Received: 19 February 2020 | Accepted: 3 August 2020

DOI: 10.1111/2041-210X.13473

RESEARCH ARTICLE

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

Detecting plant species in the field with deep learning and drone technology

Katherine James | Karen Bradshaw

Department of Computer Science, Rhodes University, Grahamstown, South Africa

Correspondence
Katherine James
Email: katherine.mf.james@gmail.com

Funding information
Rhodes University

Handling Editor: Sean McMahon

Abstract

1. Aerial drones are providing a new source of high-resolution imagery for mapping of plant species of interest, amongst other applications. On-board detection algorithms could open the door to allow for applications in which drones can intelligently interact with their environment. However, the majority of plant detection studies have focused on detection in post-flight processed orthomosaics. Greater research into developing detection algorithms robust to real-world variations in environmental conditions is necessary, such that they are suitable for deployment in the field under variable conditions.
2. We outline the steps necessary to develop such a system, show by example how real-world considerations can be addressed during model training and briefly illustrate the performance of our best performing model in the field when integrated with an aerial drone.
3. Our results show that introducing variations in brightness as an additional augmentation strategy during training is beneficial when dealing with real-life data. We achieved a 27% improvement in the F1-score obtained on the unseen test set when using this approach. Further improvements to the model performance were obtained through the use of weight map-based loss, accounting for uncertainty in the annotation masks due to the indistinct nature of the edges of the target plants using weighting. This resulted in a 15% improvement in precision for the best configuration of hyper-parameters, yielding a final model with an F1-score of 83% and accuracy of 96%. Finally, results computed on the fly show that such a system is deployable in the field.
4. This study shows that it is possible for a commercially available drone, integrated with a deep learning model, to detect invasive plants in the field and demonstrates methodology which could be applied to developing similar systems for other plant species of interest. The ability to perform detection on the fly is necessary for future applications in which intelligent interaction between a drone and its environment is required.

KEYWORDS

aerial imagery, deep learning, drones, *Hakea*, invasive alien plants, segmentation, U-Net, unmanned aerial vehicle

1 | INTRODUCTION

High-resolution aerial imagery is frequently used for studies in which particular plant species of interest must be identified. Examples of such studies include invasive alien plant detection for mapping and monitoring (Bradley, 2014; Huang & Asner, 2009), detection of flowering plants (de Sá et al., 2018), applications in precision agriculture such as weed mapping, crop coverage estimation or crop health inspection (Pajares, 2015) and censuses of endangered plants in fragile habitats (Rominger & Meyer, 2019).

In these studies, plants are detected in digital images, either manually or using automated analysis techniques (Dash, Watt, Paul, Morgenroth, & Hartley, 2019). Unmanned aerial vehicles, commonly referred to as aerial 'drones', have provided a more readily available source of high-resolution imagery and object-based image analysis (OBIA) has become the dominant analysis technique for detection of targets in this imagery (Blaschke, 2010). However, pixel-based image analysis may still be preferable in some scenarios (Dvořák, Müllerová, Bartaloš, & Brůna, 2015). Machine learning classifiers, such as random forest and support vector machines, are well established for automatic classification in both analysis paradigms (Belgiu & Drăguţ, 2016). These classifiers are trained to identify which of a set of manually selected features, extracted from objects or pixels, are important for distinguishing between classes.

Some studies have, however, begun to consider deep learning as an alternative to OBIA for classification (Guirado, Tabik, Alcaraz-Segura, Cabello, & Herrera, 2017), which has been shown to be more successful provided sufficient training data is available (Liu, Abd-Elrahman, Jon, & Wilhelm, 2018). Deep learning is a machine learning approach whereby a model is trained by supervised example, given both the input and the expected output, such that it learns distinguishing features from the data, rather than these features being manually selected (Zhang, Zhang, & Du, 2016). Deep learning has begun to be used widely in remote sensing (Zhang et al., 2016), though applications of particular interest in this study are the use of deep learning for woody vegetation mapping (Flood, Watson, & Collett, 2019) and in agricultural applications, where Kamilaris and Prenafeta-Boldu (2018) observed that deep learning methods almost always outperform other image processing techniques.

The predominant workflow for the production of distribution maps, after the collection of drone sourced high-resolution imagery, currently involves the creation of an orthomosaic and the subsequent application of OBIA with a machine learning classifier for automated classification (Alvarez-Taboada, Paredes, & Julián-Pelaz, 2017; Chabot, Dillon, Shemrock, Weissflog, & Sager, 2018; Mafanya et al., 2017; Martin et al., 2018; Wu et al., 2019). However, the intensive processing necessary to produce mosaics acts as a bottleneck, restricting further research (Dash et al., 2019). The end product of such analysis is a map showing the prevalence, distribution and density of the species of interest, which can help support land management decisions and monitoring of populations.

These maps can also be used to guide subsequent interactions between a drone and the environment, such as pre-defined point-to-point delivery of bio-control to stressed regions of crop-fields (Rangel, 2016) and the spraying of aquatic weeds at human-defined spray points (Göktogan et al., 2010). While these studies demonstrate some of the types of interactions drones can have with their environments, more complex in-field applications are restricted due to the off-board computation of plant targets of interest. On-board computation and identification of targets while in flight would allow for immediate interaction with the environment, for example flying closer to an identified target for the purpose of closer inspection, taking a high-resolution photograph or spraying weeds with herbicide, as demonstrated by Alsalam, Morton, Campbell, and Gonzalez (2017). In this study, an OBIA-based colour thresholding technique was used to detect spear thistle weed in a grassy field. The authors noted the detection component of their system could be improved upon using deep learning.

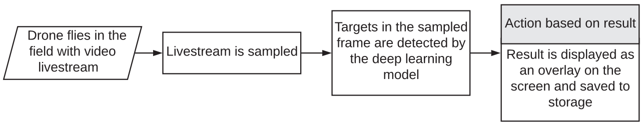

Drone flies in the field with video livestream
Livestream is sampled
Targets in the sampled frame are detected by the deep learning model
Action based on result
Result is displayed as an overlay on the screen and saved to storage

FIGURE 1 A high-level overview demonstrating how detection may be performed in the field

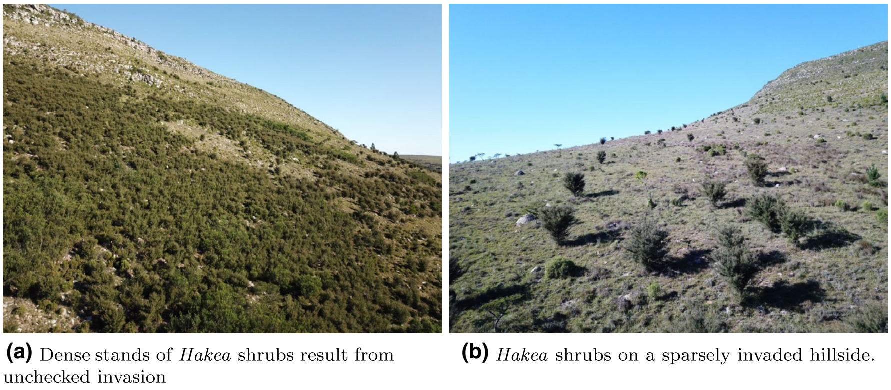
Although drones provide a useful remote sensing platform, we contend that many more applications in which interaction with the environment is required could be facilitated by the development of an intelligent drone system capable of performing detection while in flight. The development and demonstration of a suitable detection approach is a prerequisite for such applications.

