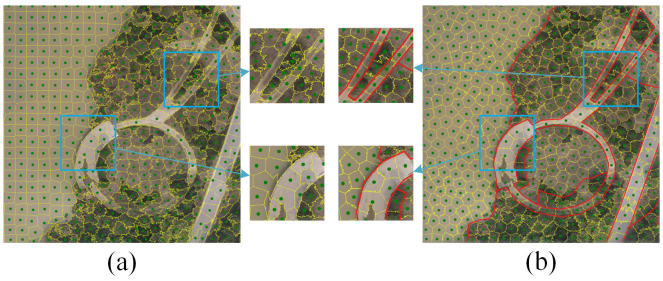


Fig. 15. Visual comparison of segmentation methods. (a) Segmentation results of the original SLIC. (b) Segmentation results of the improved method in this article.

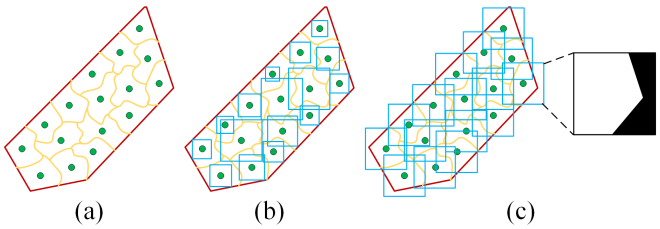


Fig. 16. Comparison of sample clipping methods. (a) Original segmentation map. (b) Illustration of adaptive clipping. (c) Illustration of cropping combined with inpainting.

polygons. This situation can result in the inclusion of additional land class information in the samples.

In order to address this issue, there are two commonly employed approaches: adaptive cropping [73] and cropping with inpainting. Fig. 16 illustrates the superpixel segmentation simulation map. In Fig. 16(a), the red border depicts the boundary of the vector polygons, the yellow line segment represents the boundary of the segmented superpixel, and the green point denotes the superpixel's center. On the other hand, Fig. 16(b) demonstrates the adaptive cropping technique, which dynamically adjusts the size of the clipping area when it exceeds the vector boundary, ensuring an appropriate size. The precise adjustment method is given as follows:

$$S = \begin{cases} S_0, & d_{\min} > S_0 \\ \sqrt{2}\, d_{\min}, & d_{\min} \leqslant S_0 \end{cases} \tag{17}$$

where S_0 is the preset crop size and $d_{\min}$ is the shortest distance from the superpixel center to the vector boundary. The cropping-with-inpainting method, depicted in Fig. 16(c), enables the cropping window to extend beyond the vector polygon's boundary. It assigns zero value to the region outside the boundary before feeding the incomplete image into the image repair network for restoration. A comparison between adaptive cropping and cropping with inpainting reveals a significant difference. In the former, the cropping window fails to encompass the entire vector polygon, possibly resulting in the omission of particular ground objects. Conversely, the latter ensures comprehensive sampling by fully covering the vector polygon when using an appropriate preset size. To comprehensively account for varying scales of ground objects and select appropriate sizes, this study employs multiscale cropping in addition to repair-based clipping. This approach aims to enhance the reliability of the cropped samples.

Fig. 17. Example of image inpainting results. (a) and (c) Outcomes of Dataset 1. (b) and (d) Results of Dataset 2.

TABLE III

COMPARISON OF CLASSIFICATION PERFORMANCE BEFORE AND AFTER PURIFICATION OF THE TWO DATASETS

Dataset			Epoch				
			10	20	30	40	50
Dataset 1	Initial	OA(%)	91.19	92.74	92.86	94.08	94.56
	Purified	OA(%)	98.66	98.72	98.89	99.21	99.42
Dataset 2	Initial	OA(%)	96.69	97.46	97.80	97.91	97.95
	Purified	OA(%)	97.90	98.68	99.56	99.62	99.69

This article utilizes the DeepFill v2 network for image repair, and the corresponding repair outcomes are presented in Fig. 17. The results demonstrate that the inpainting process successfully resolves the interference caused by land classes outside the vector polygons' boundary, thus validating the feasibility of our approach.

