Mapping of land cover with open-source software and ultra-high-resolution imagery acquired with unmanned aerial vehicles

Ned Horning, Erica Fleishman, Peter J. Ersts, Frank A. Fogarty & Martha Wohlfeil Zillig

Center for Biodiversity and Conservation, American Museum of Natural History, New York, New York 10024
Department of Environmental Science and Policy, University of California, Davis, California 95616
Department of Fish, Wildlife and Conservation Biology, Colorado State University, Fort Collins, Colorado 80523

Keywords
Aerial photography, machine learning, neural networks, open source, UAV, ultra-high resolution

Abstract
The use of unmanned aerial vehicles (UAVs) to map and monitor the environment has increased sharply in the last few years. Many individuals and organizations have purchased consumer-grade UAVs, and commonly acquire aerial photographs to map land cover. The resulting ultra-high-resolution (sub-decimeter-resolution) imagery has high information content, but automating the extraction of this information to create accurate, wall-to-wall land-cover maps is quite difficult. We introduce image-processing workflows that are based on open-source software and can be used to create land-cover maps from ultra-high-resolution aerial imagery. We compared four machine-learning workflows for classifying images. Two workflows were based on random forest algorithms. Of these, one used a pixel-by-pixel approach available in ilastik, and the other used image segments and was implemented with R and the Orfeo Toolbox. The other two workflows used fully connected neural networks and convolutional neural networks implemented with Nenetic. We applied the four workflows to aerial photographs acquired in the Great Basin (western USA) at flying heights of 10 m, 45 m and 90 m above ground level. Our focal cover type was cheatgrass (Bromus tectorum), a non-native invasive grass that changes regional fire dynamics. The most accurate workflow for classifying ultra-high-resolution imagery depends on diverse factors that are influenced by image resolution and land-cover characteristics, such as contrast, landscape patterns and the spectral texture of the land-cover types being classified. For our application, the ilastik workflow yielded the highest overall accuracy (0.82–0.89) as assessed by pixel-based accuracy.

Introduction
Several peer-reviewed papers (e.g. Horning 2018; Singh and Frazier 2018; Jiménez López and Mulero-Pázmány 2019) and books (Calvo and Lovejoy 2018; Wich and Koh 2018) illustrate realized and potential applications of unmanned aerial vehicles (UAVs) in conservation and ecology, often with a focus on spatial assessment of land cover. Much of that literature emphasizes techniques that stitch together hundreds of aerial photographs to create orthophoto mosaics that are geometrically correct and can be used in a geographic information system (GIS). Few publications provide pragmatic guidance on use of methods other than visual interpretation to extract extremely detailed land-cover information from images acquired by UAVs.

There is tremendous value in visual or manual interpretation of true-color images (Ghorbani and Pakravan 2013). However, automating some of the information-extraction steps offers potential for analyzing much larger volumes of data, which might encompass larger areas, longer time periods or more-frequent sampling. Here, we test four different machine-learning workflows to extract land-cover features from low-altitude aerial imagery. Our results may be applicable to the design of future research.
Our work was informed by 3 years of experiments to automate the classification of land-cover types in sagebrush (Artemisia tridentata) shrubsteppe in the Great Basin of the western United States. We sought to improve the ability to transition from field measurements to mapping of land cover on the basis of moderate-resolution images from satellite-mounted sensors. Low-altitude aerial imagery can be a valuable source of intermediary data when transitioning from field measurements to predictions that are based on coarser-resolution imagery collected by satellite sensors (Wang et al. 2017; Leitão et al. 2018). Although kite or pole aerial photography platforms and hand-held digital cameras are valuable when drone flights are prohibited or for long-term monitoring, UAVs often are preferred because they are readily available, they can be programmed to fly precise autonomous missions, and systems with integrated cameras are widely available.

We define features (predictor variables) as dimensions of features space derived from an aerial image, objects as physical objects in an image and sub-objects as components of an object. Examples of features are spectral bands, indices derived from spectral bands and computed textures. Each of these features is a dimension in the feature space used to classify image pixels. Examples of objects are contiguous land-cover types or discrete objects, such as a rock or shrub. Sub-objects of a shrub, for example, might include leaves, branches, and shadow and soil throughout or under the shrub.

We focused on evaluating methods that are applicable to imagery acquired from a UAV at flying heights < 122 m above ground level, an upper limit set by a US federal regulation with a nadir (vertical) view. Images acquired with point-and-shoot or action cameras, which often are supplied with consumer UAVs at these flying heights, have a spatial resolution of 10 cm (decimetre resolution) or finer. At these ultra-fine resolutions, which are cases of H-resolution (Strahler et al. 1986), high intra-object spectral variability makes it difficult to identify or classify an object in its entirety with conventional classification methods. Object-based segmentation methods that use region-growing algorithms are effective with coarser-resolution (<1–4 m) satellite imagery (Yu et al. 2006). However, these segmentation methods tend to be less effective for classifying images with sub-decimeter resolutions and high intra-object spectral variability unless classification results are adjusted manually post-processing (Pande-Chhetri et al. 2017). Object-based segmentation works best when intra-object variability is low and the objects strongly contrast with the background and neighboring objects (Hsieh et al. 2017; Kalantar et al. 2017).