The aim of this paper is to determine whether commercially available quadcopters, when integrated with a deep learning model, can be used to assist on the fly detection of plant species of interest, as illustrated in Figure 1. We outline the necessary steps for developing such a system (Section 2), show the necessity of data augmentation to address fluctuations due to environmental factors (Section 3.1), illustrate loss function hyper-parameter tuning (Section 3.2), and briefly demonstrate the integrated system (Section 3.3), as well as outlining challenges encountered (Section 4). To demonstrate the methodology, a case study was performed for a particular genus of shrub, which is invasive in South Africa. The action taken by the system based on the detection result was to simply display the detection result, although in future applications this action could be replaced by a sub-routine to perform an interaction with the environment.

2 | MATERIALS AND METHODS

2.1 | Target plant

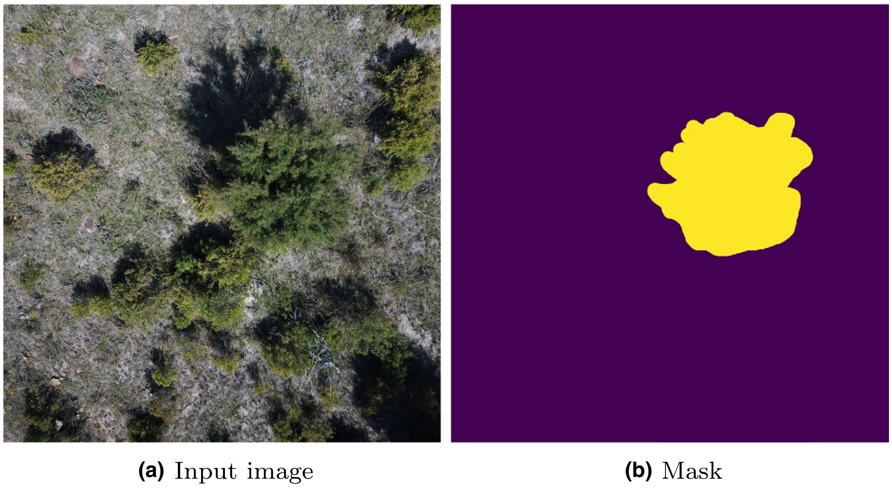
Shrubs of the *Hakea* genus are indigenous to Australia, and are one of the most prevalent woody invaders in South Africa's fynbos biome (Richardson, van Wilgen, & Mitchell, 1987). The shrub has needle-like leaves and grows between 3 and 5 m high (Kluge & Neser, 1991). Unlike indigenous members of the Protacea family, shrubs of the *Hakea* genus do not suffer from seed predation, resulting in the germination of large seed banks after fire and the formation of dense stands, as seen in



(a) Dense stands of *Hakea* shrubs result from unchecked invasion

(b) *Hakea* shrubs on a sparsely invaded hillside.

FIGURE 2 Typical invasion of *Hakea* on a grassy slope with other scattered shrubs



(a) Input image

(b) Mask

FIGURE 3 An example of semantic annotation on an image from the dataset, where a binary mask is produced by an annotator as a ground truth, separating the image into regions belonging to the two classes. In this image, yellow depicts an object of the target class while purple indicates the background class

Figure 2a, which alter the composition of the ecosystem and crowd out indigenous fynbos (van Wilgen & Richardson, 1985). This genus of shrub was selected as the target shrub for this research.

2.2 | Dataset collection and annotation

A DJI Mavic Pro,[1] flown with its camera pointing vertically downwards, was used to collect the dataset. This is a small quadcopter measuring just under 20 cm in its largest dimension, with a maximum flight time of 27 min and maximum flight distance of 13 km under ideal conditions. It carries a gimbal mounted camera of 12 MP for photographs, up to 4 K resolution for stored videos and 2,048 × 1,488 pixels for the sampled livestream. The drone was flown using the DJI GO app, which ran on an Asus ZenPad3s10[2] tablet, with a hexa-core processor and 4 GB of RAM.

Imagery of the target shrub was collected from a farm southwest of Grahamstown, in the Eastern Cape, South Africa. The site falls within a hilly area, where two species of *Hakea* have been documented to have invaded, namely *Hakea suaveolens* and *Hakea sericea* (Palmer, 2004). Data collection was performed twice, with approximately a month's separation and was conducted in regions where other shrubs were also present, such as shown in Figure 2b. The first dataset collected contains 52 images, each with at least one target shrub present, while the second contains 48 images similarly with at least one target shrub. Due to collection on different days, there is natural environmental variation captured in the datasets, both in terms of illumination and appearance of the vegetation. The datasets were each annotated semantically into two classes, foreground (the target plant), and background (all other landcover), as shown in Figure 3.

2.3 | Deep learning architecture

A particularly successful deep learning algorithm when it comes to image analysis is the convolutional neural network (CNN), which takes images an inputs. This architecture consists of hierarchical layers, consisting of artificial neurons, which perform different operations on the data to extract features (convolutional layers), apply mathematical transformations necessary for training (non-linearity layers) and reduce dimensionality (pooling layers). Convolutional layers map inputs to feature maps through the application of a convolution operator between the input and a small two-dimensional matrix called a filter. Feature maps are simply the output of this operation and shrink through the network as a result of filter sampling. Each filter has weights which during supervised training are updated such that the network learns to extract relevant discriminant features

[1] https://www.dji.com/uk/mavic/info#specs

[2] https://www.asus.com/us/Tablets/ASUS-ZenPad-3S-10-Z500M/

from the data (see Zhang et al., 2016). Early features are likely to be simple ones, like edges, blobs and colour (Nogueira, Penatti, & dos Santos, 2016), while features extracted in deeper layers of the architecture are more abstract. Typically, these extracted features are then mapped through a final few layers to a classification result, indicating the likelihood of the input belonging to a particular class. This result is compared to the ground truth result and a measure of error called the 'loss' of the network is then computed to guide how the weights are updated.

A variant of the CNN takes this process a step further, and instead of predicting a classification for an entire input image, predictions are made for each pixel in the image. This is called image segmentation, as opposed to image classification. Instead of the loss being computed for a single class label as in classification, the loss in image segmentation must now be computed across all pixels in the image. For this reason, it is necessary to provide annotated masks showing the desired segmentation result as the ground truth, corresponding to the training samples.

A particular image segmentation architecture that has risen to attention is the U-Net (Ronneberger, Fischer, & Brox, 2015). The U-Net architecture combines a regular CNN with subsequent additional layers, which up-sample the feature maps to increase their resolution again over successive layers. These two components of the network are thus referred to as the 'contracting' and 'expanding' parts, due to the behaviour of the feature map size across them. The increase in resolution in the expanding part is performed by combining features from the corresponding layers in the contracting part of the network. Through this process, localisation is thus performed, and a final layer in the network maps the resulting feature map to class predictions to produce the final segmentation. For more detail, see (Ronneberger et al., 2015).

