


BTS: a binary tree sampling strategy for object identification based on deep learning


Xianwei Lv, Zhenfeng Shao, Xiao Huang, Wen Zhou, Dongping Ming, Jiaming Wang & Chengzhuo Tong

To cite this article: Xianwei Lv, Zhenfeng Shao, Xiao Huang, Wen Zhou, Dongping Ming, Jiaming Wang & Chengzhuo Tong (2021): BTS: a binary tree sampling strategy for object identification based on deep learning, International Journal of Geographical Information Science, DOI: [10.1080/13658816.2021.1980883](https://doi.org/10.1080/13658816.2021.1980883)

To link to this article: <https://doi.org/10.1080/13658816.2021.1980883>

 Published online: 24 Sep 2021.

 Submit your article to this journal [↗](#)

 Article views: 131

 View related articles [↗](#)

 View Crossmark data [↗](#)

RESEARCH ARTICLE



BTS: a binary tree sampling strategy for object identification based on deep learning

Xianwei Lv ^a, Zhenfeng Shao ^a, Xiao Huang ^b, Wen Zhou^c, Dongping Ming^d,
Jiaming Wang ^a and Chengzhuo Tong^e

^aState Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China; ^bThe Department of Geosciences, University of Arkansas, Fayetteville, AR, USA; ^cFaculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands; ^dSchool of Information Engineering China, University of Geosciences (Beijing), Beijing, China; ^eThe Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong, China

ABSTRACT

Object-based convolutional neural networks (OCNNs) have achieved great performance in the field of land-cover and land-use classification. Studies have suggested that the generation of object convolutional positions (OCPs) largely determines the performance of OCNNs. Optimized distribution of OCPs facilitates the identification of segmented objects with irregular shapes. In this study, we propose a morphology-based binary tree sampling (BTS) method that provides a reasonable, effective, and robust strategy to generate evenly distributed OCPs. The proposed BTS algorithm consists of three major steps: 1) calculating the required number of OCPs for each object, 2) dividing a vector object into smaller sub-objects, and 3) generating OCPs based on the sub-objects. Taking the object identification in land-cover and land-use classification as a case study, we compare the proposed BTS algorithm with other competing methods. The results suggest that the BTS algorithm outperforms all other competing methods, as it yields more evenly distributed OCPs that contribute to better representation of objects, thus leading to higher object identification accuracy. Further experiments suggest that the efficiency of BTS can be improved when multi-thread technology is implemented.

ARTICLE HISTORY

Received 5 August 2020
Accepted 12 September 2021

KEYWORDS

Object identification; convolutional neural network; object convolutional position; deep learning

1. Introduction

Object-based image analysis (Blaschke 2010, Blaschke *et al.* 2014, Luus *et al.* 2015) introduces a new paradigm into geographical applications, and particularly to land-cover and land-use classification. In object-based image analysis, remote sensing images are processed based on objects that represent vector polygons (often meaningful geographical units) segmented from remote sensing images. Based on these objects, a variety of geographical features can be extracted, including fractal geometry (Krummel *et al.* 1987), perimeter-area relationship (Zhang and Atkinson 2016), minimum width bounding box (Chaudhuri and Samal 2007), and morphological building and shadow indices (Huang

et al. 2014), benefiting a wide range of remote sensing applications. Therefore, object-based image analysis serves as a bridge that links remote sensing and geographical information science. As a basic step of object-based image analysis, the identification of objects is of great importance for downstream analysis.

Object-based convolutional neural networks (OCNNs) are deep learning based methods that aim to identify irregular objects (Zhang *et al.* 2018, Lv *et al.* 2018b). Taking advantage of the great object locating capability of CNNs, OCNNs incorporate object-based image analysis and CNNs that aim to extract and learn deep features for discriminative localization of objects in images (Oquab *et al.* 2014, 2015, Zhou *et al.* 2016, Bazzani *et al.* 2016). Existing studies have shown that an activation map for a particular object category points to a discriminative image region that benefits its identification, i.e. objects and their surrounding regions are observable in activation maps. For a certain object, the OCNN generates multiple locations as object convolutional positions (OCPs) to exact patches and further feed them into the CNN. During this process, OCNN assigns a unique label to each OCP based on the corresponding receptive field, and all pixels in an object are assigned to the same label with the highest frequency. Figure 1 presents an example of object identification via OCNNs.

This study serves as an extension of existing OCNN studies. OCNNs generate multiple OCPs, i.e. centroids of convolutional windows in objects, and extract the convolutional receptive fields from images based on these OCPs. However, most OCNN studies tend to focus on the optimization of network structures, largely ignoring the selection of OCPs, despite such selection plays an important role in object identification, as misclassifications via OCNNs are often caused by poor choices of OCPs (Chen *et al.* 2019). According to the principle of OCNN, in an extracted patch centered on an OCP, the center and surrounding pixels in the patch are the real object (see Figure 1). To guarantee a successful object identification, we need to make sure the center of the patch (the real object) is activated in the model (i.e. OCPs must be located within the object). Studies have shown that if the center pixel and nearby pixels are close to the object boundary, misclassifications tend to occur (Lv *et al.* 2018b, Chen *et al.* 2019). Thus, OCPs should be located away from the object boundary in order to ensure accurate object identification and classification. In most cases, objects present great homogeneity, and OCPs are able to obtain true labels with a high probability when the homogeneity of the pixels around the OCP is high. The distribution of OCPs also affects the identification of objects, as patches generated by evenly distributed OCPs can cover the object in a comprehensive manner, providing sufficient information for the CNN model.

OCPs are rarely used in traditional computer vision algorithms but are rather common in geographical applications. The OCP selection process is essentially a data sampling process. Hence, in this study, we propose a morphology-based binary tree sampling (BTS) method that considers the shape attributes (e.g. shape index (SI) and area) of segmented geographical objects to solve OCP sampling problems. The proposed BTS facilitates the generation of evenly distributed OCPs. The strategy of BTS is to select locations in objects used for standard CNNs. After image segmentation and object delineation using well-established methods, a series of image processing techniques are used to split each object into a tree structure and further identify the locations of OCPs. The goals of this study are to optimize OCP selection and to investigate the importance of OCPs in object identification. In addition, we propose a distribution index that can be used to evaluate



Figure 1. The concept of Object-based convolutional neural networks (OCNNs). Object convolutional positions (OCPs) (red dots) are distributed within a certain object (the blue polygon) and their corresponding patches (red dashed lines).

the distribution of OCPs in objects. Taking object identification in land-cover and land-use classification as a case study, we compare seven OCP generation methods using two very high resolution (VHR) images and five standard CNN models, including AlexNet (Krizhevsky *et al.* 2012), VGG16 (Simonyan and Science 2015), VGG19 (Simonyan and Science 2015), Inception_v3 (Szegedy *et al.* 2015), and ResNet (Kaiming *et al.* 2016).

The main contributions of the proposed method are as follows:

- (1) We propose a morphology-based binary tree sampling (BTS) method that provides a reasonable, effective, and robust strategy to select OCPs for identifying objects based on OCNNs.
- (2) We propose a distribution index to evaluate the distribution of OCPs in objects.

- (3) We illustrate the advantages and shortcomings of OCNN and a typical end-to-end network (U-Net). The results confirm that our OCNN incorporated with the BTS algorithm outperforms U-Net in full-element mapping from large-scale remote sensing images.

This paper is divided into seven sessions. Following the Introduction session (Section 1), Section 2 introduces relevant works. Section 3 documents our methodology in detail. Section 4 introduces the study area. Section 5 presents the classification results. Section 6 presents a discussion on the results, followed by Section 7 that concludes this study.

2. Related works

Data obtained from advanced earth observation techniques provide rich spatial and spectral information. Compared with images with median and low spatial resolution, remote sensing images with high resolution contain abundant information and thus are advantageous for geographical applications (Huang *et al.* 2020) that require precise thematic maps (e.g. land-cover and land-use classification). VHR image processing requires the extraction of deep features, posing challenges for conventional classification methods (Zhao and Du 2016).

CNNs contain multiple convolutional and pooling layers with hundreds or even thousands of filters (Krizhevsky *et al.* 2012). Based on the deep structures, CNNs are ideal in terms of extracting deep features from VHR images (Zhao *et al.* 2015) and thus have been applied to many geographical applications (Ma *et al.* 2019). In recent years, CNN-based algorithms have been greatly improved with respect to scene classification (Zou *et al.* 2015), image fusion (Shao and Cai 2018, Lu *et al.* 2019), object identification (Huang *et al.* 2020), land-cover and land-use classification (Wang *et al.* 2020, Martins *et al.* 2020), change detection (Yaqian *et al.* 2020), and semantic segmentation (Zhang *et al.* 2018, Guo and Feng 2020). As one of the most popular architectures, U-Net achieves high semantic segmentation and category mapping accuracy using natural remote sensing images (Alhassan *et al.* 2019, Flood *et al.* 2019). The object-based image analysis involves four key processes (Phiri *et al.* 2018): 1) segmenting images to generate objects, 2) designing object features, 3) training classifiers, and 4) classifying objects. The third step, i.e. the feature design process of object-based image analysis, can be replaced by CNNs to achieve better object identification accuracy. Hence, OCNN (the combination of object-based image analysis and CNN) was proposed to facilitate the extraction and utilization of deep features (Zhang *et al.* 2018, Fu *et al.* 2018, Lv *et al.* 2018a, 2018b, Zheng *et al.* 2020).

In OCNNs, OCP selection plays an important role in object identification. To date, two methods are widely adopted in OCP generation. Certain studies selected object (polygon) centroids (CID) as OCPs (Fu *et al.* 2018, Lv *et al.* 2018a, Chen *et al.* 2019, Zhang *et al.* 2020, Ghorbanzadeh *et al.* 2020). However, such an approach may lead to misclassification for complex objects. Another approach is to generate multiple OCPs in objects, notably the random generation method (RAND_L) proposed by Lv *et al.* (2018b). However, such an approach tends to general OCPs that are located on objects' boundaries, potentially leading to increased uncertainties. Based on random

generation methods, an overall position generation method (RAND_Z) combined with the object's CIDs has been proposed for urban functional zone classification (Zhou *et al.* 2019). Several morphology-based approaches for the generation of OCPs have been proposed. For example, the OCP analysis (OCPA) method based on the minimum moment bounding (MB) box was proposed to overcome the uncertainties of random methods (Zhang *et al.* 2018); the OCPs are distributed near the long axis of the moment box. The patch-based method (PBM) was proposed to integrate checkboard segmentation, ignoring the objects in the images (Sharma *et al.* 2017). In another study (intersect of patch and object, IPO) (Tong *et al.* 2020), the results of multiresolution segmentation and checkboard segmentation were combined. The multiresolution segmentation is used to generate the objects in the image, while checkboard segmentation is applied to generate patches that are used as the CNN input. Objects are identified according to the intersection by the multiresolution segmentation and checkboard segmentation (Tong *et al.* 2020). However, the aforementioned methods fail to meet the requirements of a robust representation of the entire object. A method that is able to generate OCPs with even distribution to represent the whole object is needed.

