ISPRS Journal of Photogrammetry and Remote Sensing 139 (2018) 154–170

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Deep convolutional neural network training enrichment using multi-view object-based analysis of Unmanned Aerial systems imagery for wetlands classification

Tao Liu, Amr Abd-Elrahman *

University of Florida, School of Forest Resources and Conservation, Gainesville, FL 32611, USA
University of Florida, Gulf Coast Research Center, Plant City, FL 33563, USA

ARTICLE INFO

Article history:
Received 21 August 2017
Received in revised form 22 January 2018
Accepted 5 March 2018
Available online 18 March 2018

Keywords:
Deep learning
Convolutional neural network
Object-based classification
Random forest
Support vector machine
small UAS
OBIA

ABSTRACT

Deep convolutional neural network (DCNN) requires massive training datasets to trigger its image classification power, while collecting training samples for remote sensing application is usually an expensive process. When DCNN is simply implemented with traditional object-based image analysis (OBIA) for classification of Unmanned Aerial systems (UAS) orthoimage, its power may be undermined if the number training samples is relatively small. This research aims to develop a novel OBIA classification approach that can take advantage of DCNN by enriching the training dataset automatically using multi-view data. Specifically, this study introduces a **M**ulti-View **O**bject-based classification using **De**ep convolutional neural network (**MODe**) method to process UAS images for land cover classification. **MODe** conducts the classification on multi-view UAS images instead of directly on the orthoimage, and gets the final results via a voting procedure. 10-fold cross validation results show the mean overall classification accuracy increasing substantially from 65.32%, when DCNN was applied on the orthoimage to 82.08% achieved when MODe was implemented. This study also compared the performances of the support vector machine (SVM) and random forest (RF) classifiers with DCNN under traditional OBIA and the proposed multi-view OBIA frameworks. The results indicate that the advantage of DCNN over traditional classifiers in terms of accuracy is more obvious when these classifiers were applied with the proposed multi-view OBIA framework than when these classifiers were applied within the traditional OBIA framework.

1. Introduction

1.1. Background

Deep Convolutional Neural Network (DCNN) is the workhorse behind deep learning algorithm, which has seen its rapid development in the past decade thanks to its great success in computer vision community and the fast advancement of computing facilities (LeCun et al., 2015). Compared to traditional classifiers (e.g., random forest, support vector machine), DCNN does not need extraction and selection of hand-crafted features. Such advantage, together with its success in the computer vision field has motivated researchers in the remote sensing community to investigate its usefulness for remote sensing image analysis (Cheng et al., 2017a, b, 2016; Liu et al., 2018; Lu et al., 2017; Ma et al., 2016; Makantasis et al., 2015; Vetrivel et al., 2017; Yao et al., 2016; Zhang et al., 2016; Zhao and Du, 2016; Zhao et al., 2017a, b; Zhong et al., 2017). The number of parameters of DCNN with moderate depths (e.g., 30–50 layers) can easily run up to millions, posing a critical issue for collecting a large volume of training dataset to trigger the power of DCNN. Solving this problem is important for successful application of DCNN, which motivated several studies (Celikyilmaz et al., 2016; Marcos et al., 2016; Xie et al., 2015; Yu et al., 2017; Zhao and Du, 2016), including ours.

Small Unmanned Aerial System (UAS) is becoming popular for natural resource management in recent years and has been employed in an increasing number of remote sensing applications (Colomina and Molina, 2014; Liu et al., 2018; Lu and He, 2017; Pande-Chhetri et al., 2017). One of the primary approaches for

* Corresponding author at: University of Florida, School of Forest Resources and Conservation, Gainesville, FL 32611, USA and University of Florida, Gulf Coast Research Center, Plant City, FL 33563, USA.
E-mail address: aamr@ufl.edu (A. Abd-Elrahman).

https://doi.org/10.1016/j.isprsjprs.2018.03.006
0924-2716/

UAS image classification is Object-Based Image Analysis (OBIA). Object-based classification can utilize the spectral, geometrical, textural, and contextual features extracted from meaningful objects of a segmented image producing results that tend to show more appealing appearance and higher or equal accuracy compared to pixel-based classification (Cleve et al., 2008; Fu et al., 2017; Gao and Mas, 2008; Ke et al., 2010; Xun and Wang, 2015). In addition, the processing unit in the OBIA is cluster of pixels contained in each object as compared to single pixel in the pixel-based classification, potentially reducing the processing time for classification procedure in the classification phase. These attributes of OBIA make it a preferable approach over pixel-based methods when high spatial resolution images are used.

However, even though OBIA has become a standard paradigm for processing high resolution images due to its advantages compared to pixel-based methods, current applications of DCNN to processing remote sensing images for mapping task still mainly rely on pixel-based method (i.e., sliding a fixed-size window centered at the pixel under consideration) (Chen et al., 2014; Makantasis et al., 2015; Mnih and Hinton, 2010), even though applying DCNN in the OBIA context started to appear in the recent literature as in (Liu et al., 2018; Zhao et al., 2017a). The need to explore the application of DCNN in the OBIA context is highlighted in a recent OBIA application review article (Chen et al., 2018; Ma et al., 2017). The present study considers applying the DCNN for OBIA classification using content-adaptive window as input. Given an object generated from orthoimage segmentation procedure, the window exactly enclosing the object is used to extract the data to input the DCNN for classification. Obviously, window size is always adapted according to the object size, thus producing the content-adaptive window approach tested in this study.

Traditional OBIA relies only on the orthoimage for classification. However, considering the data-greedy feature of DCNN, simply applying the DCNN with content-adaptive window to the orthoimage under the traditional OBIA framework may prevent the DCNN from releasing its full power given the limited number of training samples and the cost associated with collecting training data, as shown in study by Liu et al. (2018). The related work to overcome the training data limitation issue is reviewed in Section 1.2. However, none of the previous work has taken advantage of the special characteristics of the multi-view nature of remote sensing data. The multi-view data here refers to remote sensing images that show the same object on the ground from multiple angles of views in the air. The inclusion of multi-view information has already been proved useful in vegetation classification using both satellite and manned aircraft-acquired aerial imagery (Abuelgasim et al., 1996; Camacho-de Coca et al., 2004; Chopping et al., 2003; Gatebe and King, 2016; Koukal et al., 2014; Liu et al., 2016; Su et al., 2007). However, most of the previous research relied on Bidirectional Reflectance Distribution Function (BRDF) modelling to extract BRDF parameters as features to utilize the multi-view data (Koukal et al., 2014; Longbotham et al., 2012; Su et al., 2007). BRDF only contains the bidirectional reflectance characteristics of ground objects, potentially losing other meaningful information contained in the multi-view images; thus, the efficiency for its classification effect may be limited. To more efficiently utilize the multi-view information for classification, this study proposes to train the DCNN classifier directly on the multi-view data without any feature extraction by e.g., BRDF modelling, or any feature selection procedures. Given an orthoimage object and the trained classifier, the classifier is applied to each of the multi-view objects corresponding to the orthoimage object instead of the orthoimage object itself and a majority voting of the multi-view object classification results is applied to generate the final classification label for the orthoimage object. Multi-view of small UAS can naturally improve the diversity and quantity of the training dataset from limited training samples, due to changes in sensor-object-sun geometry and is expected to trigger the power of DCNN. In this study, we will examine if the multi-view data can improve DCNN classification compared to its application using only the orthoimage in the OBIA context.

The DCNN approach requires fixed-size training frames for processing. Since OBIA objects have irregular geometry and different sizes, windows exactly enclosing these objects need to be resized to be used as input frames within the DCNN framework. The larger the size of the DCNN input frame size the more parameters that need to be adjusted and hence more computational resource are needed for network training. Such impact of object size on the DCNN training process and classification accuracy has not been thoroughly addressed in existing literature, especially in a multi-view image analysis context.

The scientific contributions of our study (referred to as research objectives in Section 3.5) can be summarized as follows:

I. *Develop a novel OBIA image DCNN classification method based on multi-view information, aiming to trigger the power of DCNN by enriching the number of training samples. We name our proposed method as **M**ulti-View **O**bject-based classification using **Dee**p convolutional neural networks (MODe) in the subsequent discussion.*
II. *Compare the DCNN with SVM and RF under traditional OBIA framework and the multi-view OBIA framework to reveal how MODe can be used to let DCNN show better classification performance than traditional classifiers (*e.g.*, RF and SVM).*
III. *Develop an operational version of MODe, aiming to reduce the computation cost significantly without compromising the classification accuracy, considering the standard version of MODe is computationally expensive, requiring the projection of all the object vertices onto UAS images to construct multi-view objects.*
IV. *Investigate how modifying the input layer size of DCNN will impact classification accuracy.*

The rest of this manuscript is organized as follows: Section 1.2 shows related work aimed to solve training data limitation issue to trigger DCNN power; Section 2 introduces the study area and data pre-processing steps; Section 3 explains methods and experiment design; Section 4 presents and discusses the results; and Section 5 discusses the results and outlines research conclusions.

1.2. Related work

The power of deep learning networks can only be triggered by massive datasets, but collecting training data for remote sensing application is often expensive and time consuming. Even though developing novel DCNN structure for better performance is important, we believe it's equally important to develop novel approaches to trigger the power of DCNN with limited training samples from the practical point of view. This study focuses on the latter issue and reviews the related work in this section. The techniques to overcome this obstacle generally fall into the following three categories.

The first one tries to augment the limited labeled samples with various transformation operations such as rotation, translation and scaling (Cheng et al., 2016; Marcos et al., 2016; Yu et al., 2017; Zhao and Du, 2016). For example, Zhao and Du (2016) transformed the original very high resolution images into Laplacian pyramid images as multiscale datasets and their results showed a significant increase in classification accuracy. Cheng et al. (2016) and Yu et al. (2017) directly applied rotation, translation and scaling to augment training dataset before training DCNN. Compared with the expanded training data resulting from

manual operations (i.e., rotation, translation and scaling), multi-view projection from orthoimage to UAS images provides a natural way to augment the training objects. The improved diversity brought about by these naturally expanded dataset not only includes the rotation, translation, and scaling changes through viewing the same object on the ground from different angles, but also contains the spectral changes due to the BRDF effect. These naturally augmented training datasets are rarely used for DCNN training and the effectiveness is unknown. In addition, multi-view information of the same object can result in multiple classifications for the same objects. Combining these multiple classifications to obtain a final label for the object may provide further improvement for the final classification results, which is rarely investigated in the literatures simply applying geometrical transformations (rotation, translation and scaling) to augment DCNN training datasets.

The second type of techniques attempts to utilize the unlabeled samples to help train the deep network. One way to exploit the unlabeled samples is by using them to pre-train the deep learning networks. This pre-training strategy using unlabeled samples accompanied the early development of deep learning around 2006 (Hinton et al., 2006) and is primarily applicable to one of the oldest deep learning networks called Deep Belief Networks (Erhan et al., 2010), even though it has also been adapted to pre-train the popular DCNN (Celikyilmaz et al., 2016) in recent years. Another way of using unlabeled data is to assign labels to the unlabeled samples based on some measurements of similarity between labeled and unlabeled samples. For example, Ma et al. (2016) pre-labeled each unlabeled sample with local and global decisions and added some unlabeled ones with high confidence to expand the training samples. Their results indicated that this was an effective way to apply the deep learning for hyperspectral image. This kind of semi-supervised classification method can be considered as two rounds of consecutive classifications, with the output of the first round used to expand the training dataset for the second round of classification. One issue with this method is that it is difficult to set up a rule to filter the output of the first round of classification to guarantee that the filtered outputs c have a reliable accuracy when used in the second round of classification. If the accuracy of the training datasets is not high, poor classification results most probably will be the outcome of this process.

The third technique deals with limited training samples through transfer learning (Pan and Yang, 2010). Supposing sufficient labeled samples are available for task A (source task), but only limited labeled samples exist for task B (target task), transfer learning is conducted by first training the model with the rich labeled samples from task A, then further training the model using labeled samples from task B and finally applying the model trained this way to classify the unlabelled samples for task B. Therefore, the knowledge learned from task A is transferred to task B; hence this technique is called transfer learning. Several researchers have used transfer learning to handle the scarce training sample issue (Kunze et al., 2017; Oquab et al., 2014; Xie et al., 2015; Yang et al., 2017). Xie et al. (2015) tried to map the poverty (task B) using high resolution images. Due to the limited training samples, they first trained the convolutional neural network to predict the nighttime light with a rich training dataset (task A). Then the model trained with the nighttime light dataset was further trained with the insufficient poverty dataset. This research demonstrated that features learned with transfer learning methods were informative for poverty mapping. However, the effectiveness of transfer learning depends on how the source task is related to the target task. If the relationship is weak, negative transfer may be incurred to decrease the classification performance for target task (Torrey and Shavlik, 2009). In addition to the studies mentioned above, one of the latest studies by Zhao et al., 2017a, b tried to improve the DCNN classification results by utilizing the object context information via conditional random field modelling.

In this study, we aim to crack the training data limitation issue using photogrammetry techniques in an attempt to trigger the power of DCNN with limited labeled training samples. By "trigger", we mean 1) DCNN applied with the proposed method shows a much better classification performance compared to when it was simply applied within the traditional OBIA framework, and 2) when DCNN, RF and SVM were applied with the proposed method and traditional OBIA framework respectively, DCNN will show more obvious advantages over RF and SVM in the former setup than in the latter one.

