ISPRS Journal of Photogrammetry and Remote Sensing 168 (2020) 56–73

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Exploring multiscale object-based convolutional neural network (multi-OCNN) for remote sensing image classification at high spatial resolution

Vitor S. Martins [a,*], Amy L. Kaleita [a], Brian K. Gelder [a], Hilton L.F. da Silveira [b], Camila A. Abe [c]

[a] Agricultural and Biosystems Engineering, Iowa State University, Ames, IA, USA
[b] Brazilian Agricultural Research Corporation, Embrapa Territorial Intelligence, Campinas, SP, Brazil
[c] Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI, USA

ARTICLE INFO

Keywords:
Deep learning
Convolutional neural network
Land cover
Aerial imagery

ABSTRACT

Convolutional Neural Network (CNN) has been increasingly used for land cover mapping of remotely sensed imagery. However, large-area classification using traditional CNN is computationally expensive and produces coarse maps using a sliding window approach. To address this problem, object-based CNN (OCNN) becomes an alternative solution to improve classification performance. However, previous studies were mainly focused on urban areas or small scenes, and implementation of OCNN method is still needed for large-area classification over heterogeneous landscape. Additionally, the massive labeling of segmented objects requires a practical approach for less computation, including object analysis and multiple CNNs. This study presents a new multiscale OCNN (multi-OCNN) framework for large-scale land cover classification at 1-m resolution over 145,740 km^2. Our approach consists of three main steps: (i) image segmentation, (ii) object analysis with skeleton-based algorithm, and (iii) application of multiple CNNs for final classification. Also, we developed a large benchmark dataset, called IowaNet, with 1 million labeled images and 10 classes. In our approach, multiscale CNNs were trained to capture the best contextual information during the semantic labeling of objects. Meanwhile, skeletonization algorithm provided morphological representation ("medial axis") of objects to support the selection of convolutional locations for CNN predictions. In general, proposed multi-OCNN presented better classification accuracy (overall accuracy ~87.2%) compared to traditional patch-based CNN (81.6%) and fixed-input OCNN (82%). In addition, the results showed that this framework is 8.1 and 111.5 times faster than traditional pixel-wise CNN_{16} or CNN_{256}, respectively. Multiple CNNs and object analysis have proved to be essential for accurate and fast classification. While multi-OCNN produced a high-level of spatial details in the land cover product, misclassification was observed for some classes, such as road versus buildings or shadow versus lake. Despite these minor drawbacks, our results also demonstrated the benefits of IowaNet training dataset in the model performance; overfitting process reduces as the number of samples increases. The limitations of multi-OCNN are partially explained by segmentation quality and limited number of spectral bands in the aerial data. With the advance of deep learning methods, this study supports the claim of multi-OCNN benefits for operational large-scale land cover product at 1-m resolution.

1. Introduction

Land cover classification is one of the most popular and challenging topics in remote sensing image processing (Gong et al., 2013; Zhu and Woodcock, 2014; Chen et al., 2015; Yang et al., 2018). Land cover maps have greatly advanced our knowledge about Earth's terrestrial surface, providing critical information on natural resources and land management (Jin et al., 2013; Jia et al., 2014; Lu et al., 2016). However, despite the advances in image processing, large-area classification remains a difficult task for high-resolution satellite and aerial imagery (Yifang et al., 2015). Finer resolution data exhibit high detail (and intra-class variance), which pose a challenge for accurate model prediction across heterogeneous landscape. Moreover, supervised classification requires i) a large annotated dataset to properly train a classifier and ii)

* Corresponding author.
E-mail address: vitors@iastate.edu (V.S. Martins).

https://doi.org/10.1016/j.isprsjprs.2020.08.004
Received 3 April 2020; Received in revised form 29 July 2020; Accepted 7 August 2020
Available online 13 August 2020

an efficient framework to manage a massive number of pixels during the classification (Yang et al., 2018). Machine learning algorithms have been widely evaluated to provide land cover maps from remotely sensed data (Lu and Weng, 2007; Mountrakis et al., 2011; Belgiu and Drăguţ, 2016; Mahdianpari et al., 2017; Ma et al., 2017).

Deep learning has rapidly become a key research field in machine learning. In the last decade, neural networks have achieved significant results in computer vision tasks (LeCun et al., 2010; Farabet et al., 2012; LeCun et al., 2015), such as object and facial recognition (Liu et al., 2015; Sun et al., 2015), self-driving cars (Tian et al., 2018), and audio recognition (Abdel-Hamid et al., 2014). The deep learning algorithms, such as convolutional neural networks (CNN or ConvNets), perform "end-to-end learning" to obtain hierarchical representation from input data. The multiple inter-connected layers provide the capability to fit a problem-specific model in a robust manner, without handcrafted features or decision rules (LeCun et al., 2010). Recently, CNN architectures have been increasingly explored for remote sensing applications, including building extraction (Xu et al.,2018; Vakalopoulou et al., 2015; Alshehhi et al., 2017), vehicle or road detection (Lv et al., 2018), agriculture mapping (Kussul et al., 2017) and land cover classification (Castelluccio et al., 2015; Längkvist et al., 2016; Nogueira et al., 2017; Hu et al., 2018).

While these studies have drawn significant attention to CNN architectures, there are certain limitations for practical land cover classification. Notably, patch-based CNN architecture takes advantage of contextual information from images to predict the class score at the end of the network. The pixel-wise classification using traditional CNNs is a redundant process using sliding window (stride = 1) across the image, and fully convolutional network (FCN) has been proposed for dense prediction output (Zhu et al., 2017). However, both CNN and FCN classification often produce coarse maps with blurred object edges (Paoletti et al., 2018). Intuitively, the input patch of CNN is often not consistent with real-world objects, leading to inaccurate classification of the edges (over-expansion or shrinkage) and geometry distortions in the final classification. To address these problems, recent studies have proposed the integration of CNN classifier and image segmentation, resulting in post-classification refinements, or the object-based CNN (OCNN) approach (Längkvist et al., 2016; Alshehhi et al., 2017; Zhao et al., 2017; Lv et al. 2018; Zhang et al., 2018; Huang et al., 2018; Liu et al., 2019; Mboga et al., 2019).

In general, OCNN approach for land cover classification consists of two main steps: (i) original image is segmented into homogeneous regions and then (ii) object-based classification is performed using CNN model. As part of object-based image analysis (OBIA), segmentation tools produce relatively homogeneous groups of pixels, known as image objects, based on the spectral, geometric and spatial properties (Hay et al., 2003; Blaschke, 2010; Blaschke et al., 2014). These objects are representations of land targets such as buildings, roads, water bodies, and others. This OCNN integrates the advantage of edge-preserving objects and capabilities of CNN classifier to generate more consistent land cover maps. A promising OCNN approach was developed by Zhao et al. (2017). The authors classified urban objects using the most frequent CNN prediction class within the segmented area. Alshehhi et al. (2017) used the CNN probability map and segmentation results from a simple linear iterative clustering algorithm to improve the shape of classified roads and buildings. Similarly, Liu et al. (2019) proposed a CNN classification with post-classification refinement to produce object-based thematic maps using multispectral and radar data. Although successful, these studies still require a pixel-by-pixel labeling using patch-based CNN across the entire image. This processing is computationally intensive and spatially redundant due to overlapping areas using sliding window.

In contrast, Zhang et al. (2018) developed an efficient object-based CNN (OCNN) approach for urban land use classification. The authors used object convolutional position analysis to classify the objects using less CNN predictions (instead of all pixels). The proposed method is more accurate and efficient than traditional pixel-wise CNN classification. Also, Lv et al. (2018) developed a CNN classification with region-based majority voting for high-resolution images. The authors performed CNN predictions in specific locations (point voters) within each segmented region, which reduces the number of class prediction per object. The authors also mentioned the benefits of this object-based method to preserve the boundary information. While OCNN is a promising approach for efficient land cover mapping, previous studies are mostly limited to small urban areas or single scenes (Lv et al. 2018; Zhang et al., 2018; Huang et al., 2018). Moreover, we argue that OCNN is highly dependent on multiple CNN models to solve the problem of observation scale in large context (Zhao and Du, 2016). The observation scale refers to the spatial extent under consideration and influences on how the objects appear in the images (Dabiri and Blaschke, 2019). Therefore, the spatial-related features (edge, contour, texture) are a scale-dependent information and the performance of CNNs is affected by the choice of window size (Zhang et al., 2018). With the growing interest in deep learning methods, we believe that further research is required to implement OCNN in large-area classification.

This study presents a new multiscale object-based CNN (multi-OCNN) approach for large-area land cover classification at 1-m resolution. The proposed multi-OCNN includes three main procedures: image segmentation, object analysis, and scene classification using multiple CNNs. The National Agriculture Imagery Program (NAIP) aerial imagery (~6100 tiles, 955 GB) were used in this study, covering the state of Iowa in the United States. The performance of multi-OCNN approach was evaluated over eight regions. The main contributions are as follows: (i) integration of multiscale CNNs and object-based approach for land cover classification; (ii) development of a new benchmark dataset, called IowaNet (1 million images with 10 land cover types); (iii) novel object analysis using skeleton-based method. Our results show that multi-OCNN provides a satisfactory framework for massive semantic segments (~1.01 billion) in a heterogeneous landscape (overall accuracy ~87.2%). As far as we know, this study is the first application of OCNN for scene classification in such a broad context. Given the challenges to turn CNN into a practical tool, this study supports the claim of capabilities of multi-OCNN for large-area classification.

The paper is structured as follows: Section 2 describes related works using deep learning and OCNN approach. In Section 3, we describe the NAIP aerial images and IowaNet dataset. An overview of the proposed CNN architectures, image segmentation and OCNN framework are presented in Section 4. Sections 5 and 6 present the results and discussion, respectively. Finally, the conclusions are drawn in the last section.

2. Related work

The promising CNN approaches were firstly presented for urban mapping using aerial imagery. In 2010, the successful application of patch-based CNN for roads and buildings extraction (Mnih and Hinton, 2010) has supported other studies using these networks (e.g.: Chen et al., 2016; Maggiori et al., 2016; Saito et al., 2016; Alshehhi et al., 2017; Nogueira et al., 2017; Sharma et al., 2017; Audebert et al., 2018; Sun et al., 2019). For instance, Saito et al. (2016) proposed a building and road extraction system using single patch-based CNN architecture. The proposed CNN predicts three classes (road, building and background) from a Massachusetts dataset. Using a multi-class method, Paoletti et al. (2018) applied 3-D CNN architecture for spatial-spectral classification of hyperspectral data in distinct areas (agricultural and urban). The authors also evaluated the performance of the deep network at different patch sizes. In another context, CNN models have been used to generate wetland inventory (Mahdianpari et al., 2018; DeLancey et al., 2020). Rezaee et al. (2018) evaluated the efficiency of patch-based CNN for wetland mapping; the classification results show a higher overall accuracy for CNN (94.82%) compared to a random forest classifier (79.11%). They mentioned a high redundancy during the classification because of overlapping input patches. Paisitkriangkrai et al.

(2016) combined both CNN-derived and hand-crafted features for semantic labeling of aerial images. The post-processing step was also implemented to improve the final classification using conditional random fields. Similarly, Längkvist et al. (2016) developed a CNN classification with post-classification refinement for better representation of real-world targets (five-classes: vegetation, ground, road/parking/railroad, building and water).

In remote sensing, CNN-based classification involves the partition of original image into small patches and the trained network predicts a single label for the central pixel in the patch. As a result, land cover classification becomes a slow and expensive process for high-resolution images, especially on large-scale applications. Fully Convolutional Network (FCN) is an extension of the traditional CNN that performs semantic segmentation of images (Long et al., 2015). The FCN-based architectures consist of encoder/decoder blocks for full-resolution segmentation map, and some examples are U-Net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), FC-DenseNet (Jégou et al., 2017), and Deeplab networks (Chen et al., 2017). By replacing fully-connected layers with up-convolutional layers, FCN maintains the 2-D structure in the output classification, and predicts certain class for each pixel of original input image. This dense prediction map reduces the overlapping patches because sliding window uses the stride with the same dimension of input image. While FCNs are typically faster compared to traditional CNNs, several studies have pointed out the needed of post-processing of FCN results to refine the object boundaries (Marmanis et al., 2016; Sherrah, 2016; Fu et al., 2017; Audebert et al., 2018; Mboga et al., 2019) and efforts are needed to improve the final quality of mapping results.

As deep learning emerges in land cover classification, the object-based CNN becomes a potential approach to improve the classification performance using image objects. Since the segmentation provides a cluster of homogeneous pixels as objects, an effective solution explores the objects for less computation. Technically, the success of OCNN approach relies on object analysis to define the convolutional locations for CNN predictions within object area. By applying the CNN in specific locations (instead of all pixels), the number of model predictions reduces substantially and speed up the overall processing. However, there are few studies on literature exploring object analysis in the OCNN framework (Lv et al. 2018; Liu et al., 2018; Zhang et al., 2018; Huang et al., 2018).

A relevant approach was implemented by Huang et al. (2018). The authors used a morphological operator to extract skeletons of mapping units (street blocks). The convolutional locations are selected along skeleton lines, and standard CNN is only applied in limited locations within the mapping unit. The final class label of object is defined by majority voting scheme. Although this procedure is useful for CNN classification, the proposed approach is still dependent on small urban features, such as street blocks to define mapping units. Likewise, Zhang et al. (2018) developed an object convolutional position analysis to define appropriate locations for CNN application within the object. The method uses the shape features (linear and non-linear) of objects to select processing units for CNN prediction. The authors also emphasized that two CNN models with different input sizes were needed for complex objects (irregular shape), and future research should consider multiple models. Likewise, some studies have also highlighted the relevance of multiple CNNs for image classification (Zhao and Du, 2016; Liu et al., 2017).