3. Methodology

This study aims to integrate object-based image analysis and CNNs. To illustrate the robustness and accuracy of our proposed method, seven comparative approaches were applied using five standard CNNs. VHR images were first segmented into representative geographical objects. Five typical CNN modes were simultaneously trained using labeled data. The OCPs within objects were generated using the proposed BTS and other comparative methods. Further, CNN models were used to extract deep features from the VHR images. Based on the positions of OCPs and pretrained CNN models, segmented objects were identified. Finally, the classification results obtained from OCNNs were compared, analyzed, and summarized.

3.1. Image segmentation

Based on the bottom-up theory that follows a minimum heterogeneity principle, the multiresolution segmentation method is often used to segment images into meaningful objects that are internally connected by guaranteeing that the pixels inside the objects are homogeneous (Rabiee *et al.* 1996). Our following analysis is based on segmented objects via multiresolution segmentation.

3.2. Binary tree sampling

As mentioned above, several approaches have been proposed to address the generation of OCPs, such as CID, RAND_L, RAND_Z, OCPA, PBM, and IPO. However, all these methods have their intrinsic limitations. To improve the robustness and universality of OCP generation, we proposed a BTS approach to generating evenly distributed OCPs within objects.

The proposed BTS generates suitable OCPs by dividing an object (polygon) into two sub-objects and recursively dividing the resulting sub-objects until the OCP number meets the requirements specified in Equation (1):

$$S(n) = \begin{cases} P_{s1} & , n = 1 \\ S(\lceil \frac{1}{a}n \rceil) + S(\lfloor \frac{a-1}{a}n \rfloor) & , n > 1 \end{cases} \quad (1)$$

where n presents the number of OCPs in an object, a is the ratio of the area of the parent object to that of the sub-object, and S^* is the kernel function for the BTS of objects. The BTS method divides an object under a divide-and-conquer strategy. The kernel function returns the centroid P_{s1} of the sub-object when n is one. The OCPs of inseparable sub-objects (the smallest sub-objects) are their centroids. The flowchart of the BTS method is shown in Figure 2.

The workflow of the proposed BTS method contains four major steps:

- Step 1. Calculating the convex hull of an object and generating the minimum bounding rectangle (MBR) of the convex hull.
- Step 2. Computing the centroid of the object and dividing the object into two sub-objects by creating a line vertical to the longest side of the MBR and passing through the midpoint of the longest side.
- Step 3. Recursively applying Steps 1 and 2 to each sub-object until the required number of OCPs is reached.
- Step 4. Local fine-tuning of OCPs (see Section 3.2.6).

3.2.1. Convex hull

Given a real vector space V and point dataset $X = (x_1, x_2, \dots, x_n)$, a convex hull is defined as the intersection of all convex sets containing X . In contrast to the convex hull of a disorder point dataset, the convex hull of a polygon is the smallest convex containing the polygon composed of order points. The convex hull of a convex polygon is the convex polygon, whereas that of a concave polygon can be calculated with a well-designed method.

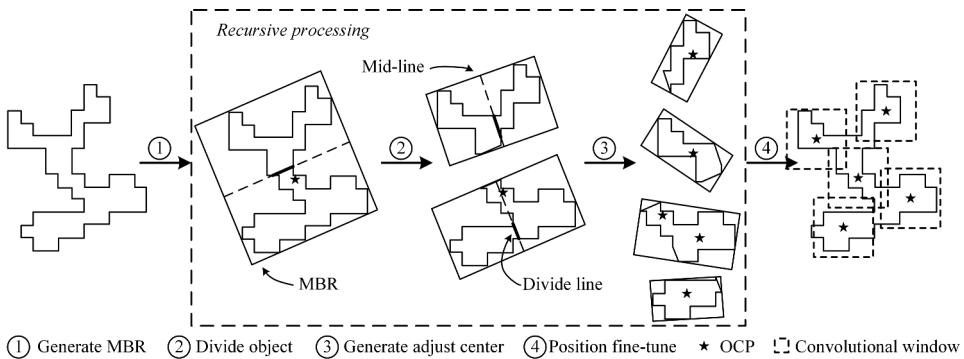


Figure 2. The workflow of the proposed binary tree sampling (BTS) method.

3.2.2. Minimum area bounding rectangle (MBR)

The MBR that suggests the bounding rectangle enclosing the minimum area (Lewis *et al.* 1997, Jiao *et al.* 2012) can be generated based on the convex hull of polygons. The MBR method is based on two axioms:

Axiom 1: The MBRs of polygons and their convex hulls are equivalent.

Axiom 2: At least one edge of the MBR of a polygon's convex hull coincides with an edge of the polygon.

The MBR of a polygon is calculated by iterating the edges of a polygon's convex hull.

3.2.3. Determining the number of OCPs

The number of OCPs for a certain object depends on its area and *SI*. For extremely small objects, one OCP is sufficient. The number of OCPs increases when an object's area increases, although their relationship does not necessarily follow a linear rule. Owing to the fact that objects usually contain homogeneous areas, the number of OCPs should no longer increase when the object's area reaches a certain level (a user-defined threshold). We define that objects' area can be divided into three categories: small, medium, and large. The *SI* indicates the smoothness of the border of an image object and is defined as the perimeter (*C*) of the object divided by four times the square root of its area (*S*), as shown in Equation (2):

$$SI = \frac{C}{4\sqrt{S}} \quad (2)$$

The smoother the border of an object, the lower the *SI* value. The more convoluted the border of an object, the higher its complexity index. Objects with a lower *SI* require only fewer OCPs. In contrast, objects with a high *SI* require more OCPs. Similar to the area, the *SI* of objects can also be divided into three categories based on a user-defined threshold. The number of OCP within objects can be calculated based on an intersection matrix composed of three categories of object areas and three *SI* categories, as shown in Table 1.

These thresholds are often defined by the user. S_i and A_i are the *i*-th interval for the *SI* and area, respectively. Thresholds between S_i and S_{i+1} (1.7 and 2.0 in this study) and between A_i and A_{i+1} (300 and 600 pixels in this study) are user-defined parameters.

3.2.4. Binary tree division

Recursion is an important divide-and-conquer strategy in the data processing. The proposed BTS can be divided into three steps: 1) dividing, 2) conquering, and 3) combining. An object is first divided into two smaller sub-objects by fusing the MBR and center-line of the MBR. The total number of OCPs required by the parent object is assigned to the

Table 1. The number of object convolutional positions (OCP) within objects using the intersection matrix.

	S_1	S_2	S_3
A_1	1	3	5
A_2	3	3	5
A_3	3	5	7

two sub-objects based on their area ratios. The sub-objects are then recursively split into two until the assigned OCP number is one (the 'base case'). In the base case, the object's centroid is considered to be one OCP. Finally, the centroids of all base cases are combined as the final OCPs of the original objects. The pseudocode below shows the recursive OCP generation progress.

Algorithm: BTS

Input: object Obj, number n, queue of OCPs Q

```

1.   Q = [];
2.   if n == 1:
3.     centroid = Obj.AdjustCentroid;
4.     Q.Add(centroid);
5.     return;
6.   (sub-object1, sub-object2) = CutObj(Obj);
7.   n1 = n * sub.object1.Area/Obj.Area;
8.   n2 = n * sub.object2.Area/Obj.Area;
9.   BTS(sub-object1, n1);
10.  BTS(sub-object2, n2);

```

In the BTS algorithm, Obj represents an object with two attributes (AdjustCentroid and Area). n is the number of OCPs. Q is the queue structure used to store the OCPs. The CutObj(*) function is used to split an object into two sub-objects.

The CutObj(*) function is based on the centerline of the MBR. An object is internally connected (as mentioned in Section 3.3); however, it is possible for the sub-objects to be unconnected after splitting. Thus, an internal adjustment approach is used, which is illustrated in Figure 3.

In Figure 3, an object is split into two parts along the cutting line. A sub-object (Part II) has multiple separate parts. By ignoring the maximum area part (lower Part II) of this sub-object, the other part (upper Part II) is merged into Part I. The adjusted final sub-objects are internally connected. The centroids of concave objects may fall outside of the objects. Illustrated in Figure 3, a position adjustment approach can be used to address this issue for both concave and convex objects. First, a vertical line (vertical to the longest side of the minimum bounding rectangle (MBR)) that passes through the centroid is constructed. The centroid further moves to the midpoint of the longest intersected line of the vertical line and the object.

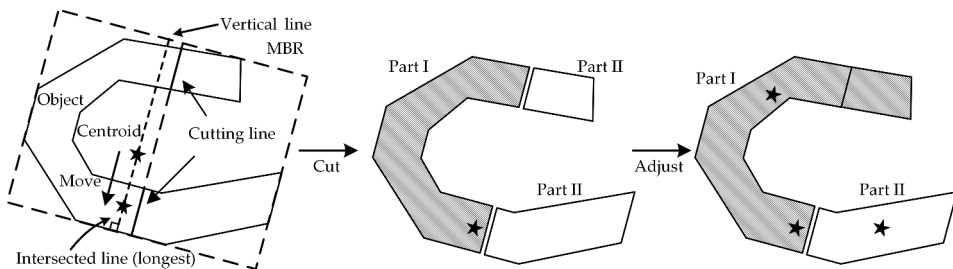


Figure 3. The flowchart of the adjustment of objects with internally unconnected parts.

3.2.5. Determining the centroid of an object

The centroids of objects are the gravity centers of the base cases. Assuming (x,y) is a point in an object, the centroid position (\bar{x}, \bar{y}) can be expressed by Equations (3) and (4) (Gere et al. 1977):

$$\bar{x} = \frac{\iint x d\sigma}{S} \quad (3)$$

$$\bar{y} = \frac{\iint y d\sigma}{S}, \quad (4)$$

where $d\sigma = d\bar{x} \cdot d\bar{y}$ is the differential area around point (x,y) .