2. Study area and experiment material

2.1. Study area

The proposed object-based MODe classification approach was tested on a 677 m × 518 m area, that is part of a 31,000-acre ranch, located in Southern Florida, between Lake Okeechobee and the city of Arcadia. The ranch is comprised of diverse tropical forage grass pastures, palmetto wet and dry prairies, pine flatwoods and large interconnecting marsh of native grass wetlands. The land also hosts cabbage palm and live oak hammocks scattering along the lengths of copious creeks, gullies, and wetlands. The study area is infested by Cogon grass (Imperata Cylindrical), as shown in the lower left corner of Fig. 1, scattered across the pasture. In this study, a Cogan grass class is defined due to its harmful effect on the region as an invasive species. Cogon grass is considered one of the top ten worst invasive weeds in the world (Holm et al., 1977). The grass is not palatable as a livestock forage, decreases native plant biodiversity and wildlife habitat quality, increases fire hazard, and lowers the value of real estate. Several agencies, including U.S Army Corps of Engineers (USACE) are involved in routine monitoring and control operations to limit the spread of Cogan grass in Florida. These efforts will greatly benefit from developing an efficient way to classify Cogan grass from UAS imageries. All other classes, except the shadow class, were assigned according to the lestandard of vegetation classification for South Florida natural areas (Rutchey et al., 2006). Our objective is to classify the Cogon grass (species level) and five other community-level classes as well as the shadow class as listed in Table 1.

2.2. UAS image acquisition and preprocessing

The images used in this study were captured by the USACE-Jacksonville District using the NOVA 2.1 small UAS. A flight mission was designed with 83% forward overlap and 50% sidelap was planned and implemented. A Canon EOS REBEL SL1 digital camera is used in this study. The CCD sensor of this camera has 3456*5184 pixels. The images are synchronized with onboard navigation grade GPS receiver to provide image locations. Five ground control points were established (four near the four corners and one close to the center of the study area) and used in the photogrammetric solution. More details on the camera and flight mission parameters are listed in Table 2.

3. Methods and experiment design

3.1. Orthoimage creation and segmentation

The UAS images were pre-processed to correct for the change in sun angle during the acquisition period before the orthoimage is created. Given an original UAS image i with

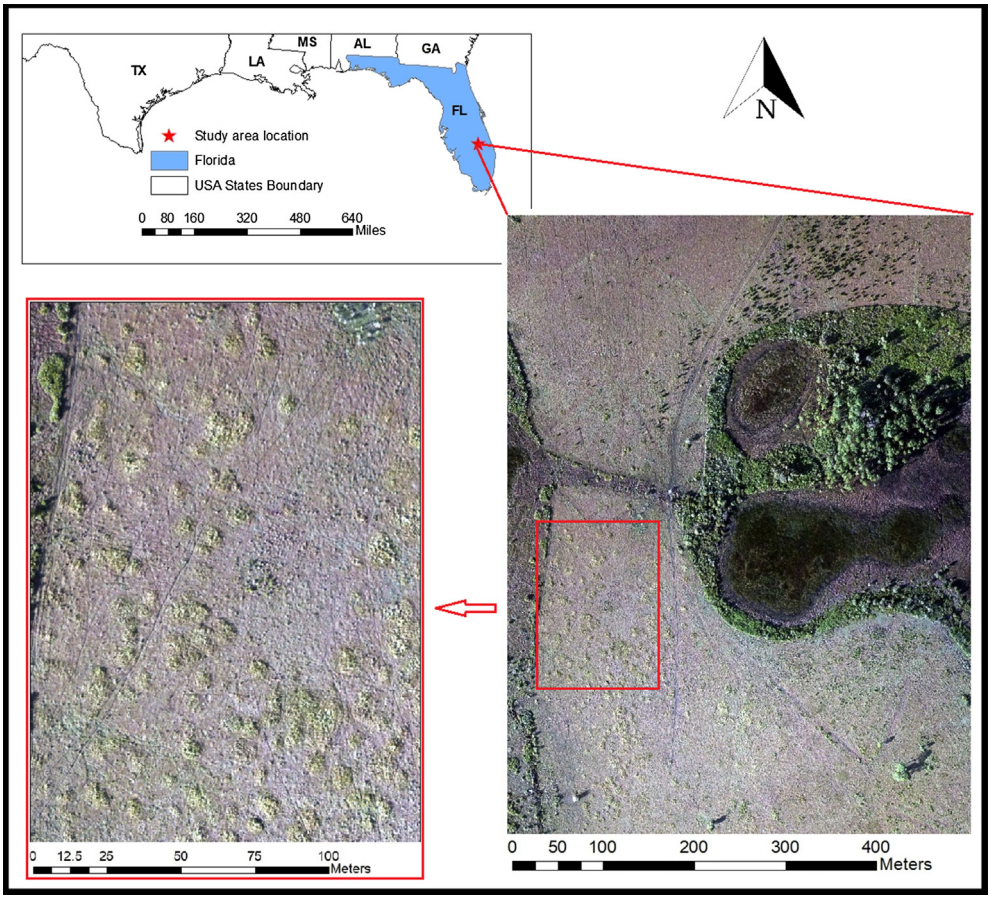


Fig. 1. Study area.

Table 1
Land cover classes in the study area.

Class ID	Class name	Description
CG	Cogon grass	Cogon grass (Imperata cylindrica) is an invasive, non-native grass which occurs in Florida and several other Southeastern US states.
IP	Improved Pasture	A sown pasture that includes introduced pasture species, usually grasses in combination with legumes. These are generally more productive than the local native pastures, have higher protein and metabolizable energy and are typically more digestible. In our case, we also assume it is not infested by Cogon grass.
SUs	Saw Palmetto Shrubland	Saw Palmetto (Serenoa repens) dominant shrubland.
MFB	Broadleaf Emergent Marsh	Broadleaf emergent dominated freshwater marsh. It can be found throughout Florida.
MFG	Graminoid Freshwater Marsh	Graminoid dominated freshwater marsh. It can be found throughout Florida.
FHp	Hardwood Hammock - Pine Forest	A co-dominate mix (40/60 to 60/40) of Slash Pine (Pinus elliottii var. densa) with Laural Oak (Quercus laurifolia), Live Oak (Q. virginiana), and/or Cabbage Palm (Sabal palmetto).
Shadow	Shadow	Shadow of all kinds of objects in the study area.

Table 2
Summary of sensor and flight procedure.

Items	Description
UAS Type	Light UAS with Fixed wing
Sensor Name	Canon EOS REBEL SL1
Sensor Type	CCD
Pixel Dimension	5184 × 3456
Length of focus	20 mm
Sensor Size	22.3 × 14.9 mm
Channels	RGB
Takeoff time	29/10/2015 16:54:51EDT[a]
Landing time	29/10/2015 17:49:33*EDT*[a]
Takeoff Latitude	27.22736549Â°
Takeoff Longitude	−81.51152802Â°
Average Wind Speed	5.1 m/s
Average Altitude	302.7 m
Average Pixel Size	6.5 cm
Forward overlap	83%
Side overlap	50%
FOV[b] across-track	58Â°
FOV[b] along-track	41Â°

a Eastern Daylight Time.
b Field of view in degree.

zenith angle θ_i, the original UAS images was corrected as $ImgCorrected_i = ImgOriginal_i(\cos(\theta_i)/\cos(75\hat{A}^\circ))$ (Koukal and Atzberger, 2012). The operation was conducted on all the UAS images. Once the images are corrected, the Agisoft Photoscan Pro version 1.2.4 software was used to implement the bundle block adjustment on a total of 1397 UAS images of the study area. The software was used to produce and export a 3 band (Red, Green, and Blue) 6cm resolution orthoimage, a Digital Surface Model (DSM), and the camera exterior and interior orientation parameters.

The Trimble's eCognition software (Developer (2012)) was used to segment the orthoimage image. One of the most important

options controlling the segmentation process is the scale parameter. Using small segmentation scale results in smaller and probably more homogeneous objects; however, very small object size may decrease separation between classes and increase the computation cost. It increases the computational burden due to a large number of created objects and may affect the quality of the information extracted from each object (e.g. textual information). On the other hand, having large objects tends to produce mixed classes within the objects.

Selecting optimal scale to enable the best segmentation has been an active research subject for probably couple of decades (Drăguț et al., 2014; Drăguț et al., 2010; Esch et al., 2008; Grybas et al., 2017; Im et al., 2014; Lowe and Guo, 2011; Xun and Wang, 2015). Wang et al. 2004 proposed to use Bhattacharya Distance to find the optimal scale to maximize segments separability. Esch et al. 2008 started from a coarse scale to a series of refined scales to decide whether an object on coarse level should be segmented into sub-objects based on certain criteria aiming at minimizing over- and under-segmentation. Drăguț et al., 2010 measured the local variance of object heterogeneity to select the scale. Lowe and Guo (2011) tried to find the optimum scale based on semivariogram plot. Even though there seems to be a consensus among the OBIA community to the need of developing a standard approach to generate the best segmentation results, the review of 76 papers conducted by Im et al. (2014) indicated that the simplest but most commonly used method in OBIA research is still through visual inspection of segmentation results.

In this study, we manually experimented with multiple segmentation scales as well as other parameters involved in the segmentation process and chose the scale (50), shape (0.20) and compactness (0.50) parameters (eCognition, 2012) that gave visually appealing segmentation results across the majority of the orthoimage. These parameters were selected to avoid the under-segmentation, and at the same time alleviate the over-segmentation as much as possible based on visual inspection. This process resulted in 40,239 objects within the study area and these objects were interpreted and manually labeled by USACE ecologist in charge of this area based on visual inspection of orthoimage. For some locations where class labels could not be determined from the orthoimage, site visits were conducted to identify/verify their land cover types. Based on the reference map prepared by USACE, 2800 orthoimage objects were randomly selected to participate the experiments.

3.2. Traditional and multi-view OBIA

The workflow of traditional object-based image classification, commonly applied to high-resolution orthoimages as implemented in the Trimble's eCognition software (Developer (2012)) can be summarized in 3 main steps: (1) Image segmentation into objects using a predefined set of parameters such as the segmentation scale and shape weight, (2) Extraction of features such as mean spectral band values and the standard deviation of the band values for each object in the segmented image and 3) Train and implement a classifier such as the support vector machine (Scholkopf and Smola, 2001), random forest (Breiman, 2001), or neural network classifiers (Yegnanarayana, 2009). In this study, we denote this traditional OBIA as **Ortho-OBIA**, since orthoimage is its only data source. We use **SVM-Ortho-OBIA** and **RF-Ortho-OBIA** to indicate the classifier type used in the Ortho-OBIA. We are also interested in investigating the performance of simple application of DCNN in the traditional OBIA framework and denote this classification method as **DCNN-Ortho-OBIA (see** Fig. 3a). Unlike the SVM-Ortho-OBIA and Ortho-Ortho-OBIA, which extract features from an orthoimage object as classifier input, DCNN-Ortho-OBIA extracts image patch exactly containing the whole orthoimage object (i.e., content-adaptive window) as the input to DCNN classifier.

Unlike traditional object-based classification that only relies on the orthoimage, we propose to conduct OBIA analysis using reconstructed objects on individual UAS images to take advantage of the multi-view information inherited in each UAS image and we denote this approach as **MV-OBIA** classification. To obtain the multi-view information for a given orthoimage object, the orthoimage object needs to be reconstructed on UAS images containing this object. Fig. 2 illustrates the multi-view objects. In Fig. 2a, an object resulting from the orthoimage segmentation procedure is at the center and is surrounded by the multi-view UAS images with highlighted reconstructed object boundaries. It can be seen from this figure that object boundary on the orthoimage is accurately reconstructed on the UAS images, even though the shape of object varies on the UAS images due to change of view geometry. To differentiate the projected objects on the UAS images from the original object on the orthoimage, projected objects are referred to as '**object instances**' and the object on the orthoimage as '**orthoimage object**', subsequently. Fig. 2b presents the mean pixel values of the red channel within each object on the UAS image and clearly shows pixel value experiencing big changes with the change of sun-object-sensor geometry. Using multi-view information one training sample object from the orthoimage corresponds to 10–14 object instances on the UAS images, depending on the amount of forward and side overlapping of the UAS images. It is our hypothesis that such spectral variation and quantity augmentation may improve the robustness of DCNN classifier trained on this dataset as compared to the classifier trained only on the orthoimage objects.

With the multi-view data introduced, MV-OBIA using traditional classifiers (e.g., SVM and RF) is conducted with the following steps: (1) segment image into objects using a predefined set of parameters such as the segmentation scale and shape weight, (2) extract object instances for all the orthoimage objects, (3) extract features separately for each object instance, (4) given an orthoimage object and assuming a trained classifier having been obtained, classify all the object instances corresponding to the orthoimage object, (5) majority voting is conducted among all the classification results of object instances to find the classification results for the orthoimage object. If SVM is used as a classifier in MV-OBIA, we denote it as **SVM-MV-OBIA.** Like the SVM-MV-OBIA notation, we use **RF-MV-OBIA** to indicate the MV-OBIA is performed using the RF classifier.

