

## Article

# Preliminary Classification of Selected Farmland Habitats in Ireland Using Deep Neural Networks

Lizy Abraham <sup>1,\*</sup>, Steven Davy <sup>1</sup>, Muhammad Zawish <sup>1</sup>, Rahul Mhapsekar <sup>1</sup>, John A. Finn <sup>2</sup> and Patrick Moran <sup>3</sup>

<sup>1</sup> Walton Institute for Information and Communication Systems Science, Waterford Institute of Technology, X91 WR86 Waterford, Ireland; steven.davy@waltoninstitute.ie (S.D.); muhammad.zawish@waltoninstitute.ie (M.Z.); rahul.mhapsekar@waltoninstitute.ie (R.M.)

<sup>2</sup> Teagasc, Environment Research Centre, Johnstown Castle, Y35 TC97 Wexford, Ireland; john.finn@teagasc.ie

<sup>3</sup> Forest Environmental Research and Services Ltd. (FERS), Kilberry, C15 R6Y3 Navan, Ireland; pat.moran@fers.ie

\* Correspondence: lizy.abraham@waltoninstitute.ie

**Abstract:** Ireland has a wide variety of farmlands that includes arable fields, grassland, hedgerows, streams, lakes, rivers, and native woodlands. Traditional methods of habitat identification rely on field surveys, which are resource intensive, therefore there is a strong need for digital methods to improve the speed and efficiency of identification and differentiation of farmland habitats. This is challenging because of the large number of subcategories having nearly indistinguishable features within the habitat classes. Heterogeneity among sites within the same habitat class is another problem. Therefore, this research work presents a preliminary technique for accurate farmland classification using stacked ensemble deep convolutional neural networks (DNNs). The proposed approach has been validated on a high-resolution dataset collected using drones. The image samples were manually labelled by the experts in the area before providing them to the DNNs for training purposes. Three pre-trained DNNs customized using the transfer learning approach are used as the base learners. The predicted features derived from the base learners were then used to train a DNN based meta-learner to achieve high classification rates. We analyse the obtained results in terms of convergence rate, confusion matrices, and ROC curves. This is a preliminary work and further research is needed to establish a standard technique.

**Keywords:** farmland habitat; habitat classification; drone images; base-learner; meta-learner; stacked ensemble model



**Citation:** Abraham, L.; Davy, S.; Zawish, M.; Mhapsekar, R.; Finn, J.A.; Moran, P. Preliminary Classification of Selected Farmland Habitats in Ireland Using Deep Neural Networks. *Sensors* **2022**, *22*, 2190. <https://doi.org/10.3390/s22062190>

Academic Editor: Ivan Andonovic

Received: 19 February 2022

Accepted: 9 March 2022

Published: 11 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Habitat mapping can be utilized in a variety of applications in nature conservation. They serve as a guiding principle for monitoring inventories of natural areas, curating the networks of protected areas, environmental impact assessment, management planning, and target setting for ecological restoration. However, most such applications still rely on field-based methods. The research in this area is increasingly focused on the data available from satellite imagery. The main works are related to forest and vegetation mapping using LANDSAT images, WorldView-2, Sentinel-2, IKONOS, GeoEye, MERIS, radar, and LiDAR images [1–7]. Earth observation data offers new opportunities for environmental sciences and is transforming artificial intelligence-based methodologies because of the massive data with spatial, spectral, and temporal variations available from satellite sensors [8–10]. The use of traditional remote sensing images for land use monitoring and mapping has been widely used in many types of research from the early 2000s onwards [11–13]. The works of Cheng et al. [14] and Xie et al. [1] describe state-of-the-art technologies and possible datasets for land cover mapping and classification. There is increasing demand for applications to monitor ecosystems and assess their seasonal variations in the twenty-first

century as climate change and global warming are making significant impacts all over the world [15,16]. To implement effective biodiversity conservation and climate protection practices on an international and national scale, accurate mapping of habitats plays a vital role. One of the prominent works in this direction is the European Union's (EU) Copernicus Programme [17]. Through this program, global data can be obtained in real-time using airborne, ground-based, and seaborne-based systems, which can also be used for local and regional needs efficiently.