3.2.6. Iterative local fine-tuning of OCPs

The OCPs are directly generated by the proposed BTS algorithm. An iterative local fine-tuning method was proposed to optimize the distribution of OCPs based on Voronoi graphs (Franz and Aurenhammer 1991, Okabe 2016). The adjusted OCPs can be calculated using Equations (5)–(7).

$$O_n = T(P_{n-1}) \quad (5)$$

$$P_n = S(O_n) \quad (6)$$

$$P_n = S(T(P_{n-1})) = (ST)^1 S(T(P_{n-2})) = (ST)^2 S(T(P_{n-3})) = \dots = (ST)^n P_0, \quad (7)$$

where P_0 represents the initial location of an OCP generated by the BTS, n is the number of iterations defined by the user, T^* is a function for the generation of Voronoi graphs based on n and the original object, and O_n is the n -th Voronoi graph. The new positions are generated using the function S^* based on the same principle as that of the object centroid. The symbol ST represents function S^* and function T^* . The final result P_n represents the locations after n iterations. Figure 4 shows the workflow of OCP optimization ($n = 5$).

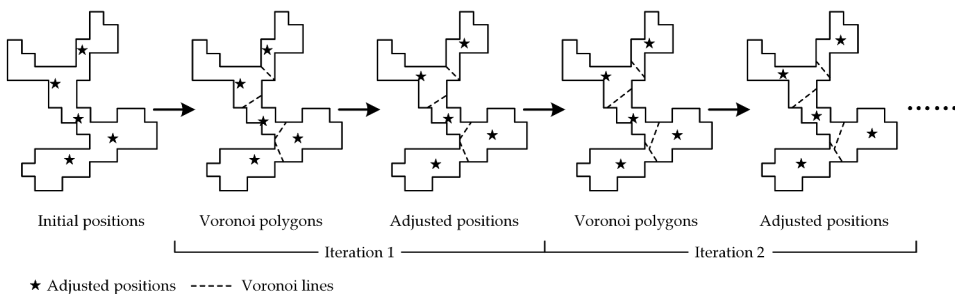


Figure 4. The workflow of iterative local fine-tuning of object convolutional positions (OCPs). Initial positions are the raw positions generated via previous steps.

3.3. Comparative methods

Six comparative OCP generation methods (CID, RAND_L, RAND_Z, OCPA, PBM, and IPO) and U-Net were implemented to illustrate the superiority of BTS and the importance of selected OCPs. U-Net is a standard and classical network with an end-to-end structure, which has been widely adopted by numerous applications. Given the popularity of U-Net, Figure 5 only shows the basic principles of the other six comparative methods.

In CID (Figure 5(a)), the centroid (gravity center) of an object is selected as the OCP (Fu *et al.* 2018). For objects with regular shapes, centroids are located inside the objects. However, for objects with irregular shapes, the centroids may be located outside of the objects, leading to great uncertainty for subsequent analysis.

RAND_L (Figure 5(b)) generates OCPs within objects via a two-step random sampling method (Lv *et al.* 2018b). An object is first partitioned into multiple triangles; the vertices of the triangles are the corner points of object boundaries. A triangle is randomly selected from the partitioned triangles. The envelope of the triangle is then used as an extent to randomly generate a point in the envelope. If the point is located on the triangle, the next point is generated under the same process. Otherwise, the point is mirrored inside the triangle.

The concept of RAND_Z (Figure 5(c)) is the same as that of RAND_L. Different from RAND_L, however, the processing unit of RAND_Z is the whole image (Zhou *et al.* 2019). RAND_Z first generates points within the whole image and then determines points that are located in the objects as OCPs. RAND_Z shares the limitations as RAND_L.

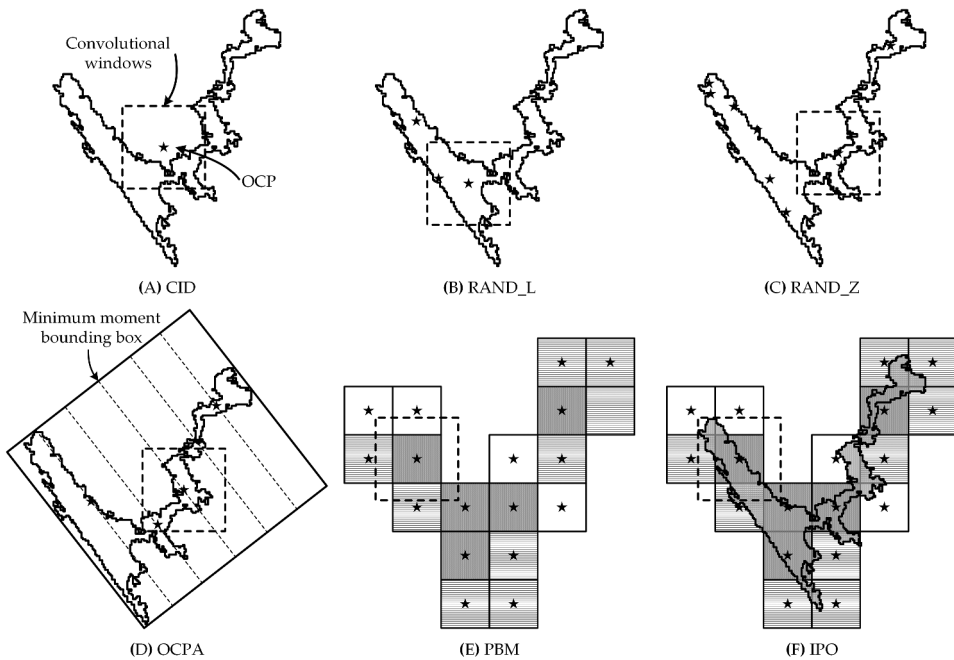


Figure 5. Six comparative OCP generation methods. (a) CID (object's centroids), (b) RAND_L (random method), (c) RAND_Z (overall position generation method), (d) OCPA (object convolutional position analysis), (e) PBM (patch-based method), and (f) IPO (intersect of patch and object).

OCPA (Figure 5(d)) calculates OCPs using the major and minor axes of the MB box of the object inertia (Zhang *et al.* 2018). The MB box is the MBR based on the moment orientation of an object. The MB box is equally divided into a specific number (depending on the number of OCPs) of sub-boxes along the major axis. Objects in the MB box are also divided into sub-objects. The middle points of lines that intersect two sub-objects are selected as OCPs. However, the OCPA method ignores the shape of the objects.

In PBM (Figure 5(e)), the sampling of OCPs is based on checkboard segmentation with equal squares (patches) in images (Sharma *et al.* 2017). Thus, the center of each patch becomes the OCP for the CNN. Theoretically, the PBM-based method does not fall into the category of object-based methods.

The IPO (Figure 5(f)) incorporates the concepts from the patch- and object-based methods. The IPO uses the patch-based classification result to identify objects that are covered by multiple patches (Tong *et al.* 2020). These objects are defined based on their patches intersecting with the highest proportion of the area. The limitation of IPO is that it fails to consider object information.

3.4. Standard CNN models

The CNNs extract abstract features using sub-images with square shapes as initial data. A CNN usually contains multiple deep layers that include the convolutional, max pooling, average pooling, batch normalization, and local response normalization layers. Convolutional layers are a series of filters with small sizes, which are used to extract spatial-spectral information from the input data. Pooling layers aim to reduce the amount of data and improve the calculating efficiency. Normalization layers are intermediate layers between convolutional layers and the activation function. They can normalize convoluted data to facilitate network convergence. The activation function is implemented after each normalization to determine whether a certain node is activated. Based on these well-designed layers, CNNs can extract deep features with useful information from raw input data. At the end of a CNN model, input data are convoluted and pooled into a one-dimensional vector, followed by several fully connected layers and classifiers (e.g. softmax) to generate the final results. In the study, we used standard CNNs that include AlexNet, VGGNet (VGG16 and VGG19), GoogleNet (Inception_v3), and ResNet.

3.5. OCNN

OCNNs that use segmented objects as processing units were proposed to address the inefficient computing of the pixel-wise CNNs and pixel-level errors in VHR image classification. However, the segmented objects are usually with complex shapes; therefore, they cannot be directly fed into the networks. OCNNs use multiple points distributed in the objects as centers of convolutional windows to predict the semantic category of each OCP. Based on the statistics of OCPs' categories, all pixels within the object are assigned to a certain category via majority voting. In this study, we fine-tuned the fully connected layers of AlexNet, VGG16, and VGG19 using transfer learning techniques (earlier layers before the fully connected layer used pre-trained parameters). The Inception_v3 and ResNet were retrained.

3.6. Accuracy assessment

The overall accuracy (*OA*) and *f1-score* based on the confusion matrix were used to measure the performance with respect to the identification of objects. The confusion matrix yields four results: true positive (*TP*), false positive (*FP*), false negative (*FN*), and true negative (*TN*). The *TP* represents the number of correctly identified samples, *FP* represents the number of misidentified samples, *FN* is the number of samples of one category misidentified as samples of another category, *TN* is the number of samples of another category that was identified correctly. The *OA*, *recall*, and *precision* can be expressed by the following equations:

$$OA = \frac{TP + TN}{TP + FP + FN + TN} \quad (8)$$

$$recall = \frac{TP}{TP + FN} \quad (9)$$

$$precision = \frac{TP}{TP + FP} \quad (10)$$

The *f1-score* is based on *recall* and *precision* and is defined by Equations (11) and (12):

$$f1_k = 2 * \frac{precision_k * recall_k}{precision_k + recall_k} \quad (11)$$

$$f1-score = \frac{1}{n} \sum f1_k, \quad (12)$$

where $f1_k$ is the k -th category and *f1-score* is the mean $f1_k$ of all categories.

3.7. Evaluation of OCPs

Three indices were utilized to measure the robustness of OCPs' distribution, i.e. the area index, The cover index, and the distribution index. The area index was used to measure the area fluctuation in the Voronoi graphs of the objects. The cover index, calculated as the ratio of the convex hull of the convolutional positions to that of the objects, measures whether objects are sufficiently covered. The distribution index aims to measure the evenness of the sample distribution. These three indices can be expressed as follows:

$$AreaIndex = \sigma^2 = \frac{\sum_{i=1}^n (\bar{s} - s_i)^2}{n - 1} \quad (13)$$

$$CoverIndex = \frac{S'}{S} \quad (14)$$

$$DistributionIndex = \frac{u - \sigma}{u} + k \cdot CI, \quad (15)$$

where n is the number of OCPs, s_i ($i= 1, 2 \dots n$) is the area of the i th Voronoi graph, and \bar{s} is the average area of all Voronoi graphs. A small area index indicates that the distribution of OCPs is close to an even distribution. Conversely, a high cover index indicates

a widespread distribution. A desirable sampling method is expected to have a low area index and a high cover index. The cover index ranges from 0 to 1. The standard deviation σ and mean u of the area of all Voronoi graphs were used to normalize the area index. As a more comprehensive measurement, the distribution index expresses the linear combination of cover index and normalized area index. In addition, k (a user-defined constant; $k = 1$ in this study) can be used to adjust the weights of two other indices. A high distribution index indicates that the generated OCPs are generally representative and evenly distributed.