Materials and Methods

Image classification algorithms

We restricted this work to nonparametric classification algorithms, which are more effective than parametric algorithms when imagery has high within-class spectral variability (Zhang et al. 2006). Parametric algorithms, such as maximum likelihood, are constrained to a specific statistical model and make assumptions about how data are distributed within each class. Nonparametric algorithms do not make such assumptions, and tend to be more successful for complex classification tasks (Zhang et al. 2006).

All four of our workflows include information from neighboring pixels in the feature set. However, all but one technically are pixel-based classifiers. Some deep-learning methods, such as convolutional neural networks, can be used to identify specific objects by drawing a bounding box, or by segmenting and labeling objects, as is done with semantic segmentation, but use of these algorithms for wall-to-wall land-cover mapping still is relatively experimental. Deep learning approaches such as U-Net and other autoencoders (Feng et al. 2018; Liang et al. 2018) have been used for land-cover mapping, but they require fully classified, ground-truthed images to train models, which is not practical for many applications. Pixel-by-pixel classification algorithms, which use surrounding pixels to add context, have existed for more than a decade (Shen and Sarris 2008).

Image acquisition

We acquired imagery for this work during the second year (2017) of our experiments in the Great Basin. We used the stock red-green-blue (RGB) camera on a DJI Phantom 3 Pro quadcopter that we flew over heterogeneous vegetation within the sagebrush shrubsteppe ecosystem. For our workflow comparison, we selected three, 12 megapixel (3000 lines by 4000 columns) images, one each from a flying height of 10 m, 45 m and 90 m above the elevation of the take-off point. Because the ground was relatively level, we refer to the flying height as above ground level (AGL), and distortion due to terrain effects was minimal. We acquired the three images on 14 June 2017 from the same area in central Nevada: 38° 53′ 57.51″ N, 117° 51′ 31.73″ W (Fig. 1). We acquired the images at 10, 45 and 90 m AGL at 9:21, 9:26 and 9:27 local time, respectively. The solar altitude was 39° and azimuth was 90°.

Images with a low sun angle are not ideal for land-cover classification because long shadows cast from shrubs and other tall vegetation may obscure a high
proportion of the ground surface. Unfortunately, as in this case, it is not always possible to acquire imagery at the optimum time of day. We used the images to develop and test the accuracy and feasibility of different image-processing workflows to classify cheatgrass and general land-cover categories, primarily herbaceous vegetation and shrubs. We used the JPEG images for classification because NH’s experience suggests that JPEG is the image format most commonly used by conservation practitioners when acquiring photographs from consumer UAVs.

Training and validation data

We classified images acquired from 10 m AGL into six cover types – cheatgrass (*Bromus tectorum*), a non-native invasive grass that changes regional fire dynamics (Balch et al. 2013; Bradley et al. 2018); other grasses and forbs; shrub; soil and rock; litter; and shadow. We did not include litter in our classification of images from 45 m and 90 m AGL because individual litter objects became too small to interpret visually.

When machine learning is used to classify land cover, the volume of training data should represent the full range of variation within each feature being used as a predictor variable. However, measuring the extent to which variability has been captured in the training data is not straightforward. The machine-learning algorithms we use here are nonparametric, and information that can be used to assess the quality of training data for nonparametric methods is limited. Therefore, the tools for collection and assessment of training data for statistical models, such as maximum likelihood, cannot easily be applied when using non-statistical learning algorithms. NH’s experience suggests that in most cases a qualitative assessment is made by the analyst collecting the training data.

To increase our confidence that the training data adequately represented feature variability, we used a k-means unsupervised classifier to partition feature space into several clusters, and then selected training data from each cluster to ensure there were no major gaps in feature variability throughout the image. We selected training data from each image. We applied six steps to extract training data for all but the segmentation workflow. First, we created 50 clusters by applying a k-means unsupervised classification algorithm to the aerial photograph. Second, we displayed each of the clusters, in turn, over the aerial photograph. Third, for each cluster, we used QGIS (QGIS Development Team 2019) to draw polygons around pixels that coincided with a single cover type, and labeled each polygon with cluster number and cover type. Fourth, for each cluster, we extracted pixels that intersected with each polygon associated with the cluster, and labeled those
pixels with the polygon’s cover type attribute. Fifth, from the set of all labeled pixels, we systematically sampled 22,000 pixels. We calculated the sampling interval to ensure that the proportion of pixels extracted from a particular cluster was proportional to the number of pixels assigned to that cluster in the entire clustered image. Sixth, we applied systematic sampling to the data created in step five to extract 20,000 pixels for model training and 2000 pixels for model validation.