Many works have been published using the Copernicus Land Monitoring Service (CLMS), due to it being free and openly accessible to users [18–20]. The main constraint is that, since the spatial resolution of CLMS is at EU-level, it has limitations for more differentiated habitat identification at the scale of individual member countries. Another distinguished EU programme is the European Nature Information System (EUNIS) which accumulates the European data from multiple databases and is extensively used as the main reference for the research in ecology and conservation [21,22]. While the EU Copernicus Programme supports research mainly in six thematic areas, namely land, marine, atmosphere, climate change, emergency management, and security, EUNIS has a dedicated directive for habitat classification. Indeed, it is the main comprehensive European hierarchical classification of habitats that covers not only marine but also terrestrial realms from early 1990 onwards [23–26]. It has become one of the key elements of the INSPIRE (Infrastructure for Spatial Information in Europe) Directive [27] which aims to create an EU spatial data infrastructure for policies or approaches that might affect the environmental structure. It is also the main contributor to Resolution 4 (1996) of the Bern Convention on endangered natural habitat types released in 1996 which was subsequently revised in 2010 and 2014 [28]. It also assists the Natura 2000 process (EU Birds and Habitats Directives), the development of indicators in the European Economic Area (EEA) core set, and the environmental reporting connected to EEA reporting activities [29]. Later, for establishing a solid database on a continental scale, it has been renewed and thus used in a major scale for studying land cover usage, vegetation, forest and habitat mapping [30–32].

However, the scarcity of publicly available datasets with remote sensing images at an appropriate scale and resolution has hindered the development of new models and methods using deep learning techniques as they demand massive heterogeneous data. Rapid advancements in unmanned aerial vehicle (UAV) technology have allowed highly accurate data collection with a wide variety of sensors. Many recent works in this area are focused on UAV images acquired using RGB, multispectral, hyperspectral, and thermal imaging cameras [33–37]. The works are not only contributing new methodologies but also high-resolution image datasets to the public domain, which in turn support applications that require in-depth analysis. The advantages of UAV imagery include ultra-high spatial resolution, low altitude images allowing detection of fine details, flexibility in using diverse sensors that can acquire different ranges of the spectrum, and ease of collection compared to data collected by fieldwork which is often limited by the logistical effort of field surveys. Images from UAVs are routinely used for monitoring diseases, crop nutrition, forest fire, hydrology, and topography analysis for drainage and road construction, creating high-resolution vegetation, forest, and habitat maps, 3-D mapping, and estimation of tree height and surveys over inaccessible areas. Rather than automated methods, these applications are still mostly dependent on fieldworks.

For all the use case scenarios, studies are concentrated on the forest, vegetation, land cover and habitat mapping; also for observing climate changes, identifying crop and soil conditions, water content determination, and drought monitoring. Extensive research is still required in automated habitat mapping methods that deal with the use of drones in farm-scale image collection and the use of deep learning techniques for the identification of habitats. Habitat mapping is always a challenging task; habitats frequently merge or grade from one to another, or form complex mosaics, with the result that a continuum of variation often exists within and between different habitat types which blur the distinctions between habitats. Although it reflects some natural phenomenon, however, in some cases,