4. Study area

We downloaded two VHR images with three bands, one located in Sacramento and the other one located in Auckland, from Google Earth provided by WorldView. Complex objects were further segmented from VHR images.

Two images in Auckland and Sacramento were captured on 17 August 2018 and 3 April 2018, respectively. The spatial resolutions of both images are the same (60 cm per pixel), and the images contain three panchromatic bands. The image in Sacramento covers $4,960 \times 6,971$ pixels, and the image Auckland covers $8,185 \times 7,850$ pixels.

Randomly stratified samples were selected from the study areas. Each category contains $\sim 1,600$ samples. The sample set was divided into two subsets: training and validation. The training set consisted of 62.5% of the samples for each category, while the validation set contained the remaining 37.5% of samples (Zhang *et al.* 2018). The detailed numbers of samples corresponding to different categories are listed in Table 2.

Sacramento contains ten categories, while Auckland contains nine categories (Bare soil is not available in Auckland). Figure 6 shows some very irregular objects segmented by the multiresolution segmentation algorithm.

As shown in Figure 6, categories greatly differ in shapes, sizes, and textures. For example, residential houses, water bodies, and wetlands have a relatively simple appearance. Asphalt roads, cement roads, shadows, and vegetation have complex shapes with a high SI . Vegetation that includes the greenbelt and trees has varying shapes with strong homogeneity. Industrial buildings are the most complex objects in terms of their textures, shapes, and materials.

Table 2. Sample numbers corresponding to different categories in two study areas.

Category	Auckland	Sacramento
Asphalt road	2,000	2,000
Bare soil	—	1,600
Cement road	1,600	1,600
Industrial building	1,598	1,598
Parking space	—	1,500
Residential house	1,600	1,600
Shadow	1,800	1,800
Truck	1,472	—
Vegetation	1,800	1,800
Water body	1,600	1,600
Wetland	692	692



Figure 6. Irregular objects with different shapes segmented by the multiresolution segmentation algorithm.

5. Results and analysis

The image segmentation process by the multiresolution segmentation algorithm involves three key parameters: scale, shape, and compactness. Efforts have been made to investigate scale selection in image segmentation (Al-Huda *et al.* 2020). Here, scale affects the size of the segmented object, while shape and compactness control the form of the object. After investigating these two study areas, we selected [scale: 25, shape: 0.5, compactness: 0.5] for Sacramento and [scale: 30, shape: 0.4, compactness: 0.5] for Auckland via a trial-and-error process.

5.1. Model training and analysis for OCNN

All five models were trained using the training dataset described in Section 3. The learning rate, initial number of epochs, and batch size dropouts of AlexNet, VGG16, and VGG19 were the same, respectively set as 0.01, 200, and 0.5. The initial learning rate, number of epochs, and batch size of Inception-v3 and ResNet were the same, respectively set as 1e-3, 200, and 50. In addition, the learning rate of Inception_v3 and ResNet was set to 1e-4 when epoch > 50, 1e-5 when epoch > 100, and 1e-6 when epoch > 150. We documented train loss value, validation loss value, validation accuracy, convergence epochs, and convergence time in Figure 7.

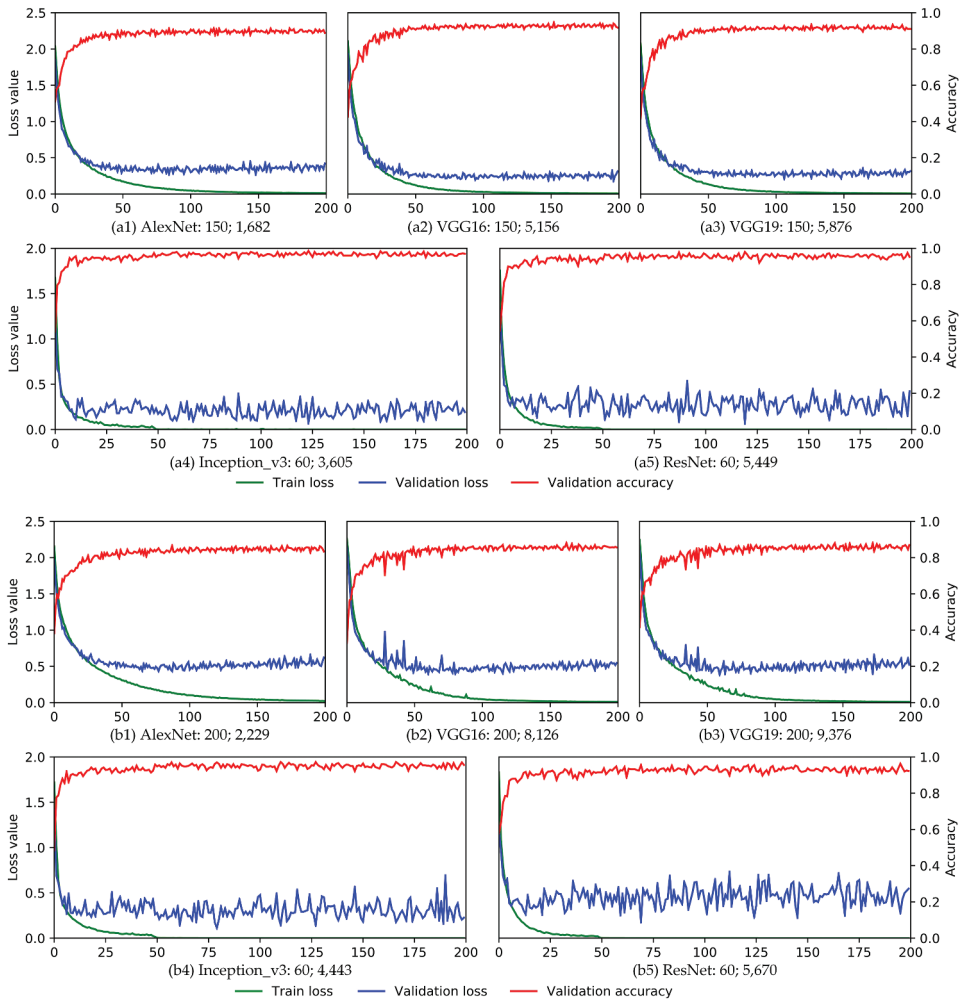


Figure 7. Training statistics for five models in two study areas. (a) Training statistics for the Auckland image; (b) training statistics for the Sacramento image. The name of sub-figures represent [model name: convergence epochs, and convergence time(seconds)]; for example, [AlexNet: 150; 1,682].

Although the initial epochs of all five CNNs were set as 200, the convergence epochs were different. The convergence epochs of AlexNet, VGG16, and VGG19 on Auckland were all 150; the convergence epochs of Inception_v3 and ResNet were both 60. The convergence epochs of AlexNet, VGG16, and VGG19 on Sacramento were all 200; the convergence epochs of Inception_v3 and ResNet were both 60. Despite their longer training time, Inception_v3 and ResNet took fewer epochs to converge compared to AlexNet, VGG16, and VGG19.

5.2. Object identification

This study aims to investigate the effects of OCPs on the OCNN in object identification. We further divided the dataset into three subsets (excluding training set and validation set) based on the required OCP numbers: 1) all objects, 2) objects with more than one OCP,

and 3) objects with more than three OCPs. The identification performance of the proposed method was tested using these three subsets of objects based on the pre-trained CNN model. Figure 8 shows the OA of the proposed BTS method and other comparative methods based on five standard CNNs.

We observe the great performance of BTS in identifying irregular objects. The highest identification accuracies of all objects obtained by BTS for Auckland were 90.24% (AlexNet), 93.18% (VGG16), 92.25% (VGG19), 94.64% (Inception_v3), and 93.93% (ResNet52). For Sacramento, BTS also reached high identification accuracies: 87.78% (AlexNet), 88.03% (VGG16), 88.57% (VGG19), 96.88% (Inception_v3), and 96.11% (ResNet52). The above results show that diverse CNNs yield very similar object identification accuracy. The performances of the competing methods can be divided into three notable echelons. The BTS and OCPA (considering the distribution of OCPs in objects) represent the first echelon. The RAND_L and CID (locating OCPs within objects) represent the second echelon. The RAND_Z, IPO, and PBM represent the third echelon characterized by their large object identification uncertainties. In general, the proposed BTS achieves the best performance, followed by OCPA, RAND_L, CID, RAND_Z, IPO, and PBM.