C. Impact of Sample Purification on Scene Classification

Further purification is necessary due to errors in the authenticity of the land class label for the vector polygons, as well as some repaired samples not meeting the required image quality. This inconsistency could affect the generated samples' alignment with their respective land class labels. This article employs ResNet-50 and EfficientNet V2-S classifiers for cross-validation to achieve automatic sample purification. Through purification, 13.53% of Dataset 1 samples and 9.95% of Dataset 2 samples were filtered out. To evaluate the importance of sample purification, this study employed prepurification and postpurification samples for training and compared the model's classification performance before and after purification (refer to Table III for details). The results indicate that sample purification improves the OA of Dataset 1 by 4.86% and Dataset 2 by 1.74%, further demonstrating that suspicious training samples can be effectively excluded through sample purification, improving accuracy and reliability during model training.

D. Impact of Classification Model on Detection Accuracy

For the postclassification comparison method, the classification effect directly affects change detection accuracy. To validate the IOCNN network model's superiority in this article, we compare its detection results with those of ResNet-50 and EfficientNet V2-S (see Table IV for details). The results indicate that the IOCNN model achieves the highest detection accuracy in both datasets, while the ResNet-50 model exhibits the lowest detection accuracy. Specifically, in Dataset 1, IOCNN's precision is 5.4% higher than ResNet-50, and its recall rate is also 5.55% higher. In Dataset 2, IOCNN's precision is 2.83% higher than ResNet-50, and its recall rate is also 2.51% higher. These findings demonstrate that models with higher classification accuracy tend to achieve superior change detection accuracy.

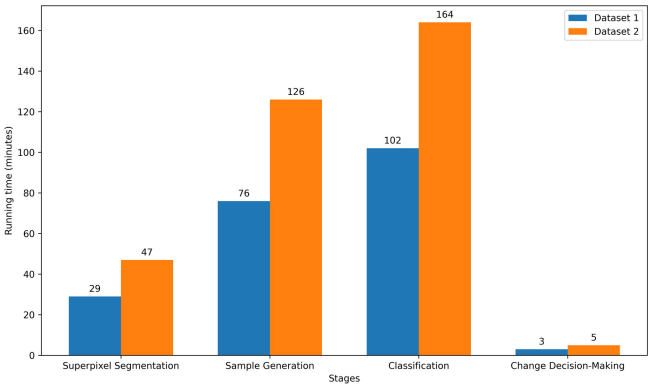


Fig. 18. Comparison of running times across different module stages.

TABLE IV

COMPARISON OF CHANGE DETECTION RESULTS FOR DIFFERENT CLASSIFICATION MODELS

Model		Dataset 1	Dataset 2
ResNet-50	Precision(%)	86.49	84.76
	Recall(%)	88.89	88.90
	F1(%)	87.67	86.78
EfficientNet V2-S	Precision(%)	89.19	85.02
	Recall(%)	91.67	89.23
	F1(%)	90.41	87.07
IOCNN	Precision(%)	91.89	87.59
	Recall(%)	94.44	91.41
	F1(%)	93.15	89.46

E. Time Consumption

To comprehensively evaluate the performance of the proposed method, we also examined the running time of each major module. Fig. 18 displays the runtime of each module on two different datasets.

From the figure, it is evident that the classification stage generally requires the most extended period. This duration is primarily due to the substantial computational resources and time required for training and predicting with deep learning models. Moreover, the training time is often determined by the number of samples under identical model and baseline conditions. The sample generation stage also demands a relatively long time, especially on Dataset 2, where the running time reaches 126 min. This extended time is mainly attributed to the high computational cost of multiscale cropping and image inpainting operations. The superpixel segmentation stage and the change decision-making stage require relatively less time. Superpixel segmentation takes 29 and 47 min on Dataset 1 and Dataset 2, respectively. The running time for the change decision-making stage is 3 and 5 min, respectively. The speed of the change decision-making stage is relatively fast because it only needs to read the prediction results and find the vector polygons that meet the conditions with the rules of the change decision-maker.