Similarly, if MV-OBIA is conducted with DCNN being the classifier, we denote this classification method as **DCNN-MV-OBIA** (see Fig. 3b). However, the workflow of RF-MV-OBIA and SVM-MV-OBIA a little bit different from DCNN-MV-OBIA. For RF-MV-OBIA or SVM-MV-OBIA, features are extracted from each object instance, while for DCNN-MV-OBIA a window exactly enclosing the object instance is formed to extract image patch as input to the DCNN classifier. The window size is changed according to the object instance size; thus, the window used in the DCNN-MV-OBIA is also content-adaptive.

The key to implement MV-OBIA is to accurately reconstruct object instance for a given orthoimage object as illustrated in Fig. 2a. The technique regarding this part is shown in the following Section 3.3.

3.3. Multi-view object instance generation

The objective of this section is to show how to generate object instances on the UAS images corresponding to this orthoimage object, to support MV-OBIA classification. This problem can be boiled down to projecting each of the vertices on the orthoimage object boundary onto UAS images. After vertex projection is done,

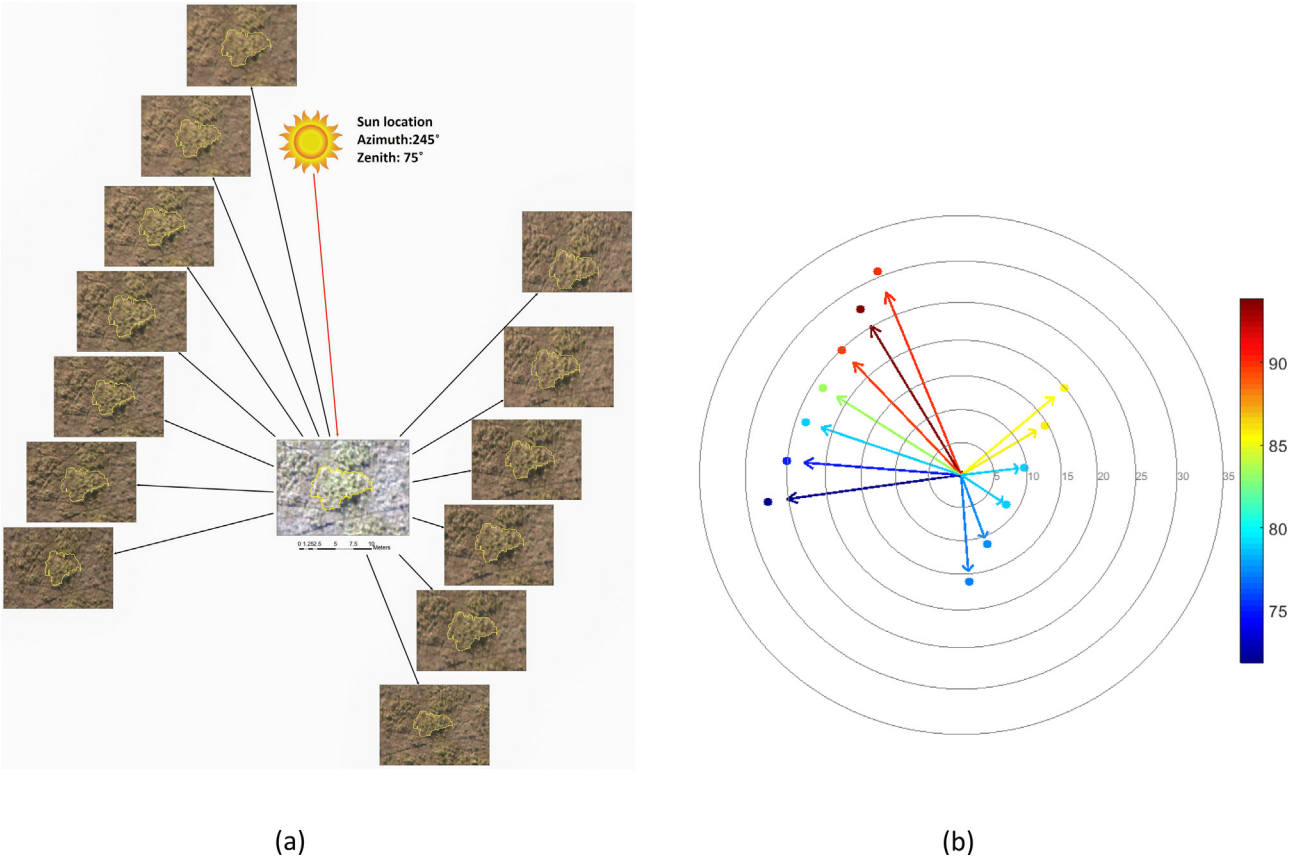


Fig. 2. Multi-view objects on the UAS images corresponding to one object on the orthoimage: (a) multi-view object instances corresponding to one orthoimage object; (b) mean value of the red band digital number (DN) of the object instances on the UAS images.

an object instance can be easily formed by threading together the projected boundary vertices.

Given the real-world coordinates, *X* and *Y*, and *Z* of an object boundary vertex on the orthoimage and the output of the bundle block adjustment results of the UAS images represented by the camera exterior orientation and self-calibration parameters, it is required to find the *x* and *y*coordinates (or row and column numbers) in the UAS image pixel coordinate system, if the boundary point exists on the UAS image. This requires converting XYZ from real-world coordinate system to camera coordinate system using Eq. (1), followed by the conversion from camera coordinate system to sensor coordinate system by Eq. (2) and then from camera sensor system to pixel coordinate system by Eq. (3). As indicated by these equations, our input includes ground point coordinate in World Coordinate System X, Y, Z, camera coordinates in World Coordinate System X_0, Y_0, Z_0, r_{ij}, $thei_{th}$ row and j_{th}column element of the camera rotation matrix R that is extracted from bundle adjustment procedure, camera focal length f, number of rows in UAS image H, UAS image pixel size p, and sensor center coordinate offset x_o, y_o.

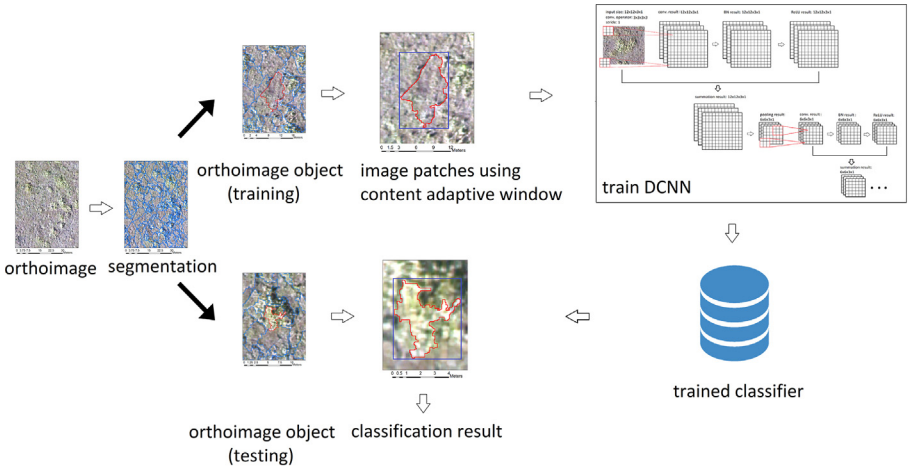
However, due to potential error coming from inaccuracies of the Digital Surface Model (DSM) used to extract Z value, camera parameters (e.g., focal length, pixel size) and camera lens distortion, a simple consecutive application of Eqs. (1)–(3) usually gave larger error. To reduce such error, we developed a two-step optimization approach to reduce the projection error. The step-one is to apply the Generalized Pattern direct Search (GPS) algorithm (Audet and Dennis Jr, 2002) to optimize the camera parameters (e.g., focal length, sensor size). The step-two is to apply random forest algorithm to model the relationship between the error and the point locations causing the error (e.g., distance from the point to UAS image center, Z value of the point and relative location of the point to the image center in terms of row distance and column distance). Average error around 1.6 pixels in the row direction and 1.8 pixels in the column direction were achieved using this method.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = [X - X_0, Y - Y_0, Z - Z_0] \times \begin{bmatrix} r_{11} r_{12} r_{13} \\ r_{21} r_{22} r_{23} \\ r_{31} r_{32} r_{33} \end{bmatrix} \tag{1}$$

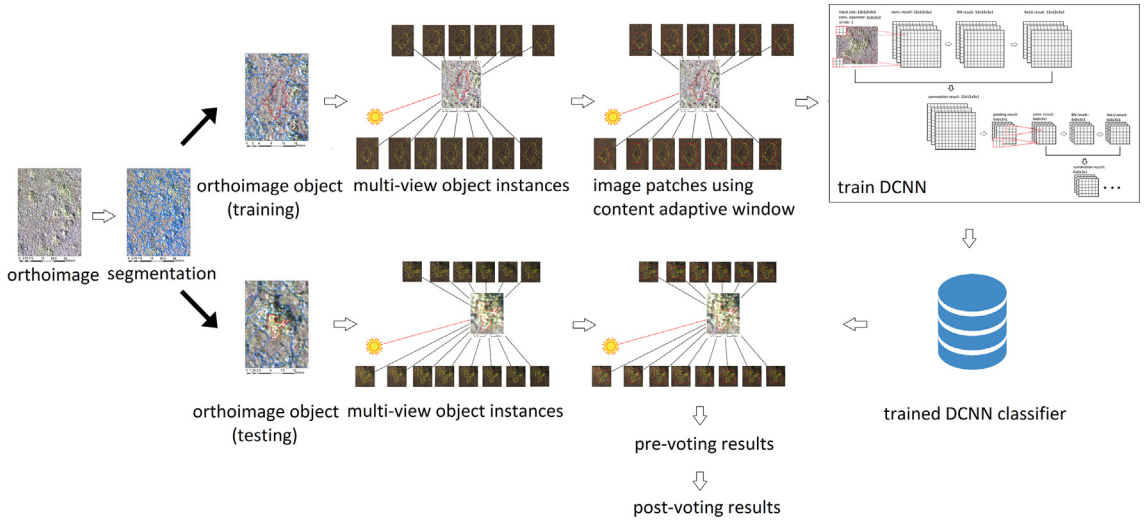
$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} x_o - f\frac{X_c}{Z_c} \\ y_0 - f\frac{Y_c}{Z_c} \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} round\left(\frac{x_s}{p}\right) \\ round\left(H - \frac{y_s}{p}\right) \end{bmatrix} \tag{3}$$

After each vertex was successfully projected onto UAS images, they were threaded together to reconstruct the complete projected boundary of the object. Projecting each object boundary vertex on to multiple UAS images is computationally expensive and may not be practical in a large-scale implementation, knowing that a typical object boundary can have more than one hundred vertices and a small study area can easily generate thousands of objects with millions of vertices (e.g., 40,239 objects generated for our study area consist of 5,313,622 vertices in total). In our study, segmentation results generated from Photoscan package (see Section 3.1) were



(a) Apply DCNN using content adaptive window within the traditional OBIA framework



(b) Apply DCNN using content adaptive window with the proposed OBIA framework

Fig. 3. DCNN implementation in traditional (a) and in the proposed (b) OBIA framework. The content-adaptive windows to extract image patches are denoted with blue rectangles in (a) and red rectangles in (b). Classification methods DCNN-Ortho-OBIA and DCNN-MV-OBIA correspond to (a) and (b) respectively. Both DCNN-Ortho-OBIA and DCNN-MV-OBIA start with an orthoimage and proceed with image segmentation, and both of them use content adaptive window as DCNN input. While DCNN-Ortho-OBIA needs only orthoimage, DCNN-MV-OBIA requires multi-view object instances for both training and classification. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

imported into ArcGIS to extract the vertices for each object and XYZ world coordinates of each vertex, which were then exported from ArcGIS to Matlab to generate the multi-view object instances.

To reduce the computational burden, we further propose to apply the MV-OBIA classification using a square window positioned at the geometrical center (centroid) of each object instance and we denote this classification approach as WMV-OBIA. Depending on the classifier used in this classification, classifier type can be affixed in the beginning of WMV-OBIA to denote the specific classification method. For example, if DCNN is used in WMV-OBIA, **DCNN-WMV-OBIA** is used to refer to this classification method. The only difference between DCNN-WMV-OBIA and DCNN-MV-OBIA is that DCNN-MV-OBIA uses content-adaptive window which always exactly encloses the object instance, while DCNN-WMV-OBIA used fixed-sized window formed by positioning a square window of a certain size at the geometrical center of the object instance. This way, only the object centroid points need to be projected on the UAS images instead of all object boundary vertices for DCNN-WMV-OBIA. Since each object only has one centroid, this method reduces the computations substantially. However, this method raises the question of how well these windows represent their corresponding objects and adds the window size as a parameter that needs to be specified by the user.

To compare the WMV-OBIA with the classification, we tested a window-based version of the Ortho-OBIA and denoted it as W-Ortho-OBIA. The W-Ortho-OBIA uses the data extracted from a square window positioned at the centroid of orthoimage objects, instead of using the complete orthoimage object. Similar to WMV-OBIA, classifier name can be affixed in the beginning of W-Ortho-OBIA to denote the specific classification. For example, **DCNN-W-Ortho-OBIA** refers to the W-Ortho-OBIA classification using DCNN.