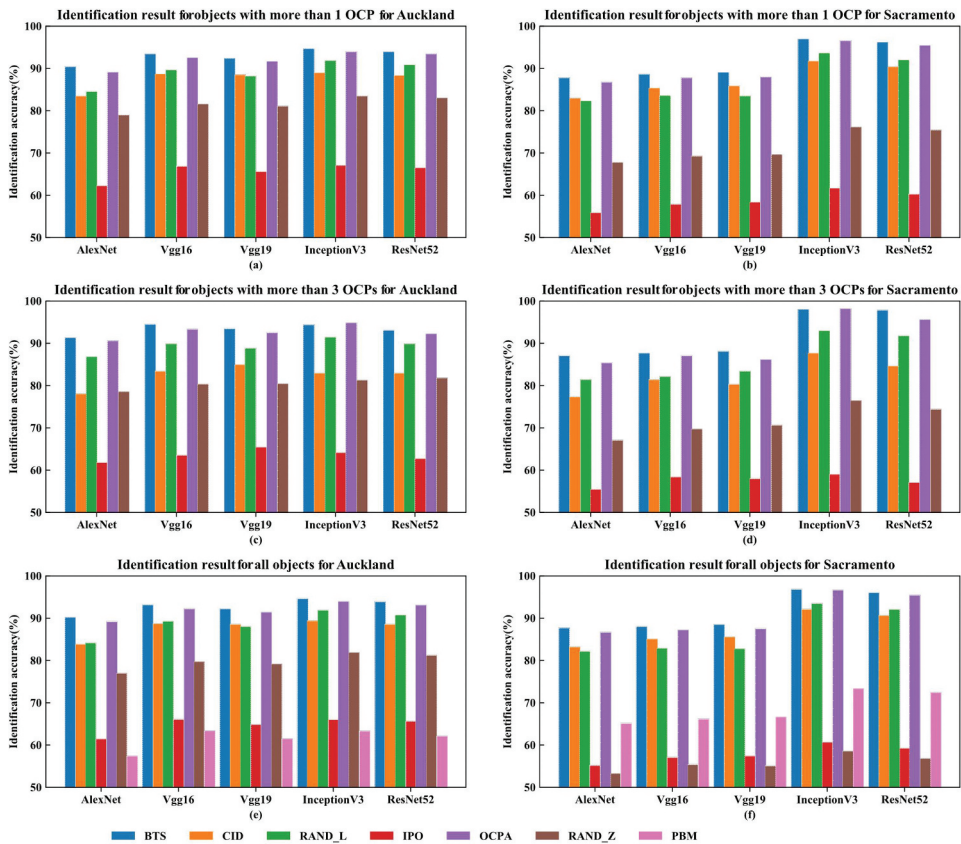


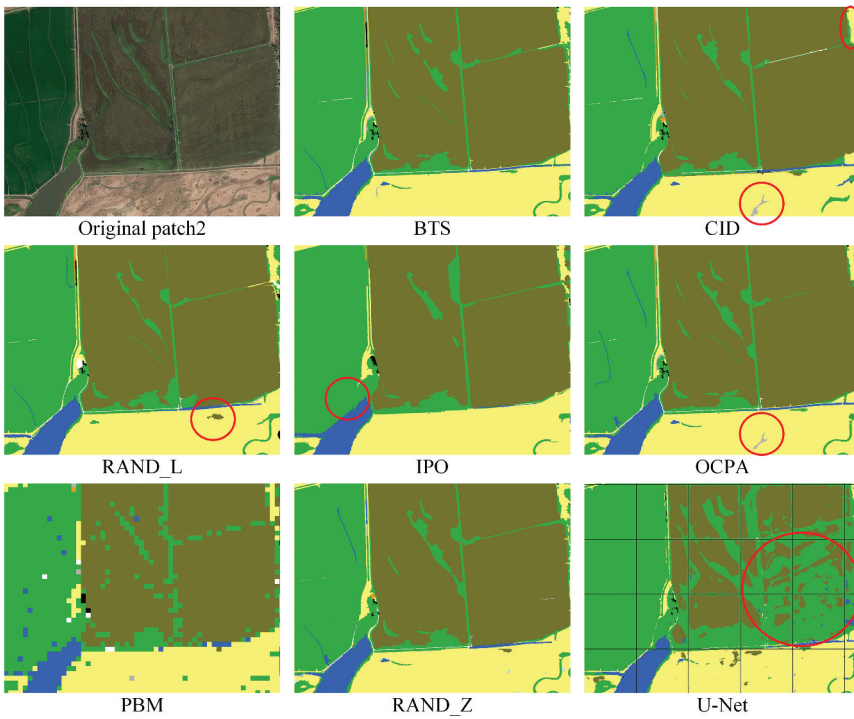
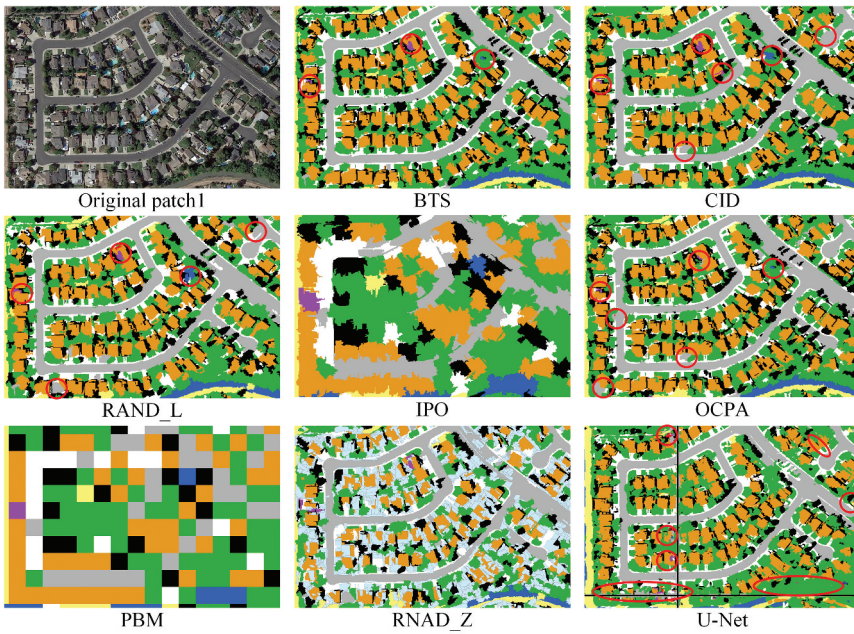
Figure 8. Overall accuracy (OA) in object identification of the proposed binary tree sampling (BTS) algorithm and comparative methods.

The identification results prove that the CNNs achieve better performance when the OCPs are located inside the objects. The BTS uses morphological features (objects' shapes and areas) to generate evenly distributed OCPs. Thus, the features of objects can be fully represented based on the derived OCPs. In comparison, OCPA only guarantees that the generated OCPs are located inside the objects but fails to consider their distribution. Despite that RAND_L can generate multiple OCPs, many OCPs are located on or near the boundaries of the objects, responsible for its great uncertainties. The CID method selects polygon centroids as OCPs. However, only when objects are concave polygons does such an approach locate OCPs within objects. Methods that include RAND_Z, IPO, and PBM do not consider the locations of OCPs at all. The generation of OCPs by RAND_Z is completely random, potentially leading to undersampled OCPs in an object, which greatly affects the object identification accuracy. The semantic segmentation results in urban and rural settings using OCNNs with different OCP generation methods and U-Net are presented in Figure 9.

From visual interpretation, the semantic segmentation performance presents notable differences between urban and rural areas. In the rural setting, U-Net achieved great performance, so did BTS, CID, RAND_L, IPO, RAND_Z and OCPA. In comparison, the performance of PBM was considered poor. In the urban setting, however, the performance of U-Net was reduced, evidenced by many misclassified residential buildings. In comparison, OCNNs coupled with OCPs generated by BTS, CID, RAND_L, and OCPA performed well, with our proposed BTS achieving the best performance in semantic segmentation.

Tables 3 and 4 show the *f1-score* of each category in the study areas. The trends of the overall *f1-score* of all categories confirmed the aforementioned object identification results. The proposed BTS was more effective than other competing methods. Methods that consider the spatial morphology (BTS and OCPA) outperformed those that do not (e.g. CID, RAND_L, and RAND_Z). For most categories, the proposed BTS achieved the highest *f1-scores*. We also observed that methods that can generate multiple OCPs generally outperformed methods that only generate a single OCP.

We observed that BTS and OCPA achieved the highest *f1-scores* for most categories, with BTS representing the majority. Despite that U-Net is able to predict tiny objects (at pixel-level) that OCNN can not, its performance in urban areas was unsatisfactory. In terms of workload in building training samples, end-to-end networks like U-Net require pixel-by-pixel labels, which are labor-intensive and time-consuming to derive. In comparison, OCNNs are based on image objects segmented by automatic segmentation algorithms, only requiring very limited workloads on sample labeling. Therefore, we believe OCNN methods are more suitable in practical applications, especially in situations that lack training samples. We observed that the object identification performance of the same category notably differed when different OCP generation methods were used. The performance gaps in water bodies and wetlands were insignificant. The performance gaps in asphalt roads and industrial buildings were notable, while those for parking spaces, residential houses, and vegetation were very notable. The performance gaps for the same category can be explained from the perspective of feature representation via OCPs, as the identification of objects is significantly affected by the OCP selection strategy. The OCPs generated from BTS, OCPA, and RAND_L ensure that centers of windows are located inside the objects, thus decreasing the possibility of



Asphalt road	Bare soil	Cement road	Container	Industrial building
Residential house	Shadow	Vegetation	Water body	Wetland

(b) The rural setting

Figure 9. Semantic segmentation results in urban and rural settings using OCNNs with different OCP generation methods and U-Net. (a) The urban setting; (b) The rural setting.

Table 3. The *f1*-scores of the OCNNs with different OCP generation methods and U-Net for each category in Auckland.

Models	Methods	Ar	lb	Ps	Rh	Sd	Cr	Vg	Wa	Wl
AlexNet	BTS	91.34**	83.82**	87.44	93.15**	93.53**	80.54	86.71**	96.89	82.29**
	CID	87.39	78.94	85.71	88.23	83.85	67.76	71.72	90.81	71.65
	RAND-L	88.48	81.22	79.40	86.49	81.47	66.00	74.54	95.55	80.45
	IPO	76.28	68.65	50.65	56.67	33.74	18.87	48.54	92.20	73.79
	OCPA	91.29	82.89	87.84**	92.52	89.84	81.85**	82.33	97.05**	76.76
	PBM	71.08	63.94	49.27	55.11	30.62	18.84	46.92	86.85	63.32
VGG16	RAND-Z	87.36	80.64	69.80	77.59	75.06	60.69	79.91	96.05	83.16
	BTS	94.52*	89.32	92.82	94.48**	94.35**	83.91**	90.83**	98.08**	87.24
	CID	92.81	86.95	90.32	91.28	87.21	69.21	78.36	95.22	78.26
	RAND-L	92.96	87.81	88.28	89.77	84.92	76.07	83.97	96.52	84.79
	IPO	78.74	73.81	60.10	61.86	38.44	24.66	53.09	95.42	82.18
	OCPA	93.98	89.77**	94.08**	93.70	92.01	82.63	86.68	97.05	88.04**
VGG19	PBM	75.14	69.91	58.47	60.85	37.31	26.64	54.77	92.44	71.92
	RAND-Z	90.25	83.16	74.03	80.36	77.57	65.71	83.89	96.91	86.46
	BTS	93.93**	88.19	92.58**	93.05	92.56**	83.33	89.91**	97.73** ()	86.91**
	CID	92.57	86.66	90.05	91.19	86.53	74.10	80.24	93.13	71.96
	RAND-L	91.81	87.26	85.29	89.25	82.25	76.16	81.57	96.18	82.54
	IPO	76.63	72.49	56.27	63.11	37.85	22.33	53.57	93.97	78.64
Inception-V3	OCPA	93.83	88.52**	92.43	93.40**	89.51	84.41**	85.04	97.08	81.29
	PBM	73.39	69.36	57.14	60.99	34.94	23.07	51.89	90.12	65.09
	RAND-Z	89.65	83.76	73.99	79.62	75.46	66.47	82.48	96.91	86.91
	BTS	95.54**	93.34**	93.75**	97.53**	94.17**	86.88	89.12**	98.62**	89.77**
	CID	93.63	90.30	90.54	91.60	83.03	71.38	78.37	96.23	83.24
	RAND-L	93.98	92.06	89.01	95.20	88.33	78.93	85.13	97.60	88.76
ResNet	IPO	78.05	77.11	55.04	64.41	29.71	23.10	54.11	95.36	86.21
	OCPA	95.40	92.59	93.30	97.06	91.99	88.83**	88.26	97.76	85.08
	PBM	75.24	73.55	56.17	62.56	28.48	23.02	51.12	93.63	79.35
	RAND-Z	91.24	88.76	74.53	84.05	80.60	66.26	84.98	96.61	94.74
	BTS	89.65**	91.40**	91.42**	96.35**	95.58**	84.66**	91.28**	97.94**	88.37**
	CID	92.44	88.74	91.46	91.49	84.3	69.21	78.25	96.15	75.79
U-Net	RAND-L	92.49	90.45	86.93	94.16	89.92	76.53	85.72	96.94	83.33
	IPO	78.12	76.79	56.63	63.36	30.26	22.54	51.11	96.10	89.39
	OCPA	94.5	91.08	90.77	95.64	93.25	84.43	89.43	97.93	83.80
	PBM	73.86	72.63	54.76	62.38	28.54	22.56	49.86	92.78	75.27
	RAND-Z	90.36	88.20	74.84	83.78	79.77	64.16	85.23	97.42	91.71
	U-Net	90.44	62.69	83.04	82.18	72.04	62.69	68.91	92.77	82.43

Table 4. The *f1*-scores of the OCNNs with different OCP generation methods and U-Net for each category in Sacramento.

