remote sensing

Article

Drone Image Segmentation Using Machine and Deep Learning for Mapping Raised Bog Vegetation Communities

Saheba Bhatnagar *, **Laurence Gill** and **Bidisha Ghosh**

Department of Civil, Structural and Environmental Engineering, Trinity College, University of Dublin, D02 PN40 Dublin, Ireland; laurence.gill@tcd.ie (L.G.); bghosh@tcd.ie (B.G.)

* Correspondence: sbhatnag@tcd.ie

Received: 12 June 2020; Accepted: 11 August 2020; Published: 12 August 2020

Abstract: The application of drones has recently revolutionised the mapping of wetlands due to their high spatial resolution and the flexibility in capturing images. In this study, the drone imagery was used to map key vegetation communities in an Irish wetland, Clara Bog, for the spring season. The mapping, carried out through image segmentation or semantic segmentation, was performed using machine learning (ML) and deep learning (DL) algorithms. With the aim of identifying the most appropriate, cost-efficient, and accurate segmentation method, multiple ML classifiers and DL models were compared. Random forest (RF) was identified as the best pixel-based ML classifier, which provided good accuracy (≈85%) when used in conjunction graph cut algorithm for image segmentation. Amongst the DL networks, a convolutional neural network (CNN) architecture in a transfer learning framework was utilised. A combination of ResNet50 and SegNet architecture gave the best semantic segmentation results (≈90%). The high accuracy of DL networks was accompanied with significantly larger labelled training dataset, computation time and hardware requirements compared to ML classifiers with slightly lower accuracy. For specific applications such as wetland mapping where networks are required to be trained for each different site, topography, season, and other atmospheric conditions, ML classifiers proved to be a more pragmatic choice.

Keywords: semantic segmentation; machine learning; random forest; deep learning; CNN

1. Introduction

The use of drones for different types of vegetation classification has increased many folds over the last decade. This is due to the technological development of affordable and lightweight drones. With drones, a very high and flexible spatial resolution can be achieved, which is not possible with satellite imagery due to their fixed orbits. Satellites are both open-source and commercial. Some of the most popular open-source satellites include the Sentinel and Landsat series. These satellites provide global information but lack high spatial resolution with the best resolution possible of 10 m using Sentinel-2 (S2). The S2 imagery has been widely used for classifications; for example, a study carried out by [1] used S2 imagery for temporal mapping of wetland vegetation communities. However, one conclusion from the study was that accuracy decreases for smaller wetlands. In many cases in Ireland, at least, the area of wetlands can be relatively small, whereby satellite-based classification is not sensitive enough and can produce large errors. One of the significant problems is pixel-mixing; when the size of the pixel is 10 m, for example, each pixel can have a combination of species present in it. This affects the overall reflectance value of the pixel, and hence, a good boundary or extent of the species cannot be achieved. There are several ways to reduce the error in satellite images, but most of them require extensive hyperspectral bands. However, another method to get detailed monitoring of small areas is to use unmanned aerial vehicles (UAVs), more commonly known as drones.

Most drones typically carry optical cameras (RGB) and occasionally can support a thermal sensor, but some drones can also support more expensive hyperspectral cameras. The presence of a thermal/hyperspectral sensor allows more details to be gathered and improves spectral resolution. However, the dilemma about spectral versus spatial remains unanswered. Missions like Airborne Visible InfraRed Imaging Spectrometer (AVIRS) and hyperspectral satellite Hyperion provides high spectral resolution. A study [2] has used AVIRIS hyperspectral data with 224 bands and 20 m spatial resolution to detect invasive plant species (*Colubrina asiatica* (Brongniart, Adolphe Théodore) in Florida. Another study [3] used Hyperion (30 m) hyperspectral data to detect *Phragmites australis* (Steudel, Ernst Gottlieb) in coastal wetlands and states that, due to low spatial resolution, the analysis was affected by pixel mixing. Therefore, apart from high spectral resolution, a proper spatial resolution is also required for monitoring vegetation communities closely. Drone images have much higher spatial resolution when compared to satellite images. Drones have been explicitly used for species detection [4–7]. A study by [8] states many advantages of using a drone over satellite imagery for identification of land cover communities such as water, land *Avicennia alba* (Blume, Carl (Karl) Ludwig), *Nypa fruticans* (Verh. Batav. Genootsch. Kunst.), *Rhizophora apiculata* (Blume, Carl (Karl) Ludwig), and *Casuarina equisetifolia* (Linnaeus, Carl). Drone imagery has also been applied for specific applications such as analysing vegetation under shallow water, tracking waterbirds, and their habitats [9,10]. A study by [11] concluded that a thermal (infrared) sensor on its own performs comparable to an RGB sensor, but a multispectral sensor (with multiple spectral bands and indices) is required for the best analysis of nitrogen on rice fields. Multiple spectral sensors, however, although useful, are costly [12]. A study by [13] supports the hypothesis that proper spatial resolution with an RGB sensor is sufficient for the analysis of wetland delineation, classification, and health assessment. Therefore, taking all the points into consideration, as an alternative to an expensive camera, an RGB camera was used in this study.

For the analysis of drone data, many techniques are available. The state-of-the-art techniques in drone image analysis consist of both machine learning (ML) and deep learning (DL). A study by [14] demonstrates the application of ML techniques to classify drone images into roads, vineyards, asphalt, and roofs. The study uses ensemble decision trees with an object-oriented approach. The study [15] has used object-based multi-resolution segmentation (eCognition software) of UAV imagery for the segmentation of the agricultural field. Other than object-based, there are multiple pixel-based studies, such as [16,17] use support vector machine (SVM) classifier for agricultural mapping and reef monitoring using drone imagery. The study [18] applies multiple ML algorithms including random forest (RF), SVM, and gradient boosting decision tree (GBDT) to classify trees, grasses, bare gravel/sand bed, and water surface. The study achieved a high accuracy of up to 98% using RF classifier on UAV images. ML algorithms have also been used for vegetation segmentation. A study by [19] has used simple linear iterative clustering (SLIC) for mangrove segmentation. Another study by [1] has used graph cut for the segmentation of vegetation communities in wetlands. Hence, the segmentation of drone images can help in the identification of subtle changes in vegetation communities. Apart from ML, advanced deep learning techniques are also now being widely applied for drone image segmentation. A recent state-of-the-art review [20] shows the surge of applying DL in the field of RS. It also gives details about various convolutional neural network (CNN) models and suggests that ≈20% of all studies since 2012, uses DL with UAV imagery. The study [21] has used DL to segment concrete-cracks in drone images. The segmentation using a CNN is known as semantic segmentation. It has been applied for various applications like urban land classification [22,23], forest cover classification [24], and wetland type classification [25,26]. A study by [27] uses ResNet50 and UNet for classification of forest tree-species, and [28] has used transfer learning to get the best semantic segmentation of the aerial images AeroScapes dataset. Both [27,28] suggest that the usage of transfer learning enhances the analysis. A study by [29] has utilised both ML (linear regression) and DL (neural network) for predicting water and chlorophyll content in citrus leaves. The study suggested that both ML and DL give comparable results for predictions using UAV images. From the literature, it is apparent that both ML and DL can be applied for drone image segmentation. However, it is not clear which technique, the traditional state-of-the-art machine learning or the advanced deep learning is better for the identification of the communities. Therefore, in this study, we applied both ML and DL techniques for vegetation classification of different vegetation communities on a raised bog wetland. Our study also demonstrates the pros and cons of both methods. It also gives a clear insight into both the techniques and their applicability for future studies on vegetation identification.

2. Study Area and Materials

The area of study is one of the largest intact raised bogs present in Ireland, covering approximately 460 ha area located in the midlands called Clara bog. The two sides of the bog are divided by a road: East Clara is a restored bog (after years of drainage and peat cutting), whereas, the West Clara remains a natural active raised bog. This study concentrates on a small part of the bog located in West Clara bog (as shown in Figure 1). The different vegetation species have been grouped into different communities on the basis of similar habitats, which are termed 'ecotopes' [30].

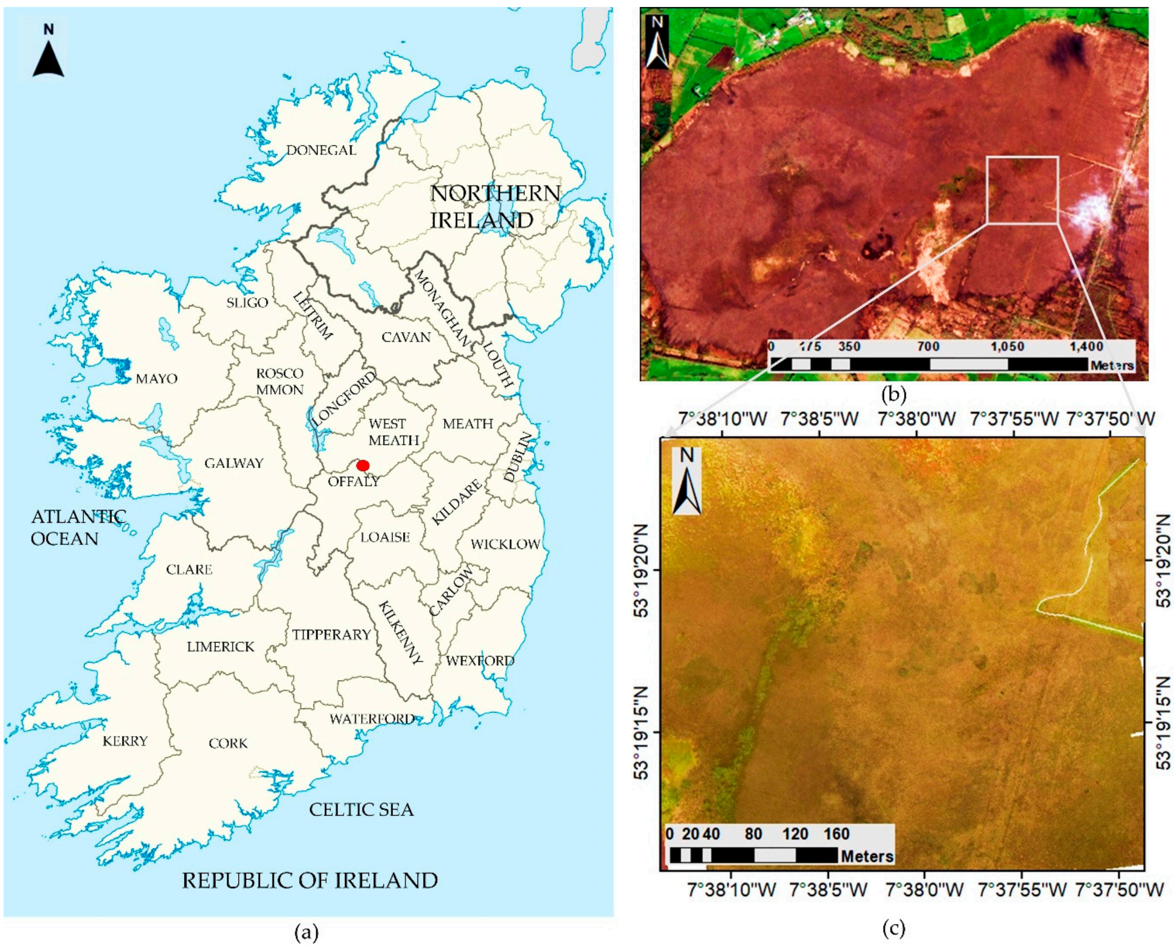


Figure 1. Study area: (**a**) map of Ireland (with the highlighted area: Clara Bog). (**b**) West of Clara Bog, County Offaly (with the highlighted area covered by drone). (**c**) Area covered by DJI Inspire 1™ drone.