The U-Net was designed to utilise small datasets for training and to produce segmentation outputs in a short run-time. This architecture relies on a high level of data augmentation, which allows the network to develop robustness against deformations in the images.

2.4 | Loss functions

A core aspect of training a deep learning model is the loss function, which computes a measure of error between the predicted output and the annotation mask, which is the ground truth. A simple and commonly used loss function is the cross-entropy (CE) loss, which is defined in Equation1, where N is the total number of pixels and C is the number of classes. The CE loss is computed for each pixel in the image as the product of the ground truth label $g_{nc} \in G$ and the log of the predicted probability $p_{nc} \in P$ of that pixel belonging to class c. This is averaged across all pixels to produce the loss for the image. A small epsilon value is used in the implementation of the loss to avoid infinite loss in the case where $p_{nc} = 0$.

$$L_{\mathrm{CE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} g_{nc}\log(p_{nc}). \tag{1}$$

We address a two-class, or binary, segmentation problem, where pixels belonging to the background and target classes have values 0 and 1 respectively. The loss, in this case, is referred to as binary cross-entropy (BCE) loss.

When considering that BCE is calculated across a batch as an average, it is clear that the dominant background class may dominate the contribution to the total loss. A possible way of reducing this effect is to make use of a weighted BCE (WBCE) loss, in which each class' contribution is weighted by the inverse frequency of its occurrence. This is seen in Equation 2.

$$L_{\mathrm{WBCE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \omega_c g_{nc}\log(p_{nc}). \tag{2}$$

In the case of our binary problem,

$$\omega_c = \begin{cases} 0.08 & \text{if background class} \\ 0.92 & \text{if target class} \end{cases}.$$

Furthermore, in the case where objects have indistinct edges, the use of a pre-computed weight map to reduce the contribution of edge pixels with low annotation confidence can be used (James & Bradshaw, 2019). This is defined as in Equation 3.

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \omega_{\mathrm{map}} g_{nc}\log(p_{nc}), \tag{3}$$

where ω_{map} is defined with two additional hyper-parameters α and β as,

$$\omega_{\mathrm{map}} = \begin{cases} \alpha & \text{if pixel falls within edge of thickness } t \\ \beta & \text{otherwise} \end{cases}.$$

The inverse class frequency weighted variant of this loss function was used, defined in Equation 4, where ω_c has the same values as defined for WBCE.

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \omega_{\mathrm{map}}\omega_c g_{nc}\log(p_{nc}). \tag{4}$$

2.5 | Training scheme

When training a deep learning architecture, it is important that the weights in the network are initialised such that they do not skew the training process. Additionally, the loss function and optimiser must be chosen and associated hyper-parameters must be set. The optimiser controls the optimisation process of finding weights which give the minimum loss.

The hyper-parameters and initialisations chosen for the model were those given in the original U-Net paper (Ronneberger et al., 2015). Weights were drawn from a Gaussian distribution with a standard deviation of $\sqrt{2/N}$, where N is the number of inputs to a neuron. The optimiser used was stochastic gradient descent with momentum

0.99. High momentum is used when the batch size is small. In our study, a batch size of 2 was used, which means that the loss is computed across two images at a time as an average, but the high momentum value ensures that many training samples are considered when updating the weights of the network.

The number of filters to apply to the first layer must also be specified as a hyper-parameter, which is continually doubled or halved in the respective contracting and expanding parts of the network. This hyperparameter determines how many feature extractors are used at each layer, and was set to 16. A process called drop-out is used to prevent over-fitting, in which some outputs of the layer are randomly ignored during training. This hyper-parameter was set to 0.05, which means that at this layer of the network, 5% of the input units are ignored during training. These input units are the 'pixels' from the feature map of the previous layer. It is left for future work to tune these hyper-parameters.

One of the strengths of the U-Net is its low requirement for the number of training samples, instead relying on heavy data augmentation. It is very common to use data augmentation when training CNNs, as this process allows for an artificial increase in training data, additionally ensuring that the network becomes invariant to the orientation, position and scale of the objects of interest. This process occurs only during training, and the transformation is applied to the entire image. To achieve sufficient data augmentation, functions to introduce 187 flips, shifts, rotations and scale were used. The tolerance learnt by the classifier from such transformations 188 during the augmentation process is particularly necessary for remotely sensed data (Li et al., 2018).

First the use of brightness as an additional augmentation strategy was assessed, in which two models were trained using WBCE loss. The first was trained using only the set of data augmentation functions already mentioned, while the other was trained using brightness augmentation in addition to these. Important metrics for the comparison of models are the precision, recall, F1 score and accuracy. The F1 score, defined in Equation 5, is a performance metric which provides a balance between recall and precision scores.

$$\mathrm{F1} = \frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \tag{5}$$

Thereafter, we show how the model may be improved through the use of weight map-based loss (WMB), defined in Equation 4. A series of different combinations of the hyper-parameters α, β and t were used to train a set of models, specifically, each weight combination w of set $W = \{1, 0; 1, 0.2; 1, 0.5; 2, 0.5; 5, 1\}$ along with each thickness t of set $T = \{1;3;5\}$. This allowed for manual tuning of the hyper-parameters. A more comprehensive approach using Bayesian optimisation may be used to tune hyper-parameters further (Yu & Zhu, 2020), but this is left for future work. From this trained set of models, the most robust model was selected and integrated with the drone via a custom Android application, as discussed in Section 2.6.

All models were first trained using 10-fold validation, using dataset 1, reserving different portions as the validation set each time so as to gain a general overview of performance that was not validation set dependent. Subsequently, models were trained using the whole of dataset 1 to obtain a final model and thereafter tested on the unseen test set, dataset 2. Henceforth, dataset 1 will be referred to as the 'training' set and dataset 2 as the 'test' set. Models were trained using TensorFlow[3] and Keras.[4]

2.6 | Integration of model with drone

DJI provides a mobile software development kit (SDK) for Android and iOS to allow developers to make use of user interface (UI) elements for the core functionalities available in DJI GO. This allows the developer to focus on the high-level functionality of the application, building on a mini version of the DJI GO app, rather than focusing on low-level control. The SDK provides access to flight control, telemetry and sensor data, access to the on-board camera and gimbal control, as well as the live video stream and additionally control of pre-defined mission functions. The custom Android application was developed using the DJI SDK UXSDKDEMO[5] application as a basis.

A process called 'freezing' converts the saved, trained model weights and model architecture and replaces all the variables with constant values, as well as removing any nodes not used for forward inference. The resulting model is saved as a protobuf file, which is deployed as part of the custom Android application. The TensorFlow library was integrated with the SDK code to facilitate this deployment.

The Open Source Computer Vision Library[6] (OpenCV) provides support for building real-time machine vision and machine learning applications. A vast library of computationally efficient algorithms provides functionality for these applications across C++, Python, Java and MATLAB platforms. This library allowed the necessary functionality for preprocessing the images within the application prior to segmentation.

The trained model was integrated to interact with the drone as demonstrated in Figure 4.