Although our method may not be the most time-efficient, it achieves reasonable time efficiency while maintaining high accuracy. Future work will explore the possibility of using faster hardware and other potential optimization strategies to further enhance the running speed of our method. Particularly in the classification and sample generation stages, we will consider employing more efficient algorithms or models to reduce the required time.

VI. Conclusion

This article proposes a change detection method for vector polygons based on high-resolution remote sensing images and deep learning. This method divides the change detection process into three parts: segmentation, classification, and detection. First, the dual-temporal remote sensing images were combined with the land-cover vector data, respectively, and the training set and the test set were automatically generated using the boundary constraint SLIC with texture features and the cropping-with-inpainting method. Then, the training set was purified by two-classifier cross-validation, and the IOCNN network classified the training set and the test set. Finally, the changed vector polygons were detected by combining the change rules. To verify the effectiveness of the proposed scheme, we conducted several ablation experiments and comparison experiments. The experimental results show that the improved segmentation method can segment more accurately boundaries than the original SLIC algorithm. The cropping-with-inpainting method used in this article can achieve complete coverage sampling within vector polygons. The sample purification method can further improve the accuracy of the model. The OA of the IOCNN model reached 91.2% and 88.92%, respectively, in the classification experiment of the two datasets, which has better classification performance compared with other advanced models. Finally, the results of visual interpretation show that the precision and recall rate of Dataset 1 are 91.89% and 94.44%, and the precision and recall rate of Dataset 2 are 87.59% and 91.41%, respectively. It can be seen that the change detection method proposed in this article can effectively detect the changed vector polygons and further reduce the manual input compared with the traditional method of manually updating the vector polygons. In the future, we will conduct further research on sample purification to achieve more demanding "nongrain" monitoring research of cultivated land.

References

[1] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, Jun. 1989.

[2] Z. Lv et al., "Land cover change detection with heterogeneous remote sensing images: Review, progress, and perspective," *Proc. IEEE*, vol. 110, no. 12, pp. 1976–1991, Dec. 2022.

[3] Z. Lv, T. Liu, J. A. Benediktsson, and N. Falco, "Land cover change detection techniques: Very-high-resolution optical images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 44–63, Mar. 2022.

[4] Z. Lv, F. Wang, G. Cui, J. A. Benediktsson, T. Lei, and W. Sun, "Spatial–spectral attention network guided with change magnitude image for land cover change detection using remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4412712.

[5] S. Tian, Y. Zhong, Z. Zheng, A. Ma, X. Tan, and L. Zhang, "Large-scale deep learning based binary and semantic change detection in ultra high resolution remote sensing imagery: From benchmark datasets to urban application," *ISPRS J. Photogramm. Remote Sens.*, vol. 193, pp. 164–186, Nov. 2022.

[6] H. Fang, P. Du, and X. Wang, "A novel unsupervised binary change detection method for VHR optical remote sensing imagery over urban areas," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 108, Apr. 2022, Art. no. 102749.

[7] Q. Ding, Z. Shao, X. Huang, and O. Altan, "DSA-Net: A novel deeply supervised attention-guided network for building change detection in high-resolution remote sensing images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 105, Dec. 2021, Art. no. 102591.

[8] S. Ye, J. Rogan, Z. Zhu, and J. R. Eastman, "A near-real-time approach for monitoring forest disturbance using Landsat time series: Stochastic continuous change detection," *Remote Sens. Environ.*, vol. 252, Jan. 2021, Art. no. 112167.

[9] K. S. Willis, "Remote sensing change detection for ecological monitoring in United States protected areas," *Biol. Conservation*, vol. 182, pp. 233–242, Feb. 2015.

[10] B. Yuan et al., "Spatiotemporal change detection of ecological quality and the associated affecting factors in Dongting lake basin, based on RSEI," *J. Cleaner Prod.*, vol. 302, Jun. 2021, Art. no. 126995.

[11] X. Wang, X. Fan, Q. Xu, and P. Du, "Change detection-based co-seismic landslide mapping through extended morphological profiles and ensemble strategy," *ISPRS J. Photogramm. Remote Sens.*, vol. 187, pp. 225–239, May 2022.