Models	Methods	Ar	Bs	Cr	Ct	lb	Rh	Sd	Vg	Wa	Wl
AlexNet	BTS	87.41**	93.76**	81.01**	84.15**	92.34**	90.32**	84.55**	80.87**	97.32**	83.05**
	CID	84.17	90.87	74.09	79.80	88.12	85.85	77.59	76.92	93.50	76.09
	RAND-L	83.04	91.91	72.79	80.67	89.88	83.48	76.59	72.20	91.57	79.76
	IPO	57.86	82.91	30.84	46.05	75.14	44.47	27.95	44.44	76.06	82.08
	OCPA	87.12	93.52	79.16	82.93	90.50	89.58	84.07	79.06	95.30	80.90
	PBM	54.31	81.73	28.46	44.17	72.53	44.78	27.43	42.77	72.41	81.56
VGG16	RAND-Z	80.28	89.05	53.12	52.34	78.97	64.31	58.03	62.43	91.78	82.95
	BTS	89.84**	95.47**	77.54	91.93**	94.30**	95.07**	79.76	77.81	95.94	90.24**
	CID	87.26	93.04	76.09	75.88	91.15	89.88	75.34	78.73	95.00	80.63
	RAND-L	84.29	92.26	71.29	77.65	90.23	87.07	74.38	74.04	93.67	79.76
	IPO	59.99	82.44	37.84	48.23	80.78	48.27	25.39	41.63	80.73	80.93
	OCPA	88.51	94.21	77.98**	80.54	91.95	90.95	81.92**	80.66**	96.69**	83.33
U-Net	PBM	55.77	81.11	35.81	44.79	79.22	49.28	26.4	40.44	76.69	80.23
	RAND-Z	81.57	88.93	52.28	49.22	81.65	67.06	57.55	65.52	94.12	85.08

(Continued)

Table 4. (Continued).

Models	Methods	Ar	Bs	Cr	Ct	Ib	Rh	Sd	Vg	Wa	Wl
VGG19	BTS	89.64**	95.17**	80.47**	82.22**	94.77**	91.41	83.16**	80.65**	98.43**	87.06
	CID	86.95	93.21	77.88	75.07	91.15	90.44	74.50	80.12	95.29	85.87
	RAND-L	83.23	90.95	71.79	79.54	92.72	86.84	75.31	72.51	93.98	84.15
	IPO	59.87	82.71	36.55	45.00	79.40	47.34	28.60	45.46	81.61	80.72
	OCPA	88.76	94.17	78.64	80.98	92.78	92.46**	82.44	79.49	96.55	87.42**
	PBM	55.19	80.38	33.92	42.77	78.21	48.84	28.60	42.64	77.03	78.83
	RAND-Z	83.30	88.33	53.57	48.03	81.41	67.76	57.92	65.65	94.43	88.24
Inception-V3	BTS	97.26**	98.71**	95.73**	98.45**	97.50**	98.72**	96.75	94.00**	98.27	84.70**
	CID	95.16	97.43	88.22	86.69	94.77	93.65	84.27	88.18	97.50	83.80
	RAND-L	94.27	98.07	89.95	94.83	95.65	96.88	90.21	88.25	96.68	84.85
	IPO	64.21	85.50	40.54	45.05	84.41	51.14	27.22	46.60	84.84	86.90
	OCPA	96.97	98.51	95.44	96.59	97.14	98.43	97.25**	93.89	98.59**	83.23
	PBM	60.86	83.43	39.04	44.75	82.30	51.89	25.13	44.79	82.49	82.76
	RAND-Z	90.24	93.11	70.53	59.5	85.34	74.17	69.62	74.31	96.00	84.39
ResNet	BTS	96.07**	98.46**	94.95**	96.00**	97.03**	97.56**	96.05**	93.39**	97.63	85.38
	CID	93.44	97.47	84.53	85.88	94.10	92.51	83.54	86.33	95.74	83.42
	RAND-L	92.69	97.05	88.16	92.16	94.52	95.11	88.32	87.09	95.51	83.53
	IPO	63.26	85.64	39.06	40.66	81.48	50.47	25.56	45.24	83.77	80.46
	OCPA	95.88	98.32	93.65	93.69	96.08	96.41	94.79	92.84	97.80**	87.72**
	PBM	59.23	83.03	36.74	40.39	79.21	51.67	24.13	42.99	79.53	74.42
	RAND-Z	89.39	92.61	69.33	57.62	85.08	72.48	66.27	74.31	94.44	86.04
U-Net		85.63	94.19	70.40	54.05	91.11	87.29	60.00	80.56	94.14	74.67

Footnote: Ar: asphalt road; Bs: bare soil; Cr: cement road; Ct: container; Ib: industrial building; Ps: parking space; Rh: residential house; Sd: shadow; Vg: vegetation; Wa: water body; Wl: wetland.

misidentification caused by other objects in the patches. The above results confirmed that OCNN coupled with the proposed BTS strategy is with high robustness in identifying various types of objects.

5.3. OCP generation via BTS in extreme cases

In our experiments, the largest number of OCPs in an object was seven, and the median was three. However, other applications, such as skeleton detection, may require more OCPs. In this session, we conducted additional experiments on BTS, OCPA, and RAND_L in generating multiple OCPs within objects in extreme cases. Figure 10 presents the OCP generation results with a number of 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, for geographical objects and hand-drawn objects.

We observe that the proposed BTS is able to generate OCPs with an even distribution in all scenarios. The superiority of BTS is more notable when the number of required OCPs increases. The performance of OCPA is acceptable when the OCP number is less than seven. However, when the OCP number further increases, the OCPA algorithm assigns OCPs to the centerlines of objects. Compared with OCPA, the OCPs generated by the proposed BTS are able to represent the whole object in a comprehensive manner. In comparison, the RAND_L algorithm has notably large uncertainty given its randomness: 1) OCPs appear on the boundary when the number is three, and 2) OCPs are overconcentrated when the number of OCPs is larger than five. These uncertainties of RAND_L become more dominant with an increasing number of OCPs.

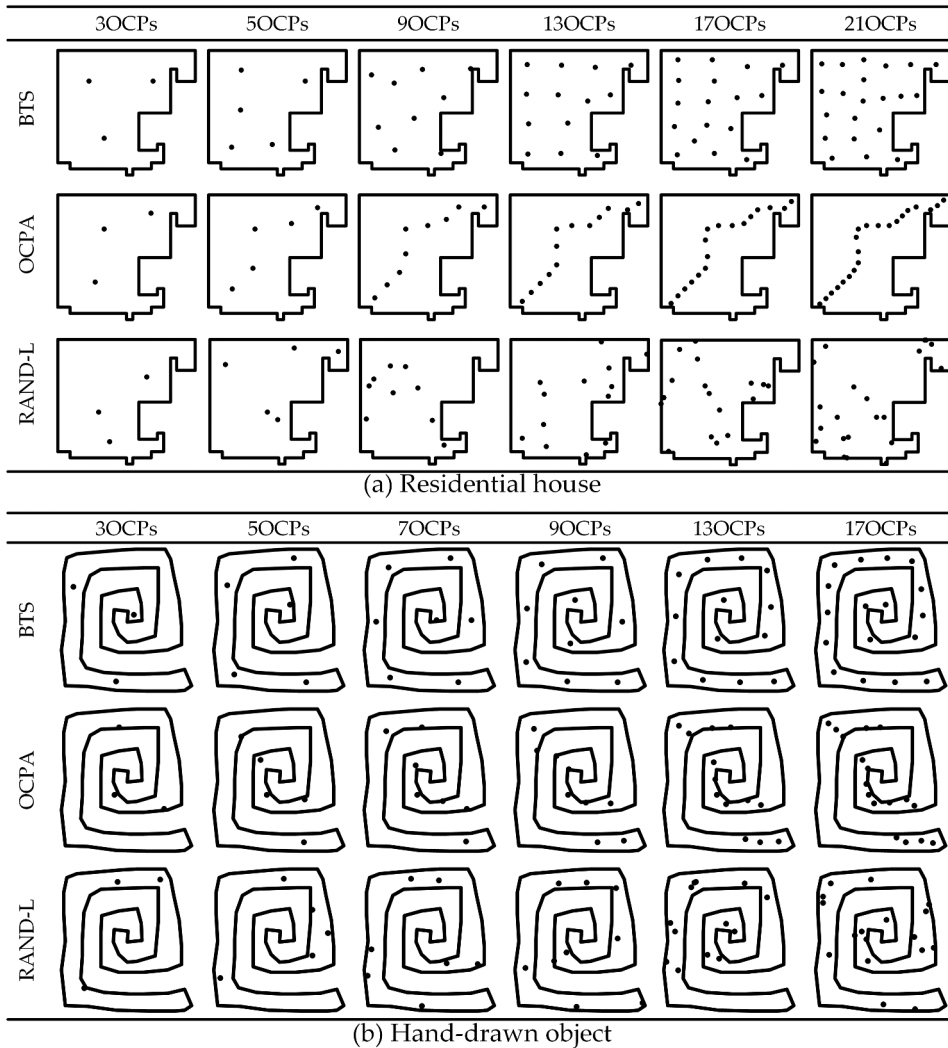


Figure 10. The object convolutional positions (OCP) generation results for (a) geographical objects (residential house), (b) hand-drawn objects with extremely irregular shapes. OCPs were selected with a series of numbers (i.e. 3, 5, 7, 9, 11, 13, 15, 17, 19, 21) for this experiment.