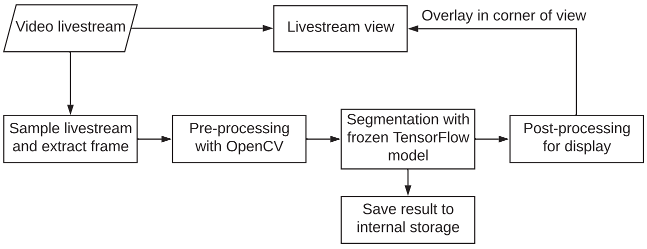

Video livestream
Livestream view
Overlay in corner of view
Sample livestream and extract frame
Pre-processing with OpenCV
Segmentation with frozen TensorFlow model
Post-processing for display
Save result to internal storage

FIGURE 4 The livestream provided to the application through the DJI SDK was sampled and the image preprocessed using OpenCV. The segmentation result from the frozen TensorFlow model was then produced and post-processed for display as an overlay in the corner of the livestream view. In addition, the segmentation result was saved to internal storage for later perusal

[3] https://www.tensorflow.org/

[4] https://keras.io/

[5] https://github.com/DJI-Mobile-SDK-Tutorials/Android-UXSDKDemo

[6] https://opencv.org/about.html

3 | RESULTS

Quantitative and qualitative results showing how two real world considerations, namely variation in brightness and uncertainty in edge annotation, may be addressed as part of the training process are presented in the first two subsections. Thereafter, a demonstration of the system when deployed in the field and results thereof are given.

3.1 | Consideration 1 – Brightness variations

When considering the 10-fold results in Table 1, it is seen that the addition of brightness augmentation results in an improvement in all metrics. This suggests that brightness augmentation assists the model in learning to make fewer false positive predictions and more correct ones, possibly by allowing the model to better cope with the slight fluctuations that exist in the training set.

When trained on the entirety of the training set and then applied to the unseen test set, we see that while similar results to those from 10-fold are achieved by the model augmented using brightness variations, the baseline performance changes (Table 2). The test set contains images that are slightly darker than the training set due to the conditions on the day of collection and the consistent performance of the augmented model shows that this approach is beneficial and produces a more robust model.

Qualitative assessment of the two models visualised in Figure 5, one trained with brightness augmentation and the other without, confirms that brightness augmentation is beneficial to model performance. With augmentation, a far better precision rate is seen and although numerous false positives are still detected, these are restricted to other shrubs rather than other landcover to a greater degree and shadows are no longer misclassified as part of the target plants.

3.2 | Consideration 2 – Uncertainty in annotation and hyper-parameter tuning

In this section, we evaluate models trained using WMB loss, using different hyper-parameter configurations and make a comparison to the WBCE loss. A distinction is made between WBCE loss and WMB loss. WBCE loss only makes use of class weights that adjust the contribution that the pixels of the target and other class each make, whereas WMB loss additionally makes use of a weight map to weight individual pixels based on the confidence of their annotation in the image.

There is minimal variation in performance between the different hyper-parameter configurations in Table 3, with all models achieving between 71% and 78% in F1-score. If we consider the WBCE as a baseline, we observe a 1% average improvement in F1-score by both the $w = 1, 0.2$ $t = 3$ and $w = 1, 0$ $t = 5$ models, which additionally have a slightly lower standard deviation than WBCE.

Substantially more fluctuation in performance is visible when considering the test results in Table 4, with a range in F1 score of 0.49 to 0.85. A number of models performed better than WBCE on the test set, with the top F1 score using WMB loss beating that of WBCE loss by 5%. This suggests that WMB loss can result in models which generalise better than models trained with equal contributions from edge pixels and others.

We select as the final configuration of the model that which obtains the highest accuracy and highest F1-score, this being WMB loss with the configuration w = 2, 0.5 t = 5. This model has good recall and reasonable precision. A selection of qualitative results are shown in Figure 6, where it is seen that fewer false positives are predicted by this configuration of WMB loss than by the WBCE loss (see Figure 5 for input images and masks).

3.3 | In-field results

The system was trialled in the field and the model was able to successfully segment the image while the drone was in flight, to produce an output image which was both displayed on the screen (see Figure 7) and saved to internal storage on the device.

Other segmentation results computed in the field are seen in Figure 8, along with their corresponding input images, both of which were saved to internal storage. For these ten images, a high recall rate is present, with only seven of the 28 target plants missed, yielding a recall of 75%. Unfortunately a high number of false positives were also predicted, with the majority of these being other shrubs.

The raw output of the network is a probability heatmap in the range [0:1]. In this study, this heatmap was thresholded at 0.5 to separate the output into the two classes. In future work it would be possible to adjust this threshold, such that only predictions of

TABLE 2 Performance of the brightness augmented inference model on the unseen test set

Average	Precision	Recall	F1	Accuracy
Baseline WBCE	0.25	0.93	0.53	0.75
WBCE with brightness augmentation	0.44	0.94	0.80	0.93

TABLE 1 Average metrics with standard deviation over 10-fold validation to assess the effects of brightness augmentation

	Precision	Recall	F1	Accuracy
Baseline WBCE	0.39 ± 0.10	0.78 ± 0.10	0.66 ± 0.09	0.89 ± 0.03
WBCE with brightness augmentation	0.46 ± 0.09	0.92 ± 0.06	0.77 ± 0.06	0.93 ± 0.03