[12] Y. Qing et al., "Operational earthquake-induced building damage assessment using CNN-based direct remote sensing change detection on superpixel level," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102899.

[13] P. Xiao, X. Zhang, D. Wang, M. Yuan, X. Feng, and M. Kelly, "Change detection of built-up land: A framework of combining pixel-based detection and object-based recognition," *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 402–414, Sep. 2016.

[14] M. Han, C. Zhang, and Y. Zhou, "Object-wise joint-classification change detection for remote sensing images based on entropy query-by fuzzy ARTMAP," *GISci. Remote Sens.*, vol. 55, no. 2, pp. 265–284, Mar. 2018.

[15] T. Gao, H. Li, M. Gong, M. Zhang, and W. Qiao, "Superpixel-based multiobjective change detection based on self-adaptive neighborhood-based binary differential evolution," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118811.

[16] L. Khelifi and M. Mignotte, "Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis," *IEEE Access*, vol. 8, pp. 126385–126400, 2020.

[17] L. Zhang, X. Hu, M. Zhang, Z. Shu, and H. Zhou, "Object-level change detection with a dual correlation attention-guided detector," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 147–160, Jul. 2021.

[18] Q. Shu, J. Pan, Z. Zhang, and M. Wang, "DPCC-Net: Dual-perspective change contextual network for change detection in high-resolution remote sensing images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102940.

[19] Z. Guo and S. Du, "Mining parameter information for building extraction and change detection with very high-resolution imagery and GIS data," *GISci. Remote Sens.*, vol. 54, no. 1, pp. 38–63, Jan. 2017.

[20] W. Feng, H. Sui, J. Tu, W. Huang, C. Xu, and K. Sun, "A novel change detection approach for multi-temporal high-resolution remote sensing images based on rotation forest and coarse-to-fine uncertainty analyses," *Remote Sens.*, vol. 10, no. 7, p. 1015, Jun. 2018.

[21] C. Zhang, R. Wu, G. Li, W. Cui, and Y. Jiang, "Change detection method based on vector data and isolation forest algorithm," *J. Appl. Remote Sens.*, vol. 14, no. 2, p. 1, May 2020.

[22] D. Wei, D. Hou, X. Zhou, and J. Chen, "Change detection using a texture feature space outlier index from mono-temporal remote sensing images and vector data," *Remote Sens.*, vol. 13, no. 19, p. 3857, Sep. 2021.

[23] Q. Zhu, X. Guo, Z. Li, and D. Li, "A review of multi-class change detection for satellite remote sensing imagery," *Geo-Spatial Inf. Sci.*, pp. 1–15, Oct. 2022.

[24] K. Jiang, W. Zhang, J. Liu, F. Liu, and L. Xiao, "Joint variation learning of fusion and difference features for change detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4709918.

[25] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[26] T. Chen, Z. Lu, Y. Yang, Y. Zhang, B. Du, and A. Plaza, "A Siamese network based U-Net for change detection in high resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2357–2369, 2022.

[27] H. Li, F. Zhu, X. Zheng, M. Liu, and G. Chen, "MSCDUNet: A deep learning framework for built-up area change detection integrating multispectral, SAR, and VHR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5163–5176, 2022.

[28] Z. Lv, H. Huang, L. Gao, J. A. Benediktsson, M. Zhao, and C. Shi, "Simple multiscale UNet for change detection with heterogeneous remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[29] M. Han, R. Li, and C. Zhang, "LWCDNet: A lightweight fully convolution network for change detection in optical remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[30] Z. Zheng, Y. Wan, Y. Zhang, S. Xiang, D. Peng, and B. Zhang, "CLNet: Cross-layer convolutional neural network for change detection in optical remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 247–267, May 2021.

[31] R. Zhang, H. Zhang, X. Ning, X. Huang, J. Wang, and W. Cui, "Global-aware Siamese network for change detection on remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 199, pp. 61–72, May 2023.

[32] Q. Zhu et al., "Land-use/land-cover change detection based on a Siamese global learning framework for high spatial resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 63–78, Feb. 2022.