We implemented step five with QGIS and steps four, five and six with an R script (see supplemental material). We used the validation data to assess accuracy of all workflows. We collected training data for the segmentation workflow using QGIS to label image segments (450 segments at 10 m, 130 at 45 m and 150 at 90 m).

Classification

The workflows on which we focused are ilastik, segmentation, fully connected neural networks (FCNs), which process one-dimensional vectors; and convolutional neural networks (CNNs), which process two-dimensional image chips (subsets). All of the software for these workflows is open source and available, at no cost, for Windows, Mac and Linux operating systems. Image and training data and the variables and scripts used for each of these workflows are in the supplemental material. We ran all classifications on an Intel® Core™ i7-7820X CPU @ 3.60 GHz × 16 and 32 GB of RAM and a GeForce GTX 1080 graphics card (used by Nenetic). Image size was 4000 × 3000 × 3 bands.

Ilastik workflow

Ilastik software (Sommer et al. 2011) (https://www.ilastik.org/) was created to support image classification and segmentation for biological applications, primarily cell counting. Its interface is easy to use, even for those with limited image classification experience. The Ilastik classification workflow uses a random forest algorithm to classify an image. Random forest (Breiman 2001) is a nonparametric, machine-learning method with a well-documented record of high classification accuracy for remote sensing applications (Rodriguez-Galiano et al. 2012). Additionally, the ability of the random forest algorithm to accommodate many dimensions in the data (e.g. several dozen features) allowed us to supplement our red, green and blue spectral layers. Ilastik users may select a minimum of nine features related to color, edge detection and texture. These features can be calculated at multiple scales (variance of Gaussian smoothing). A random forest model is created from the selected features and then used to predict a class label for each image pixel. Training data for the model are selected with a paintbrush that is similar to drawing tools in graphics software such as Adobe Illustrator or Photoshop.

Ilastik can be configured to update the classification automatically when the training data are modified, providing near real-time feedback on the effect of adding or removing training data. Ilastik calculates a layer depicting the probability of correct prediction, which allows one to assess the quality of training data. Pixels with low confidence indicate that more training data that are representative of those pixels should be collected. Once created, the model can be applied to one or more images with a batch feature. Training data from multiple images can be used to create a model.

The ilastik workflow allows one to create a classified image quickly. The ilastik website provides a short guide (Ilastik 2019) to facilitate learning the workflow. Although ilastik classifies images on a pixel-by-pixel basis, its use of coarse-resolution features generates an output that is less speckled than that typically associated with pixel-by-pixel classification algorithms.

Segmentation workflow

The segmentation workflow includes two distinct steps, image segmentation and classification of image segments. After visually assessing the results from watershed and mean-shift segmentation algorithms, we selected the ‘Large Scale Mean Shift (LSMS) image segmentation’ application in the open-source Orfeo ToolBox (OTB) (https://www.orfeo-toolbox.org/) library (Grizonnet et al. 2017). The mean-shift algorithm uses spatial and range (spectral) parameters to control segment size and shape (Comaniciu and Meer 2002), and can be used to segment remotely sensed multispectral images (Michel et al. 2015). The spatial parameter controls the size of the neighborhood that becomes the basis for averaging. The range parameter controls the search distance in feature space; longer distances tend to be less sensitive than shorter distances to edges between land-cover types. The mean-shift algorithm also incorporates a minimum segment size parameter that can be used to eliminate segments with fewer than a specified number of pixels. Tuning the segmentation algorithm until the outcome is satisfactory usually requires several iterations with different parameters (Grippa et al. 2017). Tuning on subsets of images can accelerate the process. After parameters have been selected, they often can be applied to other photographs with similar qualities (e.g. region, vegetation and flying height).

After an image has been segmented and training data selected (see ‘Training and validation data’ above), the segments must be labeled with a classification algorithm.
We used a script that we wrote in R (R Development Team, 2008), which was based in part on the randomForest package (Liaw and Wiener 2007), to apply a random forest classification algorithm and label each image segment with a land-cover class. Within the R script, we calculated summary statistics (mean, median, standard deviation and coefficient of variation) for each segment within each image band. The scripts and a user guide are available at https://github.com/nedhorning/RandomForestForRemoteSensing.