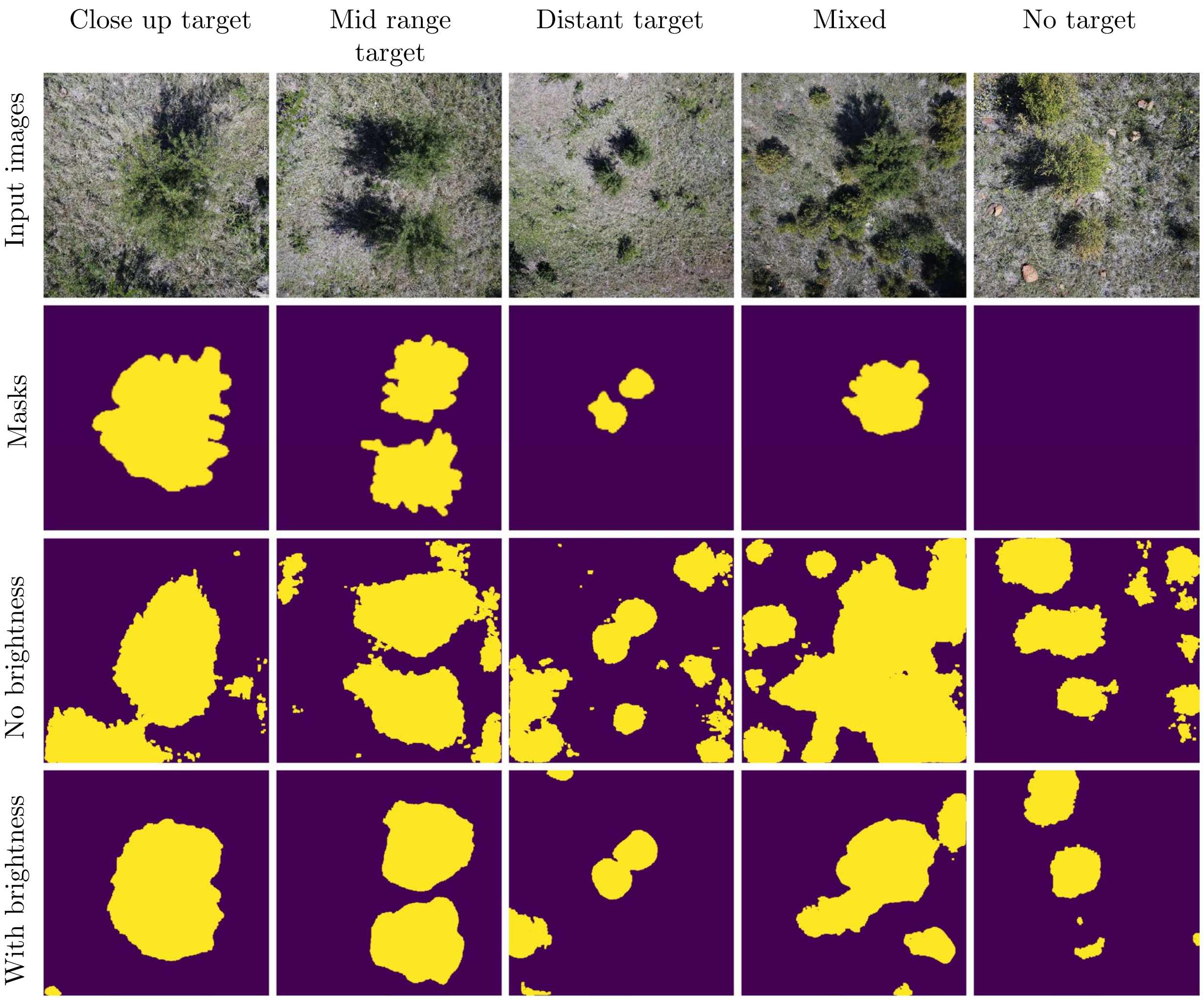


FIGURE 5 A selection of input images for different cases of image composition, along with their binary masks (ground truth) and the predicted segmentation for top-performing models. Used for brightness augmentation comparison

TABLE 3 Average metrics with standard deviation over 10-fold validation to assess the performance of WMB loss, with different hyper-parameter configurations, compared to the baseline WBCE loss

		Precision	Recall	F1	Accuracy
WBCE		0.46 ± 0.09	0.92 ± 0.06	0.77 ± 0.06	0.93 ± 0.03
WMB ($t = 1$)	$w = 1, 0$	0.46 ± 0.12	0.90 ± 0.07	0.76 ± 0.07	0.92 ± 0.04
	$w = 1, 0.2$	0.41 ± 0.11	0.87 ± 0.06	0.71 ± 0.08	0.89 ± 0.04
	$w = 1, 0.5$	0.41 ± 0.09	0.90 ± 0.06	0.73 ± 0.07	0.91 ± 0.03
	$w = 2, 0.5$	0.44 ± 0.11	0.92 ± 0.04	0.77 ± 0.05	0.93 ± 0.02
	$w = 5, 1$	0.44 ± 0.10	0.93 ± 0.04	0.76 ± 0.04	0.92 ± 0.08
WMB ($t = 3$)	$w = 1, 0$	0.43 ± 0.09	0.89 ± 0.06	0.74 ± 0.08	0.92 ± 0.03
	$w = 1, 0.2$	0.46 ± 0.10	0.90 ± 0.07	0.78 ± 0.05	0.93 ± 0.02
	$w = 1, 0.5$	0.44 ± 0.09	0.90 ± 0.06	0.75 ± 0.05	0.92 ± 0.02
	$w = 2, 0.5$	0.43 ± 0.11	0.92 ± 0.05	0.75 ± 0.06	0.92 ± 0.03
	$w = 5, 1$	0.43 ± 0.11	0.93 ± 0.03	0.76 ± 0.08	0.91 ± 0.02
WMB ($t = 5$)	$w = 1, 0$	0.50 ± 0.09	0.86 ± 0.07	0.78 ± 0.03	0.94 ± 0.02
	$w = 1, 0.2$	0.46 ± 0.08	0.88 ± 0.07	0.77 ± 0.06	0.93 ± 0.03
	$w = 1, 0.5$	0.47 ± 0.10	0.89 ± 0.05	0.77 ± 0.06	0.93 ± 0.03
	$w = 2, 0.5$	0.46 ± 0.10	0.90 ± 0.05	0.77 ± 0.06	0.93 ± 0.02
	$w = 5, 1$	0.44 ± 0.14	0.92 ± 0.04	0.76 ± 0.09	0.90 ± 0.04

higher confidence would be predicted to be positive if an application needed to minimize false positives. This may, however, result in a drop in recall.

4 | DISCUSSION

To the best of our knowledge, there has been no previous use of intelligent drones for in-field detection of invasive plants, although there has been substantial research within the remote sensing community in detecting alien vegetation in orthomosaics (Bradley, 2014; Huang & Asner, 2009) and on the fly detection has been done by Alsalam et al. (2017) for detecting spear thistle weed in a grass field using an OBIA approach. The approach taken in our research of classifying or segmenting a single image rather than an orthomosaic is not a common practice in literature pertaining to the detection of invasive plants, although this approach was used by Baron, Hill, and Elmiligi (2018) in their research on the detection of *Iris pseudacorus*. Their study calculated a set of 68 features per pixel from RGB imagery, which were classified in a pixel-based approach using a random forest classifier. In contrast to Baron et al. (2018), the features used in this study were learnt, rather than hand-crafted. This was done through the deep learning approach which also produced the final prediction, as opposed to a pixel-based image analysis approach. The study by Baron et al. (2018) achieved an overall accuracy of 99%, precision of 4.8% and recall of 93%, whereas in our study an accuracy of 96% was achieved on the test set, with an average precision and recall of 59% and 81%, respectively. Although the recall and accuracy achieved in our research are lower, our precision value is considerably higher. In both pieces of research, however, precision is the metric which suffers compared to the others, indicating a likelihood for the detection of false positives, which was manifest during field testing.

TABLE 4 Metrics obtained for trained models trained applied to the unseen test set. Loss functions WBCE and WMB are considered, with different parameter configurations for the latter

		Precision	Recall	F1	Accuracy
WBCE		0.44	0.94	0.80	0.93
WMB ($t = 1$)	$w = 1, 0$	0.55	0.88	0.83	0.95
	$w = 1, 0.2$	0.52	0.88	0.85	0.95
	$w = 1, 0.5$	0.41	0.94	0.77	0.92
	$w = 2, 0.5$	0.47	0.85	0.81	0.93
	$w = 5, 1$	0.65	0.61	0.75	0.95
WMB ($t = 3$)	$w = 1, 0$	0.67	0.68	0.78	0.96
	$w = 1, 0.2$	0.68	0.68	0.78	0.96
	$w = 1, 0.5$	0.34	0.96	0.67	0.87
	$w = 2, 0.5$	0.55	0.85	0.83	0.94
	$w = 5, 1$	0.31	0.86	0.60	0.84
WMB ($t = 5$)	$w = 1, 0$	0.21	0.98	0.49	0.68
	$w = 1, 0.2$	0.58	0.73	0.82	0.96
	$w = 1, 0.5$	0.58	0.84	0.80	0.96
	$w = 2, 0.5$	0.59	0.81	0.83	0.96
	$w = 5, 1$	0.62	0.73	0.81	0.95