[33] H. Xia, Y. Tian, L. Zhang, and S. Li, "A deep Siamese postclassification fusion network for semantic change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5622716.

[34] Z. Zheng, Y. Zhong, S. Tian, A. Ma, and L. Zhang, "ChangeMask: Deep multi-task encoder-transformer-decoder architecture for semantic change detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 183, pp. 228–239, Jan. 2022.

[35] L. Wan, Y. Xiang, and H. You, "A post-classification comparison method for SAR and optical images change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1026–1030, Jul. 2019.

[36] L. Wan, Y. Xiang, and H. You, "An object-based hierarchical compound classification method for change detection in heterogeneous optical and SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9941–9959, Dec. 2019.

[37] N. Dahiya et al., "Detection of multitemporal changes with artificial neural network-based change detection algorithm using hyperspectral dataset," *Remote Sens.*, vol. 15, no. 5, p. 1326, Feb. 2023.

[38] X. Wei, Q. Yang, Y. Gong, N. Ahuja, and M.-H. Yang, "Superpixel hierarchy," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4838–4849, Oct. 2018.

[39] M. Wang, X. Liu, Y. Gao, X. Ma, and N. Q. Soomro, "Superpixel segmentation: A benchmark," *Signal Process., Image Commun.*, vol. 56, pp. 28–39, Aug. 2017.

[40] R. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 10–17.

[41] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[42] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[43] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3317–3330, Jul. 2017.

[44] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4895–4904.

[45] J. Nowosad and T. F. Stepinski, "Extended SLIC superpixels algorithm for applications to non-imagery geospatial rasters," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102935.

[46] T. Zhan, M. Gong, X. Jiang, and W. Zhao, "Transfer learning-based bilinear convolutional networks for unsupervised change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[47] Q. Li et al., "A superpixel-by-superpixel clustering framework for hyperspectral change detection," *Remote Sens.*, vol. 14, no. 12, p. 2838, Jun. 2022.

[48] H. Zhang, M. Lin, G. Yang, and L. Zhang, "ESCNet: An end-to-end superpixel-enhanced change detection network for very-high-resolution remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 28–42, Jan. 2023.

[49] L. Yan, J. Yang, and J. Wang, "Domain knowledge-guided self-supervised change detection for remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4167–4179, 2023.

[50] Q. Zhu et al., "Knowledge-guided land pattern depiction for urban land use mapping: A case study of Chinese cities," *Remote Sens. Environ.*, vol. 272, Apr. 2022, Art. no. 112916.

[51] Z. Lv, P. Zhong, W. Wang, Z. You, and N. Falco, "Multiscale attention network guided with change gradient image for land cover change detection using remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[52] B. Bai, W. Fu, T. Lu, and S. Li, "Edge-guided recurrent convolutional neural network for multitemporal remote sensing image building change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5610613.

[53] J. Yin, T. Wang, Y. Du, X. Liu, L. Zhou, and J. Yang, "SLIC superpixel segmentation for polarimetric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5201317.

[54] W. Huang, Y. Huang, H. Wang, Y. Liu, and H. J. Shim, "Local binary patterns and superpixel-based multiple kernels for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4550–4563, 2020.

[55] F. He, M. A. Parvez Mahmud, A. Z. Kouzani, A. Anwar, F. Jiang, and S. H. Ling, "An improved SLIC algorithm for segmentation of microscopic cell images," *Biomed. Signal Process. Control*, vol. 73, Mar. 2022, Art. no. 103464.

[56] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th ACM-SIAM Symp. Discrete Algorithms*, Jan. 2007, pp. 1027–1035.

[57] Z. Guo et al., "Land type authenticity check of vector patches using a self-trained deep learning model," *Int. J. Remote Sens.*, vol. 43, no. 4, pp. 1226–1252, Feb. 2022.

[58] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4470–4479.

[59] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[60] M. Shao, C. Wang, T. Wu, D. Meng, and J. Luo, "Context-based multiscale unified network for missing data reconstruction in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[62] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," 2021, *arXiv:2104.00298*.