3.4. Deep convolutional neural network

The building block of the DCNN used in this study is illustrated in Fig. 4, including the convolutional operation, batch normalization (Ioffe and Szegedy, 2015), activation function Rectified Linear

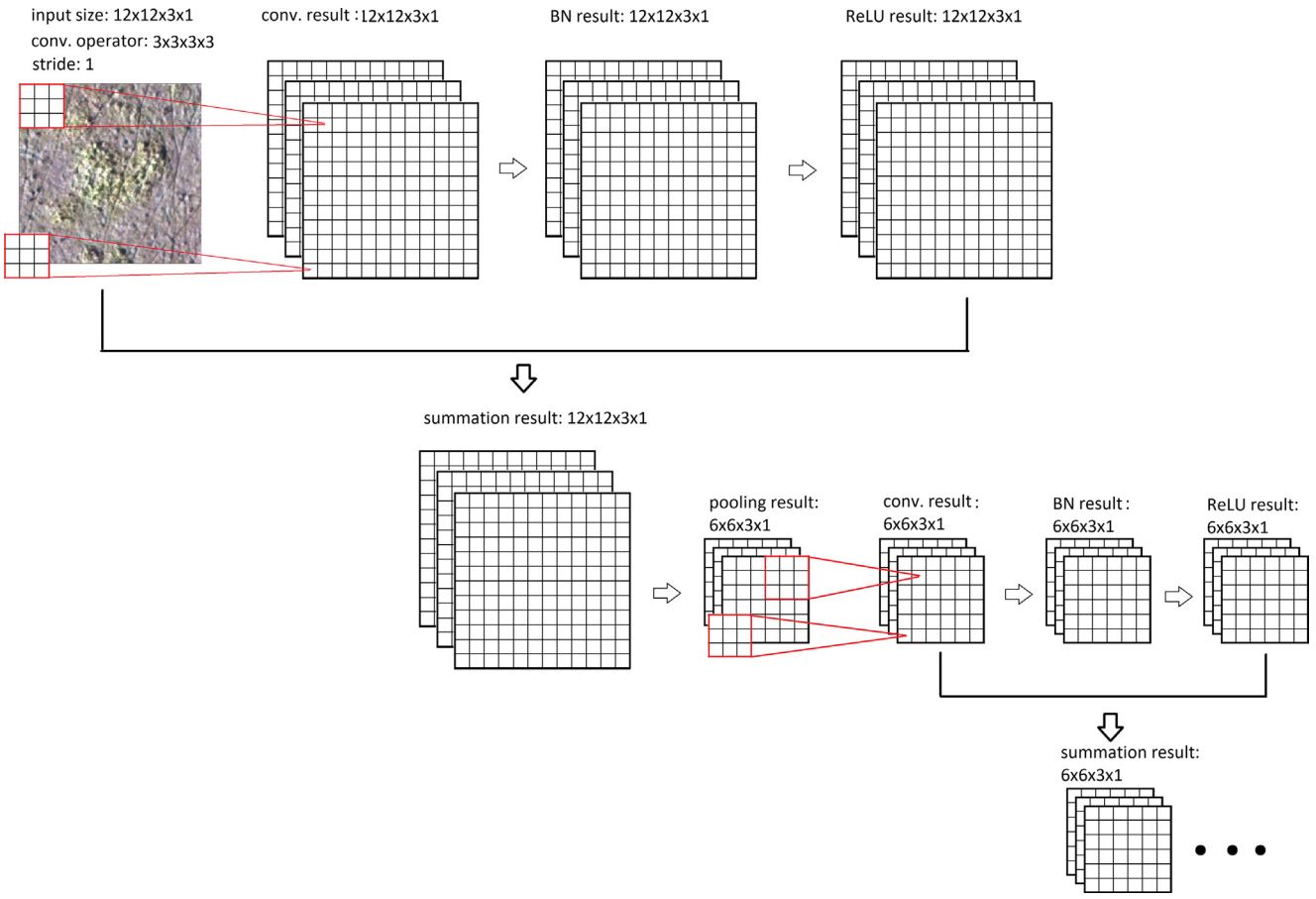


Fig. 4. Simplified illustration of the DCNN building block.

Unit (ReLU) (Nair and Hinton, 2010), summation (He et al., 2016) and max pooling. The summation operation was based on the deep residual net (ResNet) introduced by He et al. (2016). In deep residual nets, instead of directly learning the underlying mapping $\mathcal{H}(x) : x \rightarrow label(x)$, learning is performed to obtain the residual mapping $\mathcal{F}(x) := \mathcal{H}(x) - x$. The desired underlying mapping is then obtained by $\mathcal{F}(x) + x$, which forms the building block of deep residual nets. Fig. 4 illustrates the DCNN structure with a simplified version of building block and can be represented by Eq. (4).

$$x_2 = h(\mathcal{F}(x_1) + x_1) \tag{4}$$

$x_1$in Eq. (4) can be any layer in the DCNN, but for illustration purpose it is assumed to be the input image having size $12 \times 12 \times 3 \times 1$. $\mathcal{F}(x_1)$ in this case includes convolutional operation, batch normalization and ReLU activation function. Summation operation requires output of $\mathcal{F}(x_1)$ having the same size as x_1, which is the reason stride 1 and padding was used for convolutional operation to preserve the dimension of x_1. For illustration purpose, $h(x)$ in this case only includes pooling operation; however, for the real DCNN, $h(x)$ may include other operations shown in $\mathcal{F}(x_1)$ as well. $h(x)$ outputs x_2 , which is then used as the starting point for the next round of block building for $\mathcal{F}(x_2)$. Like$_{x1}$, dimension of x_2 also needs to be preserved in order to do summation operation with the output of $\mathcal{F}(x_2)$. However, it should be noted that if the output of $\mathcal{F}(x_2)$ does not have the same dimension of x_2, summation can still be made, as long as a certain function $l(x_2)$ exists to make its output having the same dimension as $\mathcal{F}(x_2)$. In this case, $\mathcal{F}(x_2)$will be added to the output of $l(x_2)$, instead of x_2. Examples of operations in $l(x_2)$ may include convolutional operation, ReLU activation, etc.

A DCNN based on this framework won the 1st place on the ILSVRC 2015 competition and was also used by DeepMind, one of leading AI companies in industry, in their latest pioneering work to develop an artificial intelligence system to enable the machine master the game of GO without the human knowledge (Silver et al., 2017). In this study, this building block was used to build a network of 50 layers. Only one fully connected layer is attached at the end of the network, allowing the convenience for experimenting with input layer size in DCNN, since changing the input layer size only impacts the number of parameters in the last fully connected layer. We modified the DCNN to allow it to have different input layer size ($15 \times 15, 33 \times 33, 65 \times 65, 124 \times 124, 224 \times 224$ and 325×325) to investigate how these modifications impact the classification performance for DCNN-MV-OBIA. Input layer size is appended to DCNN to denote which version of DCNN is used in the DCNN-MV-OBIA. For example, **DCNN15-MV-OBIA** indicates the input layer of DCNN used in DCNN-MV-OBIA is 15×15. If input layer size is omitted in the DCNN-MV-OBIA notation, it means input layer size 224×224 is used by default. DCNN in this study was implemented using the MatConNet software (Vedaldi and Lenc, 2015).

The DCNN cost is cross-entropy calculated with the output from softmax layer and ground truth label of training samples, as presented by Eq. (5).

$$C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(1_{\{1\}}\left(y_i^j\right)\ln\left(a_i^j\right)\right) \tag{5}$$

where $1_A(x) = \begin{cases} 1 & if\ x \in A \\ 0 & if\ x \notin A \end{cases}$, $a_i^j = \frac{e^{z_i^j}}{\sum_{k=1}^{m} e^{z_i^k}}$, m is the total number of classes, equal to 7 for our study, a_i^j is the softmax output for

sample i, class j, $y_i^j \in$ (Sheikh, 2002) indicating whether the ground truth class ID for sample i is j(1 means true and 0 means false).

The training of DCNN is performed through gradient descent, as shown in Eq. (6).

$$w_{updated} = w_{current} - \lambda \frac{\partial C}{\partial w} \tag{6}$$

where $w_{updated}$ is the updated parameter value, $w_{current}$ is the current value, λ is the learning rate, and $\frac{\partial C}{\partial w}$ is the gradient of w (i.e., derivative of parameter w) when cost value is C.

Instead of calculating the true gradient, DCNN was trained by approximating the true gradient by randomly shuffling the total training and calculating the gradient using subset of the randomly shuffled samples (called mini-batch), i.e., stochastics gradient descent (SGD) (Bottou, 2010).

The parameter derivatives are obtained by alternatively conducting forward propagation (Eq. (7) and backward propagation (Eqs. (8) and (9).

$$y^l = h(y^{l-1}) \tag{7}$$

$$\frac{\partial C}{\partial y^{l-1}} = \sum_{k=1}^{n} \frac{\partial C}{\partial y_k^l} \frac{\partial y_k^l}{\partial y^{l-1}} \tag{8}$$

$$\frac{\partial C}{\partial w^l} = \sum_{k=1}^{n} \frac{\partial C}{\partial y_k^l} \frac{\partial y_k^l}{\partial w^l} \tag{9}$$

In Eq. (7), y^l and y^{l-1} represent variable values in layer l and $l-1$ respectively, connected with function $h(x):=y^{l-1} \rightarrow y^l$. The function can be convolutional operation, ReLU activation, max pooling, batch normalization and sum operation, depending on the layer type used in the DCNN structure.

Eqs. (8) and (9) are general form of implementations of chain rule for training DCNN. Eq. (8) needs to be implemented first backward layer by layer to calculate the variable derivatives. Then the variable derivatives for a given layer are used to calculate the parameter derivatives for that layer using Eq. (9). If there are no parameters involved in mapping function $h(x) : y^{l-1} \rightarrow y^l$, only Eq. (8) is implemented to obtain the variable derivatives. For the DCNN used in this study, except max pooling, ReLU activation and summation functions, all the other functions have parameters to be adjusted, including convolutional operation, batch normalization and fully connected layer operation. Different functions have their specific forms of functions for forward and backward propagation.

3.5. SVM and RF

SVM and RF were selected as representatives of conventional classifiers to compare with DCNN due to their extensive adoption and reliable performance for various remote sensing applications such as individual tree crown delineation (Liu et al., 2015), vegetation classification (Li et al., 2013; Lu et al., 2014), forest aboveground biomass estimation (Ajaz Ahmed et al., 2017; Gleason and Im, 2012; Li et al., 2014), convective cloud detection (Lee et al., 2017), drought forecasting and assessment (Park et al., 2016; Rhee and Im, 2017), soil moisture monitoring (Im et al., 2016), etc. In this study, mean, standard deviation, maximum and minimum values for red, green and blue bands were extracted for classifications using RF and SVM. Gray-Level Co-Occurrence Matrix (GLCM) features were excluded from the features list for classification after it was tested and found unuseful to improve the classification accuracy. Geometric features including object boundary, object area, shape index and boundary index as defined in eCognition (2012) were neither considered for classification in our study according to our preliminary test results and previous studies (Yu et al., 2006) showing they were not helpful for OBIA classification of natural land cover scenes.

The RF and SVM parameters were adjusted to make sure their performance as good as possible for our dataset. For example, different numbers of RF trees were tested (50 to 150 trees at 10 tree intervals). Classification accuracy did not change much as the number of trees changed within this range. Three types of kernels (Gaussian, linear to polynomial kernels) for SVM were tested in our preliminary experiments with little impact on the resulting classification accuracy. We adopted the one-versus-one option of the SVM classifier instead of the one-versus-all strategy based on previous studies (Hsu and Lin, 2002). Seven class types in our study resulted in 21 binary classifiers for multi-class SVM classification with the one-versus-one scheme and final classification results in were obtained via majority voting among the 21 binary classifiers. Finally, RF with 50 trees and SVM with Gaussian kernel was implemented in Matlab to generate the classification results presented in this study.

3.6. Experiment design

This section is to show the experiment design aimed to achieve all the study objectives stated in the introduction section. Fig. 5 summarizes the experiments conducted using all the methods explained throughout previous sections. After UAS mission was completed to obtain the raw UAS images (see Section 2.2), all the original UAS images were pre-processed to remove the sun location change impact on spectral values (see Section 3.1). Based on the corrected UAS images, bundle adjustment was performed using the Photoscan package to obtain the orthoimage, DSM and camera rotation matrices (see Section 3.1). Orthoimage was segmented to generate orthoimage objects (see Section 3.1).

Using the orthoimage objects alone, three types of classification were performed, including RF-Ortho-OBIA, SVM-Ortho-OBIA and DCNN-Ortho-OBIA (see Section 3.2). Also using the orthoimage objects, a fixed-size square window positioned at each object centroid was used for the DCNN classifier, producing the DCNN-W-Ortho-OBIA results. Eight window sizes ranging from 21 × 21, 41 × 41 to 161 × 161 were used for the DCNN-W-Ortho-OBIA classification (see Section 3.3).