The value of positive detections in real-time, both true positives and false positives, depends on the application of the study. True positive detections are inherently useful in a management campaign where, say, bio-control or herbicide is delivered. False positive detections are by their nature undesirable and far less useful, and in applications like those just mentioned, should be minimised. However, if a study were, for example, collecting higher resolution

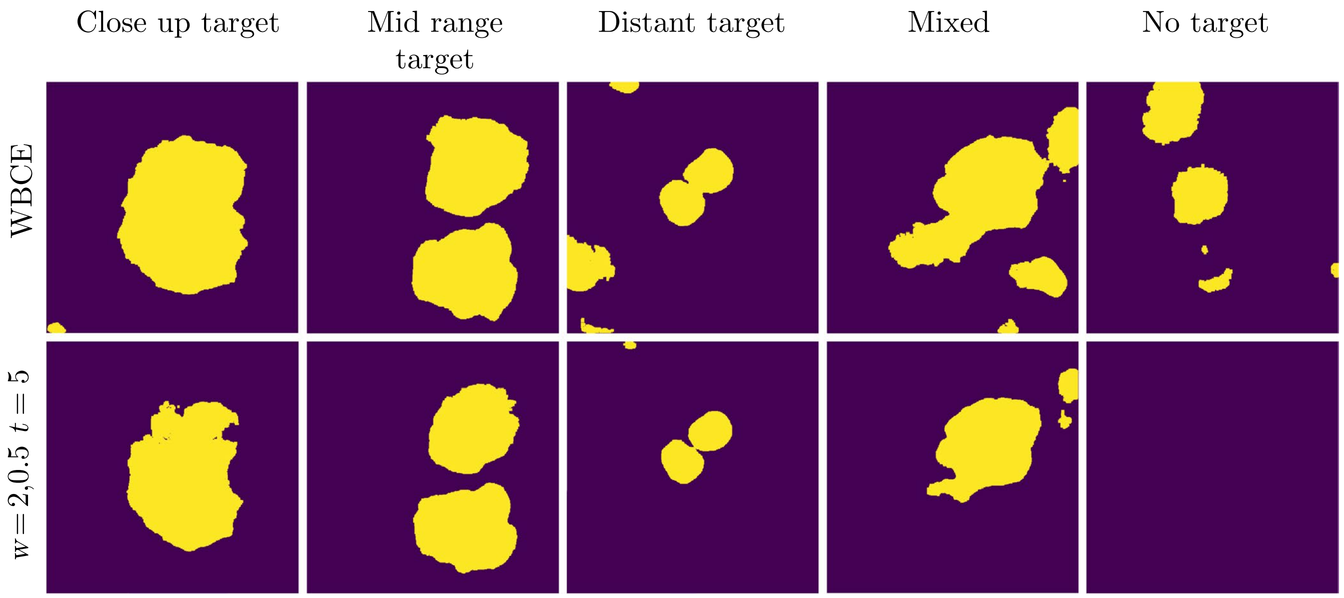
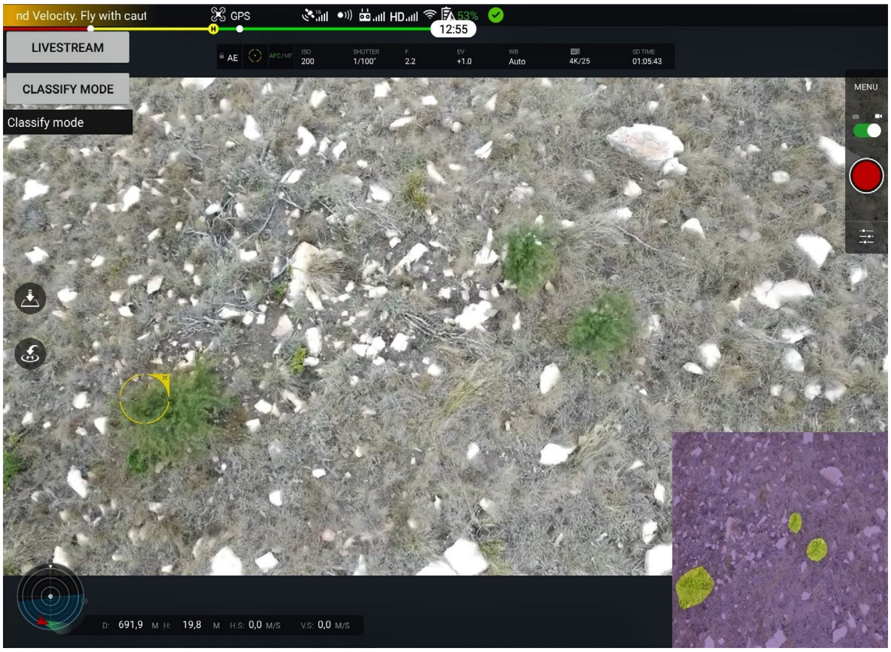


FIGURE 6 The top-performing model makes use of WMB loss, with $w = 2, 0.5$ $t = 5$. This model clearly produces fewer false positive objects than WBCE loss. This is particularly clear in the 'no target' column. For comparison, see the ground truth values (masks) in Figure 5

FIGURE 7 A screenshot of the application when the drone is in flight in the field. Three *Hakea* shrubs are successfully detected, as shown in the purple and yellow overlay