The major ecotopes present in Clara bog are Central (C), Subcentral (SC), Submarginal (SM), Marginal (M), and Active flush (or flush) (AF). Other ecotopes like Inactive flush (IAF), Facebank (FB) are also present in this bog but have not been considered in this study due to their low ecological impact. Out of all these ecotopes, the main focus is on the conservation of the active peat-forming areas [1,30,31], which are considered to be C, SC, and AF ecotopes. These areas have high sphagnum moss coverage, with hummocks, hollows, lawns, and many pools. The SM ecotope that appears at the boundaries of the SC ecotope can appear to be almost homogenous, which makes it hard to distinguish between them. The SM and M ecotopes are located on drier areas with vegetation reflective of such conditions.

For capturing high-resolution images, a DJI Inspire 1™ drone was used. The camera used with the drone was Zenmuse X3. It is an optical camera with 100–1600 ISO range (for photo) and 94° field of view (FOV). The lens is anti-distortion and autofocus (20mm of 35mm format equivalent). The aspect ratio, while clicking the images, was kept at 4:3. The images were captured on 21st April 2019 at around noon time. The highest temperature on the day was recorded at 19 °C. The height of the flight was ≈100m, and the spatial resolution of the images captured was 1.8 cm. The drone mission was pre-loaded using Google maps in Pix4DCapture application to capture ≈8 ha of the area using an iOS-12 device. The images were captured individually with 70% frontal and 80% sideways overlap at an average speed of 3 m/s. Figure 1c provides the drone imagery of the study area. For georeferencing, the drone imagery had geo-tags (lat-long locations) present in it. For better orientation, imagery was overlayed on high-resolution DigitalGlobe World Imagery (spatial resolution = 30cm) available as a base map in ArcMap v.10.6.1 [32,33]. Using 'georeferencing' toolbox present in [32], 3–4 ground control points (GCPs) were identified for every image, and projection was rectified to Geographic Coordinate System—World Geodetic System 84 (GCS WGS 84). In this study, C, SC, SM, M, and AF ecotopes were all captured using high-resolution drone imagery (Figure 2).

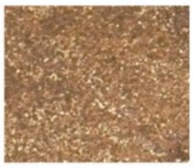
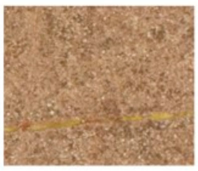
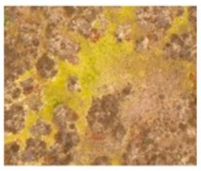
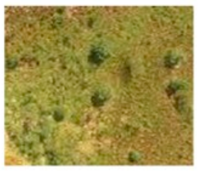
Submarginal	Subcentral	Marginal	Central	Flush
				

Figure 2. Ecotopes in Clara bog. Drone images, April 2019.

The SM and SC ecotopes are highly homogenous and appear to be mixed throughout the bog [1]. These communities were therefore merged for the rest of the study. In total, around ≈75 images of dimension 3000 × 4000 were captured. Out of these images, 15 images were discarded due to differences in light intensity, motion blur, and camera tilt. The usable 60 images were divided into 70% training and 30% testing randomly, which is around 40 images for training and 20 images for testing. In order to have a correct idea of mapping accuracy, all the images were labelled for the four vegetation communities (M, SMSC, C, AF). For ML only a part of the labelled training data was required, whereas for DL fully labelled images were used. This is discussed further in Sections 3 and 4. For the creation of a training dataset, it is essential for all the images to have a similar intensity range. Depending on the lighting situation when the picture was taken, the colour properties may be changed, even though the textural properties remain unchanged. In a temperate climate like Ireland, this change in sunlight while capturing drone images is unavoidable. Therefore, going forward in future studies, the usage of colour correction techniques for drone mages is recommended such that all the captured images can be used.

3. Segmentation Using Machine Learning

The segmentation of the images using machine learning techniques utilises combinations of intensity, colour, texture, and motion attributes to come up with hierarchical segments [34]. The drone images used for this study have intensity and colour information. Although textural information is not present in the original image, textural features were subsequently calculated using the parameters mentioned in Table 1, [35]. This was done by converting the RGB image into a grayscale image. The textural information presented in Table 1 was added as features along with the RGB layers.

The entire computation of machine learning techniques and the steps described below was performed using MATLAB v.2019b using image processing toolbox [36].

Table 1. Textural properties calculated using drone imagery.

Property	Description
Contrast	Intensity difference between pixels compared to its neighbour for the whole image [37].
Correlation	Correlation of a pixel and its neighbour for the whole image [38].
Energy	Sum of squared elements in gray level co-occurrence matrix (GLCM) [39].
Homogeneity	Closeness of the distribution of pixels in the GLCM to its diagonal [40].
Mean	Mean of the area across the window
Variance	Variance of the area across the window
Entropy (e)	Statistical measure of randomness $e = -\sum\left(h \times \log_2 h\right)$; where h contains the normalised histogram counts
Range	Range of the area across the window [41].
Skewness (S)	Asymmetry of the data over the mean value [42]. $S = E(p - \mu)^3/\sigma^3$, where μ is the mean of the pixel p, σ is the standard deviation of p, and E represents the expected value.
Kurtosis (K)	Distribution to be prone to outliers [42]; $K = E(p - \mu)^4/\sigma^4$

The segmentation technique used in this study, called graph cut, is based on max-flow min-cut [43]. This is done using posterior probabilities associated with every pixel for every class. In order to calculate the posterior probabilities, an initial classification of the drone images was carried out. Based on the texture and colour intensity, a total of 13 bands are used for further classification of the drone images. The type and choice of classifier used are discussed in the following subsection.

3.1. Choice of the ML Classifier

For efficient classification, the choice of the classifier is the most crucial decision that has to be made. Multiple studies have applied hyperspace based SVM [44,45] for image classification. Other studies, like [46], have used decision trees. Studies [47,48] suggest that there is an advantage of using ensemble classifiers over other state-of-the-art classifiers. The most commonly used ensemble classifier consists of a tree model. The tree models are easy to understand and could be used for both classification and regression. There is no need for variable selection (since it is automatic) or variable transformation. They are robust to outliers and missing data, and particularly useful for large datasets.

In this study, in order to provide proper comparative analyses, the drone images captured on 21st April 2019, were classified using multiple classifiers. The training dataset (≈12k pixels from 40 images) was the input for all the classifiers. The classifiers were tested on model accuracy, misclassification cost (i.e., the total number of incorrectly identified pixels per 10,000 pixels), and training time (time taken by the classifier for training). The model accuracy for each ML model was calculated using 5-fold cross-validation for the entire 70% training dataset. This accuracy indicates the capability of the model to label the pixels correctly. The results (Table 2) describes all the classifiers and the corresponding accuracy metric. All the calculations were performed using MATLAB v.2019b [36].

The preliminary comparison was made using six classifiers, namely, decision trees [49], naïve Bayes [50], discriminant analysis [51], SVM [52], k-nearest neighbour (KNN) [53], and random forest (RF) [54]. Based on the misclassification rate, model accuracy, and training time (see Table 2), RF was found to be best classifier. Random forest or bagging is a general-purpose procedure for reducing the variance of a predictive model. When applied on trees, the number of trees (t) is bootstrapped, each having a variance σ^2. In RF each tree can split on only a random subset of the samples (hence, the name). RF requires an attribute (sample) selection and a pruning method. Information gain ratio criterion [55] and Gini Index [56] are the most common attributes selection methodology. For this study, the Gini index criterion was used to decide the attributes. The Gini index (G) is given in Equation (1). Based on the value of G, the attribute was decided automatically.

$$G = \sum_{n} \sum_{i=1}^{N} (p_i \times (1 - p_i))_n \tag{1}$$

where p_i is the proportion of the pixel (i = 1 to N) belonging to a particular class n, i.e., it is the prior probability. A minimum of 10% of the entire ground truth image should be given as training and rest could be used for testing [1]. The samples were divided into 100 random subsets (with repetition), and for each tree, and the attributes (splitting criteria: which of the RGB bands) were decided using Equation (1). The final class selection for every pixel was made using majority voting. The workflow of the RF classifier is given in Figure 3.

Table 2. Comparison of ML classification techniques.

Name	Parameter	Model Accuracy	Misclassification Cost	Training Time (s)
Decision trees	Max. no. of splits = 20; split criterion = Gini's diversity index	87.4	736	7.3
Discriminant analysis	Kernel = quadratic	89.4	618	8.6
Naïve Bayes	Kernel = Gaussian	78.3	1271	19.5
Support vector machine	Kernel = radial basis function (rbf) = 0.25	91.9	472	112.5
K nearest neighbour	No. of neighbours = 2; distance = Euclidean	91.0	528	378.8
Random forest	No. of trees (t) = 100 (1000 samples with repetition); total no. of splits = 5853	92.9	454	59.2

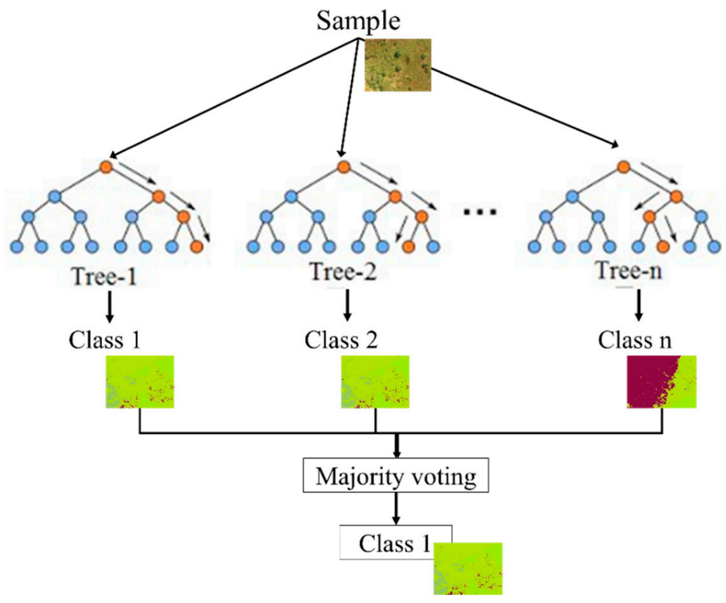


Figure 3. Workflow—random forest classifier.