The development of multiscale CNNs supports the extraction of deep spatial-related features in different observational scales (Liu et al., 2017). According to Zhao and Du (2016), multiple CNN models allow the framework to capture better contextual information during object classification. Längkvist et al. (2016) claimed the relevance of multiscale CNNs to consistently extract appropriate spatial relationships for CNN classification. Paisitkriangkrai et al. (2016) proposed a multiscale CNN that predicts an output based on the 16×16, 32×32, and 64×64 input patches. Following the recent progress, we proposed a new multi-OCNN approach with both object analysis and multiscale CNNs to improve the classification performance at 1-m resolution.

3. Material

3.1. NAIP aerial imagery

3.1.1. Data description

National Agriculture Imagery Program (NAIP) is a nationwide program that provides high 1-meter resolution aerial ortho-imagery in the agricultural growing season across contiguous United States. This program is administered by USDA Farm Service Agency. Although NAIP coverage varies across the U.S., this program typically delivers an annual dataset for most regions. Several studies have used the NAIP aerial imagery for land cover classification (Li et al., 2014; Maxwell et al., 2014, 2017; Basu et al., 2015; Nagel and Yuan, 2016). In this study, Iowa NAIP 2015 dataset was used to (i) create a large training dataset (Section 3.1.2), and (ii) evaluate the application of OCNN for land cover mapping. The statewide dataset consists of ~6100 image tiles, approximately 955 GB (140–170 MB per tile). The NAIP program provides clear-sky images with four spectral bands (blue, green, red and near-infrared) and 8-bit of radiometric resolution.

All NAIP imagery were obtained via USGS Earth Explorer and delivered in GeoTiff format with UTM North American Datum 1983 (NAD 83). These images were acquired in late summer and early fall of 2015, mostly August and September, and Fig. 1 shows the study area with NAIP 2015 data. Although tiles are collected on different flight dates, the sensor quality and pre-processing assure the consistency of this publicly available dataset. The NAIP images were mosaicked into the county level, and later, all processing is implemented for each county.

3.1.2. IowaNet training dataset

The IowaNet is a new multiscale benchmark dataset that comprises 1 million images with 10 land cover classes (100,000 points per class). This dataset was developed as part of our efforts to implement this land cover classification, including intensive manual label (total: ~650 h of work). The sample images were derived from Iowa NAIP 2015 data. The land cover classes were defined as follows (Fig. 2 and Table 1): (1) structures (e.g., residential and commercial buildings); (2) roads; (3) river; (4) pond (e.g., lakes, reservoirs); (5) cultivated crops; (6) fallow; (7) shadow; (8) forest (e.g., deciduous, evergreen, and mixed forests); (9) grassland and herbaceous; and (10) barren land. These classes were defined according to other reference studies (Gong et al., 2013; Homer et al., 2015; Yang et al., 2018). Note that most studies represented river and lake classes as “water”, but they present different geometry and size, and we decided to separate them for better CNN performance. Each class has 100,000 patch images, and the dataset consists of two sets of spectral bands: natural color (blue, green, and red) and false-color bands (green, red, and near-infrared). The IowaNet is now available as open-access dataset (Martins et al., 2019).

We performed an intensive manual labeling of 1 million points across Iowa. The label of each point was visually interpreted using 1-m NAIP 2015 image and ArcGIS online services. These points are known as “seed points” because each one is the central pixel to extract different patch sizes (growing window). By doing that, we created a multiscale dataset with six patch sizes (8×8, 16×16, 32×32, 64×64, 128×128, 256×256 pixels) with label of central pixel. For each patch size, the training dataset is a pair of labeled images $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_n is the i-th input data (patch image), y_n is the i-th label data (land cover class) and n is 1 million images. Since patch images were samples in a distinct context, the dataset is spatially representative and includes the intra-class variability for large-area classification. We should also mention that the implications of atmospheric effects on classification results are considered negligible because aerial images are typically acquired at clear-sky days and low altitudes (↓ atmospheric path), statewide

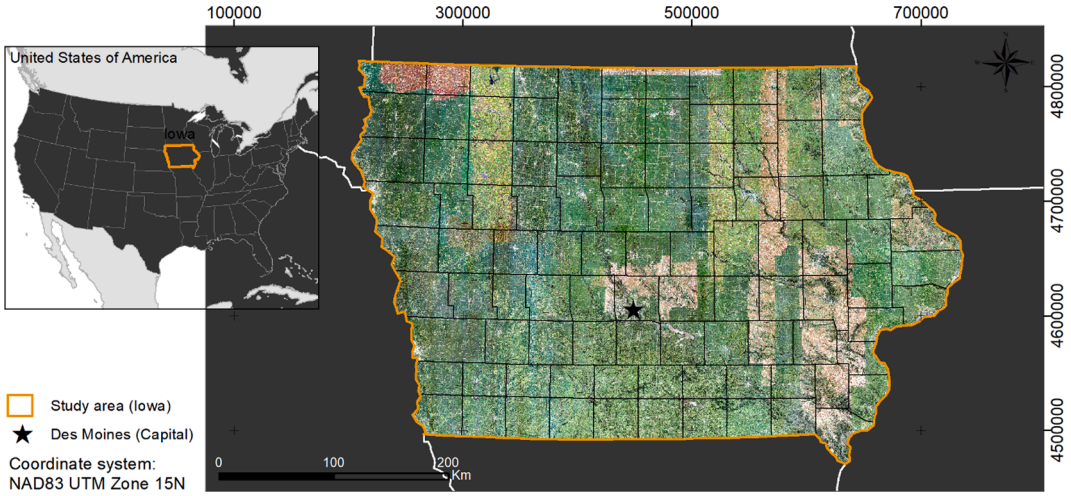

Fig. 1. Study area. The 1-m resolution NAIP (National Agriculture Imagery Program) imagery from 2015 are used in this study. Note that large banding between counties is caused by different flight dates during data acquisition.

Fig. 2. Examples of land cover classes in IowaNet dataset (1 million samples, 10 land-cover classes). The training dataset was derived from NAIP (National Agriculture Imagery Program) aerial imagery and there are two sets for users: natural color (blue, green, and red) and false-color (green, red, and near-infrared) datasets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sampling produces high intra-class spectral variability, and CNNs explore spatial patterns besides the spectral information. Due to 1 million samples and six resolutions, the IowaNet dataset has a potential to become a benchmark remote sensing dataset for deep learning applications (see others in Zhou et al., 2018). Finally, we trained the proposed CNN models using a false-color IowaNet dataset (Section 4.1.2).

4. Method

In this section, we describe a new multiscale OCNN framework for large-scale classification of aerial images. The proposed framework is shown in Fig. 3. First, six CNN models were trained using the IowaNet dataset. Second, image segmentation is performed for each county image across Iowa. Third, object analysis is implemented for CNN prediction of land cover class. Finally, the labeled objects are merged in the final aerial scene classification. Further details of these steps are discussed in the following sections.

4.1. Convolutional neural network (CNN) architecture

4.1.1. Background

Convolutional Neural Network (CNN) is a specific type of feed-forward neural network used for image recognition and classification

Table 1
Description of land cover classes.

No.	Land cover	Description
1	Barren land	Bare soil land (80% soil background), sandy land, cattle corrals, mining areas and riverbanks
2	Cropland	Cultivated crop areas (vegetated), and irrigated farmlands
3	Fallow	Post-harvest, plant senescence, and abandoned arable lands
4	Forest	Deciduous, evergreen, and mixed forest areas
5	Grassland	Herbaceous cover, pasture hay, meadow, natural pasture (open spaces)
6	Lake	Reservoirs, ponds, lakes and wetlands
7	River	Natural stream of flowing water, such as canals, streams
8	Road	Road, lane, highway, pavements, and rail tracks
9	Shadow	Tree and building shadows
10	Structures	Residential, commercial, and industrial buildings, farmhouses, barns and silos

(LeCun et al., 1998; 2010). A typical CNN architecture consists of a series of layers such as (i) convolutional, (ii) pooling, and (iii) fully-connected (FC) layers. Also, dropout and batch normalization layers can be used to avoid overfitting and improve the generalization of the model (Srivastava et al., 2014). This multi-layer architecture extracts abstract features from input data and then returns the class score at the end of the network. A convolutional layer is the central building block in the network and has a set of trainable weights to learn spatial features in images, from low- (edge, corner, contour) to high-level (complex structures). The convolutional layer receives a three-dimensional array and computes the output feature maps. These feature maps are the result of dot product of receptive field and a set of weights in filters (or kernels). In the learning process, these weights in convolutional and FC layers are adjusted to capture the most relevant features for each class. This learning procedure is performed using a backpropagation algorithm and stochastic gradient descent (SGD). Further details of this procedure are given in LeCun et al. (1998). In each convolutional layer, the number of filters (M) is equivalent to the number of output feature maps. Therefore, a convolutional layer relates the tensor $x_j \in \mathbb{R}^{m \times n \times j}$ of j-th feature map in previous layer $(i-1)$ to the output feature map $z \in \mathbb{R}^{\overline{m} \times \overline{n} \times \overline{j}}$ of current layer (i) as follows:

$$z_k^i = \mathrm{pool}_{\max}\left(\sigma\left(b_k^i + \sum_{j=1}^{M^{i-1}} W_{k,j}^i * x_j^{i-1}\right)\right) \quad (1)$$

where i-th is the convolutional layer, $W_{k,j} \in \mathbb{R}^{l \times l \times k}$ is a k-th convolutional filter (weights), (*) is a two-dimensional convolutional operator, b_k is a bias term, $\sigma(\bullet)$ is an element-wise nonlinearity function (e.g. REctified Linear Unit (ReLU): $\sigma(x) = max(0, x)$), and $pool_{max}$ is a max pooling function. A max pooling layer computes the maximum values of rectangular regions of its input. The max pooling is a typical operation after the convolutional layer and becomes an important layer to provide a translational invariant feature and to reduce the number of trainable weights. The dimensions $\overline{m}$ and $\overline{n}$ of outputs rely on pooling dimension, stride and padding parameters. In the first layer, input dimensions (m, n, and j) of image represent row, column, and number of spectral, respectively. After a series of convolution layers, the flatten layer converts the feature maps from 2-dimensional arrays to 1-D vector for FC layers. At the end of network, the last FC layer has a multi-class *softmax* function that computes the class probabilities. Once the architecture is defined, the next step is the optimization of network parameters using a backpropagation algorithm with gradient descent (Johnson and Zhang, 2013). In this context, the "learning" involves the minimization of error between target and predicted value calculated by loss function, such as categorical cross-entropy (multi-class problem).

4.1.2. Multiscale CNNs: development and training

The multiscale CNN concept has been increasingly used for remote sensing image classification (Zhao & Du, 2016; Li et al., 2016; Deng et al., 2018; Fu et al., 2018; Chen et al., 2019). In these studies, the multiscale concept is typically related to CNN applications with different window sizes. The reason for multiscale CNNs is that evaluation of real-world objects with different patch sizes improves the model performance in the understanding of target features and its contextual information. Since landscape targets present a variety of geometry and size, the object label is limited when fixed input size is used in large area classification. For example, spatial-related features of small objects can be extracted when we observe it closer, while they are hardly identified when it is observed from a distance. To capture the scale-dependent information, the selection of input window size needs to be explicitly associated with target under investigation and multiscale CNNs give such capability during object-based classification.

In this study, we developed a multiscale approach with six CNN architectures for land cover classification (10 classes). These architectures follow the well-known AlexNet (Krizhevsky et al., 2012) and LeNet-5 networks (LeCun et al., 1998). Several studies have developed CNNs for remote sensing data based on these two architectures (Hu et al., 2015; Zhang et al., 2018; Lv et al., 2018; Li et al., 2019). For multiscale framework, six CNN architectures were developed and trained with different input resolutions: CNN8 (8 × 8), CNN16 (16 × 16), CNN32 (32 × 32), CNN64 (64 × 64), CNN128 (128 × 128) and CNN256 (256 × 256 pixels). Overall, AlexNet and LeNet architectures were modified for our purpose, and several experiments were performed to define the number of convolutional layers as well as the regularization/normalization layers. The summary of all models is shown in Table 2 and the example of CNN16 network is presented in Fig. 4. For instance, the basic architecture of CNN16 consists of an input layer (16 × 16 × 3 bands), four convolution layers (+ReLU and batch normalization), two max-pooling + dropout (2nd and 4th layers), flatten layer, and two fully-connected layers. The last layer contains 10 neurons and delivers the probability vector with *softmax* function.

While the hidden layers vary for these architectures, the hyper-parameters are quite similar among them. For each model, these parameters were tuned empirically through cross-validation accuracy. For example, the number of epochs was tested from 25 to 200. The kernel

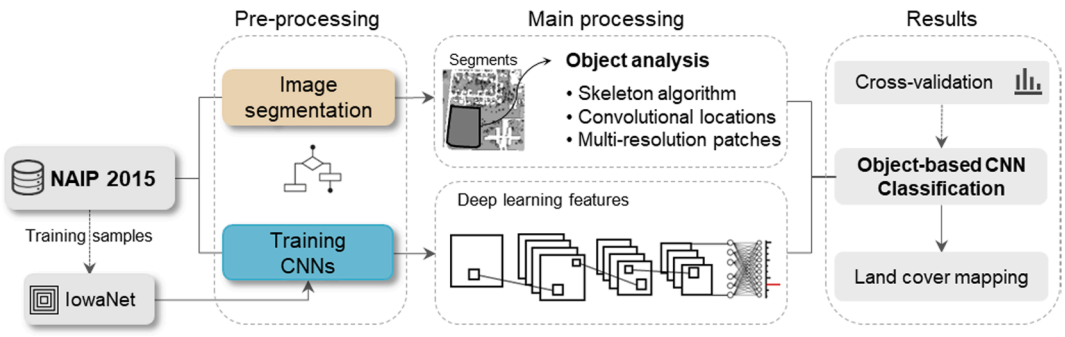

Fig. 3. Overview of the multiscale object-based Convolutional Neural Network (multi-OCNN) framework.