AE AFC/MF ISO 200 SHUTTER 1/100" F 2.2 EV +1.0 WB Auto 4K/25 SD TIME 01:05:43

D: 691,9 M H: 19,8 M H.S: 0,0 M/S V.S: 0,0 M/S

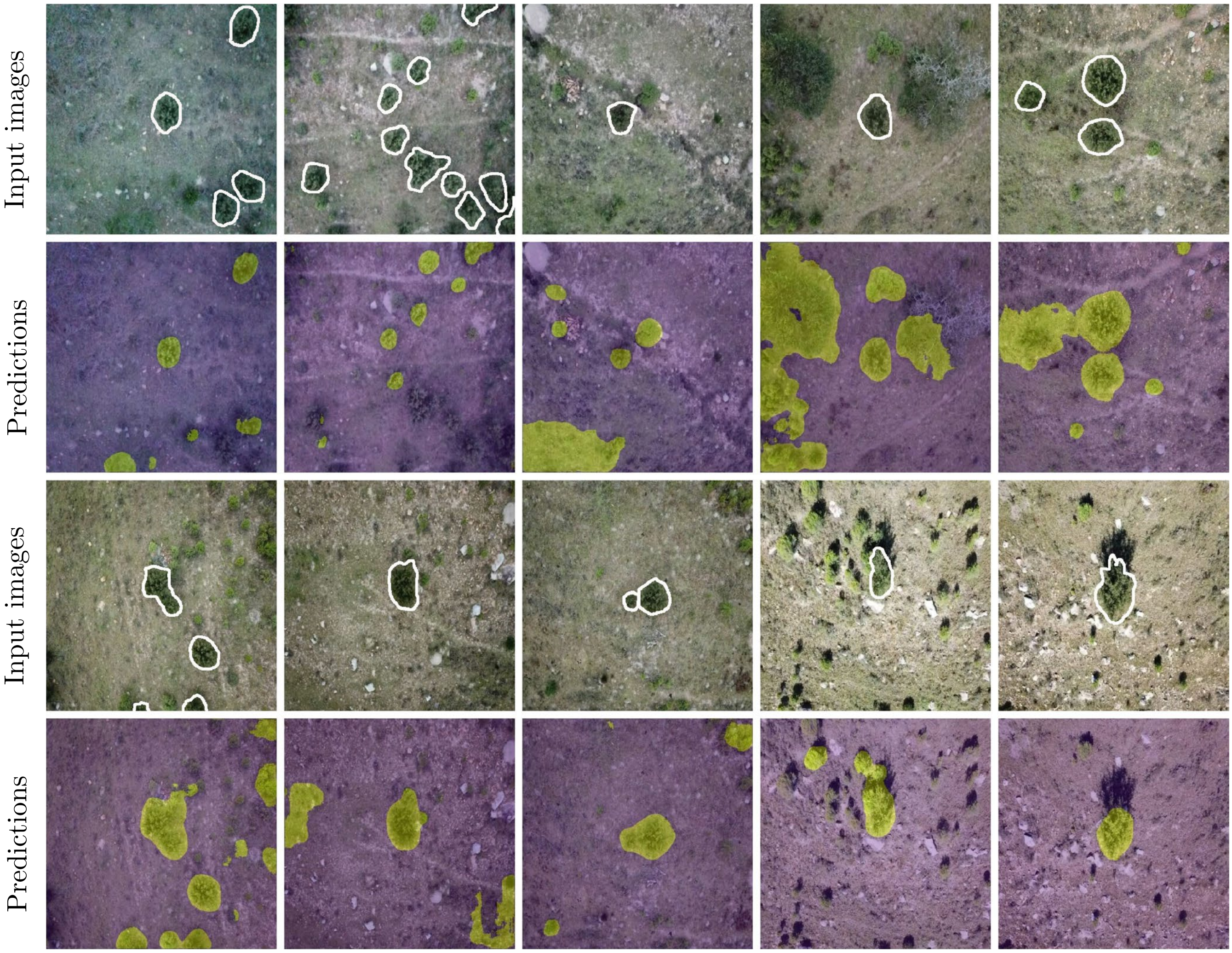


FIGURE 8 In-field segmentation results, with *Hakea* shrubs outlined in white on the input image for ease of visibility to the reader

images of positive detections, false positives would pose minimal detraction from the usefulness of detection in flight, as they could easily be manually filtered out at a later stage when the data was analysed.

Research on mapping *Hakea sericea*, one of two of the species of *Hakea* prevalent in our study area, as one of seven land cover classes was conducted by Alvarez-Taboada et al. (2017), in which precision and recall above 75% were achieved on the orthophoto created from drone sourced imagery. This result is similar to the over 80% precision and recall achieved by our model on the test set. Alvarez-Taboada et al. (2017) noted that woodland was often misclassified as *Hakea*, which indicates that false positives were a shortcoming of their approach as well. In their study, a set of chosen features were extracted from objects, some of which drew on the available near infra-red band, which was not available in our study.

When operating the system in a real outdoor environment, factors such as season and cloud cover have impacts on model performance. Season changes affect multiple variables, such as the dryness of the environment (which in turn affects the colour of the vegetation in the area), the angle of the sun at midday (having an effect on shadow prominence), the light intensity and phenological traits such as flowering. Cloud cover and illumination intensity affect the brightness of the landscape and the intensity of shadows, particularly of taller vegetation such as shrubs. Some of these challenges can be addressed during the training process using appropriate augmentation strategies, as was shown in this article with regard to brightness augmentation, resulting in a model that offered 27% better F1-score on the test set. However, other variations such as shadow prominence are harder to synthesise using augmentation strategies and as a result must be reflected in the dataset for the model to learn invariance to them. The low sun elevation angle present in the winter months results in visible shadows even at midday, and has previously been noted as unfavourable for data collection by Lehmann et al. (2017) and Chabot et al. (2018) who recommended that flights be conducted within two hours of the sun's zenith to achieve the best possible illumination. Variation in annotation confidence in the masks used during training, due to the indistinct edges associated with vegetation, was also shown to affect model performance and was overcome through the use of WMB loss.

As the dataset used in this study was collected over two days in the summer months, this limited the ability of the model to be robust to seasonal variation. However, field testing was conducted in early winter, and most of those images assessed in Section 3.3 achieved perfect recall rates, which bodes well for its performance in summer months. Unfortunately, a high number of false positives were detected, which indicates that further model refinement is necessary. It is possible, but untested, that if tested in the field in summer months when conditions are more similar to those in the dataset, that the false positive detections would be reduced and the recall rate increased. The target plant flowers in winter, a season not reflected in the dataset, and as such the model developed is not ideally suited for use during the flowering period.

Despite these challenges and limitations, the system demonstrated that modern commercially available drones are suitable for detection of plant species of interest on the fly, with an integrated deep learning segmentation model. A deep learning model suitable for detection of *Hakea* can be produced using the U-Net architecture, trained using weight map-based loss. This model achieved an F1-score and accuracy of 83% and 96%, respectively, on the unseen test set. This highlights the potential of such systems for use in applications where intelligent interaction between a drone and the environment is necessary, where the simple action of displaying the segmentation result could be replaced by a sub-routine to perform this interaction.

ACKNOWLEDGEMENTS

Thanks must go to the Rhodes University Ian Mackenzie and Henderson Scholarship funds for the financial support that has made this work possible. This work was undertaken in the Distributed Multimedia COE at Rhodes University, with financial support from Telkom SA and CORIANT.

AUTHORS' CONTRIBUTIONS

K.J. conducted the experimentation and analysis and wrote the original draft of the manuscript; K.B. assisted in conceptualisation of experimentation, and reviewed, edited and made contributions to the text of the final version.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13473.

DATA AVAILABILITY STATEMENT

The datasets (James & Bradshaw, 2020a) and source code (James & Bradshaw, 2020b) used in this paper have been made publicly available online at Zenodo.org.

ORCID

Katherine James https://orcid.org/0000-0003-0901-3791
Karen Bradshaw https://orcid.org/0000-0003-3979-5675

REFERENCES

Alsalam, B. H. Y., Morton, K., Campbell, D., & Gonzalez, F. (2017). Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture. In *2017 IEEE Aerospace Conference* (pp. 1–12). Big Sky, MT: IEEE. https://doi.org/10.1109/AERO.2017.7943593

Alvarez-Taboada, F., Paredes, C., & Julián-Pelaz, J. (2017). Mapping of the invasive species *Hakea sericea* using Unmanned Aerial Vehicle (UAV) and WorldView-2 imagery and an object-oriented approach. *Remote Sensing*, *9*(9), 913. https://doi.org/10.3390/rs9090913

Baron, J., Hill, D., & Elmiligi, H. (2018). Combining image processing and machine learning to identify invasive plants in high-resolution images. *International Journal of Remote Sensing*, *39*(15–16), 5099–5118. https://doi.org/10.1080/01431161.2017.1420940

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(1), 2–16. https://doi.org/10.1016/j.isprsjprs.2009.06.004