The vertices and centroids of the orthoimage objects were projected onto the UAS images (see dashed rectangle in Fig. 5) to extract the multi-view information for MV-OBIA and WMV-OBIA classification experiments, respectively using the projection method explained in Section 3.3. After object instances were projected onto the UAS images, three types of classification methods including RF-MV-OBIA, SVM-MV-OBIA and DCNN-MV-OBIA (see Section 3.2) were performed. For the DCNN-MV-OBIA classification, 6 experiments were performed separately with each of them using different DCNN with 6 different input layer sizes 15, 33,.. to 325 (see Section 3.4). The orthoimage object centroid projected onto the UAS images was used to perform the DCNN-WMV-OBIA classification experiments. Similar to DCNN-W-Ortho-OBIA, 8 window sizes ranging from 21 × 21, 41 × 41, 61 × to 161 × 161 were used (see Section 3.3).

Table 3 summarizes the experiments conducted in this research. A total of eight groups of experiments were performed, indicated by letters *a, b, c,…,h* at the bottom of Fig. 5. Group *g* (i.e, DCNN-MV-OBA) is developed to achieve study objective I: "develop the MODe classification approach, a DCNN classification based on multi-view information of UAS images". Comparing group *a, b, c, e, f* and *g* is intended to fulfill the study objective II: "compare the performance of the developed MODe with the SVM and RF classifier results and also simple application of DCNN for

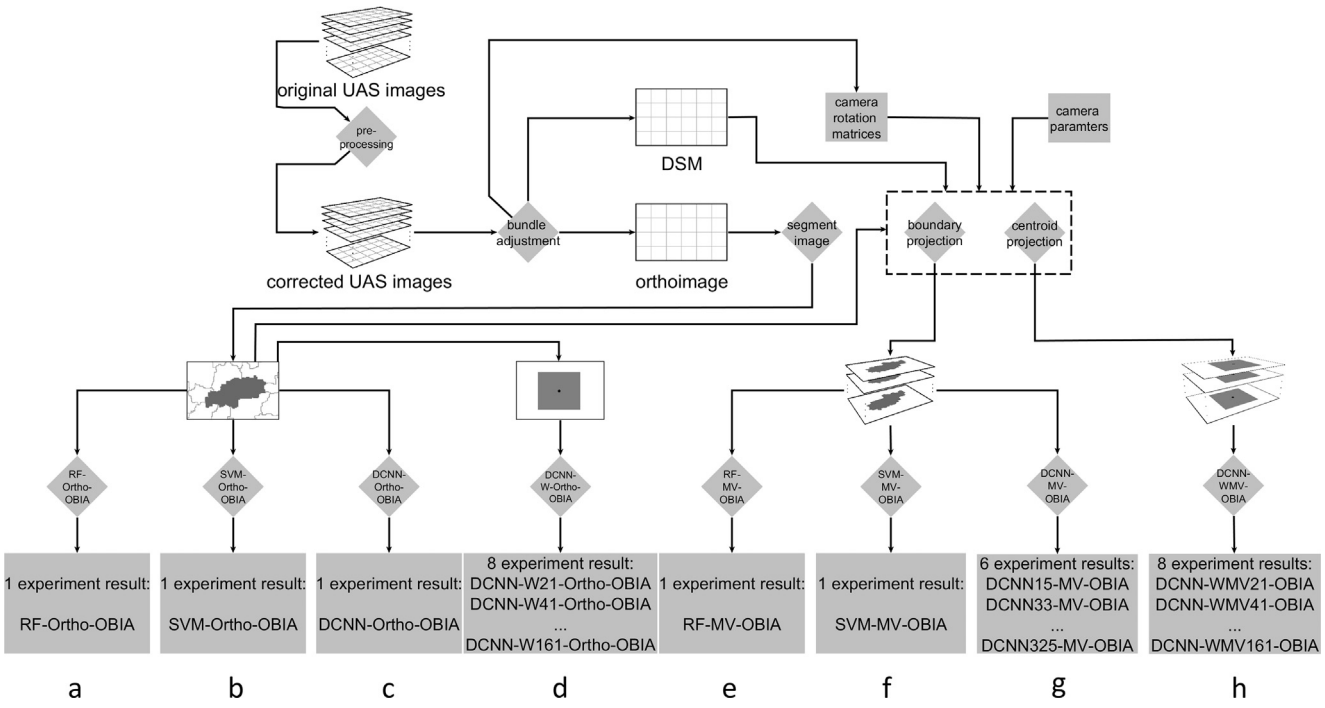


Fig. 5. Experiment flowchart.

Table 3
List of classification approaches and experiment result names.

Group ID	Name	Number of experiments	Notation of experiment results	Comments
a	RF-Ortho-OBIA	1	RF-Ortho-OBIA	Traditional OBIA classification based on orthoimage object with random forest as classifier. Given an orthoimage object, features are extracted from it as RF input.
b	SVM-Ortho-OBIA	1	SVM-Ortho-OBIA	Traditional OBIA classification based on orthoimage objects using the SVM classifier. Given an orthoimage object, features are extracted as SVM input.
c	DCNN-Ortho-OBIA	1	DCNN-Ortho-OBIA	OBIA classification based on orthoimage object with DCNN as classifier. Given an orthoimage object, rectangle exactly covering it is used as patch for DCNN input
d	DCNN-W-Ortho-OBIA	8	DCNN-W21-Ortho-OBIA, DCNN-W41-Ortho-OBIA, DCNN-W61-Ortho-OBIA, ..., DCNN-W161-Ortho-OBIA	OBIA classification based on orthoimage square window with DCNN as classifier. Given a reconstructed object, square window positioned on its centroid is used as patch for DCNN input. DCNN-W21-Ortho-OBIA means using winsow of size 21.
e	RF-MV-OBIA	1	RF-MV-OBIA	OBIA classification based on multi-view data extracted from reconstructed objects with random forest as classifier. Given an orthoimage object, features are extracted from this object as RF input.
f	SVM-MV-OBIA	1	SVM-MV-OBIA	OBIA classification based on multi-view data extracted from object instances with support vector machine as classifier. Given an orthoimage object, features are extracted from this object as SVM input.
g	DCNN-MV-OBIA	6	DCNN15-MV-OBIA, DCNN33-MV-OBIA, DCNN65-MV-OBIA, ..., DCNN325-MV-OBIA	OBIA classification based on multi-view data extracted from reconstructed objects with DCNN as classifier. DCNN15-MV-OBIA means the input layer requires size 15×15.
h	DCNN-WMV-OBIA	8	DCNN-WMV21-OBIA, DCNN-WMV41-OBIA, DCNN-WMV61-OBIA, ..., DCNN-WMV161-OBIA	OBIA classification based on multi-view data from square window on UAS images with DCNN as classifier. DCNN-WMV21-OBIA means using window size of 21.

OBIA classification". Comparing group *d* and *h* is to accomplish study objective III: "provide the guideline for operational MODe to implement MODe with substantially reduced the computation cost without compromising classification accuracy". Comparing the classification results within group *g* aims to realize study objective IV: "investigate how modification of DCNN (i.e., changing the input layer size) may impact the classification performance".

3.7. Model training and accuracy assessment

Four hundred objects were randomly created for each of the seven classes listed in Table 1, resulting in a total of 2800 (7% of 40,239 objects in the study area) randomly selected objects for all the classes. Using these samples, a 10-fold cross validation approach was conducted to experiment with both the traditional

and the proposed methods. For each fold, about 30,000 and 3500 multi-view object instances were automatically extracted for the 2520 and 280 training and validation orthoimage objects respectively using the method explained in Section 3.3.

Let $D_{train_{o}rtho} = \{(I_i, Y_i)|i \in N^+, i = 1, 2, 3, \cdots 2520\}$represent the set of orthoimage objects with (I_i, Y_i) denoting the i_{th} pair of orthoimage training sample including object $I_i$and its associated ground truth label Y_i. Let $D_{train_{m}ulti} = \{(I_{i_j}, Y_i)|i \in N^+, j \in N^+, m_i \in N^+, i = 1, 2, 3, \cdots 2520, j = 1, 2, 3, \cdots m_i, m_i \leq 14\}$ represent the set of multi-view object instances corresponding the 2520 orthoimage training objects with (I_{i_j}, Y_i) denoting one pair of multi-view training sample including the multi-view object instance I_{i_j}(i.e., the j_{th} multi-view object instance corresponding to i_{th} orthoimage object), and its associated ground truth label Y_i. It should be noted that regardless of value for j, all the multi-view object I_{i_j} share the same ground truth label Y_i. m_i is the number of multi-view objects for the i_{th} orthoimage object. We put $m_i \leq 14$ constraint to indicate the number of multi-view object instances for any orthoimage object is<14 in our dataset. $D_{train_{o}rtho}$ has 2520 elements and $D_{train_{m}ulti}$ has 30,807 elements. For all the classifications that don't consider multi-view data (i.e., RF-Ortho-OBIA, SVM-Ortho-OBIA, DCNN-Ortho-OBIA, DCNN-W-Ortho-OBIA), training was conducted using dataset $D_{train_{o}rtho}$, while for all the multi-view approach classifications (i.e., RF-MV-OBIA, SVM-MV-OBIA, DCNN-MV-PBIA, and DCNN-WMV-OBIA), training was conducted using dataset $D_{train_{m}ulti}$.

Similarly, $D_{test_{o}rtho} = \{(I_i, Y_i)|i \in N^+, i = 1, 2, 3, \cdots 280\}$ represents the set of testing orthoimage objects with (I_i, Y_i) denoting the i_{th} pair of testing sample including orthoimage training object I_i and its ground truth label Y_i. $D_{test_{m}ulti} = \{(I_{i_j}, Y_i)|i \in N^+, j \in N^+, m_i \in N^+, i = 1, 2, 3, \cdots 280, j = 1, 2, 3, \cdots m_i, m_i \leq 14\}$ represent the set of multi-view object instances corresponding to the 280 orthoimage testing objects with (I_{i_j}, Y_i) denoting one pair of testing sample including the multi-view object I_{i_j} , (i.e., the j_{th} multi-view object instance corresponding to the i_{th} orthoimage object), and its corresponding ground truth label Y_i. m_i represents the number of multi-view object for the i_{th} orthoimage object. $D_{test_{o}rtho}$ include 280 elements, while $D_{test_{m}ulti}$ include 3447 elements.

For all the classifications that don't consider multi-view data (i.e., RF-Ortho-OBIA, SVM-Ortho-OBIA, DCNN-Ortho-OBIA, DCNN-W-Ortho-OBIA), the trained classifiers were applied to the $D_{test_{o}rtho}$ and evaluated against their ground truth contained in $D_{test_{o}rtho}$.

In contrast, for all the multi-view approach classifications (i.e., RF-MV-OBIA, SVM-MV-OBIA, DCNN-MV-PBIA, and DCNN-WMV-OBIA), the trained classifier was applied to multi-view object instances contained in $D_{test_{m}ulti}$. This set of classification results is referred to as **pre-voting** classification results. Given one of the 280 orthoimage objects, its classification result was obtained by majority voting of its object instances classification results. After applying the majority voting process to the pre-voting result, we obtain the **post-voting** classification results, which was evaluated against ground truth contained in $D_{test_{o}rtho}$.

It should be noted that our final objective is to improve the classification performance evaluated against the ground truth dataset in $D_{test_{o}rtho}$ regardless of whether the classifier is trained with dataset $D_{train_{o}rtho}$ or $D_{train_{m}ulti}$. However, evaluating the classifiers trained with $D_{train_{m}ulti}$ using the $D_{test_{m}ulti}$ dataset will cast some light on how differently these classifiers trained with multi-view object instances resolve the multi-view information. Therefore, for the classifiers trained with $D_{train_{m}ulti}$, we not only evaluated their post-voting classification results against ground truth contained in $D_{test_{o}rtho}$ to study the effectiveness of the proposed image classification method MODe, but also evaluated their pre-voting classification results against the ground truth contained in $D_{test_{m}ulti}$ to study the response of different classifiers to the expanded datasets. For objective I and II, we run 10-fold cross validation to test the effectiveness of the proposed method. For objective III and IV, we used the same set of training and testing datasets (90% for training and 10% for testing) without cross validation, since the primary concerns of these two objectives are to investigate the effect of the change in image patch size on WMV-OBIA performance and DCNN input layer size on MV-OBIA performance.

4. Results and discussion

4.1. Traditional and multi-view object-based classification results

This section is to provide experimental results related to study objective I and II, showing whether the proposed DCNN-MV-OBIA is an effective image classification approach, and highlighting the improved advantages over the SVM and RF traditional classifiers due to the used of the proposed method. Fig. 6 shows the box plot of overall accuracies for 10-fold cross-validation results obtained using the RF, SVM and DCNN classifiers applied on the orthoimage objects and the multi-view object instances, including the SVM-Ortho-OBIA, SVM-MV-OBIA, RF-Ortho-OBIA, RF-MV-OBIA, DCNN-Ortho-OBIA, and DCNN-MV-OBIA classifications. Regarding the MV-OBIA classifications (i.e., SVM-MV-OBIA, RF-MV-OBIA and DCNN-MV-OBIA), both pre- and post-voting results are presented in Fig. 6. To quantitatively evaluate the results, we also provide mean values and pairwise t testing results in Table 4.