FCN and CNN workflows

To implement FCNs and CNNs, we used the Neural Network Image Classifier (Nenetic – (https://github.com/persts/Nenetic), an open-source software package under development at the American Museum of Natural History. To the best of our knowledge, Nenetic is the first publicly available software application for land-cover classification that applies neural network algorithms to vectors (FCN) or image chips (CNN) that are derived from user-selected regions around a central pixel. Training data extraction locations are collected via points that can be selected individually or in a stream mode. Cre- ation of training data for each of these extraction locations is controlled with a dialog that allows the user to select features and specify the size of the region around each extraction location. Available feature-selection rules include calculating averages of pixels within regions of different sizes, selecting all pixels in a neighborhood around a central pixel and calculating a series of RGB indices for all pixels in the neighborhood defined by the user.

When running the FCN workflow, we included all pixels in a 5 × 5 pixel area as the input vector for the central pixel. In ilastik, each of the calculated feature layers is smoothed (blurred) by a Gaussian filter with standard deviation (sigma) values from 1 to 10. This smoothing uses a weighted average of neighboring pixels to recalculate the value of the center pixel. Pixels in the CNN output were relatively aggregated, a result of our use of a 31 × 31 pixel image chip (subset).

Accuracy assessment

We assessed the accuracy of each workflow on the basis of 2000 pixels selected from the original set of labeled data for each flying height. The total number of original labeled pixels from the images acquired at 10 m, 45 m and 90 m flying heights was 880 267, 267 195 and 230 670, respectively. Therefore, the number of training points equated to less than one per cent of the labeled data. Because our test data were a small fraction of the original labeled pixels, spatial autocorrelation between training and validation data likely was minimal.

We calculated overall accuracy at each flying height. We also assessed accuracy per class at each flying height by calculating user’s and producer’s accuracy (Story & Congalton 1986) and balanced accuracy (Velez et al. 2007). For a given class, user’s accuracy is the proportion of pixels attributed to that class that were classified correctly, whereas producer’s accuracy is the proportion of reference pixels for the class that were classified correctly. User’s accuracy is map-based, and producer’s accuracy is reference-based. Balanced accuracy, which compensates for differences in sample sizes among classes, is calculated as 0.5 * (proportion of all positives that are true pos- itives + proportion of all negatives that are true nega- tives).

Results

The images we used for this work and the original metadata are included with the supplemental material. The time and effort required to import an image and select training data are roughly the same for the ilastic, FCN and CNN workflows because the processes are similar. The segmentation workflow includes an image segmentation step before the classification model can be run. This step can be time-consuming because considerable trial and error can be required to select suitable segmentation parameters. Model training and prediction processing times for ilastic and segmentation each were 5 min. Processing times for FCN and CNN were 15 and 40 min, respectively. These values should be interpreted as relative; processing time will vary substantially depending on the number of classes and the number and size of features in the data, and on the hardware specifications.

On the basis of overall accuracy, the ilastic workflow yielded the most accurate results (Tables 1–4; the confusion matrix and additional accuracy metrics for each classification method are in the supplemental material). Accuracy as measured by the per-class metrics was more equivocal. At the 10 m flying height (Table 1), the FCN workflow most accurately classified the shrub class, but the ilastic workflow generally was the most accurate classifier. At the 45 m flying height (Table 2), ilastic most accurately classified all but the shrub class. At the 90 m flying height (Table 3), the CNN workflow most accurately classified grasses and forbs, shrubs and cheatgrass.

Visually, the outputs from ilastic and FCN were quite similar (Fig. 2), which may reflect the way they use information from neighboring pixels. Incorporation of neighboring pixel information also may explain why FCN and CNN workflows classified shrubs, the most heterogeneous class, more accurately than ilastic. The segmentation
output appeared quite different from that of the other workflows, in part because the training data were independent of the data to which the other three workflows were applied but also because image-segmentation algorithms group pixels into irregularly shaped segments before classification. The difference between the segmentation output and the outputs of the other workflows increased as the flying height increased; as height increased, many smaller land-cover patches appeared to merge with surrounding cover types.

All of the workflows have the potential to accurately classify cheatgrass when it is spectrally distinct from the

| Class accuracy metrics for classification of an image taken 10 m above ground level. Boldface indicates highest value for a class and accuracy metric. | Grass and forb | Litter | Shadow | Shrub | Soil and rock | Cheatgrass |
|---|---|---|---|---|---|---|
| User's accuracy | | | | | | |
| Fully connected neural networks | 0.54 | 0.68 | 0.93 | 0.86 | 0.74 | 0.83 |
| Convolutional neural networks | 0.53 | 0.69 | 0.89 | 0.88 | 0.75 | 0.76 |
| Ilastik | **0.72** | **0.81** | **0.94** | **0.90** | **0.77** | **0.79** |
| Segmentation | 0 | 0.29 | 0.93 | 0.65 | 0.53 | 0.74 |
| Producer's accuracy | | | | | | |
| Fully connected neural networks | 0.39 | 0.51 | 0.89 | 0.81 | 0.83 | 0.85 |
| Convolutional neural networks | 0.20 | 0.39 | **0.91** | 0.65 | 0.78 | 0.88 |
| Ilastik | **0.48** | **0.61** | 0.91 | 0.59 | 0.85 | **0.89** |
| Segmentation | 0 | 0.38 | 0.69 | 0.72 | **0.86** | 0.57 |
| Balanced accuracy | | | | | | |
| Fully connected neural networks | 0.69 | 0.75 | 0.93 | **0.90** | 0.87 | 0.88 |
| Convolutional neural networks | 0.60 | 0.69 | 0.94 | 0.82 | 0.85 | 0.87 |
| Ilastik | **0.74** | **0.80** | **0.95** | 0.79 | **0.89** | **0.89** |
| Segmentation | 0.50 | 0.67 | 0.84 | 0.84 | 0.81 | 0.73 |