These extreme cases demonstrate the robustness and effectiveness of the proposed BTS that is expected to benefit other fields such as skeleton line detection and data sampling.

5.4. Evaluating the distribution of OCPs

To evaluate the distribution of OCPs, we proposed a distribution index measurement and further derived the distribution index of generated OCPs from multi-OCP methods, i.e. BTS, OCPA, and RAND_L. The higher the distribution index, the more evenly distributed the convolutional positions. The distribution index of the OCPs in the two study areas is shown in Figure 11. We noticed that the distribution index of the two study areas was similar. The distribution index obtained via the proposed BTS method (median values of

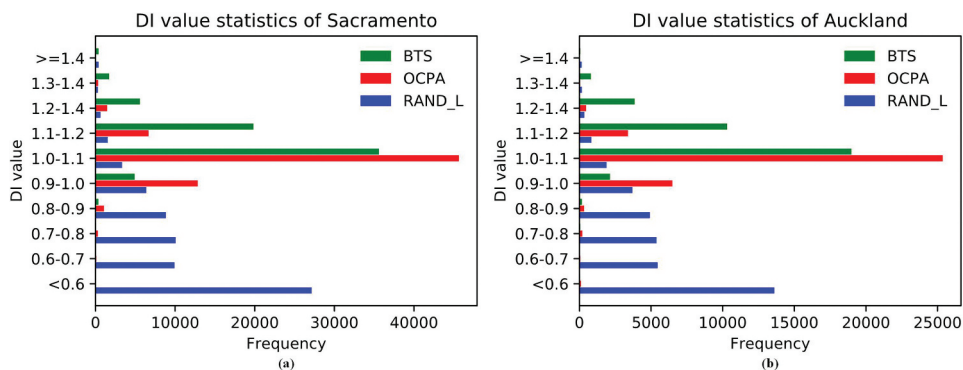


Figure 11. Statistics of distribution index values for the selected two images. (a) Sacramento; (b) Auckland.

1.082 and 1.085 for the two cases) were generally higher than those obtained with OCPA (median values of 1.025 and 1.023) and RAND_L (median values of 0.70 and 0.69). The above results again demonstrate the robustness and efficiency of the proposed BTS method in a quantitative manner. In general cases, it is easy to generate OCPs for regularly shaped objects such as ribbons, circles, and squares, while difficult for objects with complex shapes. Thus, the proposed distribution index serves as a novel measurement to evaluate the generation of OCPs, especially for objects with complex shapes.

6. Discussion

6.1. Deep feature representation through BTS

Deep features extracted by CNNs significantly differ from well-designed features. Given the varying sizes and shapes, these objects cannot be directly fed into CNNs. Hence, multiple partial deep features of objects are extracted by OCP-based CNNs to characterize the entire object. However, partial features have notable gaps with respect to the representation of the whole object. The object identification accuracy greatly varies when objects fail to be fully represented by the deep features obtained from OCP selection methods. Morphological attributes are important for deep feature representation. By considering the morphological attributes, the proposed BTS algorithm is able to generate OCPs with optimized distributions that benefit deep feature extraction, leading to high object detection accuracy. Figure 12 shows the attention maps of patches from BTS-generated OCPs, extracted from the layer 'Mixed_6e' of pre-trained Inception_v3.

A class activation map for a particular category indicates the discriminative image regions used by the CNN to identify that category (Zhou *et al.* 2016). In Figure 12, warm colors denote activated regions, and cool colors denote otherwise. The activated regions are generally the central and nearby pixels of patches; however, some activated regions fall outside of the objects. This means that, when classifying a certain object, CNNs can consider the features within that object as well as nearby features outside that object, as long as these features are activated. Building roofs generally share similar RGB reflectances as the sidewalks due to their similar materials (cement). If the activated

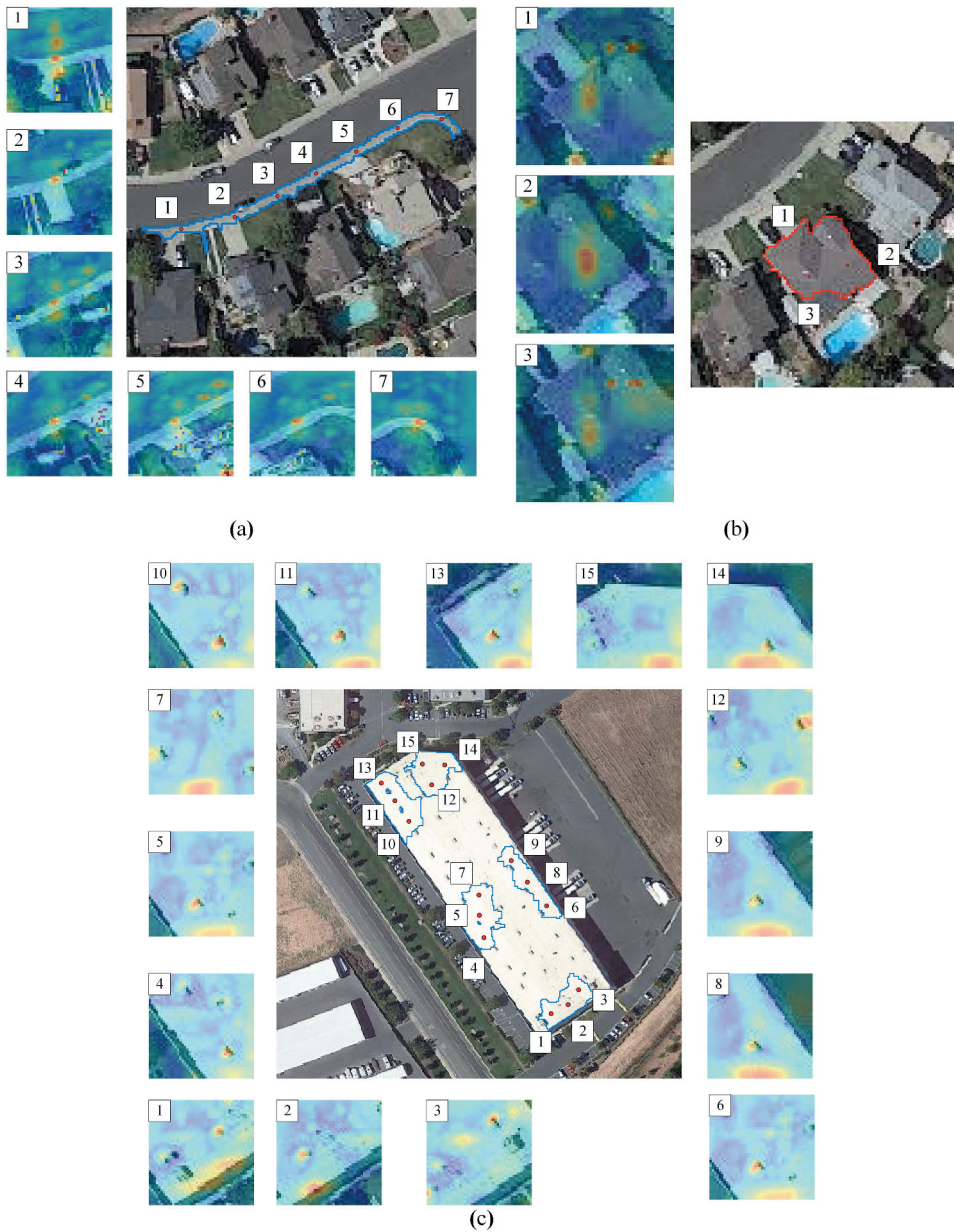


Figure 12. Example attention maps of patches from BTS-generated OCPs. (a) Cement road in a residential area with seven OCPs; (b) a residential building with three OCPs; (c) an industrial building with fifteen OCPs.

region of an industrial building is the same as sidewalks, misclassification is inevitable. Thus, accurate object identification requires sufficient activated regions that cover the object and its surroundings. The activated regions of Figure 12(c) are consistent with what we expected. Not only are the local regions of objects activated, but also these surrounding deep features. Earlier studies showed that, despite being trained on image-

level labels, CNNs have great capability in localizing objects, particularly important for fine-grained recognition (e.g. sidewalks and some factory buildings) where the distinctions between categories are subtle and focused image cropping (selecting convolutional positions) allows for better discrimination.

6.2. Efficiency issues of BTS

Given its binary tree structure, the time complexity of BTS is $O(mn\log_2(n))+\theta(m)$, where n is the OCP number of each iteration, m is the object number in the study area, O is the order of magnitude of the algorithm, and $\theta(m)$ is the time complexity of iterative local fine-tuning of OCPs. The space complexity of BTS is $O(\log_2(n))$ in terms of results stored in the global variable and the binary tree structure. We compared the time and space complexity of the proposed BTS with OCPA, which is the most advanced competing method to our best knowledge. Their comparison is presented in Table 5. A total of 9300 objects were implemented in this experiment (~10% of the objects in Sacramento).

We observe that the efficiency of the proposed BTS is lower than that of OCPA. However, we implemented data-parallel multi-threading technology to accelerate BTS. Table 6 shows the comparison between OCPA and multi-thread BTS.

In this experiment, there are two parameters that determine the running time for the proposed BTS, i.e. the batch size that controls the size of data written to disk, and the thread amount that defines the number of workers (related to the number of CPU cores). We notice that the running efficiency of BTS is very close to that of OCPA under the 1-thread scenario. With the increase in the number of threads, the running time of BTS continues to decrease until 16 threads. The running efficiency of 16-thread BTS is about 6.75 times that of single-thread BTS and about 5.85 times that of OCPA. Due to the thread scheduling issue, the running time of each thread differs, so does the running time of each batch. The CPU of our computer is Inter Core i7-7180X (8 physical cores, 16 virtual cores optimized on the physical cores by windows operation system) with 32GB memory. An excessive number of threads increases the management overhead. Thus, we recommended the number of threads as *physical cores* \times 2 (16 in our study) and the batch size as 500 ~ 1,000.

Table 5. The time and space complexity for OCPA and BTS.

Methods	Time complexity	Space complexity
OCPA	$O(mn)$	$O(n)$
BTS	$O(mn\log_2(n))+\theta(m)$	$O(\log_2(n))$

Table 6. Running time comparison between OCPA and multi-thread BTS.