Bradley, B. (2014). Remote detection of invasive plants: A review of spectral, textural and phenological approaches. *Biological Invasions*, *16*(7), 1411–1425. https://doi.org/10.1007/s10530-013-0578-9

Chabot, D., Dillon, C., Shemrock, A., Weissflog, N., & Sager, E. (2018). An object-based image analysis workflow for monitoring shallow-water aquatic vegetation in multispectral drone imagery. *ISPRS International Journal of Geo-Information*, *7*(8), 294. https://doi.org/10.3390/ijgi7080294

Dash, J. P., Watt, M. S., Paul, T. S. H., Morgenroth, J., & Hartley, R. (2019). Taking a closer look at invasive alien plant research: A review of the current state, opportunities, and future directions for UAVs. *Methods in Ecology and Evolution*, *10*(12), 1–14. https://doi.org/10.1111/2041-210X.13296

de Sá, N. C., Castro, P., Carvalho, S., Marchante, E., López-Núñez, F. A., & Marchante, H. (2018). Mapping the flowering of an invasive plant using unmanned aerial vehicles: Is there potential for biocontrol monitoring? *Frontiers in Plant Science*, *9*, 293. https://doi.org/10.3389/fpls.2018.00293

Dvořák, P., Müllerová, J., Bartaloš, T., & Brůna, J. (2015). Unmanned aerial vehicles for alien plant species detection and monitoring. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(1), 83–90. https://doi.org/10.5194/isprsarchives-XL-1-W4-83-2015

Flood, N., Watson, F., & Collett, L. (2019). Using a U-Net convolutional neural network to map woody vegetation extent from high resolution satellite imagery across Queensland, Australia. *International Journal of Applied Earth Observation and Geoinformation*, *82*, 101897. https://doi.org/10.1016/j.jag.2019.101897

Göktogan, A., Sukkarieh, S., Bryson, M., Randle, J., Lupton, T., & Hung, C. (2010). A rotary-wing unmanned air vehicle for aquatic weed surveillance and management. *Journal of Intelligent and Robotic Systems*, *57*(1), 467–484. https://doi.org/10.1007/s10846-009-9371-5

Guirado, E., Tabik, S., Alcaraz-Segura, D., Cabello, J., & Herrera, F. (2017). Deep-learning versus OBIA for scattered shrub detection with Google Earth imagery: *Ziziphus Lotus* as case study. *Remote Sensing*, *9*(12), 1220. https://doi.org/10.3390/rs9121220

Huang, C., & Asner, G. (2009). Applications of remote sensing to alien invasive plant studies. *Sensors*, *9*(6), 4869–4889. https://doi.org/10.3390/s90604869

James, K., & Bradshaw, K. (2019). Segmenting objects with indistinct edges, with application to aerial imagery of vegetation. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists* (pp. 1–7). Skukuza, RSA: ACM. https://doi.org/10.1145/3351108.3351124

James, K., & Bradshaw, K. (2020a). Hakea vegetation dataset for training and evaluating segmentation network. *Zenodo*, https://doi.org/10.5281/zenodo.3976850

James, K., & Bradshaw, K. (2020b). KatherineJames/AI Drone vegetation detection: MEE paper release. *Zenodo*, https://doi.org/10.5281/zenodo.3987188

Kamilaris, A., & Prenafeta-Boldu, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

Kluge, R., & Neser, S. (1991). Biological control of *Hakea sericea* (Proteaceae) in South Africa. *Agriculture, Ecosystems & Environment*, *37*(1–3), 91–113. https://doi.org/10.1016/0167-8809(91)90141-J

Lehmann, J., Prinz, T., Ziller, S., Thiele, J., Heringer, G., Meira-Neto, J., & Buttschardt, T. (2017). Open-source processing and analysis of aerial imagery acquired with a low-cost unmanned aerial system to support invasive plant management. *Frontiers in Environmental Science*, *5*, 44. https://doi.org/10.3389/fenvs.2017.00044

Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., & Li, W. (2018). DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(11), 3954–3962. https://doi.org/10.1109/JSTARS.2018.2833382

Liu, T., Abd-Elrahman, A., Jon, M., & Wilhelm, V. (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 Sensing*, *55*(2), 243–264. https://doi.org/10.1080/15481603.2018.1426091

Mafanya, M., Tsele, P., Botai, J., Manyama, P., Swart, B., & Monate, T. (2017). Evaluating pixel and object based image classification techniques for mapping plant invasions from UAV derived aerial imagery: *Harrisia pomanensis* as a case study. *ISPRS Journal of Photogrammetry and Remote Sensing*, *129*, 1–11. https://doi.org/10.1016/j.isprsjprs.2017.04.009

Martin, F.-M., Müllerová, J., Borgniet, L., Dommanget, F., Breton, V., & Evette, A. (2018). Using single- and multi-date UAV and satellite imagery to accurately monitor invasive knotweed species. *Remote Sensing*, *10*(10), 1662. https://doi.org/10.3390/rs10101662

Nogueira, K., Penatti, O. A., & dos Santos, J. A. (2016). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, *61*, 539–556. https://doi.org/10.1016/j.patcog.2016.07.001

Pajares, G. (2015). Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). *Photogrammetric Engineering & Remote Sensing*, *81*(4), 281–330. https://doi.org/10.14358/PERS.81.4.281

Palmer, T. (2004). *Vegetation of Makana* (Technical Report). Grahamstown, South Africa: ARC – Range and Forage Institute.

Rangel, R. (2016). Development of an UAVS distribution tools for pest's biological control 'Bug Bombs!'. In *2016 IEEE Aerospace Conference* (pp. 1–8). Big Sky, MT: IEEE. https://doi.org/10.1109/AERO.2016.7500685

Richardson, D., van Wilgen, B., & Mitchell, D. (1987). Aspects of the reproductive ecology of four Australian *Hakea* species (Proteaceae) in South Africa. *Oecologia*, *71*(3), 345–354. https://doi.org/10.1007/BF00378706

Rominger, K., & Meyer, S. E. (2019). Application of UAV-based methodology for census of an endangered plant species in a fragile habitat. *Remote Sensing*, *11*(6), 719. https://doi.org/10.3390/rs11060719

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241). Munich, Germany: Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

van Wilgen, B., & Richardson, D. (1985). The effects of alien shrub invasions on vegetation structure and fire behaviour in South African fynbos shrublands: A simulation study. *Journal of Applied Ecology*, *22*, 955–966. https://doi.org/10.2307/2403243

Wu, Z., Ni, M., Hu, Z., Wang, J., Li, Q., & Wu, G. (2019). Mapping invasive plant with UAV-derived 3D mesh model in mountain area: A case study in Shenzhen Coast, China. *International Journal of Applied Earth Observation and Geoinformation*, *77*, 129–139. https://doi.org/10.1016/j.jag.2018.12.001

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv Preprint* arXiv:2003.05689.

Zhang, L., Zhang, L., & Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, *4*(2), 22–40. https://doi.org/10.1109/MGRS.2016.2540798

How to cite this article: James K, Bradshaw K. Detecting plant species in the field with deep learning and drone technology. *Methods Ecol Evol*. 2020;11:1509–1519. https://doi.org/10.1111/2041-210X.13473