[63] Z. Li, E. Li, A. Samat, T. Xu, W. Liu, and Y. Zhu, "An object-oriented CNN model based on improved superpixel segmentation for high-resolution remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4782–4796, 2022.

[64] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[65] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022, *arXiv:2201.03545*.

[66] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 947–955.

[67] C. Wu, B. Du, X. Cui, and L. Zhang, "A post-classification change detection method based on iterative slow feature analysis and Bayesian soft fusion," *Remote Sens. Environ.*, vol. 199, pp. 241–255, Sep. 2017.

[68] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNNs for fine-grained visual recognition," 2015, *arXiv:1504.07889*.

[69] H. Touvron et al., "Augmenting convolutional networks with attention-based aggregation," 2021, *arXiv:2112.13692*.

[70] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.

[71] Y. Tang et al., "An image patch is a wave: Phase-aware vision MLP," 2021, *arXiv:2111.12294*.

[72] T. C. Ng and S. K. Choy, "Variational fuzzy superpixel segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 1, pp. 14–26, Jan. 2022.

[73] J. Shi et al., "Fine object change detection based on vector boundary and deep learning with high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4094–4103, 2022.

Hui Zhang received the B.S. degree in computer science and technology from Zhejiang A&F University, Hangzhou, China, in 2021. He is currently pursuing the M.S. degree in cartography and geographic information engineering with Jiangsu Normal University, Xuzhou, China.

His research interests include remote sensing image processing and change detection.

Wei Liu received the M.S. and Ph.D. degrees in cartography and geographic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2007 and 2010, respectively.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou. His research interests include spatial data quality checking, high-resolution remote sensing image processing, and GIS development and applications.

Hao Niu received the B.S. degree in computer science and technology from Henan Finance University, Zhengzhou, China, in 2021. He is currently pursuing the M.S. degree in cartography and geographic information engineering with Jiangsu Normal University, Xuzhou, China.

His research interests include remote sensing image processing and change detection.

Pengcheng Yin received the M.S. degree in land resource management and the Ph.D. degree in cartography and geographic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2006 and 2012, respectively.

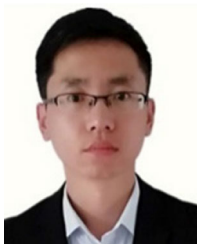
He is currently with the Bureau of Natural Resources and Planning, Xuzhou. His research interests include natural resources survey and monitoring, basic surveying and mapping, and geographic information systems.

Shiling Dong received the master's degree in software engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2004.

She is currently the Deputy Director of the Xuzhou Surveying and Geographic Information Center, a subsidiary of the Bureau of Natural Resources and Planning, Xuzhou, China. Her professional journey has been marked by a strong focus on land resource management. Her expertise encompasses natural resource informatization, surveying, and geographic information systems. She has played a pivotal role in numerous research projects, many of which have earned National and Provincial Science and Technology Progress Awards or Outstanding Engineering Awards.

Jialin Wu received the B.S. degree in surveying and mapping engineering from Nanjing Forestry University, Nanjing, China, in 2017. He is currently pursuing the M.S. degree in cartography and geographic information engineering with Jiangsu Normal University, Xuzhou, China.

His research interests include remote sensing image processing and change detection.



Erzhu Li received the M.S. degree in photogrammetry and remote sensing from the China University of Mining and Technology, Xuzhou, China, in 2014, and the Ph.D. degree in cartography and geographic information systems from Nanjing University, Nanjing, China, in 2017.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou. His research interests include high-resolution image processing and computer vision in urban remote sensing applications.

Lianpeng Zhang received the M.S. degree in geodesy and survey engineering from the Shandong University of Science and Technology, Taian, China, in 1989, and the Ph.D. degree in photogrammetry and remote sensing from the Shandong University of Science and Technology, Qingdao, China, in 2003.

He is currently a Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include high-resolution image processing and computer vision in urban remote sensing applications.

Changming Zhu received the Ph.D. degree in photogrammetry and remote sensing from the Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently a Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include remote sensing technologies and their applications to environments, land use/cover change, arid area water resources and hydrology, and wetland change monitoring.