3.2. Segmentation

Once the drone images were classified, they were segmented using the maximum-a-posterior (energy minimisation) technique. The technique uses contextual (area) information to form proper segments from pixels. The pixels, therefore, are no longer treated as a single entity but part of a more significant segment. It can be considered as a post-classification smoothing process based on spatial similarities. The formation of segments was done using a max-flow min-cut algorithm, commonly known as graph cut. This algorithm uses data cost and smoothness costs [57]. The graph cut segmentation was performed in MATLAB v.2019b [40] using MATLAB wrapper mex file function that enables the user to call C/C++ files [58]. The steps for the segmentation include calculation of the data cost, smoothness cost, and energy using posterior probability from the pixel-based classification map. Based on the maximum probability of the pixels, the segments were formed, and the pixels were joined.

The data cost ($\boldsymbol{D_p}$) is based on individual labels of pixels and their likelihood function. The data cost $\boldsymbol{D_p}$ measures the cost of assigning the class n to the pixel p for a given set of features U_N in the vectorised image having N pixels. In image processing $\boldsymbol{D_p}$ can be typically expressed as [59], given by Equation (2).

$$\boldsymbol{D_p} = \left\|U_p(n) - I(p)\right\|^2 \tag{2}$$

where, $I(p)$ was the observed reflectance of the pixel p.

The smoothness cost ($V_{p,q}$) on the other hand, was used to promote groups. It was assumed that the neighbouring pixels should belong to the same class, and hence, this cost was given based on the likelihood of pixels $p,\ q$ belonging to same class n. n_p , n_q are labels of pixels $p,\ q$ respectively. It was defined using described in Equation (3).

$$Vp,q\left(n_p\ ,\ n_q\right) = c \times exp\ (-\Delta(p,\ q)/\sigma) \times T\left(n_p\ \neq\ n_q\right) \tag{3}$$

where $\Delta(p,\ q) = \left|I(p) - I(q)\right|$ denotes how different the reflectance values of p and q are, $c > 0$ is a smoothness factor, standard deviation $\sigma > 0$ is used to control the contribution of $\Delta(p,\ q)$ to the penalty, and $T = 1$ if $n_p \neq n_q$ and 0 otherwise.

As described in [1], the steps followed for drone images were the same as for the satellite image segmentation. The main difference comes in the choice of the smoothness factor. Since a drone image was much more detailed for forming distinct segments compared to a satellite image, a high smoothness factor was required. After an iterative parametrisation optimisation exercise, a smoothness factor (c) of $c > 5$ was chosen for the drone images. This can be compared to the optimum value of $c < 1$ when processing satellite images [1]. Therefore, it was seen that for a high resolution (1.8 cm), a higher value of c was required, whereas, when working with the 10 m spatial resolution from satellite images, a small value of c suffices.

The pioneering work done by [59] explains energy (E) minimisation can be interpreted directly as posterior maximising. Using probability functions from previous steps, we get the energy function as described in Equation (4).

$$E\ (U_N, n) = \sum_{p \in N} Dp + \sum_{p,q \in N} Vp,q \tag{4}$$

Therefore, $E(U_N,\ n)$, i.e., energy for the image vector with total N pixels (U_N) for all n classes is minimised, leading to the formation of smooth segments. The pixels with least E are joined together to form the segments depending on their initial labels as obtained from the pixel-based RF classification. The results of the segmentation are further discussed in Section 5.1.

4. Segmentation Using Deep Learning

4.1. Parameters in Convolutional Neural Network

Convolution neural networks (CNNs or Covnets) have caused a step-change in pattern recognition progress. Here each neuron is connected to a local region of the input only, making the network faster and less prone to overfitting for a large dataset. Therefore, CNNs, when compared to traditional NNs, can have fewer parameters. In addition, the same parameters are used in more than one place on CNN, making the model both statistically and computationally efficient. The initial layers of the CNN identify lines, corners, edges, textures, and then the deeper the network goes, the more precisely it can learn from the features, as shown in Figure 4, which gives the architecture of CNN. The different layers used in CNN are described in detail in the following subsections.

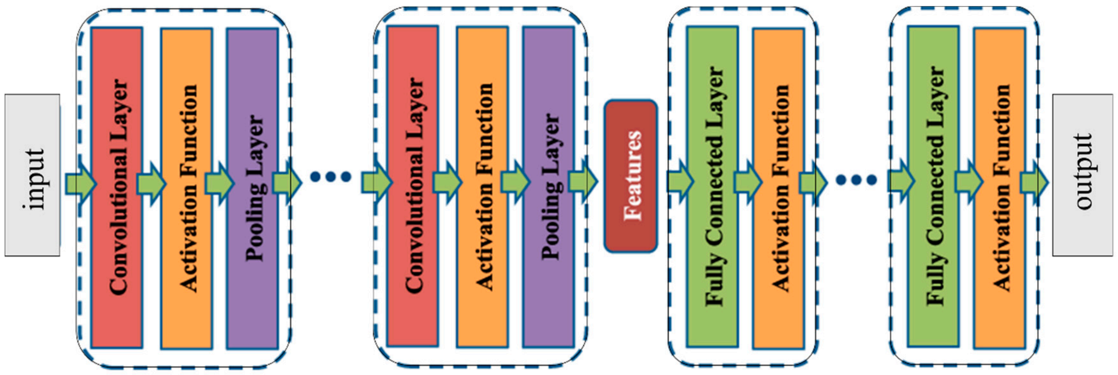


Figure 4. The general architecture of convolutional neural network (CNN).

4.1.1. Convolutional Layer

Convolution in CNN is the mathematical operation that combines signals a and b [$a * b$], i.e., filtering input a with kernel b. It is a process to overlay 'b' on 'a', multiply the numbers and sum the products and move. In CNN the convolutional layer is used instead of only fully connected layers. For visualising, convolution may look like a sliding window operation, but it is implemented as matrix multiplication. The input is divided into arrays as well as the kernels and rearranged into columns.

4.1.2. Pooling Layer

The pooling layer downsamples the input by locally summarising the data in it. The two types of pooling are shown in Figure 5.

1. Max Pooling: where the local maxima of the filtered region are carried forward.
2. Average pooling: where the local average of the filtered region is carried forward.

Of the two methods, max-pooling was used for this study, as it is a more efficient pooling technique [60]. A feature existing in the input layer is fed forward regardless of its initial position (as the local maxima will still make it to the next layer). The advantages of pooling include decreasing the size of the activation layer that is fed forward to the next layer and increasing the receptive field of the subsequent units.

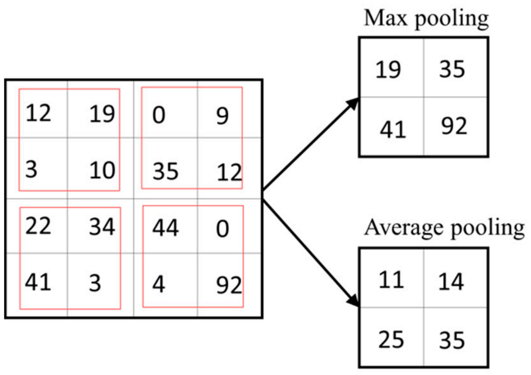


Figure 5. Types of pooling used in CNN.

4.1.3. Kernel Size

Kernels or the filters are used in order to down-sample the layers in CNN. It is preferable to use smaller kernels stacked on top of one another than using a large kernel [61]. Using smaller kernels decreases the number of parameters and also increases the nonlinearity (see Section 4.1.6). For example, a stack of two 3×3 kernels and one 5×5 kernel will have the same receptive field. However, 3×3 will have fewer parameters (as the same kernel is used twice) and more nonlinearity. Therefore, in this study, the kernel of size 3×3 was used.

4.1.4. Stride

Stride defines by how much the kernel will move in the convolution layer. The stride can be used to increase the receptive field. Example, stride = 2. Using stride > 1 provides a down-sampling effect and can be used as an alternative to the pooling layer.

4.1.5. Padding

Padding is required to maintain the spatial resolution of the input image. Padding can be of two types, valid and same. In valid padding, the spatial dimension of the output shrinks by one pixel less than the kernel spatial dimension. Whereas, in same padding, the input is surrounded with zeros such that the spatial dimension of output is the same as the input layer. Therefore, the same padding was used in this study in order to maintain the same dimension between input and output.

4.1.6. Activation Function

The activation function ($f(x)$) defines the output for a given input. It also imparts nonlinearity to the input.

Why do we need nonlinearity?

Combining linear functions yields a linear function; however, in order to compute more in-depth features, nonlinearity is required. With just linear functions, the model is no more expressive than a logistic regression model without any hidden layer. Hence, without any nonlinearity, the entire network behaves as a single linear function.

The study [62] describes the types of activation functions. Some of the most commonly used and well-known activation functions are identity (when linear relation is required), binary step (nonlinear, good for binary classification), sigmoid (nonlinear function, ranging from 0 to 1), tangent hyperbolic (tanH) (same as sigmoid, but ranges from −1 to +1), rectified linear unit (ReLu) (nonlinear function, removes all the –ve part of the input). Sigmoid, tanH, and ReLu also has other variants, see [63]. Other studies like [64,65] compare the various activation functions. A study by [66] presents a comparison between 11 activation functions and suggests ReLu to be the best. Additionally, the ReLu function is much more computationally effective, and therefore, for this study, the ReLu activation function was used. Equation (5) describes the ReLu function.

$$\begin{aligned} f(x) &= 0\,; x < 0 \\ f(x) &= x\,; x \geq 0 \end{aligned} \tag{5}$$

4.1.7. Softmax Classifier

Softmax Classifier is an activation function, typically used as the top layer (after a fully connected layer). It imparts probabilities of each input belonging to each output when there are more than two outputs. For the n number of classes, the Softmax activation (σ) can be defined by Equation (6).

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{n} e^{x_k}}\,; j = 1, \ldots, n \tag{6}$$

4.1.8. Batch Normalisation

It is apparent that each layer is dependent on its previous layer; therefore, even the smallest error in one layer can be magnified in further layers, causing much more significant errors in the final output. To avoid this, a batch normalisation layer is used. This layer normalises the hidden nodes before they are fed into an activation function.

4.1.9. Additional Parameters in CNN

An essential parameter in CNN is optimisation. Training a network can be considered to be an optimisation problem where the goal is to minimise the loss function. There are various optimisation algorithms that can be used to minimise the loss function, such as online learning [67], batch learning [68], and stochastic gradient descent (SGD) [69]. As described in [70] for faster and efficient processing, a subset of the data is taken one at a time, and therefore a stochastic gradient descent was used for optimisation in this study. The subset of data is called a mini-batch, and the number of samples in a mini-batch is called batch size.

Another important parameter in CNN is regularisation. Regularisation of the model can be carried out to make the model simple but effective. This reduces overfitting and adds additional information. This ensures that augmenting the input will not change the quality of the output. Regularisation can be done by adding a weight penalty term to the loss function (Equation (7)).

$$Loss = Loss + weight\ penalty\ (w) \tag{7}$$

L2 or ridge regularisation leads to the formation of small weights [71]. Additionally, L2 regularisation never causes a degradation in performance, even with the addition of kernels [72]. Therefore, L2 regularisation was used in CNNs for this study. For a given input x and its corresponding output $\hat{x}$ the regularisation function is given in Equation (8).

$$Loss = \sum_i (x_i - \hat{x}_i)^2 + \alpha \sum_i {w_i}^2 \tag{8}$$

A third, important parameter for CNN architecture is the learning rate (LR). The LR is defined as the rate at which the weights are updated during the training of the network. The study [73] suggests to start with a bigger learning rate and gradually decrease the gradient when getting closer to the local minima of the loss function. Since adaptive momentum estimation (ADAM) is fast and requires low memory for computation [74], it was selected as the optimisation parameter for the network used in this study. ADAM is a method that learns the LR on a parameter basis and is a combination of both adaptive gradient (AdaGrad) and root mean square (RMSProp).