| Class accuracy metrics for classification of an image taken 45 m above ground level. Boldface indicates highest value for a land-cover class and accuracy metric. | Grass and forb | Litter | Shadow | Shrub | Soil and rock | Cheatgrass |
|---|---|---|---|---|---|---|
| User's accuracy | | | | | | |
| Fully connected neural networks | 0.73 | 0.91 | 0.77 | 0.83 | 0.84 | 0.84 |
| Convolutional neural networks | 0.77 | 0.89 | 0.85 | 0.88 | 0.84 | 0.84 |
| Ilastik | **0.81** | **0.92** | **0.85** | **0.89** | **0.87** | **0.87** |
| Segmentation | 0.60 | **0.95** | 0.44 | 0.59 | 0.72 | 0.72 |
| Producer's accuracy | | | | | | |
| Fully connected neural networks | 0.77 | 0.90 | 0.67 | 0.75 | 0.92 | 0.92 |
| Convolutional neural networks | 0.76 | 0.94 | **0.82** | 0.80 | 0.88 | 0.88 |
| Ilastik | **0.84** | **0.97** | 0.72 | **0.87** | **0.91** | **0.91** |
| Segmentation | 0.49 | 0.56 | 0.73 | 0.61 | 0.73 | 0.73 |
| Balanced accuracy | | | | | | |
| Fully connected neural networks | 0.84 | 0.94 | 0.82 | 0.86 | 0.93 | 0.93 |
| Convolutional neural networks | 0.85 | 0.96 | **0.90** | 0.89 | 0.91 | 0.91 |
| Ilastik | **0.89** | **0.97** | 0.85 | **0.92** | **0.93** | **0.93** |
| Segmentation | 0.70 | 0.78 | 0.78 | 0.76 | 0.81 | 0.81 |

| Class accuracy metrics for classification of an image taken 90 m above ground level. Boldface indicates highest value for a land-cover class and accuracy metric. | Grass and forb | Litter | Shadow | Shrub | Soil and rock | Cheatgrass |
|---|---|---|---|---|---|---|
| User's accuracy | | | | | | |
| Fully connected neural networks | 0.70 | 0.90 | 0.83 | 0.79 | 0.84 | 0.84 |
| Convolutional neural networks | **0.91** | **0.92** | 0.85 | 0.88 | 0.91 | 0.91 |
| Ilastik | 0.84 | 0.91 | **0.88** | **0.90** | 0.90 | 0.90 |
| Segmentation | 0.27 | 0.78 | 0.42 | 0.87 | 0.80 | 0.80 |
| Producer's accuracy | | | | | | |
| Fully connected neural networks | 0.60 | **0.91** | 0.75 | 0.90 | 0.80 | 0.80 |
| Convolutional neural networks | 0.72 | 0.86 | **0.91** | 0.95 | 0.89 | 0.89 |
| Ilastik | **0.72** | 0.91 | 0.85 | **0.97** | **0.90** | **0.90** |
| Segmentation | 0.56 | 0.57 | 0.75 | 0.44 | 0.57 | 0.57 |
| Balanced accuracy | | | | | | |
| Fully connected neural networks | 0.79 | **0.95** | 0.86 | 0.89 | 0.88 | 0.88 |
| Convolutional neural networks | **0.86** | 0.92 | **0.94** | 0.95 | **0.93** | **0.93** |
| Ilastik | 0.85 | 0.95 | 0.91 | **0.96** | 0.93 | 0.93 |
| Segmentation | 0.69 | 0.77 | 0.76 | 0.71 | 0.76 | 0.76 |
surrounding vegetation. In the set of photographs we used, cheatgrass was redder than the surrounding vegetation. Classification becomes more difficult when the density of the cheatgrass within a patch is quite low or when it is shaded. The CNN workflow tended to increase the size of cheatgrass patches slightly and it missed some of the fine linear objects, such as litter, but in general its output was less noisy (less speckled) than that of the FCN and ilastik workflows.