Table 2
Summary of CNN architectures. The model name refers to input image size (e.g., CNN8 has input of 8 × 8 pixels). The acronyms are defined as follows: Conv is convolutional layer, FC is fully-connected layer, MP is maximum pooling layer, BNM is batch normalization layer, and DP is dropout. All Convs have Rectified Linear Unit as activation function. The number of filters and its size per convolutional layer are presented in the first line of each model, such CNN8 with 64 filters of 3 × 3 pixels (64@ 3 × 3) in the Conv1 layer. The dropout rate and filter dimension of max-pooling are presented in the parenthesis.

Models	Conv1	Conv2	Conv3	Conv4	FC1	FC2	FC3
CNN8	64@3 × 3	32@3 × 3	64@3 × 3	32@3 × 3	n = 512	n = 256	n = 10
	BNM	MP (2 × 2)	–	BNM	–	–	SoftMax
	–	DP (0.2)	–	DP (0.2)	–	–	
CNN16	64@5 × 5	32@5 × 5	64@3 × 3	32@3 × 3	n = 1024	n = 512	n = 10
	BNM	BNM	BNM	BNM	–	–	SoftMax
	–	MP (2 × 2)	–	MP (2 × 2)	–	–	–
	–	DP (0.2)	–	DP (0.2)	–	–	–
CNN32	32@3 × 3	32@3 × 3	32@3 × 3	32@3 × 3	n = 512	n = 10	–
	BNM	BNM	–	MP (2 × 2)	DP (0.5)	SoftMax	–
	–	MP (2 × 2)	–	DP (0.2)		–	–
	–	DP (0.2)	–	–	–	–	–
CNN64	64@5 × 5	64@5 × 5	64@3 × 3	64@3 × 3	n = 512	n = 10	–
	–	MP (2 × 2)	–	MP (2 × 2)	DP (0.25)	SoftMax	–
	–	DP (0.25)	–	DP (0.25)		–	–
CNN128 & CNN256	32@5 × 5	32@3 × 3	32@3 × 3	–	n = 512	n = 10	–
	BNM	MP (2 × 2)	MP (2 × 2)	–	DP (0.5)	SoftMax	–
	MP (2 × 2)	DP (0.25)	DP (0.25)	–			–

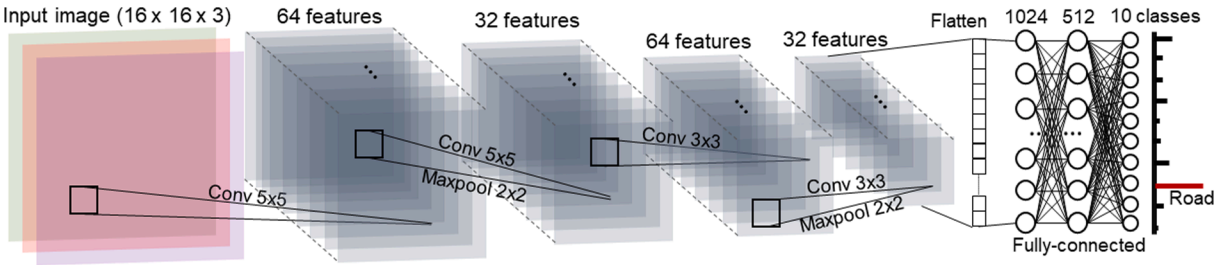

Fig. 4. The architecture of CNN16 model.

size of convolutional filters was evaluated for 3 × 3, 5 × 5 and 7 × 7 with stride = 1 (Table 2). The pooling layer was fixed with 2 × 2 window for all models. The dropout regularization was tested with different rates (0.1–0.5). Following Zhong et al. (2019), Adam parameters were fixed as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate = 0.001 with decay set to 0. These parameter settings are similar to all models.

The CNN architectures were fully trained from scratch using false-color IowaNet dataset. In the training step, the dataset was subdivided into three sets: training (90%), validation (7.5%), and test dataset (2.5%). All labeled images were normalized between 0 and 1. The training and validation datasets adapt the weights of the CNN and validate the model result in each epoch. The validation provides further evaluation of a model fit on the training dataset at the end of each epoch. Once the models are completely trained, the test dataset is used for final evaluation of predictions using "unseen" images. Six models were trained using mini-batch stochastic gradient descent (batch size was set to 64). The CNN networks were trained in the High-Performance Computing (HPC) cluster at Iowa State University. The HPC cluster contains two 8-Core Intel E5 2650 with two NVIDIA Tesla K20 GPU 5 GB cards. The computational performance is significantly faster with GPU card due to hundreds of cores for matrix and vector operations (instead of CPU with a few cores), but the limited memory constraints the maximum size of the CNN architecture and the number of bands (here: false-color bands). All these experiments are conducted with Keras/TensorFlow module in Python environment and its open-source libraries, such as GDAL, PIL, NumPy, Rasterio, and Scikit-image.

4.2. Multi-layer perceptron network

Although this study is focused on the integration of CNN models, we also argue that observation scale plays a critical role for successful classification since the object details and spatial context are dependent on window size (Dabiri and Blaschke, 2019). In some cases, tiny objects (e.g., individual trees, small barren areas, and noise polygons) with few pixels become a challenging task for CNNs because they represent a small fraction of patch area, and contextual information can lead to misclassification. Instead of 2-D images, pixel-level information is more consistent for land cover classification. In this context, we developed and trained a Multi-Layer Perceptron (MLP) network to support the multi-OCNN framework, especially for tiny areas. In general, MLP is a feedforward artificial neural network that contains fully-connected layers (or dense layers) with arbitrary numbers of neuron units for data classification. Each neuron has a learnable weight with non-linear activation functions in the hidden layers. The network weights are adjusted in a supervised way. The training procedure minimizes the difference between outputs and correct values using back-propagation algorithm.

In this study, MLP network has pixel-level input values extracted from 1 million reference points of IowaNet (see Section 3.1.2). For each point, we stored the values of four spectral bands (blue, green, red, and NIR) with corresponding label, and they were used to train the MLP network. Regarding the model architecture, five hidden layers and its neurons (1st layer: 256, 2nd layer: 512, 3rd layer: 1024, 4th layer: 512, and 5th layer: 256) were defined with some experiments through cross-validation. All hidden layers have ReLU as non-linear activation function. The dropout of 0.5 is included in 5th layer to improve the

classification results. The learning rate of Adam was set to 0.015. The output layer gives the class probability for each land cover. The application of MLP in the multi-OCNN framework is presented in Section 4.4.

4.3. Image segmentation

The image segmentation is a pre-processing step for OCNN application. In this study, the image segmentation was performed by Mean-Shift algorithm (Fukunaga and Hostetler 1975; Comaniciu and Meer, 2002). This popular OBIA algorithm produces segmented images with homogeneous spatial and spectral information from input data. The mean-shift algorithm presents the advantage of simple parametrization with great edge detection in high-resolution images (Wang et al., 2012; Ming et al., 2012; Su et al., 2015; Sun et al., 2019). Meanwhile, this segmentation tool has proven to be useful in both natural and urban contexts (Wang et al., 2015). The region growth is defined by specific homogeneity criterion and all pixels are grouping when they are closer in both spatial and spectral domain (Hossain and Chen, 2019). The mean-shift algorithm has three scale parameters: spatial scale (h_r), spectral scale (h_s), and minimum segment size (M_s).

In this study, false-color bands (NIR, red, green) were used as input data in Mean-Shift algorithm, and the optimal scale values were defined by quantitative and visual assessments over eight testing counties (see counties in Section 4.5). Several combinations of scale parameters were tested by varying h_r and h_s values from 10 to 20 (step = 0.5). Following Zhang et al. (2018), M_s value was predefined as 25, and this minimum region keeps small objects in urban area and avoids under-segmentation results. A total of manually delineated polygons (n = 1200) from different land cover classes were compared to segmented areas in these experiments. The Jaccard index ($J = |G \cap R|/|G \cup S|$) was calculated to measure the similarity of segmented (R) and reference (G) polygons. The Jaccard index, also known as Intersection-Over-Union, ranges from 0 (worst; no match) to 1.0 (best; perfect match) and can penalizes both over- and under-segmentation (Polak et al., 2009). Note that under-segmentation process is characterized by single segment (large polygon) representing many real-world objects, while the over-segmentation is exemplified by many segments (small polygons) representing one object. The description of J-index can be found in Ge et al. (2007). In addition, we visually assessed the segmentation scale effects for further consideration. Based on the results (Section 5.2), the global scale parameters were selected as [h_r, h_s, M_s] = [16.0, 16.5, 25] for segmentation of NAIP images. This segmentation step creates ~1.01 billion segments for entire Iowa 2015 data, ranging from 2.7 to 24.5 million segments per county. The processing time for segmentation is 2.0–3.0 h per county using Intel Xeon(R) CPU E3-1270 @ 3.80 GHz (36 GB memory). In the next section, we explain the procedures for categorical label of these semantic-free segments using trained CNNs.

4.4. Multi-OCNN approach for scene classification

This section describes the multiscale OCNN (multi-OCNN) framework for land cover classification of high-resolution imagery. Essentially, the proposed multi-OCNN assigns a semantic label for image objects using multiple CNNs and has two advantages. First, object-based classification reduces the number of CNN predictions compared to pixel-based approach. Second, the multiscale approach increases the agreement of object shape and CNN input patch in OCNN classification. Again, note that the term "multiscale" is associated to multiple CNNs with different input sizes. In the pre-processing step, mean-shift segmentation generates functional objects for NAIP Iowa 2015 counties (Section 4.3). The semantic-free segments are input polygons for this object analysis, where the algorithm defines a number of convolutional locations and input patch size for CNN predictions.

In general, the object analysis includes (i) the application of

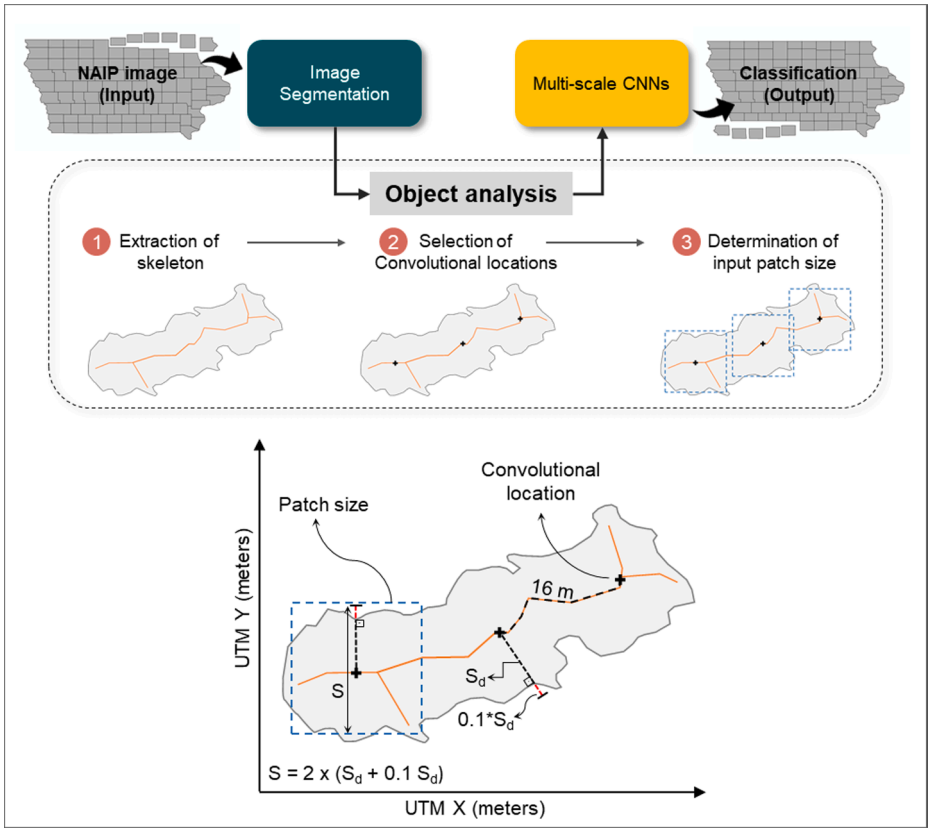

Fig. 5. Framework of the object analysis. This analysis generates skeleton lines, convolutional locations and patch sizes for multiscale CNN predictions per object. S_d represents the shortest distance between convolutional location and object boundary, and S is the patch size to support the selection of CNN model.

skeletonize algorithm, (ii) selection of convolutional locations and (iii) determination of input patch size (Fig. 5). The object classification relies on multiple locations (or processing units) for CNN predictions, and later, the final object label is defined as the highest-class membership computed from all predictions. Each image-object is treated independently as mapping unit. The object analysis starts by generating morphological lines ("medial axis") of object with skeletonize algorithm. This method performs a successive evaluation of border pixels and removes those pixels until that connected line is created (Zhang and Suen, 1984). So, this algorithm is a simple and fast technique to extract skeleton representation of any object (complex shapes with different sizes). The skeleton lines are then used to support the determination of multiple locations within the image-object. In this stage, convolutional locations are created with a 16-m interval along the skeleton line (Fig. 5). This interval is a "user-defined constant", but it should be close to the smallest input window to ensure the largest coverage across the object. Also, users should not expect an abrupt change in the performance by adjusting this parameter in a few meters. When the object area is lower than 256 m^2, the geometric centroid of object is defined as a single convolutional location. This location is a central coordinate for patch extraction from original data.