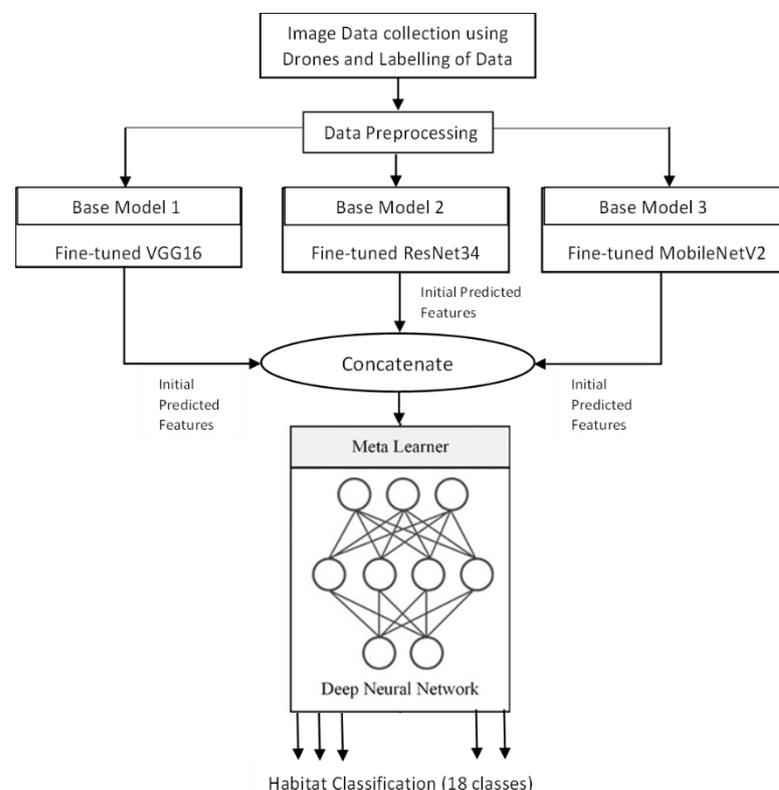
disturbance and damage blur the contrast among habitats. Another complicating factor is that typical farmland habitats have a dynamic nature due to their varying attributes across different seasons of the year e.g., hay meadows, reedbeds, areas of dense bracken, or turloughs. Fertilizer use and heavy grazing could also be the reason for a significant alteration in habitat structure, function, quality, and species composition. Therefore, even the complex deep learning algorithms fail to distinguish between different habitat types. The lack of availability of image datasets in the public domain is another problem. Yasir et al. [38] describe available datasets that can be used for various habitat mapping problems but it doesn't contain any farmland datasets. Some of the recent works in this regard are habitat mapping of marine, land, and benthic zones [39–49].

In this work, we tried to classify Irish farmlands using self-collected drone images. Farming in most of Ireland ranges from highly managed arable land in the east to smaller wetter fields in the west [50,51] and farmland habitats are an integral part of Irish biodiversity [52,53]. However, there is nearly no authentic work or datasets available for the automated classification of Irish farmlands in the public domain.

The specific objectives of this study were (i) to collect and label habitat images of representative Irish farmlands collected using UAVs, (ii) to develop a machine learning method for the digital classification of farmland habitat types, and (iii) assess the effectiveness of the developed classification scheme for habitat identification in the Irish farmlands. The paper is organized as follows: Section 2 describes the developed habitat classification method systematically. The method is validated in Section 3 using various experiments conducted on the dataset and comparing those with existing works. Analysis of the results is presented in Section 4. Finally, the work is concluded in Section 5.

## 2. Materials and Methods

The overall workflow of the proposed approach is presented in Figure 1. The workflow is explained in detail in the below subsections.



**Figure 1.** Workflow of the System.

### 2.1. Data Collection

The task comprised collecting images of a variety of selected habitats [48] in Ireland from a drone using a high-resolution imaging camera. For this purpose, a DJI Mavic 2 Zoom drone was used. It has a CMOS imaging sensor of 1/2.3-inch, 12-megapixels resolution with up to four-times zoom, including a two-times optical zoom (24–48 mm) which makes it suitable for aerial photography. The field of view (FOV) is about 48° to 83°. The lens is auto-focus with a focal length of approximately 4×. The aspect ratio was kept at 3:2 for capturing the images. Different fields within the farm have been identified, and the drone collected a low number (approximately fewer than 10) of replicate images from that field, before moving on to the next field within the farm. Thus, the method sampled different fields with different habitat types within the farm. For this work, the images were captured from July to September 2021 at around noon. The height of the flight was set at approximately 20 m above ground level.

A total of 2233 images were collected ranging across 18 different classes. Habitat categories that belong to these 18 classes can be broadly grouped into 6 different types [47,48]. Table 1 describes the habitat types among 18 different classes covered by the image dataset. Figure 2 provides one sample image from each of the habitats along with the habitat type of the study area.