4.1.10. Popular CNN Models

CNN models are formed using the combinations of parameters mentioned in the above subsections. The combinations of layers and the type of parameters used are often application-based and applied to solve a bigger problem. In this study, VGG16 [75] and ResNet50 [76] were applied based on the work done by [77,78], the models with their salient features are briefly discussed as follows.

VGGNet

- Stands for Visual Geometry Group
- Consists of 13 convolutional layers with three fully connected layers, hence the name VGG16.
- Each convolutional layer has kernel size = 3 with stride = 1 and padding = same.
- Each max-pooling layer has kernel = 2 and stride =2.

ResNet50

- Stands for Residual Network.
- A deep network, having 50 layers.
- It popularised batch normalisation.
- It uses skip connection to add information on output from a previous layer to the next layer.

4.2. CNN for Semantic Segmentation

Semantic segmentation is a process of assigning a label to each pixel in an image such that pixels with the same label are connected via some visual or semantic property [79]. In order to carry out semantic segmentation, the spatial information needs to be retained. Hence no fully connected layers are used, which is why they are called fully convolutional networks.

4.2.1. Moving from a Fully Connected to a Fully Convolution Network

This is where all fully connected layers are converted into 1×1 convolutional layers. In the case of labelling, the output is a 1D vector giving probabilities of the input belonging to n classes. In the case of segmentation, an output layer is a group of 2D probability-maps of each pixel belonging to each class. These are known as score maps. The score maps are coarse as throughout the network; the information (image) has been down-sampled to absorb minute details. Therefore, to make the output compatible with the input in size, up-sampling is required.

Up-sampling can be done using either bilinear interpolation or cubic interpolation (or similar techniques). One way of up-sampling is via skip-connections or shortcut connections. In skip-connection, the feature maps obtained as the output from the max-pooling layers are up-sampled using bilinear interpolation and added to the output score maps. The method works well but requires some amount of learning to up-sample the score maps and feature maps to match it to the size of the input image. In order to minimise the amount of learning, another method encoder-decoder is widely used. Here, the layers which down-sample the input are the part of the encoder and the layers which up-sample are part of the decoder. Three key fully connected models, SegNet [80], UNet [81], and Pyramid Scene Parsing Network (PSPNet) [82] are used in this study. A brief description of the models is given in the following subsections.

4.2.2. SegNet Model

SegNet works with encoder-decoder architecture, followed by a pixel-wise classification layer for multiple classes. Encoders extract the most relevant features from the given input. The decoder uses the information from encoder to up-sample the output (Figure 6).

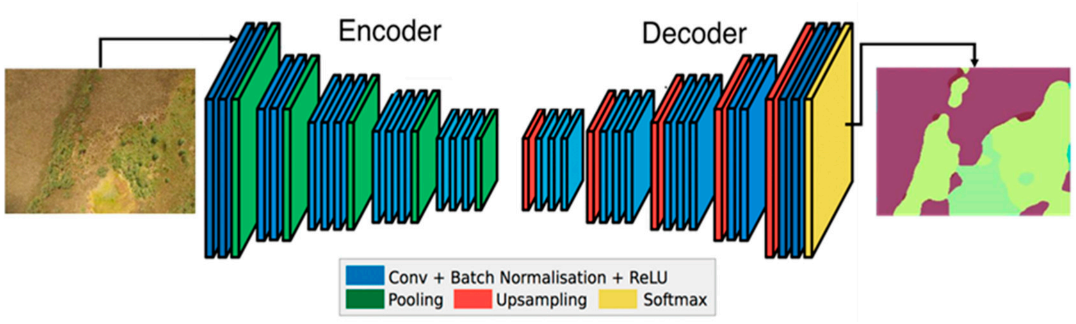


Figure 6. SegNet architecture for semantic segmentation for bog-ecotope semantic segmentation.

The up-sampling technique used by the decoder is known as max-unpooling. Max-unpooling eliminates the need for learning to up-sample (as was required in skip-connections) as shown in Figure 7. Based on the location of the maximum value, the max-pooled values are placed. The remainder of the matrix is loaded with zeros. Convolution is done using any CNN models (as discussed in Section 4.1) using this layer.

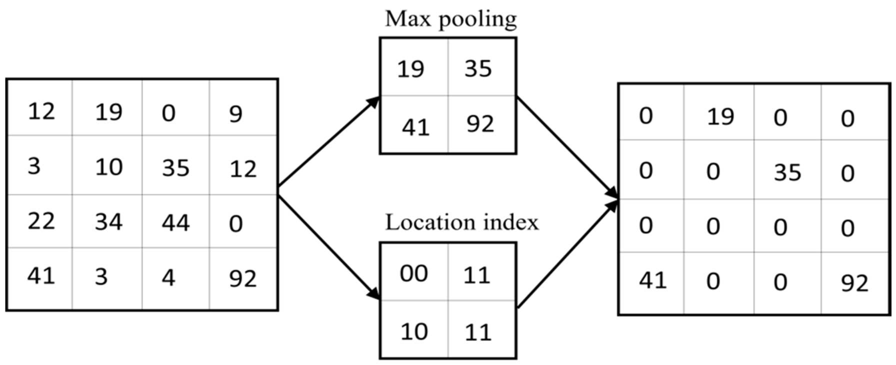


Figure 7. Pooling and unpooling for semantic segmentation.

4.2.3. UNet Model

UNet network carries out the transpose convolution (encoder-decoder) and also uses skip connections (Figure 8).

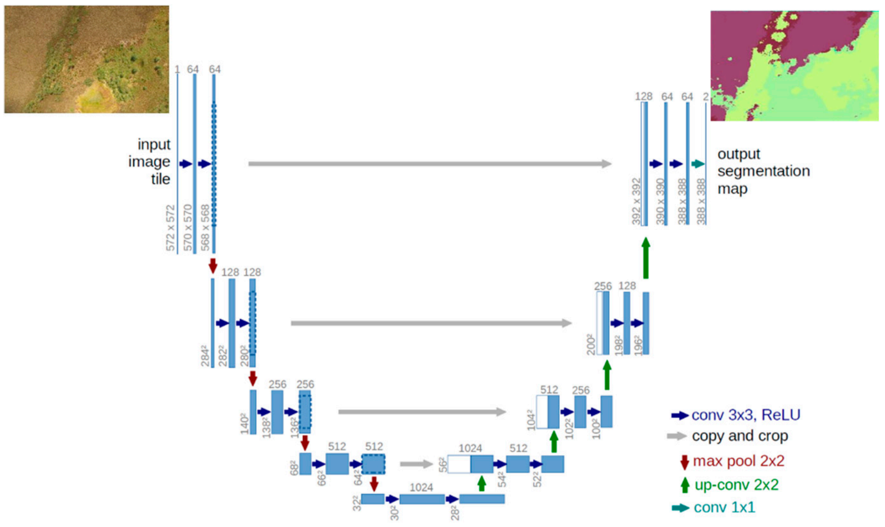


Figure 8. UNet architecture for semantic segmentation for bog-ecotope semantic segmentation.

At every layer in the decoder side, the network finds a corresponding feature map (of the same size) from the encoder and adds (1 × 1 convolution) to the score map. This way, the size of the feature map is always in sync. Due to its architecture and depth, UNet is most widely used in biomedical image analysis.

4.2.4. PSPNet Model

PSPNet stands for Pyramid Scene Parsing Network. This network incorporates the scene and global features for scene parsing and semantic segmentation as shown in Figure 9.

The pyramid pooling module in PSPNet fuses the features in four scales: coarse (1 × 1), 2 × 2, 3 × 3, and 6 × 6. The up-sampling done is a bilinear interpolation, and all the features are concatenated as the final pyramid pooling global feature [82]. The spatial pyramid pooling technique eliminates the need for using the input image of a specific size, which is used in SPPNet [83].

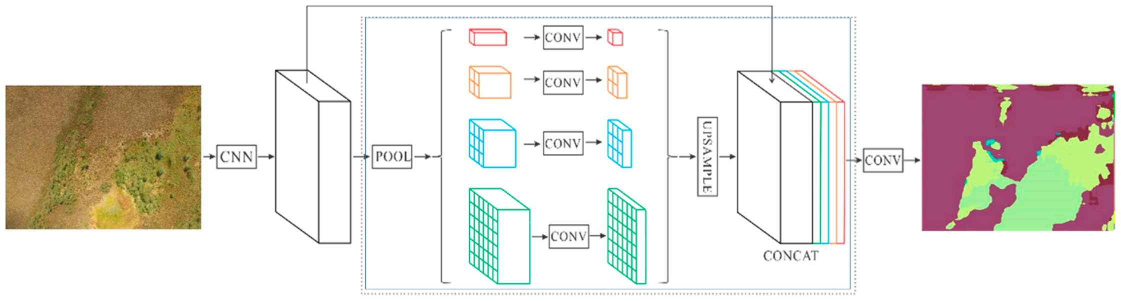


Figure 9. Pyramid Scene Parsing Network (PSPNet) architecture for semantic segmentation for bog-ecotope semantic segmentation.

4.3. Methodology for the Comparison between CNN Models for the Case Study on Raised Bog Drone Images

Using the drone images captured on 21st April 2019, semantic segmentation using various CNN architectures was applied to identify and label the ecotopes present on Clara Bog. The entire computation was performed in python v3.7 [84] using GPU (NVIDIA Tesla K40C 12GB CUDA), accessed remotely from trinity college high performing computer (TCHPC), and partly on google virtual machine (Tesla K40C 12GB). The study uses the repository in [85].

4.3.1. Training Data Preparation

In order to smoothly run the semantic segmentation, the preparation of training data was done as follows

1. Forty drone images were manually labelled using MATLAB-Image Labeler app [36].
2. The labels (in .mat format) were converted into JPG.
3. The images and labels were resized in order to use the GPU memory efficiently and to speed up the process. For resizing, the images were shrunk in the order of 2^n such that the classes were clearly distinguishable. The resizing of the images was done using a bilinear interpolation technique.
4. The images were resized from 3000 × 4000 to 512 × 1024 ($2^9 \times 2^{10}$) for further use. The size of the image is kept rectangular in order to maintain the aspect ratio of the original drone imagery. The ratio can be decided with respect to the application. For this study, to have a fair comparison between ML and DL methods, the size of the imagery was not reduced to smaller patches. Alternatively, patches of the same size ($2^9 \times 2^{10}$) can be extracted with overlapping. For this study, the small patches did not cover all the ecotopes. In a single patch, at maximum, only two ecotope classes were covered. This is due to the large size of the raised bog in the application. Therefore, to incorporate the maximum number of ecotope classes in a single image and to avoid any information loss, resizing of the images was done (instead of extracting the patches).
5. After reshaping, the images were renamed such that the images and their corresponding labels can be identified.