Once the object analysis produces the convolutional locations, the next step is the determination of optimal window size for patch extraction from original data. The algorithm computes the shortest distance (S_d) between convolutional location and object edge. The patch size (S) is defined as $S = 2 \times (S_d + 0.1 \times S_d)$, and then, the algorithm chooses the CNN model with closest larger predefined patch to S value for class prediction in the i-th location (Fig. 5). For instance, when the calculated S is 220 m, the closest larger predefined patch is 256 and CNN256 model is used for this convolutional location. The factor (0.1) assures at least 10% of the contextual information in the input patch area. However, if the convolutional location is nearby object edge ($S_d <$ 3 m), pixel-level MLP network is used for probability prediction in combination with CNNs. The reason for this that contextual information prevails in the input patch area and CNN application could be not consistent in some cases. Note that object analysis (skeletonize → convolutional location → input patch size) is implemented for massive number of objects (this study: ~1.01 billion), and this analysis has proven to be fast and robust for our application. For instance, processing times of large residence (189 m^2), natural pond (850 m^2), large cropland (259,268 m^2) are 0.002, 0.0069, and 0.49 sec, respectively. Finally, our processing system generates input patches from original NAIP image for selected locations and its patch sizes (Fig. 6). The trained CNNs compute per-class probability using these input patches (cropped images) and the object label is defined as land cover with highest class score.

In short, let us suppose that county H contains N semantic-free objects O_r, $r \in \{1, 2, \ldots, N\}$, where object O_z has M convolutional locations I_i, $i \in \{1, 2 \ldots, M\}$. At each convolution location I_i, trained CNN predicts a n-dimensional vector $\overrightarrow{P} = [c_1, c_2, \cdots, c_n]$, where c_n is the membership probability for each class $n \in \{1, 2 \ldots, 9, 10\}$. Since probability vectors are obtained in multiple locations across the object, the output class label (C) per object is defined as the highest value of added vector *P* from all locations:

$$\overrightarrow{P}_i = [c_1, c_2, \cdots, c_n] = \sum_{n=1}^{10} c_n = 1; 0 \le c_n \le 1 \tag{2}$$

$$C = \text{argmax}\left(\sum_{i=1}^{M} \overrightarrow{P}_i\right) = \text{argmax}\left(\begin{bmatrix} c_{1,1} + c_{1,2} + \cdots + c_{1,M} \\ c_{2,1} + c_{2,2} + \cdots + c_{2,M} \\ \vdots \quad \vdots \cdots \quad \vdots \\ c_{n,1} + c_{n,2} + \cdots + c_{n,M} \end{bmatrix}\right) \tag{3}$$

This classification procedure is applied for all image objects. Lastly, the multi-OCNN generates a statewide land cover product merging all county-based results.

4.5. Evaluation metrics and comparison

The classification assessment was conducted across eight regions (Fig. 7). A two-stage stratified random sampling was implemented for reference dataset (Stehman, 2009; Stehman and Foody, 2019). First, a simple random sampling creates 1250 points in each region. Since some

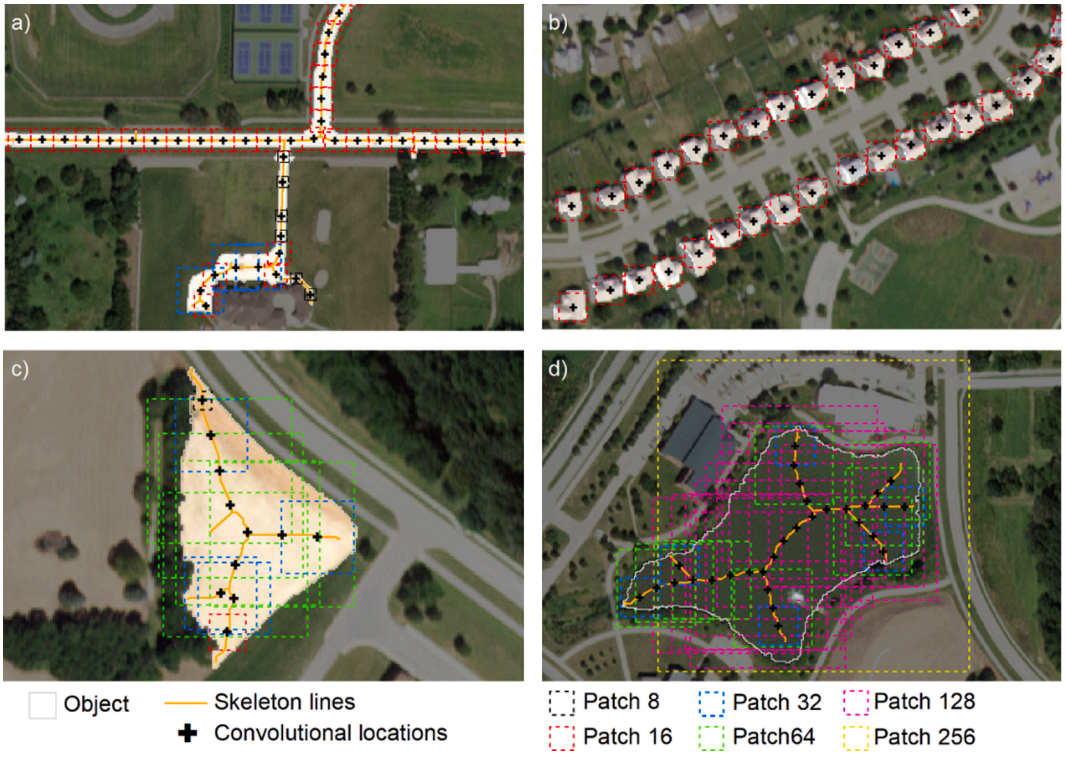

Fig. 6. Results of object analysis for different targets.

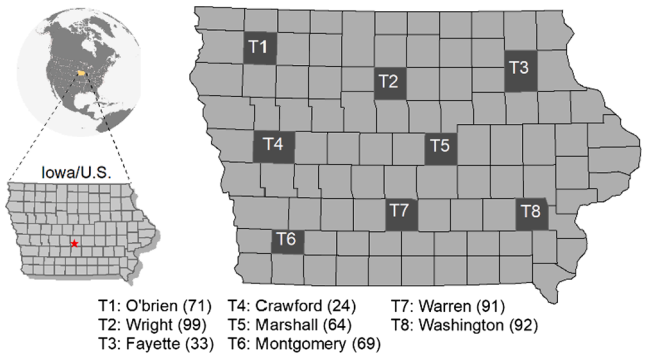

Fig. 7. Selected counties for validation experiments.

classes occur in a small proportion of area, the second step is a sampling scheme to ensure at least 125 samples per class, increasing the representation of these scarce classes. A total number of 10,000 samples were labeled as reference data (1250 samples per class). Due to spectral quality and 1-m resolution of NAIP images, the uncertainty of visual interpretation is assumed negligible. The confusion matrix and accuracy metrics were calculated for quantitative evaluation (Congalton and Green, 2002), such as overall accuracy (OA), kappa coefficient (κ), producer's (PA) and user's accuracies (UA), and per-class F-score. In this context, the confusion matrix allows the evaluation of agreement and disagreement in the classification. The producer's accuracy, which is related to omission errors, shows the proportion of the reference samples per class that is correctly classified in the map. The user's accuracy, which is related to commission error, presents the proportion of classified pixels per class in the map that is actually presented on the reference samples. The F-score is the harmonic mean of PA and UA (F-score = 2 * (PA * UA)/(PA + UA)). A larger value indicates better predictive accuracy, ranging from 0 to 100%. Additionally, other CNN-based frameworks were implemented to evaluate our proposed multi-OCNN:

- **Pixel-wise CNN**: this method is the traditional pixel-wise CNN that classifies pixel-by-pixel using single CNN model and sliding window. The traditional CNN for remote sensing becomes highly redundant because the input patch overlaps to predict the class label for central pixel of the patch. **Reason**: this comparison shows the difference between object-based and pixel-wise CNN classification. In the result section, CNN_{16} states for pixel-wise CNN with input patch of 16 × 16 pixels.
- **Fixed-OCNN**: this method uses the object classification with a single CNN model. The term "fixed" means a single model in this context. The object analysis is only partially implemented, because the determination of input patch size from the object was not used in this method. **Reason**: this comparison shows the benefits of multiscale CNNs to classify image objects. We evaluated this method using three CNN models (16 × 16, 64 × 64, 256 × 256 pixels). In the result section, $OCNN_{16}$ states for object-based CNN with fixed input patch of 16 × 16 pixels.
- **$OCNN_C$**: this method is similar to multi-OCNN, but it uses only geometric centroid for object classification. The object analysis is also partially implemented because the convolutional locations were not used in this method. The $OCNN_c$ uses a single prediction according to appropriate patch size. **Reason**: this comparison evaluates the benefits of skeletonize algorithm with multiple convolutional locations rather than a single prediction for object label.
- **$OCNN_{all}$**: this ensemble method consists of application of all six models per convolutional location. The final label of object is defined by majority voting of all predictions using all locations. **Reason**: this comparison illustrates the necessity of determination of input patch size for object-based classification.
- **$OCNN_{dense}$**: this method was inspired in the OCNN method from Zhao et al. (2017). In general, pixel-wise classification was implemented with CNN32 model, and the final label of object is defined by majority voting within the segment. This method integrates the CNN prediction and object segments in a straightforward application. **Reason**: this method is a simple strategy for refinement of CNN results.

5. Results

5.1. Performance of proposed CNNs

Table 3 shows the performance of CNN models for training, validation, and testing datasets. In general, proposed CNNs perform well in testing dataset, ranging from 83.9 (CNN8) to 93.2% (CNN256). Our findings show that increasing patch size improves model performance. In addition, we observe almost similar accuracies between training, validation and testing datasets. The agreement of training and testing accuracy is a positive measure for our study, indicating low or no overfitting in these models. In fact, the model architectures were developed to achieve a good generalization for our classification. The MLP network shows a lower accuracy (72.1%) compared to other CNNs. The pixel-level information introduces a real challenge for accurate predictions due to spectral similarity between land cover classes, such as "cropland vs. pasture" or "road vs. building roofs". Also, the computational time for model training increases from small to large patch sizes, ranging from 241 to 6544 s per epoch. The total time is dependent on model architecture (e.g., shallow or deep) and the number of epochs, such as CNN32 (~6.4 h) versus CNN256 (~90.9 h). It should be mentioned that total time only represents the model training, and the processing time for scene classification is presented in Section 5.4.

A crucial step in CNN application is the amount of training dataset required to properly learn the network parameters. The findings in Fig. 8 demonstrate the benefits of large datasets such as IowaNet for model generalization. In this experiment, CNN8 and CNN32 models were trained with different sizes (10,000; 50,000; 100,000; 250,000; 500,000; 900,000 samples). The results show that small training dataset (10,000) leads to high difference between training and testing accuracy in both models. For example, training accuracy of CNN32 is ~95.6%, while testing is 84.3%. This disagreement should be interpreted with caution because this is strong evidence of model overfitting. Overfitting is a common problem in deep learning models where the network performs well in the training dataset (closely fitted) but it has a generalization problem to make accurate predictions for new/unseen samples. In contrast, the training accuracy decreases from small to large datasets, and the overall agreement between training and testing increases for a large number of samples. We observed that the model performance for 900,000 samples is quite similar for both training and testing datasets; this is a positive measure of model generalization.

5.2. Segmentation scale effects

The summary of segmentation results with different scale parameters is presented in the Fig. 9. The Jaccard index values represent the geometric similarity of manually delineated and segmented polygons, and the largest values show more agreement between them. Overall, the best results were observed with spectral scales between 15 and 18, while the spatial values affect the segmentation, but they have less impact compared to spectral. The findings illustrate that accuracy gradually increases with increasing of spectral scale values, but it starts to decrease for spectral values higher than 18. The over-segmentation is clearly observed in the higher scales ($h_s > 18$), where object unit is presented in multiple polygons. In contrast, segmentation errors are more evident in smaller h_s scales (10–15). We observed that these low J-index values were caused by under-segmentation, where small objects are merged in large ones (e.g., houses and roads or grassland and cropland areas were

Table 3
Performance of CNN models with optimum hyperparameters.