### Discussion

A goal of automated, ultra-high-resolution mapping in the visible spectrum is to match or exceed the accuracy with which a trained human can identify and label land-cover objects, and to do so faster and more objectively. Our research indicates that automated classification of aerial photographs acquired from low flying heights with consumer UAVs and cameras remains difficult. Although much progress has been made in leveraging deep-learning algorithms to locate and identify or label features in an image, ultra-high-resolution image classification is not yet widely accessible and applicable to classification of cheatgrass, shrubs, other grasses and forbs in sagebrush shrub-steppe.

Our accuracy metrics (Tables 1–4) were generated with a pixel-based approach, which usually penalizes workflows that generalize an object rather than classify each pixel as the sub-object that it most closely resembles. The segmentation workflow tends to generalize the most, followed by the CNN and FCN workflows. The ilastik workflow tends to generalize the least.

Few existing methods for assessing the accuracy of coarse-resolution land-cover classifications (Congalton and Green 2008) are applicable to ultra-high-resolution imagery (Persello and Bruzzone 2010). Other methods for selection of validation data better represent objects, but often are more subjective, especially in landscapes with cover-class gradients. Most CNNs that classify land-cover test model accuracy on the basis of reference images and other data developed for this purpose (Long et al. 2014; Carranza-García et al. 2019). Most of those test data are from urban areas with clear delineations between land-cover objects (Volpi and Tuia 2017; Sameen et al. 2018). In undeveloped areas, by contrast, clear boundaries between cover types are uncommon. It is much more difficult to define object boundaries when the transition between objects is a gradient rather than a sharp edge. There is considerable opportunity for improving accuracy assessment protocols for land-cover maps that are based on ultra-high-resolution imagery.

The accuracy of the segmentation output was lower than that of the other workflows. In addition to the accuracy assessment challenges noted above, this low accuracy probably is a consequence of our use of the same segmentation parameters in classification of images from all flying heights. It is likely that the accuracy of the segmentation workflow would improve if the segmentation algorithm parameters (the way in which objects were defined) were better tuned to each flying height. However, even with improved parameterization, our tests (not reported) indicated that the placement of segment borders was not straightforward in cases where cheatgrass merged gradually into another cover type. The gradients between some cover types become more gradual (less stark) as flying height increases, making it more difficult to segment individual cover types. Our focal cover type, cheatgrass, can occur in relatively dense monocultures, but also co-occurs with other grasses and forbs. The latter can make it quite difficult to classify cheatgrass accurately, even with visual interpretation. Development of a workflow for ground-truthing and classifying these mixed and gradient cover types is ongoing.

For our application, accuracy assessment values for the ilastik workflow were the greatest. The overall accuracy of the ilastik and CNN workflows increased as flying height increased, whereas the FCN workflow’s overall accuracy was stable among flying heights. Overall accuracy for the segmentation workflow decreased as flying height increased. We did not expect the latter result but, as noted above, it could be due to our use of the same segmentation parameters for all flying heights.

Many of the accuracy metrics were very similar, especially those associated with the FCN and CNN workflows. Any of the other three workflows may prove to be the most accurate for other projects, and we strongly recommended that different workflows be evaluated for a particular application.

| Table 4. Overall accuracy of four classification workflows. |
|------------------------------------------------------------|
| Overall accuracy                                           |
| 10 m                                                       |
| Fully connected neural networks 0.82                       |
| Convolutional neural networks 0.79                         |
| ilastik 0.82                                               |
| Segmentation 0.65                                          |
| 45 m                                                       |
| Fully connected neural networks 0.81                       |
| Convolutional neural networks 0.84                         |
| ilastik 0.87                                               |
| Segmentation 0.62                                          |
| 90 m                                                       |
| Fully connected neural networks 0.82                       |
| Convolutional neural networks 0.89                         |
| ilastik 0.89                                               |
| Segmentation 0.56                                          |
Figure 2. Classification results for image subsets. AGL, above ground level. FCN, fully connected neural network. CNN, convolutional neural network.
We used single RGB photographs and created models for each flying height to predict land-cover classes. Adding additional spectral bands to the RGB image, including topographic layers derived from lidar, or using structure from motion can improve classification accuracy (Fraser et al. 2016; Sankey et al. 2018), but those data were not available for our study area.

We expect that the methods presented in this paper could be used to predict land cover in other images acquired during the same flight mission provided that the spectral qualities of each cover type are consistent. Models that incorporate extensive training data will make it practical to classify aerial photographs acquired over other areas in which land cover is similar but other environmental conditions, such as terrain and solar angles, are different.

Improvement of methods to select and evaluate model training data also will improve classification accuracy. Additionally, a more deliberate and objective method for tuning hyperparameters and developing network layouts substantially would advance the use of neural networks. Currently, one must test thousands of models with different sets of hyperparameters and then evaluate the results. Development of hybrid or collaborative workflows also would be worthwhile. For example, CNNs (Radovic et al. 2017) initially could be applied to an image to locate or delineate discrete objects, such as shrubs, before running a wall-to-wall land-cover classification. Moreover, it would be helpful to develop an open platform and protocol for systematically comparing different workflows for classification of land-cover from images that are challenging, such as low-altitude aerial photographs.