**Table 1.** Habitat Types of the Image Samples.

Class	Sub-Class	Species
G: Grassland and Marsh	GA1	Improved agricultural grassland
	GSi1	Dry calcareous and neutral grassland
	GS4/GSi4	Wet grassland
	GS2/GSi2	Dry meadows and grassy verges
W: Woodland and Scrub	WN2	Oak-ash-hazel woodland
	WN6	Wet willow-alder-ash woodland
	WN7	Bog woodland
	WD1	(Mixed) Broadleaved woodland
	WD4	Conifer plantation
B: Cultivated and built land	BC1	Arable Crops
	BC2	Horticulture Land
	BC3	Tilled Land
F: Freshwater	FS1	Reeds and large sedge swamps
	FL8	Other artificial lakes
P: Peatlands	PB4	Cutover Bog
H: Heath and dense bracken	HD1	Dense Bracken

The samples were labelled by JAF and PM [52]. The master information sources that have been referenced for labelling the images are the guides published by the Irish Heritage Council [52,54]. The manuals provide a standard scheme and protocol for identifying, describing, and classifying wildlife habitats in Ireland. Figure 3 illustrates the range of class types and the number of images in each type in the dataset after labelling. It is noted that the distribution of the images to class types is not ideal for training; however, this can be improved upon in the future by collecting more images. In this work, this is dealt with the task by boosting the data in the under-represented classes.

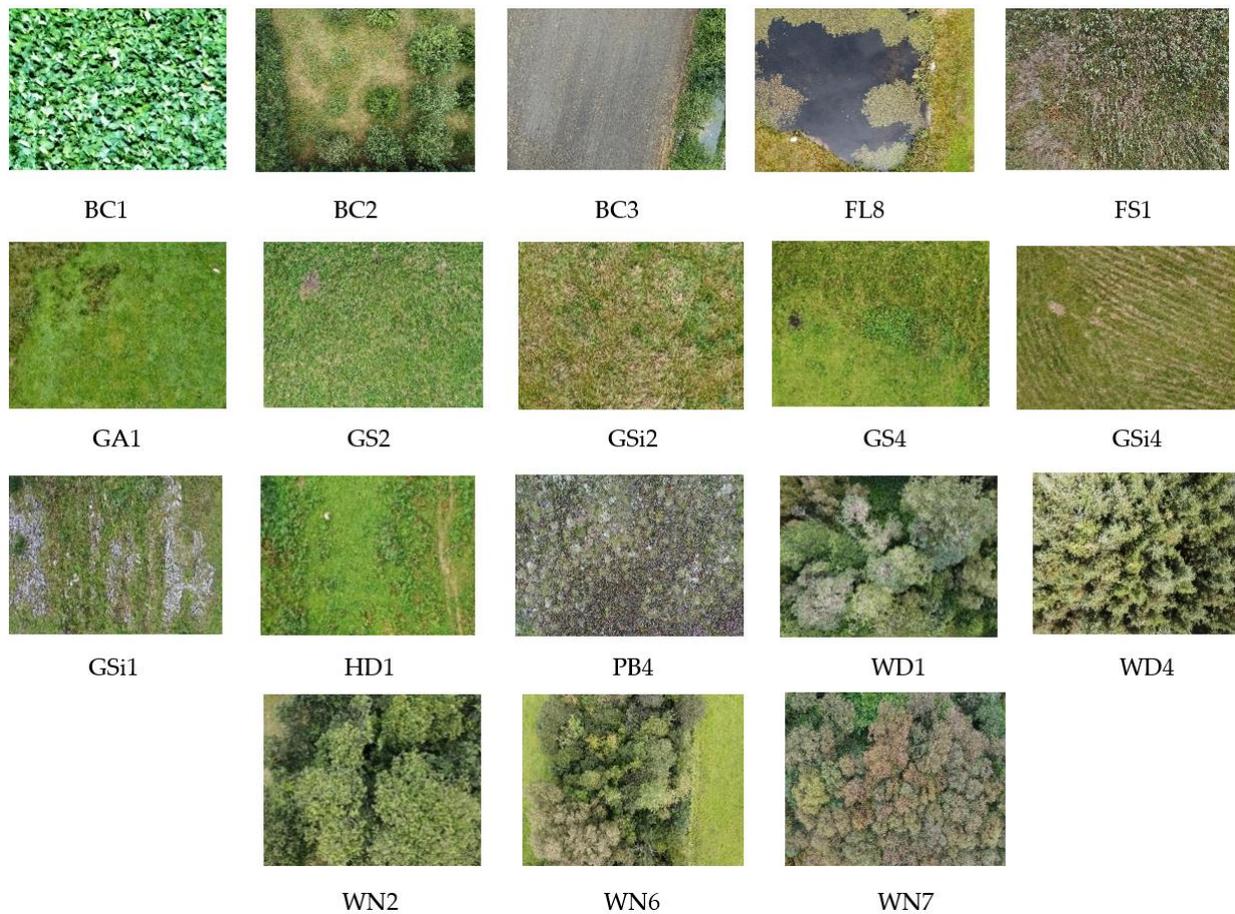


Figure 2. Sample images of 18 different habitats under study.