Steps (2–5) were repeated for all 40 images having all four ecotope classes mentioned in step 1. The final training data consisted of 40 images (both RGB and labelled) of the size $2^9 \times 2^{10,}$ which was fed to the CNN models described in the next subsection. The testing was carried out on the rest of the 20 images.

4.3.2. Models Used for Semantic Segmentation

The models were created using a base network (tested on ImageNet) along with a segmentation architecture. Since CNN takes a considerable amount of time to train, only the most frequently used and tested models (in the literature) were compared. The optimisation algorithm used was SGD Adam with initial LR = 0.05 and L2 regularisation. Initially, a high LR was used, as it is reduced throughout the epochs by a factor of 10. The max number of epochs = 100, and images were shuffled at every epoch and a mini-batch size of 64. The loss between the labels given by the model and the actual (training) label at every epoch was calculated using cross-entropy loss described in Equation (9). The cross-entropy loss is commonly applied for classification applications, whereas loss like half mean square error is more common for regression tasks. Therefore, a cross-entropy loss was used here.

$$\textit{cross entropy} = -\frac{1}{N}\sum_{i\,=\,1}^{N}\sum_{j\,=\,1}^{n}\left(\hat{x}_i \log x_{i,j}\right) + \left(1-\hat{x}_{i,j}\right)\log\left(1-x_{i,j}\right) \tag{9}$$

where N is the total number of pixels, n is the total number of classes, x is the training label (input), and $\hat{x}$ is the output label as predicted by the models. Instead of training the network from scratch, one of the most common techniques is to use a pre-trained network. The idea is to transfer the information learned by the network and then fine-tune and train the classification layer of the model for our specific task. In this manner, given that the weights are already pre-trained for a large dataset, even with a small dataset, the performance is much more improved. Pre-trained weights also speed up the convergence process (to reach local minima, i.e., to overall minimise the loss). It is also considered better than random initialisation. For the four models listed below, 'ImageNet' dataset [86] was used to initialise the weights. Other details are mentioned in detail in [85]. The architecture for these models is shown in Figure 10.

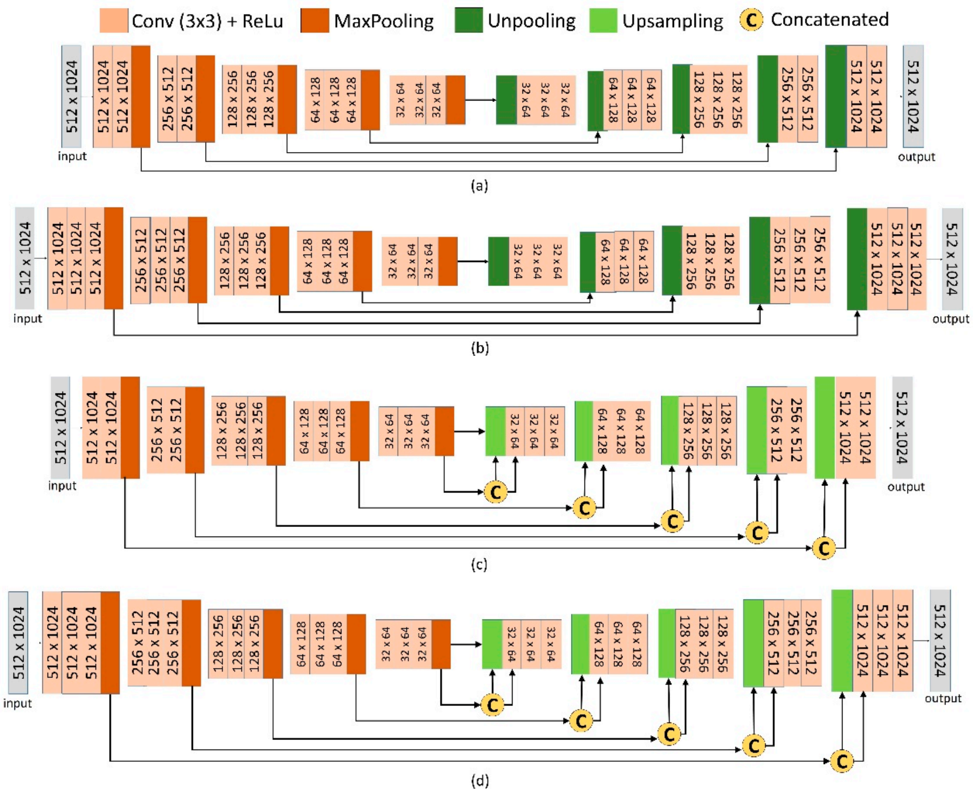


Figure 10. CNN Network architecture. (**a**) VGG16 + SegNet, (**b**) ResNet50 + SegNet, (**c**) VGG16 + UNet, (**d**) ResNet50 + UNet.

1. VGG16 base model with SegNet architecture.
2. ResNet50 base model with SegNet.
3. VGG16 with UNet.
4. ResNet50 with UNet

Figure 10a,b represents the SegNet architecture with VGG16 and ResNet50 as the base model, respectively. The left-hand side is the encoder, which has five blocks, and the layers are from the original VGG16 and ResNet models. The Max pooling operation is depicted by the red arrows. This operation reduces the image dimensions by 2×2. The Unpooling is depicted by the green arrows on the right-hand side of the figure(s). The operation ensures the size of the image was restored to the original size, and the output image has the same spatial-dimension as the input image. Figure 10c,d represents the UNet architecture with VGG16 and ResNet50 models, respectively. The network uses the original layers from the VGG16, ResNet50, with the UNet architecture. A clear U-connection can be seen in the figure. The skip connections were used, and upsampling was performed to restore the image dimensions. A concatenated operation was applied to implement the skip connections to combine them with the corresponding feature map (image). The unpooling, skip connections, and upsampling functions were used to ensure that the size of the output image is the same as the input image mentioned in Section 4.3.1.

For a specific task of semantic segmentation, dedicated segmentation based dataset was also used for initialising the weights. For the PspNet, the pre-training was done using ADE20K data [87], and Cityscapes dataset [88]. The ADE20K dataset has 21,200 images of various day to day scenes. The Cityscapes data contains images taken from video frames (≈20,000 coarse images) from 50 cities taken in spring, summer, and fall seasons. The models used are listed below, and the layers and architecture are described in Figure 9.

5. PspNet trained on ADE 20K dataset.
6. PspNet trained on Cityscapes dataset.

5. Results

Figure 11 depicts the segmentation results from both ML and DL techniques for a drone image (sized 512×1024) taken of Clara bog. The segmentation was carried out for four ecotope classes present in the drone image captured in the spring season. The accuracy and results are further discussed in this section.

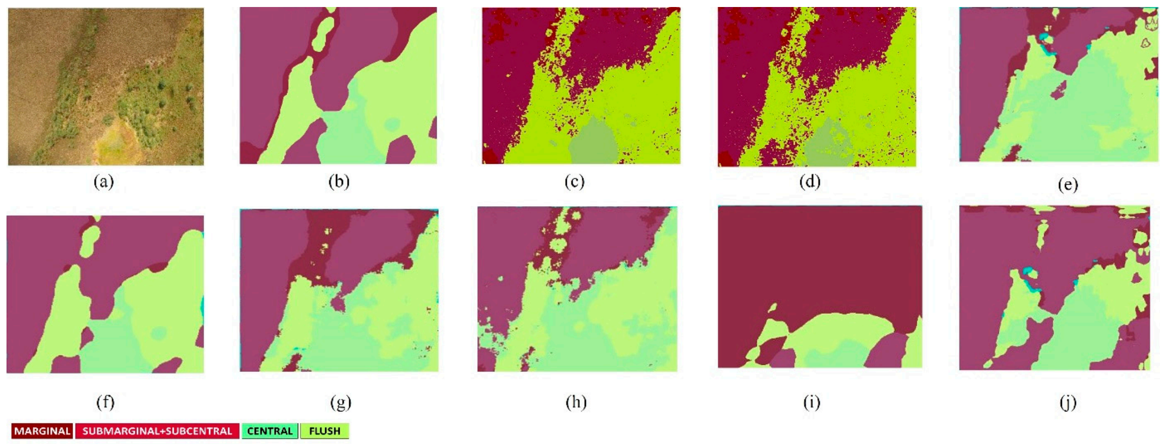


Figure 11. Segmentation results. (**a**) Drone image, (**b**) ground truth labelled image, (**c**) machine learning (ML) (random forest (RF) + Graph cut) segmentation using RGB features, (**d**) ML (RF + Graph cut) segmentation using RGB and textural features, (**e**) deep learning (DL) semantic segmentation using SegNet and VGG 16 model, (**f**) DL semantic segmentation using SegNet and ResNet50 model, (**g**) DL semantic segmentation using UNet with VGG16 model, (**h**) DL semantic segmentation using UNet with ResNet50 model, (**i**) DL semantic segmentation using PSPNet (Cityscapes), (**j**) DL semantic segmentation using PSPNet (ADE 20k).

5.1. Machine Learning

As discussed in Section 3, the ML classifiers were tested for model accuracy (5-fold validation), misclassification cost, and training time. Table 2 depicts the metric calculated over the entire 70% training data (as discussed in Section 3).

RF was chosen to be the best performing classifier, and further segmentation using Graphcut algorithm was performed using the results from RF. The segmentation is a post-classification area based smoothing process. The final segmented image was checked against a fully manually labelled image to give overall accuracy (OA). The OA is the ratio of true positives (TP) with a total number of pixels (Equation (10)).

$$OA = \frac{TP}{TP + FP + FN + TN} \tag{10}$$

where, TP = true positives, FP = false positives, FN = false negatives, and TN = true negatives. This was done for visible bands (RGB) and RGB + textural features. For proper comparison between ML and DL, the image was resampled from its original size (3000 × 4000) to a smaller scale (512 × 1024). The resampling of the image was done using bilinear interpolation [89]. Table 3 depicts the accuracies obtained by using a random forest classifier along with graph cut segmentation for both the sizes of the image. Since, the image used in segmentation using DL techniques is also resized, for an accurate comparison, the resized image (512 × 1024) was used in the further analysis.

Table 3. Segmentation accuracies using ML.

	RGB Features	RGB + Textural Features
Original size (3000 × 4000)	83.3	85.1
Resized (512 × 1024)	82.9	84.8

As can be seen from Figure 11b,c, there is not much difference in the segmentation using RGB with or without textural features. However, the textural features do add extra information and are known to be highly useful when there is a terrain variation in the scene. However, in this application, where the ecotopes under consideration are low-lying, homogenous communities, the addition of textural features did not improve accuracy very significantly—the OA only increasing by approximately 2%.

5.2. Deep Learning

The semantic segmentation using CNNs was performed for 100 epochs. The LR was decreased by a factor of 10 each time a model's accuracy was saturated. The overall accuracy performed on testing data (OA is also calculated for testing data) of all the models is shown in Figure 12.