Model	Input patch	Time per epoch	Total time	Accuracy (%) | Loss			No of Epochs
		secs	hours	Training	Validation	Testing	
MLP	d = 1 × 1(pixel)	101	8.41	71.8|0.832	71.9|0.875	72.1|0.872	300
CNN8	d = 8 × 8	241	10.0	83.5|0.450	83.8|0.446	83.9|0.442	150
CNN16	d = 16 x16	240	10.0	88.9|0.305	88.3|0.342	88.3|0.341	150
CNN32	d = 32 × 32	305	6.4	90.0|0.295	90.5|0.283	90.2|0.291	75
CNN64	d = 64 × 64	1791	24.9	92.1|0.229	90.1|0.306	90.1|0.293	50
CNN128	d = 128 × 128	1450	28.2	92.7|0.228	93.3|0.216	93.1|0.218	70
CNN256	d = 256 × 256	6544	90.9	94.2|0.182	93.2|0.218	93.2|0.217	50

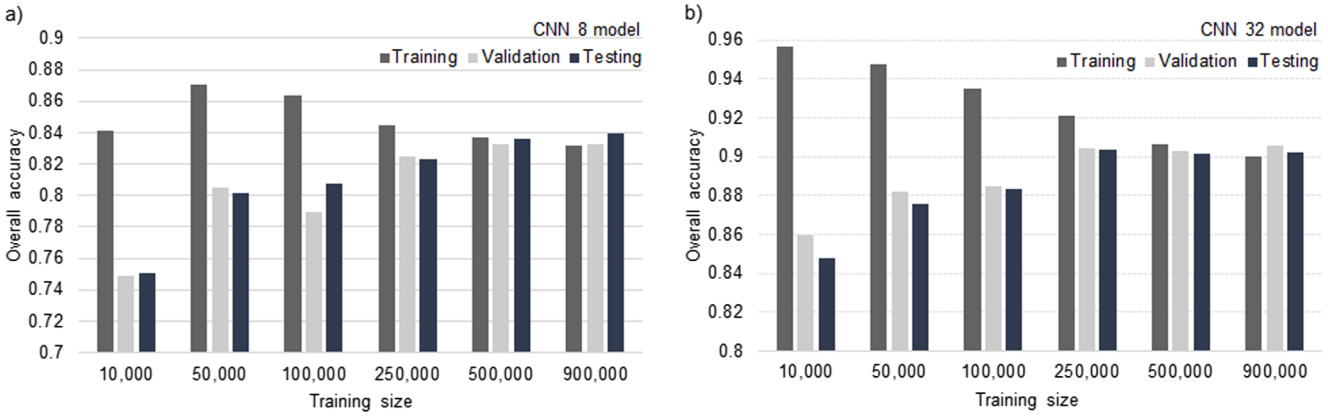

Fig. 8. Performance for (a) CNN8 and (b) CNN32 models with different training sizes. This experiment is performed up to 100 epochs. The results show the agreement of training, validation, and testing values using larger number of samples.

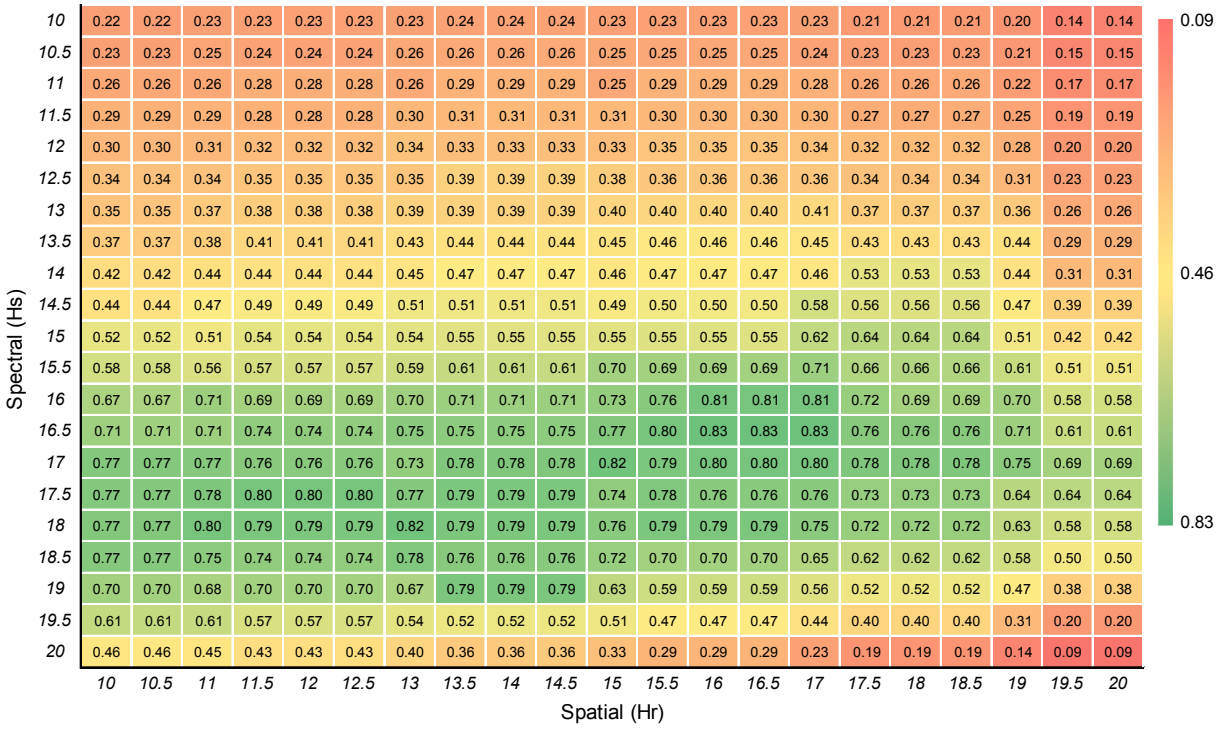

Fig. 9. Segmentation scale effects with different spatial (h_r) and spectral (h_s) values in Mean-Shift algorithm. The Jaccard index values (Section 4.3) were calculated using 1200 reference and segmented polygons over eight counties. The Jaccard index ranges from 0 (no match) to 1 (perfect match). Smaller scales tend to smooth objects, while higher scales tend to over-segment objects in image. These experiments were conducted with minimum region value of 25 pixels.

not properly separated) and this is highly penalized in this metric. In another way, segmentation results using h_s between 16 and 17.5 are near-similar and consistent when we applied h_r values around 11.0 or 16.5. These scale values are potential global parameters for our study. In addition, Fig. 10 shows the segmentation results with three combination of scales. The visual assessment corroborates with J-index results: some

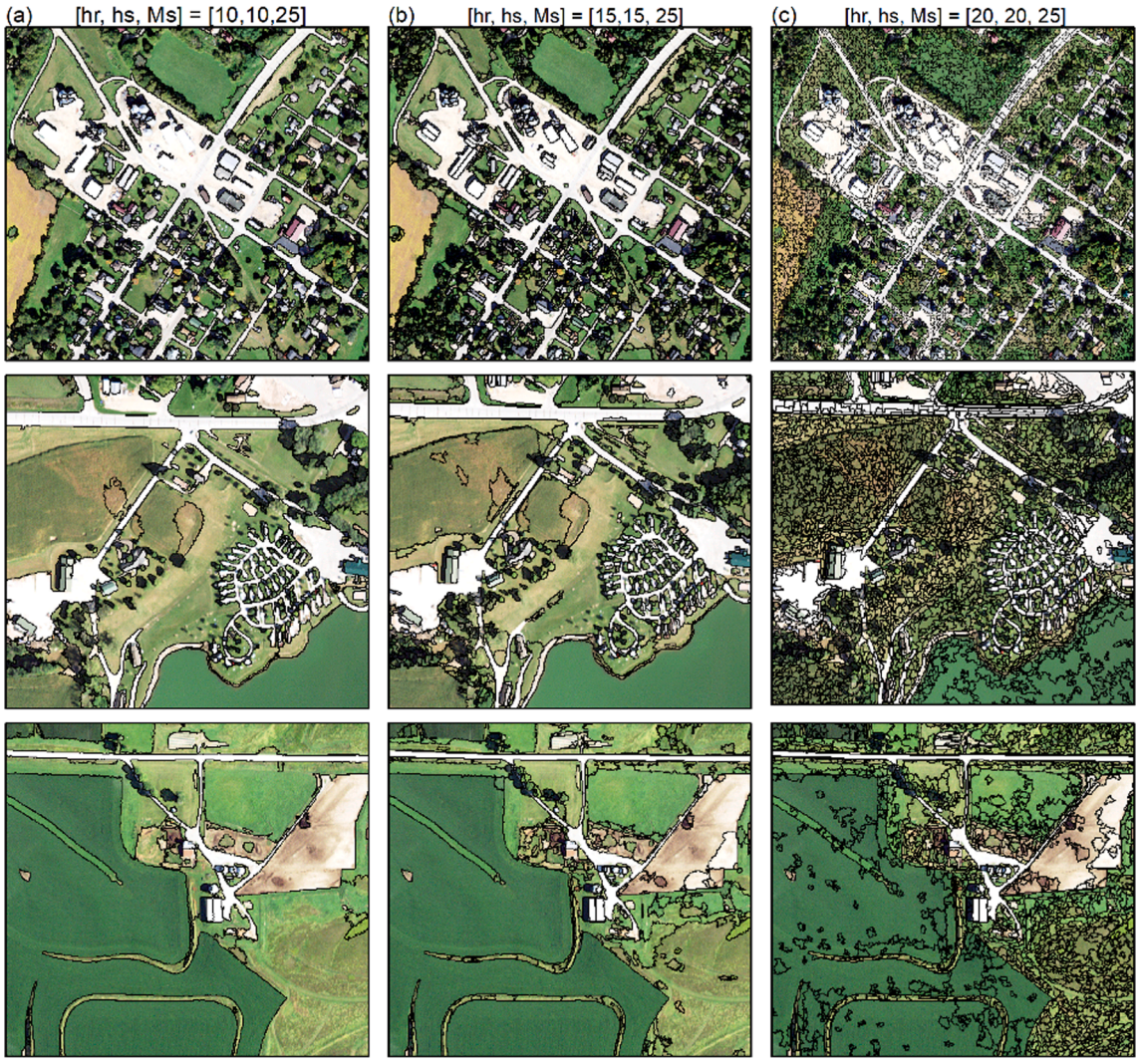

Fig. 10. Segmentation maps with three combination of input scales [h_r, h_s, M_s] in the Mean-Shift algorithm. The parameters are spatial (h_r) and spectral (h_s) and minimum region (M_s). (a) combination 1 = [10, 10, 25], (b) combination 2 = [15, 15, 25], (c) combination 3 = [20, 20, 25].

objects were not delineated in small scales (see buildings and trees), while higher scales present over-segmentation in all landscapes. The results show the capabilities to represent small-sized objects in combination 2, and this highlights the applicability of predefined $M_s = 25$. A high quality of the image segmentation is difficult with three spectral bands at 1-m spatial resolution. However, while some results might not be fully accurate, we emphasized that the proposed multi-OCNN method manages well the merging of multiple polygons in the final product. Based on these results and further investigation of segmented outputs, the global parameters were selected as [h_r, h_s, M_s] = [16.0, 16.5, 25] for image segmentation, and the segmented objects are roughly the same size or slightly smaller than real-world target to be classified in this large-area classification.

Table 4
Confusion matrix for proposed multi-OCNN framework.

No of pixels	Reference data											
Classified data	Barren	Crop.	Fallow	Forest	Grass.	Lake	River	Road	Shadow	Struct.	Total	User acc. (%)
Barren (0)	**832**	3	77	4	9	10	5	22	5	68	1035	80.4
Cropland (1)	0	**913**	20	6	28	0	1	2	0	0	970	94.1
Fallow (2)	57	9	**798**	2	22	2	4	15	0	14	923	86.5
Forest (3)	5	16	4	**884**	20	5	6	2	81	10	1033	85.6
Grassland (4)	5	57	95	81	**917**	0	3	21	4	9	1192	76.9
Lake (5)	3	1	0	0	0	**906**	20	1	30	6	967	93.7
River (6)	8	0	1	0	2	58	**948**	2	9	3	1031	91.9
Road (7)	64	0	5	1	2	0	1	**880**	3	75	1031	85.3
Shadow (8)	9	1	0	22	0	19	12	6	**854**	25	948	90.1
Structure (9)	17	0	0	0	0	0	0	49	14	**790**	870	90.8
Total	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	**10,000**	
Prod. acc. (%)	83.2	91.3	79.8	88.4	91.7	90.6	94.8	88	85.4	79.0		

Overall accuracy (OA): 87.2%; Overall kappa (κ): 85.8%.