Many of the most recent algorithms for identifying and labeling objects in an image are available as open-source software. Much of this software is distributed as libraries instead of high-level applications. However, the open-source geospatial community aims to make state-of-the-art image-processing capabilities more accessible to people whose programming experience is insufficient to use existing software libraries. New approaches to mapping land cover and innovations for improving land-cover classification are advancing rapidly. To improve access to and effective use of machine-learning image-processing methods, the AMNH developed an online portal for development, testing and comparison of new image-processing algorithms and workflows (http://biodiversityinformatics.amnh.org/ml4conservation/ml4landcover/).

Acknowledgments

This work was supported by Joint Fire Sciences Program (15-1-03-6) and by the US Geological Survey’s Northwest and Southwest Climate Science Centers via the US Fish and Wildlife Service (F16AC00025). Thanks to C. Beck, I.R. Bruen-Morningstar, J.M. DeMarines, L. Goncel, H. Herlyn, J.M. Tout and I.B. Wohlfteil-Zillig for field assistance and to B.A. Bradley, D.S. Dobkin and M. Leu for their conceptual and logistic contributions to this work. We also thank three anonymous reviewers for their constructive comments.

References

Balch, J. K., B. A. Bradley, C. M. D’Antonio, and J. Gomez-Dans. 2013. Introduced annual grass increases regional fire activity across the arid western USA (1980–2009). Glob. Change Biol. 19, 173–183.

Bradley, B. A., C. A. Curtis, E. J. Fusco, J. T. Abatzoglou, J. K. Balch, S. Dadashi, et al. 2018. Cheatgrass (Bromus tectorum) distribution in the Intermountain Western United States and its relationship to fire frequency, seasonality, and ignitions. Biol. Invasions 20, 1493–1506.

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

Calvo, K., and T. E. Lovejoy. 2018. Drones for conservation - field guide for photographers, researchers, conservationists and archaeologists. Blurb, Incorporated, San Francisco, CA.

Carranza-García, M., J. García-Gutiérrez, and J. C. Riquelme. 2019. A framework for evaluating land use and land cover classification using convolutional neural networks. Remote Sens. 11, 274.

Comaniciu, D., and P. Meer. 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 603–619.

Congalton, R. G., and K. Green. 2008. Assessing the accuracy of remotely sensed data: principles and practices, 2nd ed.. CRC Press, Bocca Raton, FL.

Feng, W., H. Sui, W. Huang, C. Xu, and K. An. 2018. Water body extraction from very high-resolution remote sensing imagery using deep U-Net and a superpixel-based conditional random field model. IEEE Geosci. Remote Sens. Lett. 16, 618–622.

Fraser, R. H., I. Olthof, T. C. Lantz, and C. Schmitt. 2016. UAV photogrammetry for mapping vegetation in the low-Arctic. Arctic Sci. 2, 79–102.

Ghorbani, A., and M. Pakravan. 2013. Land use mapping using visual vs. digital image interpretation of TM and Google Earth derived imagery in Shirvan-Darasi watershed (northwest of Iran). Eur. J. Exp. Biol. 3, 576–582.

Grippa, T., M. Lennert, B. Beaumont, S. Vanhuysse, N. Stephenne, and E. Wolff. 2017. An open-source semi-automated processing chain for urban object-based classification. Remote Sens. 9, 358.

Grisonnet, M., J. Michel, V. Poughon, J. Inglanda, M. Savinaud, and R. Cresson. 2017. Orfeo Toolbox: open source processing of remote sensing images. Open Geospat. Data Soft. Stand. 2, 15.
Ultra-High-Resolution Land-Cover Mapping

Horning, N. 2018. Remotely piloted aircraft system applications in conservation and ecology. Remote Sens. Ecol. Conserv. 4, 5–6.

Hsieh, Y.-T., C.-T. Chen, and J.-C. Chen. 2017. Applying object-based image analysis and knowledge-based classification to ADS-40 digital aerial photographs to facilitate complex forest land cover classification. J. Appl. Remote Sens. 11, 015001.

Ilastik. (2019). Available from https://www.ilastik.org/(accessed March 18, 2019).

Jiménez López, J., and M. Mulero-Pázmány. 2019. Drones for conservation in protected areas: present and future. Drones 3, 10.

Kalantar, B., S. B. Mansor, M. I. Sameen, B. Pradhan, and H. M. Z. Shafri. 2017. Drone-based land-cover mapping using a fuzzy unordered rule induction algorithm integrated into object-based image analysis. Int. J. Remote Sens. 38, 2535–2556.