Batch	BTS							OCPA
	1 thread	2 threads	4 threads	8 threads	16 threads	24 threads	32 threads	
250	270.5329	189.5830	120.0083	68.9590	44.9894	40.4144	42.1972	228.3523
500	263.0736	191.6844	117.6138	69.6178	45.6017	43.1583	46.8176	234.1499
750	259.6237	197.2496	122.3858	70.2478	46.4414	45.0426	52.3002	221.6957
1000	253.2123	202.5599	117.8896	73.0423	47.5945	47.2003	52.8278	228.1359

(Unit: Seconds)

7. Conclusions

Numerous studies have suggested that the generation of object convolutional positions (OCPs) largely determines the performance of object-based convolutional neural networks (OCNNs). In this study, we propose a morphology-based binary tree sampling (BTS) method that provides a reasonable, effective, and robust strategy to select OCPs for identifying objects via OCNNs. The proposed BTS method considers the shape attributes of segmented geographical objects and facilitates the generation of evenly distributed OCPs. Taking the object identification in land-cover and land-use classification as a case study, we compared the proposed BTS algorithm with six competing methods, i.e. CID, RAND_L, RAND_Z, OCPA, PBM, and IPO, to illustrate the superiority of BTS. The results suggest that the BTS algorithm outperforms all other competing methods, as it optimizes the distribution of OCPs, thus leading to higher performance in object identification tasks. Although BTS is less efficient given its complex design, we believe its advantages outweigh its disadvantages. Further experiments suggest that BTS can be efficient when multi-thread technology is implemented.

Data and codes availability statement

Data and codes are available from the link.

<https://doi.org/10.6084/m9.figshare.14703162>. The code file is the core file of the algorithm for binary tree sampling.

Funding

This work is supported by the National Natural Science Foundation of China with grant numbers [42090012, 41771452 and 41771454].

ORCID

Xianwei Lv  <http://orcid.org/0000-0002-1574-8500>
Zhenfeng Shao  <http://orcid.org/0000-0003-4587-6826>
Xiao Huang  <http://orcid.org/0000-0002-4323-382X>
Jiaming Wang  <http://orcid.org/0000-0001-8144-5842>

References

- Alhassan, V., Henry C, Ramanna S, *et al.*, 2019. A deep learning framework for land-use/land-cover mapping and analysis using multispectral satellite imagery. *Journal Neural Computing and Applications*, 32 (12), 8529–8544.
- Al-Huda, Z., Peng B, Yang Y, Algburi RA. 2020. Object scale selection of hierarchical image segmentation with deep seeds. *IET Image Processing*, 15 (8), 191–205.
- Bazzani, L., *et al.*, 2016. Self-taught object localization with deep networks. 2016 *IEEE winter conference on applications of computer vision (WACV)*, 1–9. Lake Placid, NY.
- Blaschke, T., 2010. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65 (1), 2–16. doi:10.1016/j.isprsjprs.2009.06.004.
- Blaschke, T., *et al.*, 2014. Geographic object-based image analysis – towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87 (100), 180–191. doi:10.1016/j.isprsjprs.2013.09.014.
- Chaudhuri, D. and Samal, A., 2007. A simple method for fitting of bounding rectangle to closed regions. *Pattern Recognition*, 40 (7), 1981–1989. doi:10.1016/j.patcog.2006.08.003.

- Chen, Y., Ming, D., and Lv, X., 2019. Superpixel based land cover classification of VHR satellite image combining multi-scale CNN and scale parameter estimation. *Earth Science Informatics*, 12 (3), 341–363. doi:10.1007/s12145-019-00383-2.
- Flood, N., Watson, F., and Collett, L., 2019. Using a U-net convolutional neural network to map woody vegetation extent from high resolution satellite imagery across Queensland, Australia. *International Journal of Applied Earth Observation and Geoinformation*, 82, 101897. doi:10.1016/j.jag.2019.101897
- Franz, and Aurenhammer, 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *Acm Computing Surveys*, 23 (3), 345–405. doi:10.1145/116873.116880.
- Fu, T., et al., 2018. Using convolutional neural network to identify irregular segmentation objects from very high-resolution remote sensing imagery. *Journal of Applied Remote Sensing*, 12 (2), 025010. doi:10.1117/1.JRS.12.025010.
- Gere, J.M., Timoshenko, S.P., and Saunders, H., 1977. *Mechanics of materials*. Boston, MA: PWS-KENT Publishing Company.
- Ghorbanzadeh, O., et al., 2020. Transferable instance segmentation of dwellings in a refugee camp - integrating CNN and OBIA. *European Journal of Remote Sensing*, 54 (1), 14.
- Guo, Z. and Feng, -C.-C., 2020. Using multi-scale and hierarchical deep convolutional features for 3D semantic classification of TLS point clouds. *International Journal of Geographical Information Science*, 34 (4), 661–680. doi:10.1080/13658816.2018.1552790.
- Huang, X., Cao, Y., and Li, J., 2020. An automatic change detection method for monitoring newly constructed building areas using time-series multi-view high-resolution optical satellite images. *Remote Sensing of Environment*, 244, 111802. doi:10.1016/j.rse.2020.111802
- Huang, X., Zhang, L., and Zhu, T., 2014. Building change detection from multitemporal high-resolution remotely sensed images based on a morphological building index. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7 (1), 105–115. doi:10.1109/JSTARS.2013.2252423.
- Jiao, L., Liu, Y., and Li, H., 2012. Characterizing land-use classes in remote sensing imagery by shape metrics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 72 (AUG.), 46–55. doi:10.1016/j.isprsjprs.2012.05.012
- Kaiming, H., et al. 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, 770–778.
- Krizhevsky, A., Sutskever, I., and Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 60 (6), 84–90. <https://doi.org/10.1145/3065386>
- Krummel, J.R., et al., 1987. Landscape patterns in a disturbed environment. *Oikos*, 48 (3), 321–324. doi:10.2307/3565520.
- Lewis, H.G., Cote, S., and Tatnall, A.R.L., 1997. Determination of spatial and temporal characteristics as an aid to neural network cloud classification. *International Journal of Remote Sensing*, 18 (4), 899–915. doi:10.1080/014311697218827.
- Lu, T., et al., 2019. Satellite image super-resolution via multi-scale residual deep neural network. *Remote Sensing*, 11 (13), 1588. doi:10.3390/rs11131588.
- Luus, F.P.S., et al., 2015. Multiview deep learning for land-use classification. *IEEE Geoscience and Remote Sensing Letters*, 12 (12), 1–5. doi:10.1109/LGRS.2015.2483680.
- Lv, X., et al., 2018a. Very high resolution remote sensing image classification with SEEDS-CNN and scale effect analysis for superpixel CNN classification. *International Journal of Remote Sensing*, 40 (2), 1–26.
- Lv, X., et al., 2018b. A new method for region-based majority voting CNNs for very high resolution image classification. *Remote Sensing*, 10 (12), 1946. doi:10.3390/rs10121946.
- Ma, L., et al., 2019. Deep learning in remote sensing applications: a meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166–177. doi:10.1016/j.isprsjprs.2019.04.015.
- Martins, V.S., et al., 2020. Exploring multiscale object-based convolutional neural network (multi-OCNN) for remote sensing image classification at high spatial resolution. *ISPRS Journal of Photogrammetry and Remote Sensing and Remote Sensing*, 168, 56–73. doi:10.1016/j.isprsjprs.2020.08.004.

- Okabe, A., 2016. Spatial tessellations. *International Encyclopedia of Geography: People, the Earth, Environment Technology: People, the Earth, Environment Technology*, 1–11. Montreal.
- Oquab, M., et al., 2014. Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1717–1724. Columbus, Ohio.
- Oquab, M., et al., 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 685–694. Boston, MA.
- Phiri, D., et al., 2018. Effects of pre-processing methods on Landsat OLI-8 land cover classification using OBIA and random forests classifier. *International Journal of Applied Earth Observation and Geoinformation*, 73, 170–178. doi:10.1016/j.jag.2018.06.014.
- Rabiee, H.R., Kashyap, R.L., and Safavian, S.R., 1996. Multiresolution segmentation-based image coding with hierarchical data structures, 1870–1873.
- Shao, Z. and Cai, J., 2018. Remote sensing image fusion with deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11 (5), 1656–1669. doi:10.1109/JSTARS.2018.2805923.
- Sharma, A., et al., 2017. A patch-based convolutional neural network for remote sensing image classification. *Neural Networks*, 95, 19–28. doi:10.1016/j.neunet.2017.07.017.
- Simonyan, K. and Science, A.Z.J.C., 2015. Very deep convolutional networks for large-scale image recognition. *ICLR,arXiv2014*, arXiv:1409.1556.
- Szegedy, C., et al., 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9. Boston, MA.
- Tong, X.-Y., et al., 2020. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, 237, 111322. doi:10.1016/j.rse.2019.111322.
- Wang, J., et al., 2020. Object-scale adaptive convolutional neural networks for high-spatial resolution remote sensing image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 283–299. doi:10.1109/JSTARS.2020.3041859.
- Yaqian, Z., et al., 2020. Simulating urban land use change by integrating a convolutional neural network with vector-based cellular automata. *International Journal of Geographical Information Science*, 34 (7), 1475–1499. doi:10.1080/13658816.2020.1711915.
- Zhang, C., et al., 2018. An object-based convolutional neural network (OCNN) for urban land use classification. *Remote Sensing of Environment*, 216, 13. doi:10.3390/rs11010013.
- Zhang, C., et al., 2020. A multi-level context-guided classification method with object-based convolutional neural network for land cover classification using very high resolution remote sensing images. *International Journal of Applied Earth Observation and Geoinformation*, 88, 102086. doi:10.1016/j.jag.2020.102086.
- Zhang, C. and Atkinson, P.M., 2016. Novel shape indices for vector landscape pattern analysis. *International Journal of Geographical Information Science*, 30 (12), 2442–2461. doi:10.1080/13658816.2016.1179313.
- Zhao, W., et al., 2015. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. *International Journal of Remote Sensing*, 36 (13), 3368–3379. doi:10.1080/2150704X.2015.1062157.
- Zhao, W. and Du, S., 2016. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113, 155–165. doi:10.1016/j.isprs.2016.01.004
- Zheng, Z., et al., 2020. FPGA: fast patch-free global learning framework for fully end-to-end hyperspectral image classification. *IEEE Transactions on Geoenvironment & Remote Sensing*, PP(99), 1–15.
- Zhou, B., et al., 2016. Learning deep features for discriminative localization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929. Las Vegas, Nevada.
- Zhou, W., et al., 2019. SO-CNN based urban functional zone fine division with VHR remote sensing image. *Remote Sensing of Environment*, 236, 111458. doi:10.1016/j.rse.2019.111458.
- Zou, Q., et al., 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12 (11), 2321–2325. doi:10.1109/LGRS.2015.2475299.