There is a jump of an average ≈32% in OA from the first to last epoch, with the PspNet model and ResNet50+SegNet showing the maximum increase in OA (≈30%, 25% respectively). The cross-entropy loss decreased by an average of ≈28% for the CNN models under consideration. This decrease happens by reducing the LR. Although accurate, a detailed analysis of per-class accuracy is required to make an informed decision about the best CNN architecture for the segmentation for this particular application in the identification of raised bog vegetation ecotopes. The per-class analysis is done to make sure there is no overfitting. As seen from Figure 11i, a model can lead to overfitting, giving sufficient accuracy but incorrect classification.

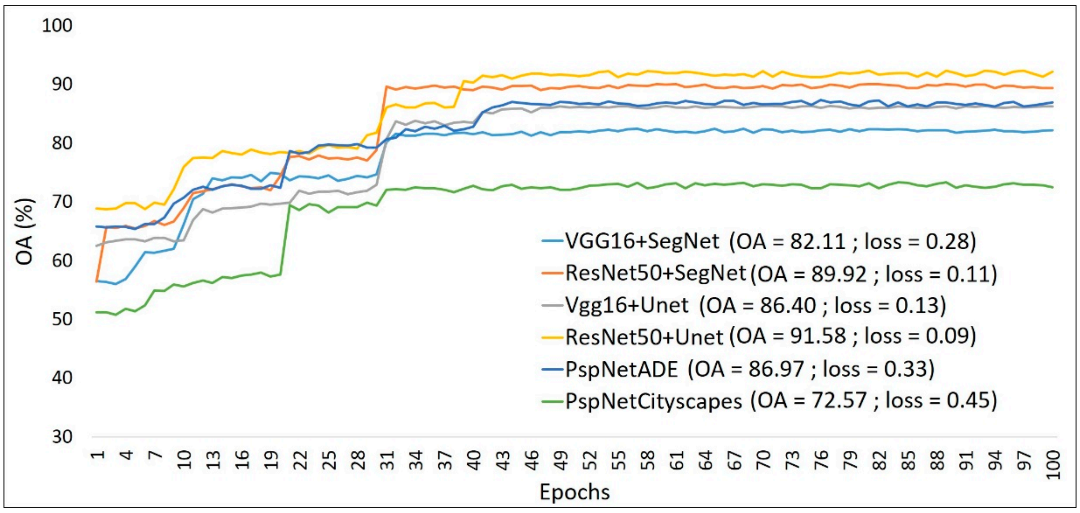


Figure 12. Overall accuracy over 100 epochs used for all CNN architectures for semantic segmentation of ecotopes in Clara Bog.

Table 4 describes the confusion matrix for every community, and both ML and DL algorithms, which is discussed further in Section 6. Other accuracy checking parameters like Precision, Recall, and F1-score were also calculated for every community (ecotope) under consideration. Equations (11)–(13) give the formula to calculate the above-stated accuracy parameters.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

Table 4. Confusion matrix per community per segmentation model (ML and DL).

	RF (RGB)				**RF (RGB + TEXTURAL)**				**SEGNET + VGG16**				**SEGNET + RESNET50**			
	M	SMSC	C	AF	M	SMSC	C	AF	M	SMSC	C	AF	M	SMSC	C	AF
M	58,405	1012	988	36,321	59,781	1598	1002	38,241	43,872	8854	2500	71,005	62,870	1631	835	16,360
SMSC	734	155,583	4979	2033	600	188,296	4608	587	7952	122,544	15,691	9514	3000	162,651	3005	2639
C	328	3862	77,939	44,321	256	4150	83,930	34,658	2831	18,529	77,369	73,108	967	4895	98,584	28,330
AF	38,010	3211	43,509	142,707	39,023	3079	32,584	112,589	65,896	21,251	64,211	99,833	18,470	14,110	10,358	148,383
Precision	0.59	0.94	0.61	0.62	0.59	0.96	0.68	0.66	0.35	0.79	0.45	0.40	0.77	0.95	0.74	0.78
Recall	0.60	0.95	0.61	0.63	0.60	0.95	0.69	0.66	0.36	0.72	0.48	0.39	0.74	0.89	0.87	0.76
F1 score	0.60	0.94	0.61	0.62	0.60	0.95	0.68	0.66	0.36	0.75	0.47	0.40	0.75	0.92	0.80	0.77
	UNET + VGG16				**UNET + RESENET50**				**PSPNET ADE20K**				**PSPNET CITYSCAPES**			
	M	SMSC	C	AF	M	SMSC	C	AF	M	SMSC	C	AF	M	SMSC	C	AF
M	82,589	9510	3258	36,951	73,897	1008	258	3371	36,351	9822	631	5311	128,890	78,353	36,118	63,001
SMSC	10,254	146,933	9800	19,759	4096	152,363	3690	15,892	15,200	210,052	6323	28,200	73,570	107,781	2988	4820
C	4523	12,967	96,582	35,489	982	5183	90,258	28,105	987	7921	96,587	3715	38,562	5815	32,510	12,377
AF	32,563	19,638	34,822	83,417	5101	14,852	22,110	155,446	3074	21,520	5300	127,296	58,360	7826	13,524	57,450
Precision	0.62	0.79	0.65	0.49	0.94	0.87	0.72	0.79	0.70	0.81	0.88	0.81	0.42	0.57	0.36	0.42
Recall	0.64	0.78	0.65	0.47	0.88	0.88	0.78	0.77	0.65	0.84	0.89	0.77	0.43	0.54	0.38	0.42
F1 score	0.63	0.78	0.66	0.48	0.91	0.87	0.75	0.78	0.67	0.83	0.89	0.79	0.43	0.55	0.37	0.42

6. Discussion

The study describes methods to map vegetation communities in a raised bog 'Clara Bog' located in Ireland using drone images from DJI Inspire 1™ drone captured during the spring season. The size of the images were 3000×4000, and 40 images were used as training. Furthermore, both ML, DL algorithms were tested for the rest of the 20 images. The study shows that high-resolution (1.8 cm) RGB images are adequate for mapping vegetation communities. However, a key challenge associated with RGB images is the change in intensity due to sunlight conditions, particularly in a temperate climate like Ireland, where sunlight levels are rarely constant for long. Therefore, in this study, all the images with significantly different light conditions were removed. The use of a colour correction technique could be a possible solution to this problem, which is a domain yet to be explored. Similarly, the addition of textural properties does create the challenge of increasing the computations (time and complexity). The segmentation is done using 13 features instead of three, thereby being more computationally expensive.

Initially, a comparative analysis of the state-of-the-art classifiers was performed (Table 2). It was seen that the RF ensemble classifier outperformed the other classifiers. The RF classifier uses bootstrapping for forming multiple trees leading to reduced possibilities of overfitting of the data. The SVM classifier with RBF kernel had similar accuracy and misclassification cost as RF, but with twice the training time. Hence, RF was deemed to be the best choice for drone image classification with model accuracy of 92%. As pixel-based segmentation often fails to take the contextual (area-based) information into account; therefore, to form segments based on area, graph cut segmentation was subsequently applied. Out of the 40 training images, only a part of labelled pixels ($\approx$12k) was input to the ML model. The entire processing time of ML segmentation was $\approx$30 min.

This was done for both RGB and RGB + textural images. The images were resampled to $2^9 \times 2^{10}$ for proper comparison with deep learning algorithms (discussed later). It has to be noted that the aspect ratio of the imagery was maintained while resampling it. This was done mainly to keep the textural properties intact. The authors of [37] explain that in order to capture textural properties, the size of the image/sliding window should be carefully chosen. Therefore, a decrease in the size of the image (or change in aspect ratio) can lead to a change in textural properties. Table 3 shows that the resampling using bilinear interpolation did not make a big difference in the OA. The resampled image with textural properties performs comparably to the original image. The OA with textural properties is also comparable to OA with just RGB for this application with a low-lying homogenous area of interest. Overall, the textural properties perform the best segmentation with both the original-sized image and the rescaled image.

From Figure 11c,d, it can be seen that using textural properties, the ecotopes like SMSC and AF are differentiated better (see Table 4). Likewise, from Table 4, it can be seen TP for the C ecotope increases with the addition of textural properties but decreases for the AF ecotope. The decrease in misclassified pixels (FP, FN) between SMSC and AF has led to an increase in precision and recall for the SMSC ecotope. There is a definite increase in accuracy for the C and AF ecotopes by using textural features, whereas, the SMSC and M ecotopes are identified with similar precision, recall, and f1 score values for both the images.

The deep learning technique used for segmentation is semantic segmentation using CNN models. In this study, six different models were tested for the semantic segmentation to identify the different bog-ecotopes. The training data for the CNN models consisted of 40 images containing all the ecotopes in different orientations and lighting (brightness). The size of the training data is a notable factor in this study, as for many applications, 100s of images are more usually required for training such CNN models. This study demonstrates the usage of minimum labelled training images for attaining the segmentation, given that 40 images seemed to be sufficient for this application as the weights were initialised using ImageNet dataset having 1000 different classes. This reduces the dependence on the extensive training dataset and also is faster [90]. All these 40 images were resized to $2^9 \times 2^{10}$ for efficiently performing semantic segmentation. For an application involving a prominent area such as this, the classes are also sparsely located. Therefore, cropping or extracting patches from the images was leading to a reduction in classes (ecotopes) covered in an image. In order to make sure that the model identifies all the ecotopes, the images were resized. Nevertheless, for an application where the classes are located close enough (spatially), cropping/extracting patches can be a viable option.

The algorithms were run for 100 epochs, after which the accuracy was becoming saturated. The computation time was ≈700 min per model for 100 epochs. It was decided not to increase the number of epochs as it may lead to overfitting of the model [91]. The LR was decreased with epochs when OA saturated. This decrease leads to faster convergence and an increase in accuracy. Without decreasing the LR, if the same LR is continued, one may still get high accuracy, but it would require a massive number of epochs; therefore, it is not recommended. There is an apparent increase in accuracy using DL methods when compared to ML methods. At the end of the epochs, it is clear that SegNet and UNet architecture with ResNet50 yield the best results for the semantic segmentation of bog-ecotopes. In comparison, the VGG16 base model has led to the over-classification of ecotopes such as M, AF. The VGG model has been shown to be effective when there is noise in the data but does not perform well when the brightness of the images changed [92]. This explains the low accuracy of the model, as the images had different lighting due to variable weather conditions. Figure 11e–j depicts the DL segmentation results. It can be seen that the segmentation using SegNet and UNet is similar for ecotopes like SMSC and C, but is different for AF and M ecotopes.

The study also demonstrates the use of transfer learning by using a segmentation specific pre-trained PspNet model. This model was pre-trained using ADE20K and cityscapes image set instead of widely applicable ImageNet. In our application, the usage of these segmentation datasets was not successful as the weights were calculated for a specific task of segmenting areas of traffic, cars, houses, pavements, etc. Additionally, due to the uniqueness of these communities, the weights transferred from the pre-trained models were not accurate. In order to use transfer learning, the model selected should be pre-trained using similar categories as the application.