Leitão, P. J., M. Schwieder, F. Pöttschger, J. R. R. Pinto, A. Teixeira, F. Pedroni, et al. 2018. From sample to pixel: multi-scale remote sensing data for upsampling aboveground carbon data in heterogeneous landscapes. Ecosphere 9, e02298.

Liang, P., W. Shi, and X. Zhang. 2018. Remote sensing image classification based on stacked denoising autoencoder. Remote Sens. 10, 116.

Liaw, A., and M. C. Wiener. 2007. Classification and regression by randomForest. R. News 2, 18–22.

Long, J., E. Shelhamer, and T. Darrell. 2014. Fully convolutional networks for semantic segmentation. arXiv:1411.4038 [cs]. Available from http://arxiv.org/abs/1411.4038(accessed February 25, 2019).

Michel, J., D. Youssefi, and M. Grizzonnet. 2015. Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images. IEEE Trans. Geosci. Remote Sens. 53, 952–964.

Pande-Chhetri, R., A. Abd-EIrahman, T. Liu, J. Morton, and V. L. Wilhelm. 2017. Object-based classification of wetland vegetation using very high-resolution unmanned air system imagery. J. Remote Sens. 50, 564–576.

Persello, C., and L. Bruzzone. 2010. A novel protocol for accuracy assessment in classification of very high resolution images. IEEE Trans. Geosci. Remote Sens. 48, 1232–1244.

QGIS Development Team. 2019. QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org.

R Development Team. 2008. R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Radovic, M., O. Adarkwa, and Q. Wang. 2017. Object recognition in aerial images using convolutional neural networks. J. Imaging 3, 21.

Rodriguez-Galiano, V. F., B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez. 2012. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS J. Photogramm. Remote Sens. 67, 93–104.

Sameen, M. I., B. Pradhan, and O. S. Aziz. 2018. Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks. Available from https://www.hindawi.com/journals/js/2018/7195432/ (accessed February 26, 2019).

Sankey, T. T., J. McVay, T. L. Swetnam, M. P. McClaran, P. Heilman, and M. Nichols. 2018. UAV hyperspectral and lidar data and their fusion for arid and semi-arid land vegetation monitoring. Rem. Sens. Ecol. Conserv. 4, 20–33.

Shen, G., and A. Sarris. 2008. Application of texture analysis in land cover classification of high resolution image. Pp. 513–517 in J. Ma, Y. Yin, J. Yu and S. Zhou, eds. 2008 IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery, October 18–20, 2008. Institute of Electrical and Electronics Engineers (IEEE), Changdng, China.

Singh, K. K., and A. E. Frazier. 2018. A meta-analysis and review of unmanned aircraft system (UAS) imagery for terrestrial applications. Int. J. Remote Sens. 39, 5078–5098.

Sommer, C., C. Straehele, U. Köthe, and F. A. Hamprecht. 2011. Ilastik: interactive learning and segmentation toolkit. Pp. 230–233 in S. Wright, X. Pan and M. Liebling, eds. Proceedings of 2011 IEEE international symposium on biomedical imaging: from nano to macro, March 30–April 2, 2011. Institute of Electrical and Electronics Engineers (IEEE), Chicago, IL.

Story, M., and R. G. Congalton. 1986. Accuracy assessment: a user’s perspective. Photogramm Eng Remote Sensing 52, 397–399.

Strahler, A., C. E. Woodcock, and J. A. Smith. 1986. On the nature of models in remote sensing. Remote Sens. Environ. 20, 121–139.

Velez, D. R., B. C. White, A. A. Motsinger, W. S. Bush, M. D. Ritchie, S. M. Williams, et al. 2007. A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. Genet. Epidemiol. 31, 306–315.

Volpi, M., and D. Tuia. 2017. Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55, 881–893.

Wang, Y., D. Ryu, S. Park, S. Fuentes, and M. O’Connell. 2017. Upscaling UAV-borne high-resolution vegetation index to satellite resolutions over a vineyard. 2nd International Congress on Modelling and Simulation, Hobart, 3 to 8 December 2017. Tasmania, Australia.

Wich, S., and L. P. Koh. 2018. Conservation drones: mapping and monitoring biodiversity. Oxford University Press, Oxford, UK.
Yu, Q., P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer. 2006. Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. *Photogramm. Eng. Remote Sens.* 72, 799–811.

Zhang, J., B. Rivard, A. Sánchez-Azofeifa, and K. Castro-Esau. 2006. Intra- and inter-class spectral variability of tropical tree species at La Selva, Costa Rica: implications for species identification using HYDICE imagery. *Remote Sens. Environ.* 105, 129–141.

**Supporting Information**

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supplemental material URL (temporary until MS accepted then transferred to AMNH server): https://resources.cbc.amnh.org/supplemental_material/doi/10.1002/rse2.144/