5.3. Performance of multi-OCNN framework

This section presents the classification performance of proposed multi-OCNN framework. Table 4 shows the confusion matrix and overall classification accuracy (OA) using 10,000 reference samples. In general, scene classification using multi-OCNN achieves satisfactory accuracy of 87.2% and $\kappa = 85.8\%$. The producer's accuracy (PA) varies among the land cover classes, from 79.0 to 94.8%. The higher PA values were observed for cropland (91.3%), grassland (91.7%), and river (94.8%), while lower values were found for structure (79%), fallow (79.8%), and barren (83.2%). Regarding the user's accuracies (UA), the values range from 76.9 to 94.1% (Table 4). User's accuracies were greater than 90% for five land cover classes: cropland (94.1), lake (93.7), river (91.9), shadow (90.1), and structure (90.8%). Note that user's and producer's accuracies are quite similar; they mostly differ between 1 and 6%. In particular, both PA and UA values were higher than 90% for cropland, which is a positive measure for crop-dominated landscape as Iowa. In contrast, some classes present persistent confusion. For instance, road class is often classified as structure and barren (and vice-versa). Also, lower UA was observed for grassland class (76.9%), where these grassland pixels are incorrectly classified as cropland, fallow and forest areas. Not surprisingly, misclassification occurs for classes with most similar

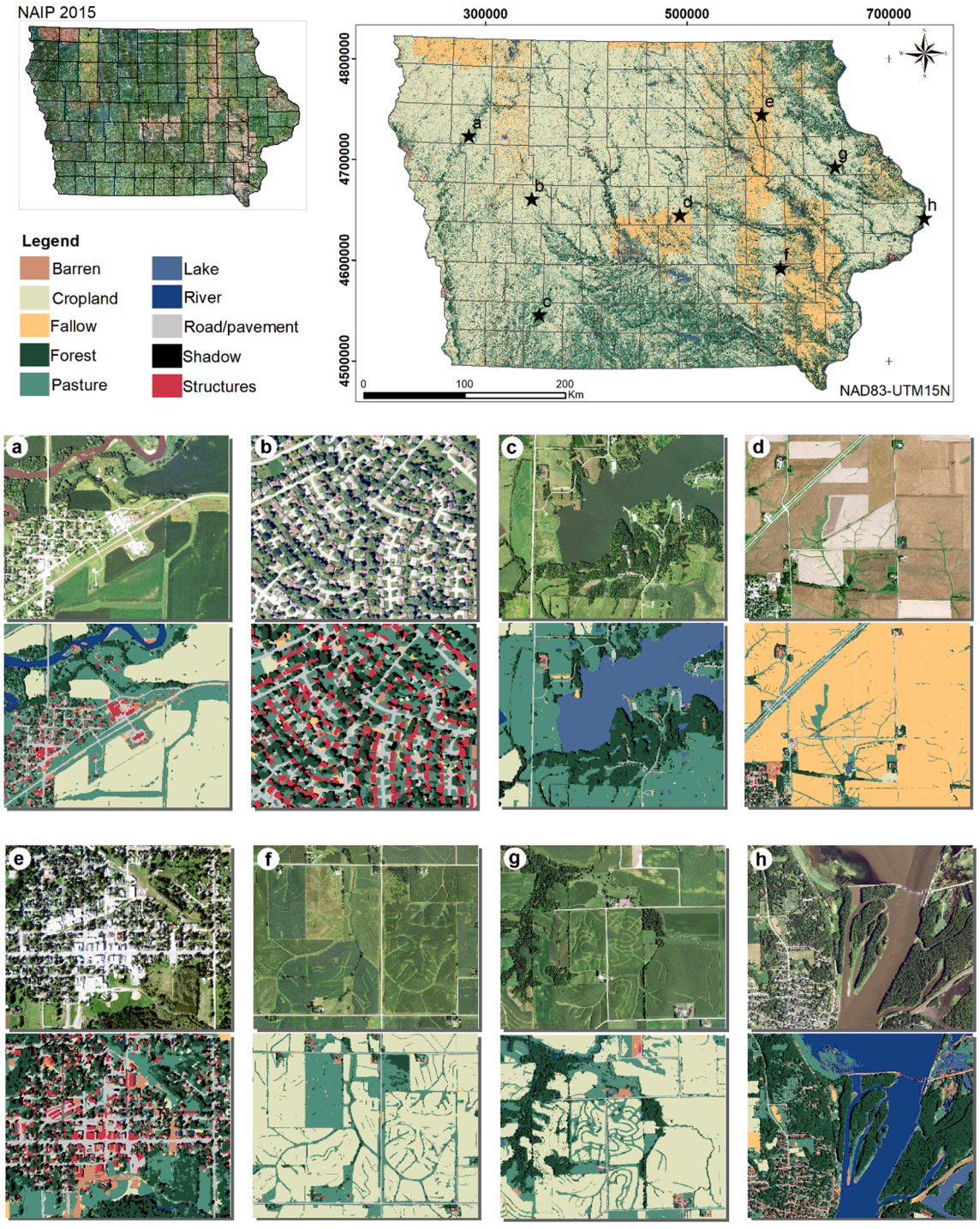

Fig. 11. Classification results of proposed multi-OCNN method. The panels are subset images with land cover result. Note that large banding of the "fallow" class is caused by different flight dates during NAIP 2015 acquisition.

spatial-spectral features. Likewise, labeled pixels as "lake" are typically mixed with river and shadow classes. The definition of river and lake classes makes sense for CNN classifier (spatial features are different), but these pixels are quite similar in spectral domain. We observed that shadow pixels might be confused in dense forest areas, while tree shadow is often confused with dark lake pixels. The shadow pixels are typically observed close to forest areas (tree shadow). The spectral mixing leads to some confusion between these classes.

Fig. 11 presents the land cover mapping derived from multi-OCNN framework. These examples show the benefits of multi-OCNN to preserve the high-level of spatial details. For example, panels (11-b) and (11-e) present a dense residential area at fine mapping. Small houses, individual trees, and grassland areas are clearly observed in these areas. Natural targets such as rivers and lake were also well-represented with the correct label (panels 11-c and 11-h). Also, conservation practices in cropland areas were correctly captured in the map, such as grassed waterways (panels 11-d, 11-f, and 11-g). Despite the spectral similarities between cropland and grassland classes, the pasture pixels were well distinguished (panels 11-f and 11-g). In panel (11-e), we observe a reasonable classification for small urban areas, but the road/pavement class was misclassified as a barren surface. This problem is observed in other scene areas. In contrast, we observed the capability for cropland delineation, providing a good distinction between grassland and cropland areas. This CNN-based cropland boundaries can support the refinement of cropland mapping, such as Cropland Data Layer product. Note that NAIP imagery are collected on different dates across the state and the land cover results are accurate for a specific moment but they are limited for comparison to the broader area. This temporal effect of image acquisition is easily observed in the land cover results; see the fallow banding with adjacent cropland areas. Fig. 12 presents other mapping results for our framework. In general, these examples show a consistent map in these panels, but some confusion is also observed for structures and roads (panel 12-b). Note that object-based classification reduces the salt-and-pepper effects in the final map. Moreover, the riparian forest is well-classified around the river. Bright areas (roads, structures, and sand) are often misclassified due to intra-class spectral similarity. For instance, panel (a) in Fig. 12 shows sand near to river is misclassified as structure. This error appears in some areas. In contrast, the shadow in forest area is clearly observed in panel 12-d. The lake area is well-classified in panel 12-e, but we also observe the river pixels in wrong area. Also, detailed mapping is observed in small city (panel 12-f).

5.4. Comparison with other CNN-based methods

Table 5 presents the comparison of proposed multi-OCNN with other methods. This summary shows per-class F-score accuracy, overall accuracy, and kappa coefficient for each method. As mentioned, pixel-wise CNN is the traditional method for land cover classification using patch-based input, while the fixed-OCNN is the object-based classification using a single model (instead of multiscale). In general, overall accuracy ranges from 68.7 to 87.2%. The proposed multi-OCNN outperforms all experimental methods (OA = 87.2%), improving 5.2, 18.5, and 5.6% regarding to fixed-$OCNN_{16}$, $OCNN_c$, and pixel-wise CNN_{16}. Also, $OCNN_{all}$ and $OCNN_{dense}$ results did not outperform the multi-OCNN approach, suggesting the relevance of input size or object analysis for this classification. In this comparison, lower input size produces slightly higher classification accuracy. In contrast, the overall accuracies of fixed-OCNN and pixel-wise CNN are almost similar; there is no significant improvement by implementing object-based classification without multiscale CNNs. In contrast, $OCNN_c$ method achieves the lowest classification accuracy (68.7%). This result suggests the importance of skeleton-based analysis and multiple convolutional locations for better prediction.

While some methods have higher F-score per class (bold text), multi-OCNN performs the best for five classes (barren, cropland, grassland, shadow, and structure). In contrast, natural targets of large areal extent are typically well-classified by large input patches ($OCNN_{256}$). We observe that both fixed-OCNN and pixel-wise CNN perform more poorly in the structure class for urban areas. Also, note that CNN256 presents a higher F-score for road class (91.1%), but we should evaluate this result with caution. It was observed that these elongated targets are over-

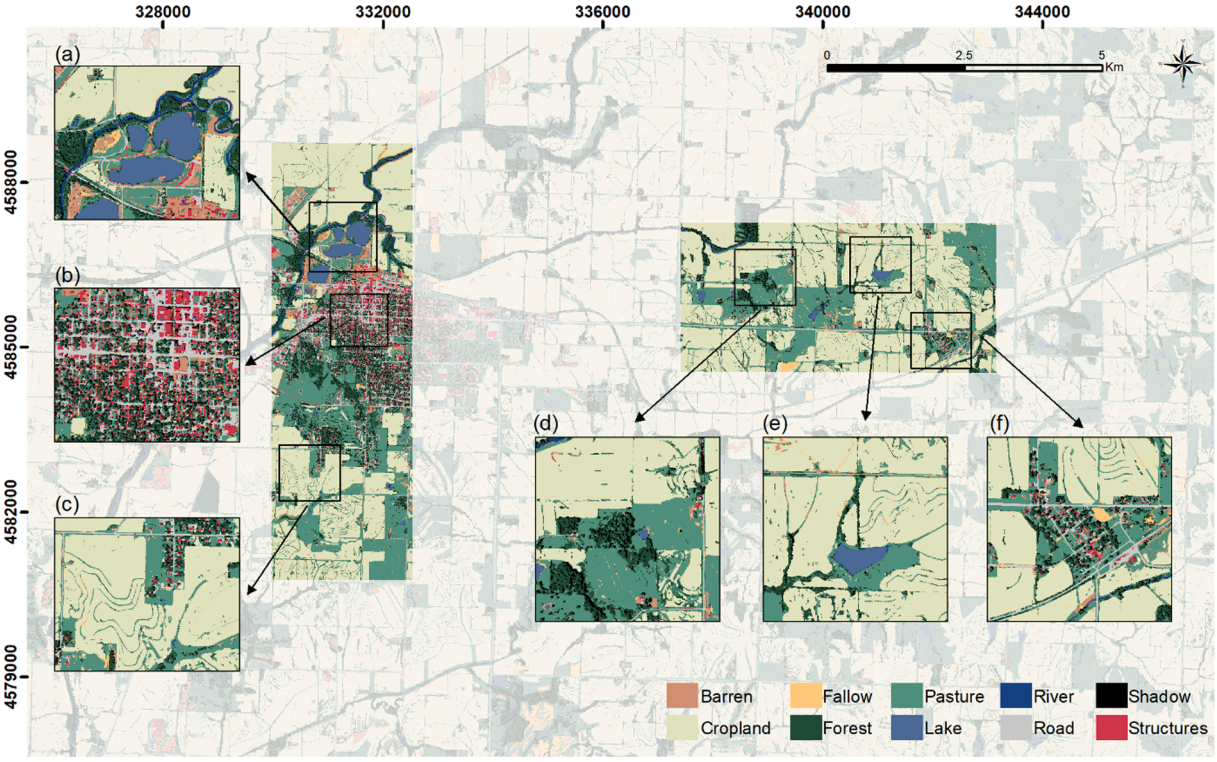

Fig. 12. Other examples of proposed multi-OCNN.

Table 5
Comparison of proposed multi-OCNN with different methods using F-score metric (Section 4.5). The F-score is the harmonic mean of producer's and user's accuracies in percentage (F-score = 2 * (PA × UA) / (PA + UA)). The best result for each class is marked in bold. The classification methods are described in Section 4.5 and represent different strategies for CNN-based land cover classification.

Land cover classes	pixel-based classification			Object-based classification				object + multiscale		
	CNN_{16}	CNN_{64}	CNN_{256}	$OCNN_{16}$	$OCNN_{64}$	$OCNN_{256}$	$OCNN_{dense}$	$OCNN_{c}$	$OCNN_{all}$	**Multi-OCNN**
Barren	73.6	71.5	60.5	76.9	71.6	57.9	76.6	66.6	78.0	**81.8**
Cropland	89.3	85.8	84.7	90.3	88.4	83.3	89.5	73.3	89.6	**92.7**
Fallow	81.0	79.6	**84.7**	84.4	77.6	82.8	83.2	68.9	84.5	83.0
Forest	**87.1**	82.7	79.8	83.9	81.1	79.3	82.4	68.5	82.3	87.0
Grassland	68.5	61.3	59.9	71.2	64.0	58.6	69.0	63.1	66.8	**83.7**
Lake	84.5	93.0	95.6	87.3	96.3	96.8	92.3	85.9	**97.0**	92.1
River	83.9	91.0	95.1	88.1	96.0	96.5	91.4	52.4	**97.1**	93.3
Road	86.8	87.7	**91.1**	84.5	84.9	87.5	84.7	38.7	87.7	86.7
Shadow	72.5	70.9	68.3	65.6	66.9	65.1	69.5	80.9	65.0	**87.7**
Structure	83.5	81.1	77.7	82.2	79.6	76.5	82.3	77.3	80.8	**84.5**
Kappa (κ):	79.5	78.8	78.7	80.0	79.0	78.7	80.4	65.2	81.7	**85.8**
OA.:	81.6	80.9	80.8	82.0	81.1	80.8	82.4	68.7	84.0	**87.2**

expanded, causing shape distortion (Fig. 13). This finding suggests the importance of correct observation scale for classification in urban areas, and consequently, the application of multiple CNNs (and pixel-based MLP).

The visual assessment of these methods is shown in Fig. 13. In these experimental areas, we illustrate the benefits of multi-OCNN method compared to pixel-wise CNN and fixed-OCNN, especially in urban area. The traditional CNN smooths small targets and introduces uncertainties in the target edges. For instance, pixel-wise CNN_{64} has salt-and-pepper effects and some mistakes are observed by misclassifying the roads as structures. Small targets, such as residences, were difficult to achieve good results with pixel-wise CNN_{64} (64 × 64 pixels). However, a similar problem is also observed for fixed-$OCNN_{64}$. The object-based classification should preserve the geometry fidelity, but fixed-$OCNN_{64}$ expands the small objects and smooths linear features such as grassed waterways. The results demonstrate that urban areas are not easy to classify with input patch of 64 × 64 pixels. Also, Fallow pixels are incorrectly labeled as barren by both fixed-OCNN and pixel-wise CNN, but these classes are commonly mixed ones. At the end, these results suggest that multiscale CNNs improve the quality of the map, and object-based classification is only relevant when multiple CNNs are implemented.