For making the final decision of the best CNN architecture, the accuracy parameters for every ecotope were considered. Table 4 shows that the SMSC ecotope is identified quite well using all the CNN models, with the exception of the PspNet model pre-trained with cityscapes images. Using the base model ResNet50, the ecologically important, peat-forming communities (the SMSC, and C ecotopes) are better identified using SegNet than UNet. Using PspNet (ADE20K), the C ecotope was identified the best, although the OA of the model is low. Therefore, taking into consideration the OA, precision, recall, and f1-score of all the communities, SegNet architecture with the ResNet50 base model appears to be the best choice for drone image segmentation in relation to identifying raised bog vegetation types.

The best OA recorded from ML was 85%, and from DL was 91%. However, the most appropriate technique for this study was not decided just on the basis of OA. For applying the technique to new applications, other parameters cannot be ignored. For example, a lot more training data was required for using DL as compared to ML. Similarly, time and hardware also play a significant role in deciding the best technique. Table 5 summarises the essential pros and cons of the two techniques.

It is clear that there are many pros and cons of both techniques, as described in this study. The main idea behind using remote sensing techniques is to reduce the amount of manual fieldwork that is required for monitoring the wetlands. This includes minimising the training data given as input to the classifiers. Additionally, the idea is to automate the process in the simplest way possible, given that the availability of high performing computers or GPUs cannot always be guaranteed, in order to optimise the speed. Keeping in mind the above requisites, the ML technique is the clear choice for our application. Whilst DL techniques can be used once there is enough labelled data created from all the wetlands such that all the species are covered, in the case of a new wetland, which contains new species to be mapped, a clear indication of the species (with full coverage) is required. Therefore, DL is more advantageous to use for more global or applied applications, whereas for a more specific application such as this where not enough training data is available, ML can produce accuracies almost comparable to the DL.

Table 5. Pros and cons of ML vs. DL for mapping ecotopes, Clara bog.

Pros	Cons
MACHINE LEARNING ALGORITHMS	
1. Very fast training (≈30 min per model)	1. Cannot be applied globally, i.e., does not work on augmented data.
2. Needs less training data (≈12k labelled pixels in 40 images)	2. In order to re-use the same model, input has to be consistent with original images or to be modified.
3. No need of size alteration; accepts the input of any size.	3. Can require parameterisation, manual handling.
4. No need for HPCs or GPUs; works well with CPU.	4. Textural and additional features have to be input separately; does not learn by itself.
5. Provides good accuracy (OA = 85%)	
DEEP LEARNING ALGORITHMS	
1. Can be applied globally for multiple applications and works well with augmented data.	1. Slower training process (≈700 min per model).
2. Works well with low resolution imagery.	2. Needs a considerable amount of training data or requires a pre-trained model (≈48 × 106 labelled pixels in 40 images).
3. Learns textural features on its own, require no additional inputs.	3. Requires alteration of images with respect to size for faster computation and analysis.
4. Provides good results (OA = 91%)	4. It is recommended the use of GPUs with good RAM for running CNN efficiently.

Finally, the drone images are very high resolution and can be captured at any given time. However, although practical, drone image capturing does have certain limitations. The first limitation of using drones is the length of the battery life. For example, on average, the drone (such as DJI Inspire 1) the battery will only last approximately 15 min and so to cover a large area, many batteries are required. Therefore, in the future, in order to make the process cost-effective, drone images can be used in conjunction with the freely available satellite images. Satellite images give excellent coverage and proper temporal resolution meaning that the usage of drones and satellites together should be advantageous.

7. Conclusions

This study aimed at providing mapping of vegetation in wetlands using image segmentation. For this, ML and DL algorithms were compared by applying them to a set of drone images of Clara Bog, a raised bog located in the middle of Ireland. The images were captured using DJI Inspire 1^{TM} drone (RGB sensor), with the open-source and freely available Pix4DCapture application. Using ML, a total of six different state-of-the-art pixel-based classifiers were compared, out of which, the best ML algorithm for the given dataset was shown to be the RF classifier (model accuracy = 92.9%). For ML image segmentation, RF classifier was used with maximum a-posteriori graph cut segmentation. It was seen that accuracy is improved by ≈2% after addition of textural features (OA = 85.1%) when compared to the original RGB image (83.3%), and ecologically important communities such as central ecotope were mapped better.

Apart from ML, the study also describes image segmentation or 'semantic segmentation' using DL methods. A detailed account on the selection of variables for the DL segmentation was presented. The study was done on a combination of six architectures and base models. For the given dataset, the ResNet50 base model with both UNet (OA = 91.5%) and SegNet (OA = 89.9%) architecture performs very well. ResNet50+SegNet model was deemed best, as it was able to identify complex vegetation communities, such as SMSC ecotope better. All the models were run for 100 epochs, and it was seen the accuracy saturated after ≈35 epochs. It was seen that for mapping ecotopes in a raised bog, transfer of initial weights from wide-ranged ImageNet outperforms the segmentation-specific datasets like ADE20K or Cityscapes.

Overall, the accuracy of the DL was ≈4% higher than the ML methods. Additionally, the DL method does not require any colour correction or the addition of extra textural features. However, DL requires a large amount of initial labelled training data ($\approx 48 \times 10^6$ pixels). On the other hand, the ML algorithm requires much less training data (≈12,000 labelled pixels) and is much faster (≈30 times) when compared to CNNs. Therefore, in retrospect, for such a specific application as the wetland mapping application, it was considered that the ML approach is more suitable. This would be particularly useful for any un-surveyed wetland, where the minimum amount of information of the vegetation communities is required to produce accurate maps.

Author Contributions: Conceptualization, S.B. and B.G.; methodology, S.B.; software, S.B.; validation, S.B., B.G. and L.G.; formal analysis, S.B.; investigation, S.B., B.G.; resources, S.B., L.G.; data curation, S.B.; writing—original draft preparation, S.B.; writing—review and editing, S.B., L.G., B.G.; visualisation, B.G.; supervision, B.G.; project administration, L.G.; funding acquisition, L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study is funded by the Environmental Protection Agency of Ireland (EPAIE) (Grant No.: 2016-W-LS-13).

Acknowledgments: The authors would like to thank Jonathan Munro (University of Bristol) and Yunis Ahmad Lone for their advice, and Trinity College High Power Computing (TCHPC) department for providing the necessary hardware and support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhatnagar, S.; Gill, L.; Regan, S.; Naughton, O.; Johnston, P.; Waldren, S.; Ghosh, B. Mapping Vegetation Communities Inside Wetlands Using Sentinel-2 Imagery in Ireland. *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *88*, 102083. [CrossRef]
2. Hirano, A.; Madden, M.; Welch, R. Hyperspectral image data for mapping wetland vegetation. *Wetlands* **2003**, *23*, 436–448. [CrossRef]
3. Pengra, B.W.; Johnston, C.A.; Loveland, T.R. Mapping an invasive plant, Phragmites australis, in coastal wetlands using the EO-1 Hyperion hyperspectral sensor. *Remote. Sens. Environ.* **2007**, *108*, 74–81. [CrossRef]
4. Álvarez-Taboada, F.; Araújo-Paredes, C.; Julián-Pelaz, J. Mapping of the Invasive Species Hakea sericea Using Unmanned Aerial Vehicle (UAV) and WorldView-2 Imagery and an Object-Oriented Approach. *Remote. Sens.* **2017**, *9*, 913. [CrossRef]
5. Baena, S.; Moat, J.; Whaley, O.; Boyd, D.S. Identifying species from the air: UAVs and the very high resolution challenge for plant conservation. *PLoS ONE* **2017**, *12*, e0188714. [CrossRef]
6. Dvořák, P.; Müllerová, J.; Bartaloš, T.; Brůna, J. Unmanned Aerial Vehicles for Alien Plant Species Detection and Monitoring. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2015**, *40*, 83–90. [CrossRef]
7. Hill, D.; Tarasoff, C.; Whitworth, G.E.; Baron, J.; Bradshaw, J.; Church, J.S. Utility of unmanned aerial vehicles for mapping invasive plant species: A case study on yellow flag iris (Iris pseudacorus L.). *Int. J. Remote. Sens.* **2016**, *38*, 2083–2105. [CrossRef]
8. Ruwaimana, M.; Satyanarayana, B.; Otero, V.; Muslim, A.M.; Muhammad, A.M.; Ibrahim, S.; Raymaekers, D.; Koedam, N.; Dahdouh-Guebas, F. The advantages of using drones over space-borne imagery in the mapping of mangrove forests. *PLoS ONE* **2018**, *13*, e0200288. [CrossRef]
9. Chabot, D.; Dillon, C.; Shemrock, A.; Weissflog, N.; Sager, E.P.S. An Object-Based Image Analysis Workflow for Monitoring Shallow-Water Aquatic Vegetation in Multispectral Drone Imagery. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 294. [CrossRef]
10. Han, Y.-G.; Yoo, S.H.; Kwon, O. Possibility of applying unmanned aerial vehicle (UAV) and mapping software for the monitoring of waterbirds and their habitats. *J. Ecol. Environ.* **2017**, *41*, 21. [CrossRef]
11. Zheng, H.; Cheng, T.; Li, D.; Zhou, X.; Yao, X.; Tian, Y.; Cao, W.; Zhu, Y. Evaluation of RGB, Color-Infrared and Multispectral Images Acquired from Unmanned Aerial Systems for the Estimation of Nitrogen Accumulation in Rice. *Remote. Sens.* **2018**, *10*, 824. [CrossRef]
12. Govender, M.; Chetty, K.; Bulcock, H. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water SA* **2009**, *33*. [CrossRef]
13. Boon, M.A.; Greenfield, R.; Tesfamichael, S. Wetland assessment using unmanned aerial vehicle (UAV) photogrammetry. *Remote. Sens. Spat. Inf. Sci.* **2016**, *XLI-B1*, 781–788.
14. Treboux, J.; Genoud, D. Improved Machine Learning Methodology for High Precision Agriculture. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 1–6.