The first prominent pattern shown in Fig. 6 and Table 4 is the significantly improved classification accuracy of the SVM, RF and DCNN classification conducted on multi-view object instances compared to the classification based on orthoimage objects (78.54% versus 65.26% for SVM, 77.21% versus 63.58% for RF and 82.08% versus 65.32% for DCNN for mean overall accuracy in Table 4) regardless of the used classifier. This demonstrates the superiority of using the multi-view approach over the widely used traditional OBIA classification relying on orthoimage data.

Fig. 6 also shows that the DCNN classifier did not obviously improve the accuracy compared to the traditional classifiers (63.58% for RF, 65.26% for SVM versus 65.32% for DCNN for mean

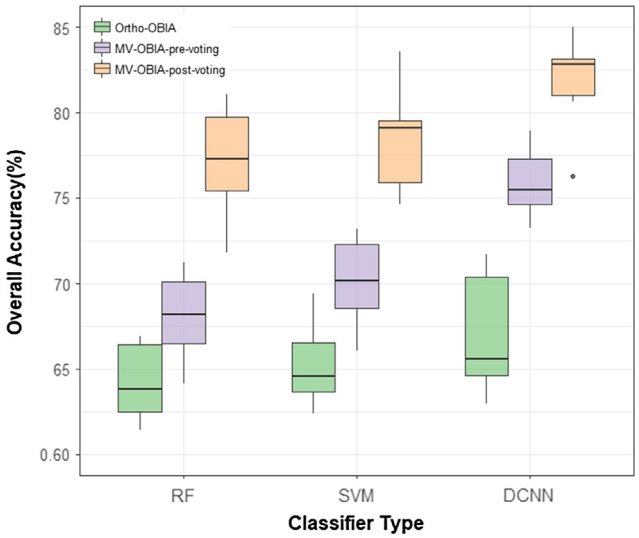


Fig. 6. Overall accuracy of the SVM, RF and DCNN classifiers applied on the orthoimage objects (SVM-Ortho-OBIA, RF-Ortho-OBIA and DCNN-Orth-OBIA) and their multi-view object instances (SVM-MV-OBIA, RF-MV-OBIA and DCNN-MV-OBIA).

Table 4
10-fold cross validation results.

	Ortho-OBIA			MV-OBIA-pre-voting			MV-OBIA-post-voting		
Classifiers	RF	SVM	DCNN	RF	SVM	DCNN	RF	SVM	DCNN
Fold #1	66.91	69.42	64.62	71.22	73.21	76.01	81.07	83.57	81.95
Fold #2	62.50	63.57	50.00	67.84	69.59	76.48	75.71	74.64	82.86
Fold #3	59.86	63.44	62.95	69.65	72.46	73.27	80.00	81.79	76.26
Fold #4	63.21	66.79	63.93	68.51	70.78	77.56	76.43	79.29	84.29
Fold #5	63.80	64.52	71.69	66.38	69.04	74.49	78.93	79.29	80.65
Fold #6	66.43	65.71	65.59	70.23	71.69	77.84	78.21	79.64	83.15
Fold #7	63.80	62.37	68.10	66.76	68.38	74.90	75.36	78.57	82.80
Fold #8	61.43	64.64	71.07	71.18	72.58	74.54	80.00	78.93	80.71
Fold #9	61.43	63.93	64.88	65.27	66.55	74.99	74.64	75.00	83.15
Fold #10	66.43	68.21	70.36	64.15	66.06	78.95	71.79	74.64	85.00
Mean	63.58	65.26	65.32	68.12	70.03	75.90	77.21	78.54	82.08
Pairwise *t* test	DCNN vs RF: p = 0.396 DCNN vs SVM: p = 0.977			DCNN vs RF: p < 0.01** DCNN vs SVM: p < 0.01**			DCNN vs RF: p = 0.013* DCNN vs SVM: p = 0.045*		

* Means there is significant difference on significance level of 0.05.
** Means there is significant difference on significance level of 0.01.

overall accuracy in Table 4) under the traditional OBIA framework. Pairwise *t* test indicates there is no significant difference between DCNN and RF or DCNN and SVM under the traditional classification framework.

On the other hand, if the DCNN classifier was applied with multi-view information, an overall accuracy of 82.08% was obtained, which is higher than the 78.58% and 77.21% accuracy achieved by the SVM and RF classifiers respectively. Pairwise *t* test indicates the differences between DCNN and RF or DCNN and SVM are significant. These results provide evidence to support our previous hypothesis that the augmented quantity and increased spectral variation of the training samples automatically generated by multi-view projection (see Section 3.3) can make it more likely to let DCNN show more obvious advantages over traditional classifiers.

Another pattern worth noticing in Fig. 6 is the difference between the pre- and post-voting results of the multi-view classification. Recalling that the pre-voting classification was trained with about 30,000 reconstructed UAS object instances, and the Ortho-OBIA classifier was trained with 2520 orthoimage training samples, where the 30,000 training samples are the multi-view object instances corresponding to the 2520 orthoimage objects selected as training samples for each fold of experiment. Similarly, the evaluation of the pre-voting results was based on roughly 3500 accuracy assessment object instances, while evaluating the accuracy of the Ortho-OBIA classifier was based on 280 orthoimage accuracy assessment objects. Fig. 6 indicates that the accuracy difference of the RF-Ortho-OBIA and pre-voting RF-MV-OBIA is not big (63.58% for RF-Ortho-OBIA versus 68.12% for pre-voting of RF-MV-OBIA respectively for mean overall accuracy in Table 4), implying that the 13.63% improvement from the RF-Ortho-OBIA to the post-voting of RF-MV-OBIA (63.58% for RF-Ortho-OBIA versus 77.21% for post-voting of RF-MV-OBIA for mean overall accuracy in Table 4) mainly came from the effect of the voting procedure, rather than the expanded training data set, as is similar to SVM.

In contrast, the DCNN classifier made better use of the multi-view data. This is evidenced by the 75.90% overall accuracy for pre-voting DCNN-MV-OBIA classification versus the 65.32% for the DCNN-Ortho-OBIA classification as shown in Table 4. Such an improvement (75.90% versus 65.32%) is in contrast with RF (63.58% versus 68.12%) and SVM (65.26% versus 70.03%). DCNN obtained significant improvement compared to RF or SVM for the pre-voting results, with p value for pairwise *t* test both smaller than 0.01, providing further evidence to show the effectiveness of multi-view data to trigger the power of deep convolutional learning. The voting procedure still played an important role in the case of DCNN-MV-OBIA, leading to 6.18% increase (75.90% versus 82.08% for mean overall accuracy in Table 4) in the mean overall accuracy after the voting procedure was applied.

However, it should also be mentioned that while DCNN provided superior performance in terms of classification accuracy as compared to traditional classifiers when massive training data become available or MODe is used, it comes at the cost of

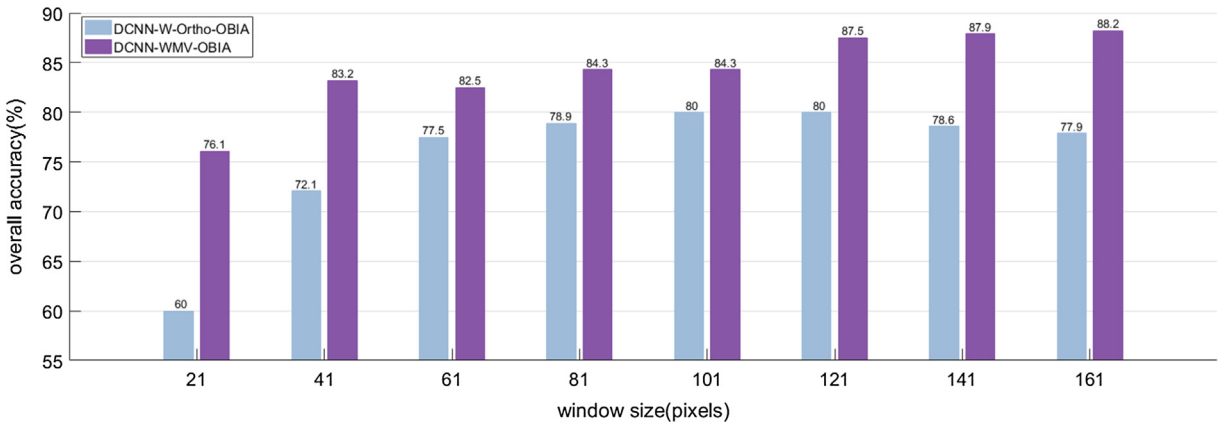


Fig. 7. Overall accuracy of DCNN-WMV-OBIA and DCNN-W-Ortho-OBIA obtained with different window sizes.

expensive computational resources. For example, training a DCNN requires 20 h using a Titan X premium GPU machine and training such a big DCNN on CPU is way slower making it practicable to work with computers that do not have independent GPU. In contrast, traditional classifiers such as RF do not require GPU and it only took few minutes to finish training. However, in contrast with its slow training procedure, inference procedure is very quick for DCNN with one image window/object only requiring 0.2 s for processing on our lab machine which has Intel Xeon E5-1620 CPU, 64G RAM and Titan X GPU.

4.2. Sensitivity analysis of window size for classification results using window-based data from multi-view

This section is related to study objective III, showing whether the proposed window-based version of the DCNN multi-view analysis (i.e. DCNN-WMV-OBIA) was able to give comparable results to DCNN-MV-OBIA. Such investigation is important for practical implementation since DCNN-WMV-OBIA only needs to project one point per object to each UAS images to achieve the multi-view information, while DCNN-MV-OBIA requires projecting all

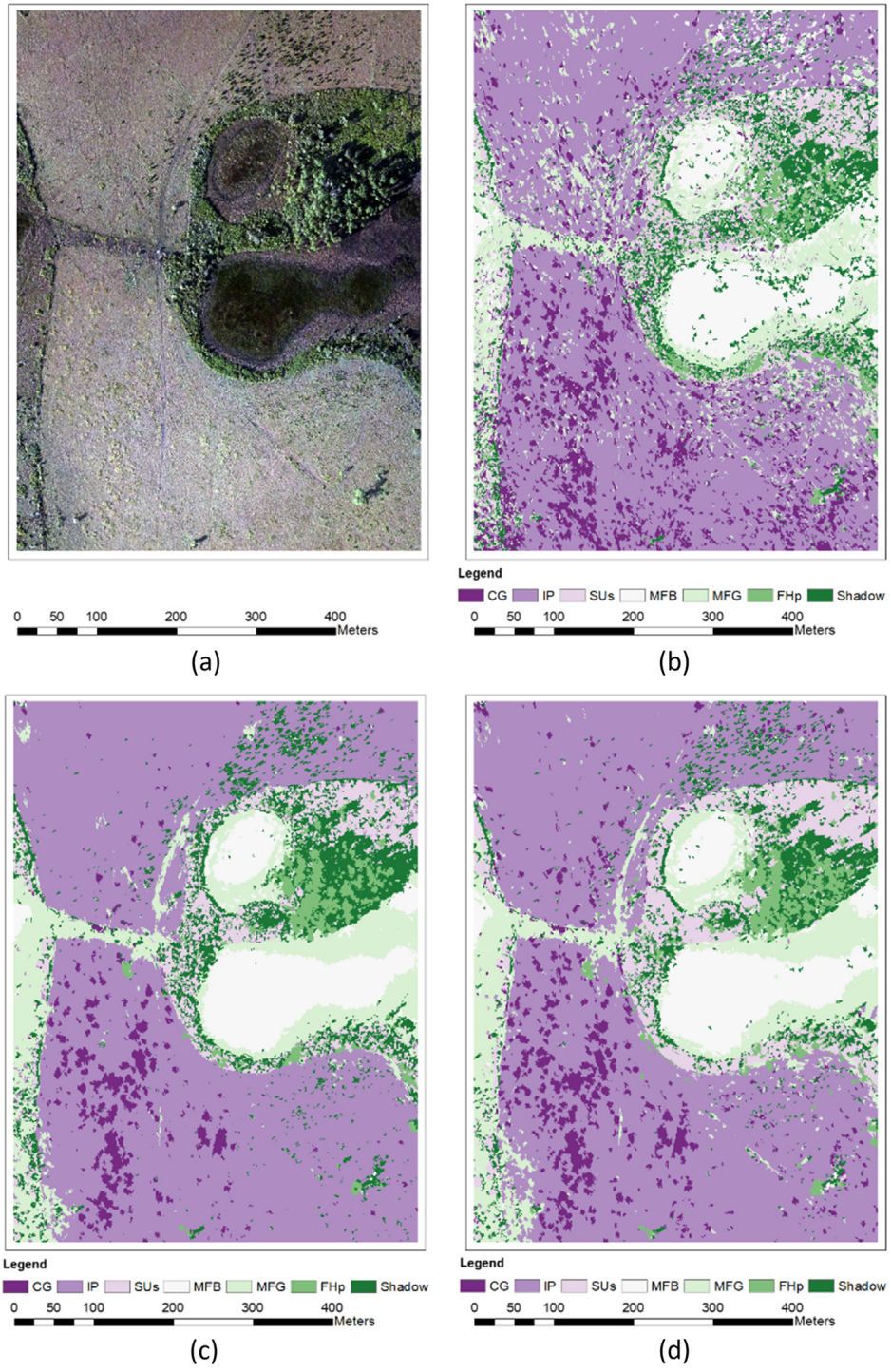


Fig. 8. Orthoimage and classification maps (whole study area): (a) orthoimage of the whole study area, (b) classification map from SVM-Ortho-OBIA, (c) classification map from DCNN-MV-OBIA, (d) classification map from DCNN-WMV-OBIA.

the object vertices onto UAS images. The computation cost for the latter can easily go beyond one hundred folds of the former approach, depending on the size and complexity of the object boundary. Results for simple application of DCNN with fixed-sized windows are also presented in this section along with DCNN-WMV-OBIA to examine how DCNN-WMV-OBIA performance is impacted by the window size. Fig. 7 shows the overall accuracy of the DCNN-WMV-OBIA and DCNN-W-Ortho-OBIA classifications obtained with window sizes ranging from $21 \times 21, 41 \times \ldots$ to $161 \times$. Fig. 7 clearly shows that DCNN-WMV-OBIA always produces higher accuracy than DCNN-W-Ortho-OBIA, regardless of used window size.