This study claims the applicability of multi-OCNN framework for large-area classification. The results of computational time are summarized in Table 6. These experimental results were developed for 100,000 m^2 area. Although the computational time dependents on number of objects and convolutional positions, the proposed multi-OCNN presents reasonable time (15.9 sec) for 100,000 m^2. There is clear evidence of time reduction for object-based compared to pixel-wise classification, varying from a few seconds to hours. For instance, the total time of multi-OCNN framework is 8.1 and 111.5 times faster than traditional pixel-wise CNN_{16} or CNN_{256}, respectively. In comparison, $OCNN_c$ presents the fast classification (total time = 3 sec) due to a single convolutional location per object. However, this method shows the lowest overall accuracy/kappa among the approaches. We observed that lower input patch has faster processing time among pixel-wise CNNs. For the same model, object-based classification (e.g., $OCNN_{16}$) is 10 times faster than pixel-wise classification (e.g., CNN_{16}). With this time of performance and a single computer, the statewide classification (e.g.: Iowa) could take 268 days for multi-OCNN compared to ~48 years for CNN_{64}. Thus, both results would be limited for operational mapping. However, note that multi-OCNN classification was implemented in HPC clusters with several computer nodes. As a result, classification result is achieved in <12 days (up to 25 nodes). Therefore, this suggests that our framework is faster than traditional CNN, but it still requires significant computational resources in large-scale applications.

6. Discussion

This study describes a new multiscale object-based CNN framework for large-area land cover classification at 1-m resolution. The proposed framework integrates (i) image segmentation, (ii) object analysis, and (iii) multiple CNNs. While previous studies have implemented OCNN classification, this study is the first large-area classification exploring multiscale CNNs and skeletonize algorithm. This multi-OCNN was successfully applied in NAIP imagery across Iowa, United States. The overall result of multi-OCNN is shown in Table 4. In general, our findings show that multi-OCNN provides a consistent land cover product with a high-level of spatial detail (Figs. 11 and 12), achieving a satisfactory accuracy (OA ~87.2% and kappa ~85.8%). The assessment in Fig. 13 showed that multi-OCNN helped to improve the classification results compared to other algorithms, such as fixed-OCNN and pixel-wise CNN. Similar benefits of OCNN were also presented in other studies (Längkvist et al., 2016; Zhao et al., 2017; Lv et al., 2018; Zhang et al., 2018; Jin et al., 2019; Liu et al., 2019). For instance, Zhao et al. (2017) presented an improvement of 0.87 to 5.46% for OCNN compared to pixel-wise CNN in different classification datasets (worldview-2, Pavia center, Vaihingen). Similarly, Liu et al. (2019) achieved the overall accuracy of 95.33% using object-based post-classification refinement of CNN maps in urban context. While these studies highlight the advantage of OCNN, we observed that these benefits are highly dependent on multiple CNNs for correct classification (Fig. 13).

Typically, standard CNN extracts deep spatial-related features for fixed input patch size. However, each object has its distinctive geometric characteristics, and spatial-related features are scale-dependent (Zhao and Du, 2016). Therefore, the classification using a single CNN (with fixed input size) will not solve the problem of observation scale across heterogeneous landscape (ranging from small tree to large cropland). The results in Table 5 confirm the relevance of multiple CNNs for object classification, where fixed-OCNN achieved lower accuracy than proposed multi-OCNN. In this context, Zhang et al. (2018) combined two CNN models with large (128 × 128) and small (48 × 48) window size for object-based classification. However, they recognized the limitation of two CNN models to represent complex geometries, recommending the integration of multiple models. The importance of multiple models was demonstrated in other studies. Längkvist et al. (2016) developed four CNNs with different input sizes (15, 25, 35, 45) and found higher accuracy for multiple CNN architecture (94.49%) compared to single CNN (90.02%). Sun et al. (2019) proposed a fusion of deep multiscale features (three CNNs) for building extraction. Our findings agree with these previous studies, and the fine-detail mapping in Fig. 11 corroborates to prove the value of multi-OCNN using six CNN models. Note that MLP network is also used to support the classification of tiny objects. However, pixel-level MLP network is only used when the convolutional

Fig. 13. A comparison of land cover classification from different methods. (a) true-color image, (b) pixel-wise CNN_{64}, (c) fixed-$OCNN_{64}$, and (d) proposed multi-OCNN. The classification methods are described in Section 4.5.

location is close to object edge ($S_d < 3$ m), and some developers might prefer to simplify our approach without MLP.

While the multiscale CNNs represent a solution for the problem of observational scale, the application of multi-OCNN is highly dependent on object analysis with decision-rules. Admittedly, this processing stage involves further efforts to create a robust and fast algorithm for heterogeneous landscape. In this analysis, skeletonize algorithm was successfully used to generate the morphological representation of segmented objects (Section 4.4). This method is essential for definition of convolutional locations, and subsequently, the determination of input patch size. The results in Table 5 also confirm the importance of multiple convolutional locations, where poor result of $OCNN_c$ illustrates the limitation of single prediction for final object label. Previously, Huang et al. (2018) proposed a skeleton-based decomposition with street blocks for land use mapping. However, the application was limited to regular urban blocks instead of any functional object (e.g. road, lake, trees). In contrast, Zhang et al. (2018) used object analysis in the OCNN approach for urban land use classification, but this method has some limitations for heterogeneous landscape. For geometrically complex objects, the object analysis does not have a specific criterion to define the appropriate window size or implementation of multiple CNN models. Therefore, our object analysis contributes to OCNN application in broader context, giving the flexibility to work for any type of land target.

Despite the recent progress, the generation of large-area land cover product using CNNs is still a challenge for high-resolution data. Some studies have used post-classification refinement with segmentation

Table 6
Computational time for different methods. The experimental results were performed for a scene area of 100,000 m^2. The classification methods are described in Section 4.5.

Method	Computation time (secs)			Speed	Total time
	Segmentation*	Object analysis*	Scene classification**	(pixels/sec)	(secs)
Pixel-wise CNN_{16}	–	–	128.6	777	128.6
Pixel-wise CNN_{64}	–	–	1040.0	96	1,040.0
Pixel-wise CNN_{256}	–	–	1772.2	56	1,772.2
Fixed-$OCNN_{16}$	1.4	4.7	6.2	8,130	12.3
Fixed-$OCNN_{64}$	1.4	4.7	30.4	2,739	36.5
Fixed-$OCNN_{256}$	1.4	4.7	90.4	1,036	96.5
$OCNN_c$	1.4	1.1	0.5	33,333	3.0
$OCNN_{all}$	1.4	4.7	218.4	445	224.5
$OCNN_{dense}$	1.4	–	269	369	270.4
Multi-OCNN	1.4	4.7	9.8	6289	15.9

* Computer Resource: Intel Xeon(R) E3 −1270 (3.80 GHz) processor.
** Computer Resource: two 8-Core Intel Xeon (R) E5 2650 (2.0 GHz) processor.

results (Langkvist et al., 2016; Zhao et al., 2017; Liu et al., 2019), but it still requires a dense pixel-by-pixel prediction. The data handling of huge number of scenes requires computational resources and integrated algorithm. The object analysis showed the advantage to reduce the number of CNN predictions and speed up the overall processing. The results in Table 6 demonstrate that multi-OCNN is 8.1 and 111.5 times faster than traditional pixel-wise CNN_{16} or CNN_{256}, respectively. These findings are similar to other studies. For instance, Zhang et al. (2018) shows that pixel-wise CNN is ~100 times longer than OCNN with limited computational resources. At the end, this time improvement is only possible because object-based CNN gives the flexibility to develop a different scheme for CNN application instead of pixel-based approach.

The segmentation step is a relevant part of object-based classification, and the effects of scale parameters are shown in Fig. 9. For mean-shift algorithm, the application of smaller spatial/spectral scales creates smoother objects, and higher scales are useful when the targets are small and spectrally similar such as streets and building roofs (Fig. 10). The optimal selection of segmentation parameters is a well-known challenge in heterogeneous landscape at high spatial resolution (Drăguţ et al., 2010; Zhang et al., 2014), and the automation of scale parameter selection is under investigation (Anders et al., 2011; Drăguţ et al., 2014; Yang et al., 2015). Ideally, the global scale parameters need to preserve the geometric representation of small objects but also avoids the over-segmentation of large ones. The applied segmentation optimization with multiple experiments has proven to be an effective way for this study. The global scale parameters produced a slight over-segmentation results in some cases but proposed multi-OCNN manages well the fragmented objects to produce meaningful unit in the final product. While our efforts were concentrated on a practical strategy for statewide application, some users/developers might adopt locally defined scales in their studies, especially when the segmentation is applied to single scene or small area.

Exploring the development of multi-OCNN, many lessons were learned to turn CNN into a practical tool. The challenges for multi-OCNN application are identified as computational resources, programming skills, development of training samples and multiple networks, and data manipulation (~1 TB). The development of multiple CNNs involves an intensive training in GPUs using different configurations. In addition, multi-OCNN for large-area classification is highly dependent on computational resources. For instance, the statewide classification required a massive object label (this study: ~1,01 billion objects), and each object will have multiple convolutional locations for CNN application. For this research HPC cluster allowed access to multiple computer nodes, which make the overall classification more reasonable in terms of time. The optimization of object analysis could be proposed to allow the same process in a single computer, including the fine-tuning of parameters (e.g., interval). Also, other segmentation algorithms can be explored in future studies to compare the benefits in terms of geometry quality and processing speed, including eCognition multi-resolution, quick-shift, and watershed algorithms.

7. Conclusions

Convolutional neural network offers the ability to explore spatial-related deep features for land cover classification. However, CNN classification introduces certain challenges for operational mapping, especially in a broader context. In this research, we presented a new multiscale object-based CNN for large-area land cover classification. This framework includes a series of stages, including image segmentation, object analysis with skeleton-based method and multiple CNNs. We also developed a new benchmark dataset ("IowaNet") with 1 million images (10 classes) for training of multi-resolution CNNs. In general, we demonstrated that multi-OCNN is a feasible method for operational land cover mapping at 1-m resolution, achieving overall accuracy of 87.2%. We found that multi-OCNN presents faster and accurate classification compared to other OCNN frameworks. The results have shown that traditional pixel-wise CNN_{64} produces blurred boundaries, while multi-OCNN preserves the target edges in different contexts. Moreover, the results indicated relevance of multiscale CNNs (six models) in object-based classification: multi-OCNN shows higher accuracy than fixed-OCNN or $OCNN_{all}$ methods. The results suggest the benefits of object analysis for appropriate selection of CNN model, preserving the geometry fidelity in this object-based classification. Also, we found that OCNN improves the speed in overall processing; multi-OCNN is 8.1 and 111.5 times faster than traditional pixel-wise CNN_{16} or CNN_{256}, respectively. While multi-OCNN offers a practical framework for large-area mapping, this application is highly dependent on computational resources for data storage and handling (this study classifies ~1.01 billion of objects). With increased availability of EO data, further research is recommended to understand the potential of multi-OCNN in satellite datasets, such as GeoEye-1 and WorldView-4 imagery.

Declaration of Competing Interest

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

Acknowledgement

We gratefully acknowledge the USDA's Farm Service Agency (FSA) for providing the NAIP 2015 dataset. The multi-resolution IowaNet dataset is available at https://doi.org/10.5281/zenodo.3385318. We thank the HPC-ISU and ResearchIT for the computer resources of Condo and Pronto. The processed data in this study are available from the corresponding author upon reasonable request. We thank the three

anonymous reviewers for their valuable comments.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.isprsjprs.2020.08.004.

References

Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Deng, L., Penn, G., Yu, D., 2014. Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 22 (10), 1533–1545.

Alshehhi, R., Marpu, P.R., Woon, W.L., Dalla Mura, M., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 130, 139–149.

Anders, N.S., Seijmonsbergen, A.C., Bouten, W., 2011. Segmentation optimization and stratified object-based analysis for semi-automated geomorphological mapping. Remote Sens. Environ. 115 (12), 2976–2985.

Audebert, N., Le Saux, B., Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogramm. Remote Sens. 140, 20–32.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Basu, S., Ganguly, S., Nemani, R.R., Mukhopadhyay, S., Zhang, G., Milesi, C., Cook, B., 2015. A semiautomated probabilistic framework for tree-cover delineation from 1-m NAIP imagery using a high-performance computing architecture. IEEE Trans. Geosci. Remote Sens. 53 (10), 5690–5708.

Belgiu, M., Drăguţ, L., 2016. Random forest in remote sensing: a review of applications and future directions. ISPRS J. Photogramm. Remote Sens. 114, 24–31.

Blaschke, T., 2010. Object based image analysis for remote sensing. ISPRS J. Photogramm. Remote Sens. 65 (1), 2–16.

Blaschke, T., Hay, G.J., Kelly, M., Lang, S., Hofmann, P., Addink, E., Tiede, D., 2014. Geographic object-based image analysis–towards a new paradigm. ISPRS J. Photogramm. Remote Sens. 87, 180–191.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv: 1508.00092.

Chen, J., Chen, J., Liao, A., Cao, X., Chen, L., Chen, X., Zhang, W., 2015. Global land cover mapping at 30 m resolution: a POK-based operational approach. ISPRS J. Photogramm. Remote Sens. 103, 7–27.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 54 (10), 6232–6251.

Chen, Y., Ming, D., Lv, X., 2019. Superpixel based land cover classification of VHR satellite image combining multi-scale CNN and scale parameter estimation. Earth Sci. Inf. 12 (3), 341–363.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 5, 603–619.