15. Pap, M.; Király, S.; Moljak, S. Investigating the usability of UAV obtained multispectral imagery in tree species segmentation. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W18*, 159–165. [CrossRef]
16. Zuo, Z. Remote Sensing Image Extraction of Drones for Agricultural Applications. *Rev. Fac. Agron. Univ. Zulia* **2019**, *36*, 1202–1212.
17. Parsons, M.; Bratanov, D.; Gaston, K.J.; Gonzalez, F. UAVs, hyperspectral remote sensing, and machine learning revolutionising reef monitoring. *Sensors* **2018**, *18*, 2026. [CrossRef]
18. Miyamoto, H.; Momose, A.; Iwami, S. UAV image classification of a riverine landscape by using machine learning techniques. *EGU Gen. Assem. Conf. Abstr.* **2018**, *20*, 5919.
19. Zimudzi, E.; Sanders, I.; Rollings, N.; Omlin, C. Segmenting mangrove ecosystems drone images using SLIC superpixels. *Geocarto Int.* **2018**, *34*, 1648–1662. [CrossRef]
20. Höser, T.; Kuenzer, C. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. *Remote. Sens.* **2020**, *12*, 1667. [CrossRef]
21. Lee, D.; Kim, J.; Lee, D.-W. Robust Concrete Crack Detection Using Deep Learning-Based Semantic Segmentation. *Int. J. Aeronaut. Space Sci.* **2019**, *20*, 287–299. [CrossRef]
22. Zhang, C.; Wang, L.; Yang, R. Semantic segmentation of urban scenes using dense depth maps. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–10 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 708–721.
23. Montoya-Zegarra, J.A.; Wegner, J.D.; Ladický, L.; Schindler, K. Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2015**, *2*, 127–133. [CrossRef]
24. Dechesne, C.; Mallet, C.; Le Bris, A.; Gouet-Brunet, V. Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery. *ISPRS J. Photogramm. Remote. Sens.* **2017**, *126*, 129–145. [CrossRef]
25. Cui, B.; Zhang, Y.; Li, X.; Wu, J.; Lu, Y. WetlandNet: Semantic Segmentation for Remote Sensing Images of Coastal Wetlands via Improved UNet with Deconvolution. In Proceedings of the International Conference on Genetic and Evolutionary Computing, Qingdao, China, 1–3 November 2019; Springer: Singapore, 2019; pp. 281–292.
26. Jiang, J.; Feng, X.; Liu, F.; Xu, Y.; Huang, H. Multi-Spectral RGB-NIR Image Classification Using Double-Channel CNN. *IEEE Access* **2019**, *7*, 20607–20613. [CrossRef]
27. Kentsch, S.; Caceres, M.L.L.; Serrano, D.; Roure, F.; Donoso, Y.D. Computer Vision and Deep Learning Techniques for the Analysis of Drone-Acquired Forest Images, a Transfer Learning Study. *Remote Sens.* **2020**, *12*, 1287. [CrossRef]
28. Nigam, I.; Huang, C.; Ramanan, D. Ensemble knowledge transfer for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 1499–1508.
29. Do, D.; Pham, F.; Raheja, A.; Bhandari, S. Machine learning techniques for the assessment of citrus plant health using UAV-based digital images. In Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping III, Baltimore, MD, USA, 15–16 April 2019; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 10664, p. 1066400.
30. Bhatnagar, S.; Ghosh, B.; Regan, S.; Naughton, O.; Johnston, P.; Gill, L. Monitoring environmental supporting conditions of a raised bog using remote sensing techniques. *Proc. Int. Assoc. Hydrol. Sci.* **2018**, *380*, 9–15. [CrossRef]
31. Bhatnagar, S.; Ghosh, B.; Regan, S.; Naughton, O.; Johnston, P.; Gill, L. Remote Sensing Based Ecotope Mapping and Transfer of Knowledge in Raised Bogs. *Geophys. Res. Abstr.* **2019**, *21*, 1.
32. ESRI. *ArcMap Desktop*; (Version 10.6.1); Esri Inc.: Redlands, CA, USA, 2019.
33. ESRI "World Imagery" [High Resolution 30 cm Imagery]. Scale ~1:280 (0.03 m). Available online: http://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08febac2a9 (accessed on 25 November 2019).
34. Shi, J.; Malik, J. Normalised cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
35. Feng, Q.; Liu, J.; Gong, J. UAV remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote Sens.* **2015**, *7*, 1074–1094. [CrossRef]

36. *MATLAB*; Version R2019b; The MathWorks Inc.: Natick, MA, USA, 2019.
37. Tavares, J.; Jorge, R.N. Computational Vision and Medical Image Processing V. In Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing, VipIMAGE 2015, Tenerife, Spain, 19–21 October 2015; CRC Press: Boca Raton, FL, USA, 2015.
38. Schwenker, F.; Abbas, H.M.; El Gayar, N.; Trentin, E. Artificial Neural Networks in Pattern Recognition. In Proceedings of the 7th IAPR TC3 Workshop, ANNPR 2016, Ulm, Germany, 28–30 September 2016; Springer: New York, NY, USA, 2018.
39. Chai, H.Y.; Wee, L.K.; Swee, T.T.; Hussain, S. Gray-level co-occurrence matrix bone fracture detection. *WSEAS Trans. Syst.* **2011**, *10*, 7–16. [CrossRef]
40. Salem, Y.B.; Nasri, S. Texture classification of woven fabric based on a GLCM method and using multiclass support vector machine. In Proceedings of the 2009 6th International Multi-Conference on Systems, Signals and Devices, Jerba, Tunisia, 23–26 March 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 1–8.
41. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
42. Mardia, K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika* **1970**, *57*, 519–530. [CrossRef]
43. Stoer, M.; Wagner, F. A simple min-cut algorithm. *J. ACM* **1997**, *44*, 585–591. [CrossRef]
44. Ishida, T.; Kurihara, J.; Viray, F.A.; Namuco, S.B.; Paringit, E.C.; Perez, G.J.; Marciano, J.J.J. A novel approach for vegetation classification using UAV-based hyperspectral imaging. *Comput. Electron. Agric.* **2018**, *144*, 80–85. [CrossRef]
45. Braun, A.C.; Weidner, U.; Hinz, S. Support vector machines for vegetation classification–A revision. *Photogramm. Fernerkund. Geoinf.* **2010**, *2010*, 273–281. [CrossRef]
46. Laliberte, A.S.; Rango, A. Texture and scale in object-based analysis of subdecimeter resolution unmanned aerial vehicle (UAV) imagery. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 761–770. [CrossRef]
47. Özlem, A. Mapping land use with using Rotation Forest algorithm from UAV images. *Eur. J. Remote Sens.* **2017**, *50*, 269–279. [CrossRef]
48. Meng, X.; Shang, N.; Zhang, X.; Li, C.; Zhao, K.; Qiu, X.; Weeks, E. Photogrammetric UAV Mapping of Terrain under Dense Coastal Vegetation: An Object-Oriented Classification Ensemble Algorithm for Classification and Terrain Correction. *Remote Sens.* **2017**, *9*, 1187. [CrossRef]
49. Friedl, M.A.; Brodley, C.E. Decision tree classification of land cover from remotely sensed data. *Remote Sens. Environ.* **1997**, *61*, 399–409. [CrossRef]
50. Cheng, J.; Greiner, R. Comparing Bayesian network classifiers. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Stockholm, Sweden, 30 July–1 August 1999; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1999; pp. 101–108.
51. Balakrishnama, S.; Ganapathiraju, A. Linear discriminant analysis-a brief tutorial. *Inst. Signal Inf. Process.* **1998**, *18*, 1–8.
52. Cortes, C.; Vapnik, V. Support vector machine. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
53. Laaksonen, J.; Oja, E. Classification with learning k-nearest neighbors. In Proceedings of the International Conference on Neural Networks (ICNN'96), Washington, DC, USA, 3–6 June 1996; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 1996; Volume 3, pp. 1480–1483.
54. Liaw, A.; Wiener, M. Classification and regression by randomForest. *News* **2020**, *2*, 18–22.
55. Ross, Q.J. *C4. 5: Programs for Machine Learning*; Morgan Kaufmann Publishers: San Mateo, CA, USA, 1993.
56. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
57. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimisation in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [CrossRef] [PubMed]
58. MATLAB Wrapper for Graph Cut. Shai Bagon. Available online: https://github.com/shaibagon/GCMex (accessed on 12 December 2019).
59. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimisation via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [CrossRef]

60. Masci, J.; Meier, U.; Ciresan, D.; Schmidhuber, J.; Fricout, G. Steel defect classification with max-pooling convolutional neural networks. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2012; pp. 1–6.
61. Li, Q.; Cai, W.; Wang, X.; Zhou, Y.; Feng, D.D.; Chen, M. Medical image classification with convolutional neural network. In Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2014; pp. 844–848.
62. Sharma, S. Activation functions in neural networks. *Towards Data Sci.* **2017**, *6*, 310–316.
63. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
64. Özkan, C.; Erbek, F.S. The comparison of activation functions for multispectral Landsat TM image classification. *Photogramm. Eng. Remote Sens.* **2003**, *69*, 1225–1234. [CrossRef]
65. Karlik, B.; Olgac, A.V. Performance analysis of various activation functions in generalised MLP architectures of neural networks. *Int. J. Artif. Intell. Expert Syst.* **2011**, *1*, 111–122.
66. Bircanoğlu, C.; Arıca, N. A comparison of activation functions in artificial neural networks. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izimir, Turkey, 2–5 May 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 1–4.
67. Bottou, L. Online learning and stochastic approximations. *On-line Learn. Neural Netw.* **1998**, *17*, 142.
68. Fukumizu, K. Effect of batch learning in multilayer neural networks. *Gen* **1998**, *1*, 1E–03E.
69. Bottou, L. Stochastic gradient learning in neural networks. *Proc. Neuro Nımes* **1991**, *91*, 12.
70. Paine, T.; Jin, H.; Yang, J.; Lin, Z.; Huang, T. Gpu asynchronous stochastic gradient descent to speed up neural network training. *arXiv* **2013**, arXiv:1312.6186.
71. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, CA, 7–12 December 2015; Neural Information Processing Systems Foundation, Inc. (NIPS): San Diego, CA, USA, 2015; pp. 1135–1143.
72. Van Den Doel, K.; Ascher, U.; Haber, E. *The Lost Honour of l2-Based Regularization*; (Radon Series in Computational and Applied Math); De Gruyter: Berlin, Germany, 2013.
73. Atienza, R. *Advanced Deep Learning with Keras: Apply Deep Learning Techniques, Autoencoders, GANs, Variational Autoencoders, Deep Reinforcement Learning, Policy Gradients, and More*; Packt Publishing Ltd.: Birmingham, UK, 2018.
74. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimisation. *arXiv* **2014**, arXiv:1412.6980.
75. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
76. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 770–778.
77. Cheng, K.; Cheng, X.; Wang, Y.; Bi, H.; Benfield, M.C. Enhanced convolutional neural network for plankton identification and enumeration. *PLoS ONE* **2019**, *14*, e0219570. [CrossRef] [PubMed]
78. Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 169–175.
79. Ghosh, S.; Das, N.; Das, I.; Maulik, U. Understanding deep learning techniques for image segmentation. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 73. [CrossRef]
80. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
81. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
82. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 2881–2890.

83. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
84. Python Software Foundation. Python Language Reference, Version 3.7. Available online: http://www.python.org (accessed on 20 June 2020).
85. Divamgupta. Image-Segmentation-Keras. 2019. Available online: https://github.com/divamgupta/image-segmentation-keras.git (accessed on 20 June 2020).
86. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Berg, A.C. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
87. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ade20k dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 633–641.
88. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 3213–3223.
89. Andrews, H.C.; Patterson, C.L. Digital interpolation of discrete images. *IEEE Trans. Comput.* **1976**, *100*, 196–202. [CrossRef]
90. Liu, Y.; Starzyk, J.A.; Zhu, Z. Optimised approximation algorithm in neural networks without overfitting. *IEEE Trans. Neural Netw.* **2008**, *19*, 983–995.
91. Grm, K.; Štruc, V.; Artiges, A.; Caron, M.; Ekenel, H.K. Strengths and weaknesses of deep learning models for face recognition against image degradations. *IET Biom.* **2017**, *7*, 81–89. [CrossRef]
92. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimisation, network minimisation and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 4133–4141.