Another obvious observation in Fig. 7 is that while DCNN-W-Ortho-OBIA shows decreasing trend after it achieved optimal accuracy 80.0% at window size 121, DCNN-WMV-OBIA does not seem to decrease the classification accuracy as the window size changes from 21 to 161, implying the improved robustness of DCNN trained with multi-view samples.

When the window size reaches 41, an overall accuracy of 83.2% was obtained for DCNN-WMV-OBIA, which is comparable to the

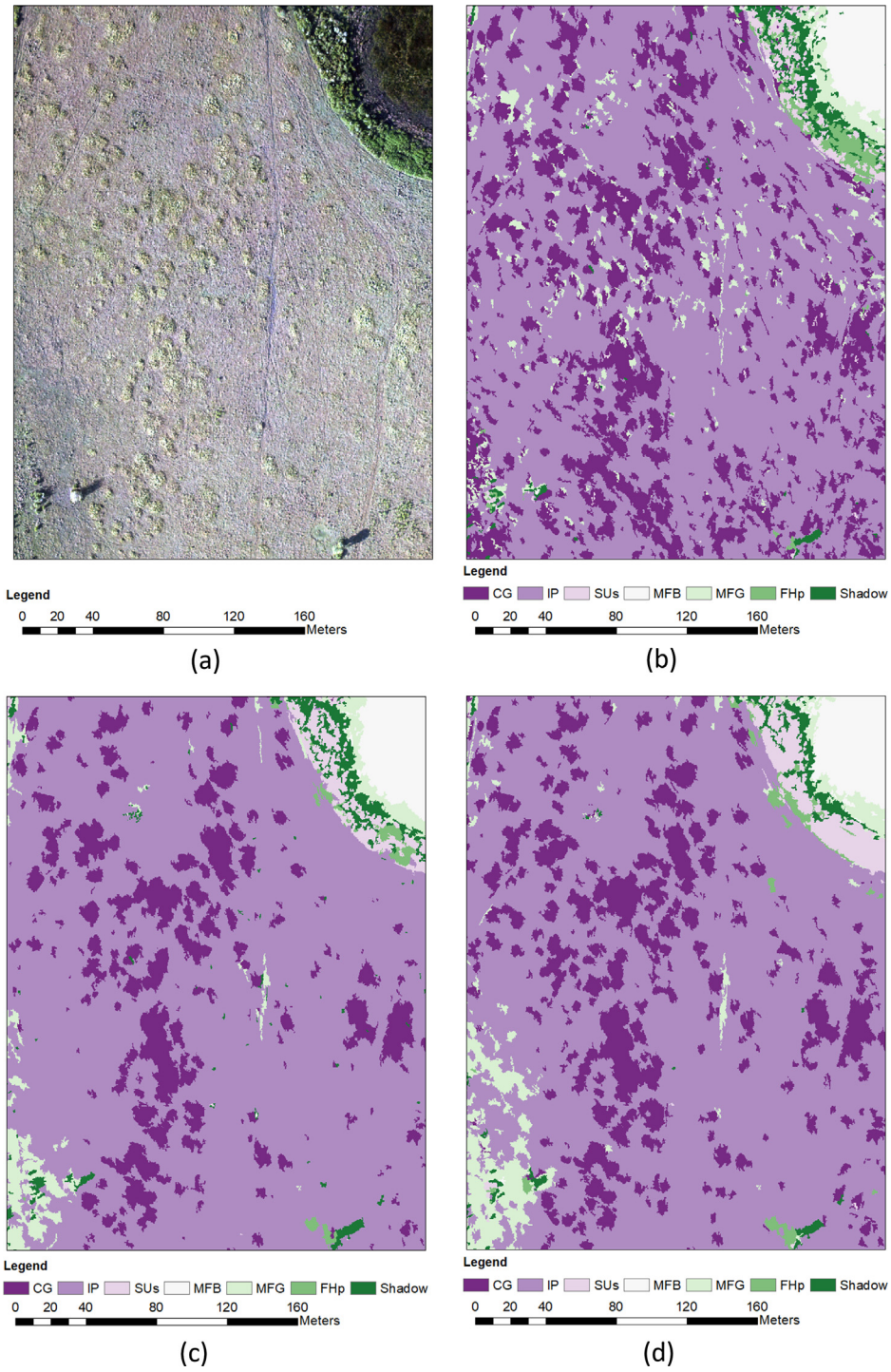


Fig. 9. Orthoimage and classification maps (Cogon Grass area): (a) orthoimage of the whole study area, (b) classification map from SVM-Ortho-OBIA, (c) classification map from DCNN-MV-OBIA, (d) classification map from DCNN-WMV-OBIA.

83.9% obtained by DCNN-MV-OBIA. Considering the average area of the objects, which is equivalent to a 43 × square window, such results may provide a tip regarding the selection of an appropriate window size to use for the DCNN-WMV-OBIA implementation. As a rule of thumb derived from Fig. 7, a square window that has the same area of the average area of the objects may be used to get the similar classification performance as the standard version of MODe. Following this rule of thumb, we applied the DCNN-WMV-OBIA with window size 43 × to classify the whole study area (see Fig. 8a), and the resulting classification map is shown in Fig. 8d, along with the maps generated with traditional method SVM-Ortho-OBIA in Fig. 8b and DCNN-MV-OBIA in Fig. 8c. Visual inspection of Fig. 8 shows, DCNN applied with multi-view information tends to generate more accurate classification maps (see Fig. 8c and d), but the map generated by traditional OBIA classification (see Fig. 8b) seems to mislabel the IP class as class MFG or CG in some area. Furthermore, while DCNN-WMV-OBIA tends to generate more CG labels compared to DCNN-MV-OBIA, they are overall qualitatively similar. Fig. 10 shows the same classification results as shown in Fig. 8, but focuses on a sub-area infested by Cogon Grass (i.e., CG). Visual inspection of Fig. 9 indicates DCNN-MV-OBIA mapped the CG accurately.

It's necessary to emphasize that we introduced the window-based version of the multi-view classification due to its desired computational efficiency. Compared to the standard multi-view OBIA, where hundreds of ground points need to be projected for a given ground object, the window-based version of the multi-view OBIA method, only requires a single point to be projected.

4.3. Sensitivity analysis of input layer size for classification results using different DCNN input layer size

This section is related to objective IV, i.e., studying how changing input layer size of DCNN would impact its performance MV-OBIA, the standard implementation of MODe. The DCNN used in this study has one fully connected layer at the end of the network, which makes changing the size of the DCNN input layer size only requires changing the height and width of the last convolutional layer. This special characteristic makes it relatively easy to configure the DCNN to enable it having different input layer sizes, since after selecting a different size for the input layer, it only needs to change the dimension of the last fully connected layer with all the other layers staying the same. We experimented with 15 × 15, 33 × 33, 65 × 65, 124 × 124, 224 × 224 (original size), and 325 × 325 input layer sizes and run each of the adapted DCNN with the same multi-view object instances dataset. The overall accuracy achieved with the different input layer sizes are shown in Fig. 10.

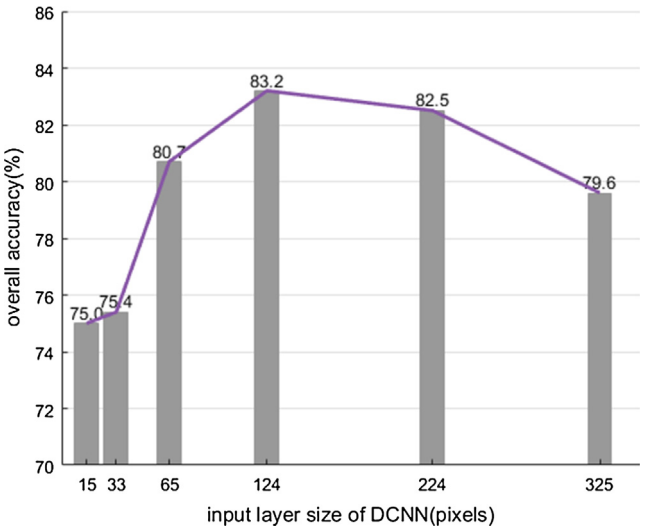


Fig. 10. Impact of image input size on overall accuracy.

Table 5
Welch's *t*-test results for the correctly and mistakenly classified object groups.

Type of classification result	Total number	Mean of ratio of object size to input layer size	Standard deviation of ratio of object size to input layer size
Correct classification	231	18.0	26.1
Wrong classification	49	25.9	23.8
p = 0.0413 for Welch's *t*-test			

Fig. 10 shows that as the input layer size increases, the overall accuracy increases until it peaks at 83.2% when the size reaches 124, after which the accuracy decreases to 79.6% as the size grows to 325. It should be noted that all the DCNNs used to generate the results in Fig. 10 used the same rectangular image patches corresponding to the complete object instances for training and testing. Therefore, the only factor causing the pattern observed in Fig. 10 is the structural change in the DCNN incurred by the change in the input layer size.

Given that our objects have different sizes and in order to investigate the effect of changing object size to fit the DCNN input layer size, we computed the ratio of the DCNN input layer size (224 × 224) to the object size in pixels for the correctly and mistakenly classified objects. The mean value of the ratios is reported in Table 5 for the correctly and mistakenly classified objects. Table 5 shows mean ratios of 18 (standard deviation equals 26) and 25.9 (standard deviation 23.8) for the correctly and mistakenly classified objects, respectively. We applied Welch's statistical *t*-test (Welch, 1947), which showed the mean of the ratio values for the mistakenly classified object group is significantly higher than the correctly classified group. Larger ratio value means larger enlargement of the original image patch to fit the DCNN input window size, which potentially resulted in more noise added to the original image patches due to the resampling operation, and hence reducing the classification accuracy. This may explain the decreasing pattern observed as the DCNN input layer size passed 124 in Fig. 10.

Enlarging the dimension of the DCNN input data layer does not always improve the classification accuracy, as shown in Fig. 10. As the size of the DCNN input layer increases, the learning power (the ability to accommodate complex pattern) of DCNN may also improve due to the increased number of parameters added to the last convolutional layer and fully connected layer, leading to an upward trend of overall accuracy in the left part of Fig. 10. However, this does not explain the right part of Fig. 10, where overall accuracy started to decrease as input layer size increased beyond 124. Recalling that the size of the objects in our dataset differs from one object to the other, and given that we have to resize the image patches to be equal in size as required for the DCNN input layer, we expect that this process can be the reason for the decreasing pattern on the right side of Fig. 10. Larger size of DCNN input layer requires larger enlargement of the original image patch, leading to more noises added to the image patch, which may reduce the classification accuracy. Even though this is just an empirical explanation, the bottom line is that an optimal size of DCNN input layer may need to be explored for optimal multi-view OBIA classification accuracy.

5. Conclusion and future study

This study proposed a novel methodology to apply DCNN of UAS images for natural land cover classification. The proposed method

is named MODe (**M**ulti-view **O**bject-based classification using **De**ep convolutional neural network), since it not only integrates the commonly used OBIA in DCNN, but also utilizes the multi-view data to satisfy the DCNN need for a large training dataset. Given limited orthoimage object training samples, multi-view projection technique was used to automatically generate object instances to form a new training set which can be 10–14 folds the original orthoimage objects. MODe triggers the power of the DCNN classifier by conducting training on the expanded training sets as compared to simple application of DCNN that conducts training on the orthoimage objects. Given an orthoimage object with unknown label, MODe conducts the classification on all the object instances corresponding to this orthoimage object and obtains the label of the orthoimage object via majority voting. This is in contrast with traditional OBIA for processing UAS data, which only relies on classifying the orthoimage objects and loses the abundant multi-view information embedded in the individual UAS images. The standard implementation of MODe requires the projection of all the boundary vertices to form complete object instance on the UAS images. To reduce the computation cost, we also proposed a window-based version of MODe, which only needs to project object centroids onto the UAS images.

Our results show that the MODe can significantly improve the classification accuracy as compared to simple application of DCNN on orthoimage. The results also show that the window-based version of MODe can achieve comparable or even better accuracy than the full object implementation of MODe. In addition, our results also imply that MODe is more likely to let the DCNN show its superiority over the RF and SVM traditional classifiers when these classifiers were trained with the multi-view object instances as compared to when these classifiers were trained with the traditional orthoimage objects. Additionally, the study showed that DCNN input layer size has an effect on the classification accuracy and using larger input layer size does not always improve the classification since accuracy started to decrease beyond certain input layer size.