Congalton, R.G., Green, K., 2002. Assessing the Accuracy of Remotely Sensed Data: Principles and Practices. CRC Press.

Dabiri, Z., Blaschke, T., 2019. Scale matters: a survey of the concepts of scale used in spatial disciplines. Eur. J. Remote Sens. 52 (1), 419–434.

DeLancey, E.R., Simms, J.F., Mahdianpari, M., Brisco, B., Mahoney, C., Kariyeva, J., 2020. Comparing deep learning and shallow learning for large-scale wetland classification in Alberta, Canada. Remote Sens. 12 (1), 2.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 145, 3–22.

Drăguţ, L., Csillik, O., Eisank, C., Tiede, D., 2014. Automated parameterisation for multi-scale image segmentation on multiple layers. ISPRS J. Photogramm. Remote Sens. 88, 119–127.

Drăguţ, L., Tiede, D., Levick, S.R., 2010. ESP: a tool to estimate scale parameter for multiresolution image segmentation of remotely sensed data. Int. J. Geographical Inform. Sci. 24 (6), 859–871.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2012. Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1915–1929.

Fu, G., Liu, C., Zhou, R., Sun, T., Zhang, Q., 2017. Classification for high resolution remote sensing imagery using a fully convolutional network. Remote Sens. 9 (5), 498.

Fu, Z., Sun, Y., Fan, L., Han, Y., 2018. Multiscale and multifeature segmentation of high-spatial resolution remote sensing images using superpixels with mutual optimal strategy. Remote Sens. 10 (8), 1289.

Fukunaga, K., Hostetler, L., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inf. Theory 21 (1), 32–40.

Ge, F., Wang, S., Liu, T., 2007. New benchmark for image segmentation evaluation. J. Electron. Imag. 16 (3), 033011.

Gong, P., Wang, J., Yu, L., Zhao, Y., Zhao, Y., Liang, L., Li, C., 2013. Finer resolution observation and monitoring of global land cover: first mapping results with Landsat TM and ETM+ data. Int. J. Remote Sens. 34 (7), 2607–2654.

Hay, G.J., Blaschke, T., Marceau, D.J., Bouchard, A., 2003. A comparison of three image-object methods for the multiscale analysis of landscape structure. ISPRS J. Photogramm. Remote Sens. 57 (5–6), 327–345.

Homer, C., Dewitz, J., Yang, L., Jin, S., Danielson, P., Xian, G., Megown, K., 2015. Completion of the 2011 National Land Cover Database for the conterminous United States–representing a decade of land cover change information. Photogramm. Eng. Remote Sens. 81 (5), 345–354.

Hossain, M.D., Chen, D., 2019. Segmentation for Object-Based Image Analysis (OBIA): a review of algorithms and challenges from remote sensing perspective. ISPRS J. Photogramm. Remote Sens. 150, 115–134.

Hu, F., Xia, G.S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. 7 (11), 14680–14707.

Hu, Y., Zhang, Q., Zhang, Y., Yan, H., 2018. A deep convolution neural network method for land cover mapping: a case study of Qinhuangdao, China. Remote Sens. 10 (12), 2053.

Huang, B., Zhao, B., Song, Y., 2018. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. Remote Sens. Environ. 214, 73–86.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 11–19.

Jia, K., Liang, S., Zhang, N., Wei, X., Gu, X., Zhao, X., Xie, X., 2014. Land cover classification of finer resolution remote sensing data integrating temporal features from time series coarser resolution data. ISPRS J. Photogramm. Remote Sens. 93, 49–55.

Jin, B., Ye, P., Zhang, X., Song, W., Li, S., 2019. Object-Oriented method combined with deep convolutional neural networks for land-use-type classification of remote sensing images. J. Indian Soc. Remote Sens. 1–15.

Jin, S., Yang, L., Danielson, P., Homer, C., Fry, J., Xian, G., 2013. A comprehensive change detection method for updating the National Land Cover Database to circa 2011. Remote Sens. Environ. 132, 159–175.

Johnson, R., Zhang, T., 2013. Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, pp. 315–323.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14 (5), 778–782.

Längkvist, M., Kiselev, A., Alirezaie, M., Loutfi, A., 2016. Classification and segmentation of satellite orthoimagery using convolutional neural networks. Remote Sens. 8 (4), 329.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

LeCun, Y., Kavukcuoglu, K., Farabet, C., 2010, May. Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems. IEEE, pp. 253–256.

Li, J., Zhang, R., Li, Y., 2016, July. Multiscale convolutional neural network for the detection of built-up areas in high-resolution SAR images. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 910–913.

Li, W., Dong, R., Fu, H., 2019. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. Remote Sens. 11 (1), 11.

Li, X., Myint, S.W., Zhang, Y., Galletti, C., Zhang, X., Turner II, B.L., 2014. Object-based land-cover classification for metropolitan Phoenix, Arizona, using aerial photography. Int. J. Appl. Earth Obs. Geoinf. 33, 321–330.

Liu, Q., Hang, R., Song, H., Li, Z., 2017. Learning multiscale deep features for high-resolution satellite image scene classification. IEEE Trans. Geosci. Remote Sens. 56 (1), 117–126.

Liu, S., Qi, Z., Li, X., Yeh, A.G.O., 2019. Integration of convolutional neural networks and object-based post-classification refinement for land use and land cover mapping with optical and SAR data. Remote Sens. 11 (6), 690.

Liu, T., Abd-Elrahman, A., Morton, J., Wilhelm, V.L., 2018. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. GISci. Remote Sens. 55 (2), 243–264.

Liu, Z., Luo, P., Wang, X., Tang, X., 2015. Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision, pp. 3730–3738.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Lu, D., Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. 28 (5), 823–870.

Lu, M., Chen, J., Tang, H., Rao, Y., Yang, P., Wu, W., 2016. Land cover change detection by integrating object-based data blending model of Landsat and MODIS. Remote Sens. Environ. 184, 374–386.

Lv, X., Ming, D., Lu, T., Zhou, K., Wang, M., Bao, H., 2018. A new method for region-based majority voting CNNs for very high resolution image classification. Remote Sens. 10 (12), 1946.

Ma, L., Li, M., Ma, X., Cheng, L., Du, P., Liu, Y., 2017. A review of supervised object-based land-cover image classification. ISPRS J. Photogramm. Remote Sens. 130, 277–293.
Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2016. Convolutional neural networks for large-scale remote-sensing image classification. IEEE Trans. Geosci. Remote Sens. 55 (2), 645–657.
Mahdianpari, M., Salehi, B., Mohammadimanesh, F., Motagh, M., 2017. Random forest wetland classification using ALOS-2 L-band, RADARSAT-2 C-band, and TerraSAR-X imagery. ISPRS J. Photogramm. Remote Sens. 130, 13–31.
Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., Zhang, Y., 2018. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sens. 10 (7), 1119.
Marmanis, D., Wegner, J.D., Galliani, S., Schindler, K., Datcu, M., Stilla, U., 2016. Semantic segmentation of aerial images with an ensemble of cnss. ISPRS Ann. Photogram. Remote Sens. Spatial Inform. Sci. 2016 (3), 473–480.
Martins, V.S., Kaleita, A., Gelder, B., Silveira, H., Abe, C., 2019. IowaNet dataset for deep learning: 1 million samples with 10 land cover classes (Version 1.0). http://doi.org/10.5281/zenodo.3385318.
Maxwell, A.E., Strager, M.P., Warner, T.A., Zegre, N.P., Yuill, C.B., 2014. Comparison of NAIP orthophotography and RapidEye satellite imagery for mapping of mining and mine reclamation. GISci. Remote Sens. 51 (3), 301–320.
Maxwell, A.E., Warner, T.A., Vanderbilt, B.C., Ramezan, C.A., 2017. Land cover classification and feature extraction from national agriculture imagery program (NAIP) orthoimagery: a review. Photogramm. Eng. Remote Sens. 83 (11), 737–747.
Mboga, N., Georganos, S., Grippa, T., Lennert, M., Vanhuysse, S., Wolff, E., 2019. Fully convolutional networks and geographic object-based image analysis for the classification of VHR imagery. Remote Sens. 11 (5), 597.
Ming, D., Ci, T., Cai, H., Li, L., Qiao, C., Du, J., 2012. Semivariogram-based spatial bandwidth selection for remote sensing image segmentation with mean-shift algorithm. IEEE Geosci. Remote Sens. Lett. 9 (5), 813–817.
Mnih, V., Hinton, G.E., 2010, September. Learning to detect roads in high-resolution aerial images. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 210–223.
Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: a review. ISPRS J. Photogramm. Remote Sens. 66 (3), 247–259.
Nagel, P., Yuan, F., 2016. High-resolution land cover and impervious surface classifications in the twin cities metropolitan area with naip imagery. Photogramm. Eng. Remote Sens. 82 (1), 63–71.
Nogueira, K., Penatti, O.A., dos Santos, J.A., 2017. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recogn. 61, 539–556.
Paisitkriangkrai, S., Sherrah, J., Janney, P., Van Den Hengel, A., 2016. Semantic labeling of aerial and satellite imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9 (7), 2868–2881.
Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., 2018. A new deep convolutional neural network for fast hyperspectral image classification. ISPRS J. Photogramm. Remote Sens. 145, 120–147.
Polak, M., Zhang, H., Pi, M., 2009. An evaluation metric for image segmentation of multiple objects. Image Vis. Comput. 27 (8), 1223–1227.
Rezaee, M., Mahdianpari, M., Zhang, Y., Salehi, B., 2018. Deep convolutional neural network for complex wetland classification using optical remote sensing imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (9), 3030–3039.
Ronneberger, O., Fischer, P., Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, Cham., pp. 234–241.
Saito, S., Yamashita, T., Aoki, Y., 2016. Multiple object extraction from aerial imagery with convolutional neural networks. Electron. Imag. 2016 (10), 1–9.
Sharma, A., Liu, X., Yang, X., Shi, D., 2017. A patch-based convolutional neural network for remote sensing image classification. Neural Networks 95, 19–28.
Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint arXiv:1606.02585.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Stehman, S.V., 2009. Sampling designs for accuracy assessment of land cover. Int. J. Remote Sens. 30 (20), 5243–5272.
Stehman, S.V., Foody, G.M., 2019. Key issues in rigorous accuracy assessment of land cover products. Remote Sens. Environ. 231, 111199.
Su, T., Li, H., Zhang, S., Li, Y., 2015. Image segmentation using mean shift for extracting croplands from high-resolution remote sensing imagery. Remote Sens. Lett. 6 (12), 952–961.
Sun, G., Huang, H., Zhang, A., Li, F., Zhao, H., Fu, H., 2019. Fusion of multiscale convolutional neural networks for building extraction in very high-resolution images. Remote Sens. 11 (3), 227.
Sun, Y., Liang, D., Wang, X., Tang, X., 2015. Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873.
Tian, Y., Pei, K., Jana, S., Ray, B., 2018, May. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering. ACM, pp. 303–314.
Vakalopoulou, M., Karantzalos, K., Komodakis, N., Paragios, N., 2015, July. Building detection in very high resolution multispectral data with deep learning features. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 1873–1876.
Wang, L., Dai, Q., Hong, L., Liu, G., 2012. Adaptive regional feature extraction for very high spatial resolution image classification. J. Appl. Remote Sens. 6 (1), 063506.
Wang, L., Dai, Q., Xu, Q., Zhang, Y., 2015. Constructing hierarchical segmentation tree for feature extraction and land cover classification of high resolution MS imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (5), 1946–1961.
Xu, Y., Wu, L., Xie, Z., Chen, Z., 2018. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. Remote Sens. 10 (1), 144.
Yang, J., He, Y., Weng, Q., 2015. An automated method to parameterize segmentation scale by enhancing intrasegment homogeneity and intersegment heterogeneity. IEEE Geosci. Remote Sens. Lett. 12 (6), 1282–1286.
Yang, L., Jin, S., Danielson, P., Homer, C., Gass, L., Bender, S.M., Funk, M., 2018. A new generation of the United States National Land Cover Database: Requirements, research priorities, design, and implementation strategies. ISPRS J. Photogramm. Remote Sens. 146, 108–123.
Yifang, B., Gong, P., Gini, C., 2015. Global land cover mapping using Earth observation satellite data: recent progresses and challenges. ISPRS J. Photogr. Remote Sens. (Print) 103 (1), 1–6.
Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., Atkinson, P.M., 2018. An object-based convolutional neural network (OCNN) for urban land use classification. Remote Sens. Environ. 216, 57–70.
Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Commun. ACM 27 (3), 236–239.
Zhang, X., Xiao, P., Feng, X., Wang, J., Wang, Z., 2014. Hybrid region merging method for segmentation of high-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 98, 19–28.
Zhao, W., Du, S., 2016. Learning multiscale and deep representations for classifying remotely sensed imagery. ISPRS J. Photogramm. Remote Sens. 113, 155–165.
Zhao, W., Du, S., Emery, W.J., 2017. Object-based convolutional neural network for high-resolution imagery classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (7), 3386–3396.
Zhong, L., Hu, L., Zhou, H., 2019. Deep learning based multi-temporal crop classification. Remote Sens. Environ. 221, 430–443.
Zhou, W., Newsam, S., Li, C., Shao, Z., 2018. PatternNet: a benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J. Photogramm. Remote Sens. 145, 197–209.
Zhu, X.X., Tuia, D., Mou, L., Xia, G.S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: a comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. 5 (4), 8–36.
Zhu, Z., Woodcock, C.E., 2014. Continuous change detection and classification of land cover using all available Landsat data. Remote Sens. Environ. 144, 152–171.