While we tried to run our experiments as thorough as we could for the study area in this paper in order to achieve our research objectives, we believe it is important for future studies to test the methods introduced in this study in other study areas. Segmentation parameters selection and/or the use of multi-resolution analysis are other aspects that need to be explored further. In this research, we selected the segmentation scale parameter based on visual inspection. Even though this is a commonly used (Im et al., 2014), we think it is also important to test the implementation of the classification methods used in this study on the results obtained using different sets of segmentation parameters.

Acknowledgements

We thank the USACE ecologist Morton Jon for helping prepare the ground truth dataset and USACE geomatics lead Victor L. Wilhelm for collecting the UAS images and preparing the orthoimage.

Finally, we thank the ISPRS Associate Editor and all the anonymous reviewers for offering valuable suggestions that enabled this paper improved substantially.

References

Abuelgasim, A.A., Gopal, S., Irons, J.R., Strahler, A.H., 1996. Classification of ASAS multiangle and multispectral measurements using artificial neural networks. Remote Sens. Environ. 57, 79–87.

Ajaz Ahmed, M.A., Abd-Elrahman, A., Escobedo, F.J., Cropper Jr, W.P., Martin, T.A., Timilsina, N., 2017. Spatially-explicit modeling of multi-scale drivers of aboveground forest biomass and water yield in watersheds of the Southeastern United States 199, 158.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Breiman, L., 2001. Random forests. Mach. Lear. 45, 5–32.

Camacho-de Coca, F., Bréon, F.M., Leroy, M., Garcia-Haro, F.J., 2004. Airborne measurement of hot spot reflectance signatures. Remote Sens. Environ. 90, 63–75.

Celikyilmaz, A., Sarikaya, R., Hakkani-Tur, D., Liu, X., Ramesh, N., Tur, G., 2016. A new pre-training method for training deep learning models with application to spoken language understanding. Interspeech 2016, 3255–3259.

Chen, G., Weng, Q., Hay, G.J., He, Y., 2018. Geographic object-based image analysis (GEOBIA): emerging trends and future opportunities. GIScience & Remote Sensing.

Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. Deep learning-based classification of hyperspectral data. Selected topics in applied earth observations and remote sensing. IEEE J. 7, 2094–2107.

Cheng, G., Han, J., Lu, X., 2017a. Remote sensing image scene classification: benchmark and state of the art. Proc. IEEE.

Cheng, G., Li, Z., Yao, X., Guo, L., Wei, Z., 2017. Remote sensing image scene classification using bag of convolutional features. IEEE Geosci. Remote Sens. Lett. 14, 1735–1739.

Cheng, G., Zhou, P., Han, J., 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens. 54, 7405–7415.

Chopping, M.J., Rango, A., Havstad, K.M., Schiebe, F.R., Ritchie, J.C., Schmugge, T.J., French, A.N., Su, L., McKee, L., Davis, M.R., 2003. Canopy attributes of desert grassland and transition communities derived from multiangular airborne imagery. Remote Sens. Environ. 85, 339–354.

Cleve, C., Kelly, M., Kearns, F.R., Moritz, M., 2008. Classification of the wildland–urban interface: a comparison of pixel-and object-based classifications using high-resolution aerial photography. Comput. Environ. Urban Syst. 32, 317–326.

Colomina, I., Molina, P., 2014. Unmanned aerial systems for photogrammetry and remote sensing: a review. ISPRS J. Photogramm. Remote Sens. 92, 79–97.

Developer, e., 2012. User guide. Trimble Documentation.

Drăguț, 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.

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

eCognition, 2012. Features Reference, http://community.ecognition.com/home/features-reference.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., Bengio, S., 2010. Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660.

Esch, T., Thiel, M., Bock, M., Roth, A., Dech, S., 2008. Improvement of image segmentation accuracy based on multiscale optimization procedure. IEEE Geosci. Remote Sens. Lett. 5, 463–467.

Fu, B., Wang, Y., Campbell, A., Li, Y., Zhang, B., Yin, S., Xing, Z., Jin, X., 2017. Comparison of object-based and pixel-based Random Forest algorithm for wetland vegetation mapping using high spatial resolution GF-1 and SAR data. Ecol. Ind. 73, 105–117.

Gao, Y., Mas, J.F., 2008. A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolutions. Online J. Earth Sci. 2, 27–35.

Gatebe, C.K., King, M.D., 2016. Airborne spectral BRDF of various surface types (ocean, vegetation, snow, desert, wetlands, cloud decks, smoke layers) for remote sensing applications. Remote Sens. Environ. 179, 131–148.

Gleason, C.J., Im, J., 2012. Forest biomass estimation from airborne LiDAR data using machine learning approaches. Remote Sens. Environ. 125, 80–91.

Grybas, H., Melendy, L., Congalton, R.G., 2017. A comparison of unsupervised segmentation parameter optimization approaches using moderate-and high-resolution imagery. GIScience Remote Sens., 1–19

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proc. IEEE Conf. Computer Vision Pattern Recog., 770–778

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554.

Holm, L.G., Plucknett, D.L., Pancho, J.V., Herberger, J.P., 1977. The World's Worst Weeds. University Press.

Hsu, C.-W., Lin, C.-J., 2002. A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Networks 13, 415–425.

Im, J., Park, S., Rhee, J., Baik, J., Choi, M., 2016. Downscaling of AMSR-E soil moisture with MODIS products using machine learning approaches. Environ. Earth Sci. 75, 1120.

Im, J., Quackenbush, L.J., Li, M., Fang, F., 2014. Optimum scale in object-based image analysis. Scale Issues Remote Sens., 197–214

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Ke, Y., Quackenbush, L.J., Im, J., 2010. Synergistic use of QuickBird multispectral imagery and LIDAR data for object-based forest species classification. Remote Sens. Environ. 114, 1141–1154.

Koukal, T., Atzberger, C., 2012. Potential of multi-angular data derived from a digital aerial frame camera for forest classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 5, 30–43.

Koukal, T., Atzberger, C., Schneider, W., 2014. Evaluation of semi-empirical BRDF models inverted against multi-angle data from a digital airborne frame camera for enhancing forest type classification. Remote Sens. Environ. 151, 27–43.

Kunze, J., Kirsch, L., Kurenkov, I., Krug, A., Johannsmeier, J., Stober, S., 2017. Transfer Learning for Speech Recognition on a Budget. arXiv preprint arXiv:1706.00290.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lee, S., Han, H., Im, J., Jang, E., Lee, M.-I., 2017. Detection of deterministic and probabilistic convection initiation using Himawari-8 Advanced Himawari Imager data. Atmos. Meas. Tech. 10, 1859.
Li, M., Im, J., Beier, C., 2013. Machine learning approaches for forest classification and change analysis using multi-temporal Landsat TM images over Huntington Wildlife Forest. GIScience Remote Sens. 50, 361–384.
Li, M., Im, J., Quackenbush, L.J., Liu, T., 2014. Forest biomass and carbon stock quantification using airborne LiDAR data: a case study over Huntington Wildlife Forest in the Adirondack Park. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. 7, 3143–3156.
Liu, L., Liu, X., Wang, Z., Zhang, B., 2016. Measurement and analysis of bidirectional SIF emissions in wheat canopies. IEEE Trans. Geosci. Remote Sens. 54, 2640–2651.
Liu, T., Abd-Elrahman, A., Jon, M., 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. GIScience Remote Sens.
Liu, T., Im, J., Quackenbush, L.J., 2015. A novel transferable individual tree crown delineation model based on Fishing Net Dragging and boundary classification. ISPRS J. Photogramm. Remote Sens. 110, 34–47.
Longbotham, N., Chaapel, C., Bleiler, L., Padwick, C., Emery, W.J., Pacifici, F., 2012. Very high resolution multiangle urban classification analysis. IEEE Trans. Geosci. Remote Sens. 50, 1155–1170.
Lowe, S.H., Guo, X., 2011. Detecting an optimal scale parameter in object-oriented classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 4, 890–895.
Lu, B., He, Y., 2017. Species classification using Unmanned Aerial Vehicle (UAV)-acquired high spatial resolution imagery in a heterogeneous grassland. ISPRS J. Photogramm. Remote Sens. 128, 73–85.
Lu, D., Li, G., Moran, E., Kuang, W., 2014. A comparative analysis of approaches for successional vegetation classification in the Brazilian Amazon. GIScience Remote Sens. 51, 695–709.
Lu, X., Zheng, X., Yuan, Y., 2017. Remote sensing scene classification by unsupervised representation learning. IEEE Trans. Geosci. Remote Sens. 55 (9), 5148–5157.
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.
Ma, X., Wang, H., Wang, J., 2016. Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. ISPRS J. Photogramm. Remote Sens. 120, 99–107.
Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks, geoscience and remote sensing symposium (IGARSS), 2015 IEEE Int. IEEE, 4959–4962.
Marcos, D., Volpi, M., Tuia, D., 2016. Learning rotation invariant convolutional filters for texture classification. arXiv preprint arXiv:1604.06720.
Mnih, V., Hinton, G.E., 2010. Learning to detect roads in high-resolution aerial images. Eur. Conf. Computer Vision. Springer, 210–223.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814.
Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks. Proc. IEEE Conf. Computer Vision Pattern Recog., 1717–1724
Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowledge Data Eng. 22, 1345–1359.
Pande-Chhetri, R., Abd-Elrahman, A., Liu, T., Morton, J., Wilhelm, V.L., 2017. Object-based classification of wetland vegetation using very high-resolution unmanned air system imagery. Eur. J. Remote Sens. 50, 564–576.
Park, S., Im, J., Jang, E., Rhee, J., 2016. Drought assessment and monitoring through blending of multi-sensor indices using machine learning approaches for different climate regions. Agric. For. Meteorol. 216, 157–169.
Rhee, J., Im, J., 2017. Meteorological drought forecasting for ungauged areas based on machine learning: using long-range climate forecast and remote sensing data. Agric. For. Meteorol. 237, 105–122.
Rutchey, K., Schall, T., Doren, R., Atkinson, A., Ross, M., Jones, D., Madden, M., Vilchek, L., Bradley, K., Snyder, J., 2006. Vegetation classification for South Florida natural areas. US Geological Survey, St Petersburg, FL.
Scholkopf, B., Smola, A.J., Snyder, J., 2001. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.
Sheikh, P.A., 2002. Florida Everglades Restoration: Background on Implementation and Early Lessons. The Libraryof Congress: CongressionalResearch Service
Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., 2017. Mastering the game of go without human knowledge. Nature 550, 354–359.
Su, L., Chopping, M.J., Rango, A., Martonchik, J.V., Peters, D.P., 2007. Support vector machines for recognition of semi-arid vegetation types using MISR multi-angle imagery. Remote Sens. Environ. 107, 299–311.
Torrey, L., Shavlik, J., 2009. Transfer learning. Handbook Res. Mach. Learn. Appl. Trends: Algorithms, Methods, Techniques 1, 242.
Vedaldi, A., Lenc, K., 2015. Matconvnet: convolutional neural networks for matlab. In: Proceedings of the 23rd ACM International Conference on Multimedia. ACM, pp. 689–692.
Vetrivel, A., Gerke, M., Kerle, N., Nex, F., Vosselman, G., 2017. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. ISPRS J. Photogramm. Remote Sens.
Welch, B.L., 1947. The generalization of 'student's' problem when several different population variances are involved. Biometrika 34, 28–35.
Xie, M., Jean, N., Burke, M., Lobell, D., Ermon, S., 2015. Transfer learning from deep features for remote sensing and poverty mapping. arXiv preprint arXiv:1510.00098.
Xun, L., Wang, L., 2015. An object-based SVM method incorporating optimal segmentation scale estimation using Bhattacharyya Distance for mapping salt cedar (Tamarisk spp.) with QuickBird imagery. GIScience Remote Sens. 52, 257–273.
Yang, J., Zhao, Y.Q., Chan, J.C.W., 2017. Learning and transferring deep joint spectral-spatial features for hyperspectral classification T2 - IEEE transactions on geoscience and remote sensing. IEEE Trans. Geosci. Remote Sens. pp. 1.
Yao, X., Han, J., Cheng, G., Qian, X., Guo, L., 2016. Semantic annotation of high-resolution satellite images via weakly supervised learning. IEEE Trans. Geosci. Remote Sens. 54, 3660–3671.
Yegnanarayana, B., 2009. Artificial neural networks. PHI Learning Pvt. Ltd..
Yu, Q., Gong, P., Clinton, N., Biging, G., Kelly, M., Schirokauer, D., 2006. Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. Photogramm. Eng. Remote Sens. 72, 799–811.
Yu, X., Wu, X., Luo, C., Ren, P., 2017. Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. GIScience Remote Sens., 1–18
Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 116, 24–41.
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., 2017a. Object-based convolutional neural network for high-resolution imagery classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.
Zhao, W., Du, S., Wang, Q., Emery, W.J., 2017b. Contextually guided very-high-resolution imagery classification with semantic segments. ISPRS J. Photogramm. Remote Sens. 132, 48–60.
Zhong, P., Gong, Z., Li, S., Schönlieb, C.-B., 2017. Learning to diversify deep belief networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.