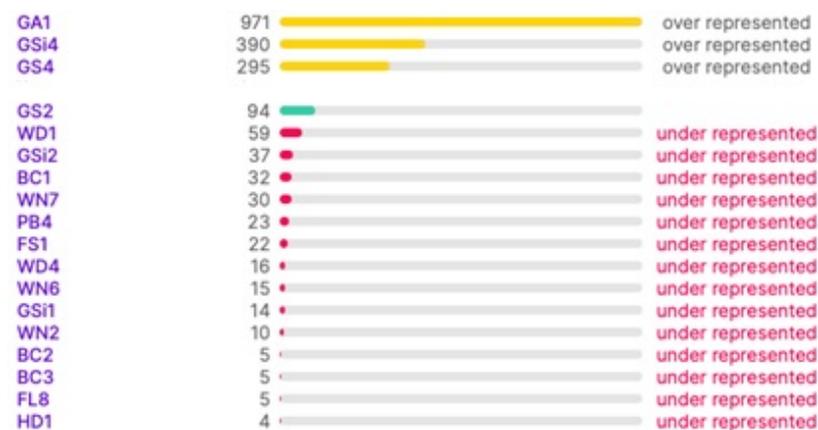


Figure 3. Class Balance between Habitat Types.

## 2.2. Data Cleaning and Preprocessing

Some datasets contain class imbalance and have far more instances than others (Figure 3). Achieving high classification accuracy with this data was difficult. However, to assess the performance with this imbalanced data, a simple classification was performed with a VGG16 pre-trained model by changing the number of nodes in the final classification layer to 18. 20% of the data has been used for validation. The results are shown in Figure 4. It can be noted that the performance is poor in terms of both accuracy and loss. The validation accuracy is less than 20%, which clearly outlines that if the imbalance in the actual data stream is reflected, it can lead to poor average precision during deep

learning classification. Therefore, it was necessary to balance the dataset with nearly equal data over different classes. The high-resolution drone images have a spatial resolution of  $4000 \times 3000$  pixels. The images were resized and scaled using bilinear interpolation [55] to make them compatible with the input size of the base models; in this work, it was set as  $150 \times 150$ . Using bilinear interpolation, the size of the images was reduced without affecting the characteristics of the actual high-resolution images and thus the features are preserved for the deep neural network classification. Then the images were boosted and augmented using 4 different transformations: rotation, flipping, shifting, and shear. The transformations were applied iteratively for different classes by changing the parameters (rotation range, flip direction, shift range and direction, shear range) of each transformation.

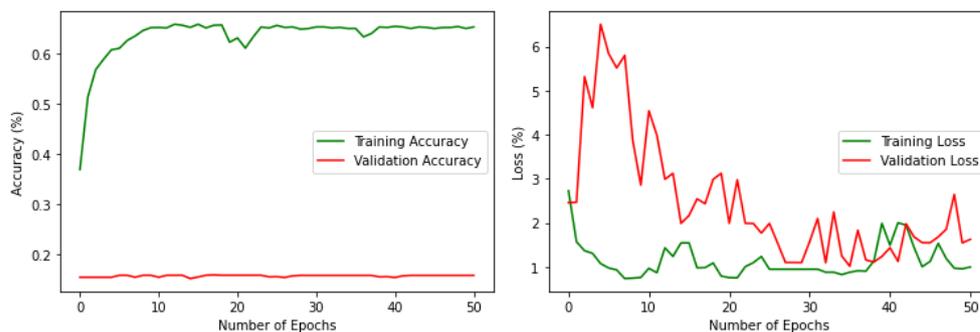


Figure 4. Accuracy and loss, respectively, for the imbalanced data.

Given  $F_n$  is the number of images in the class type  $C_i, i=[1,2, \dots, 18]$  of the original dataset and  $Y_n$  is the number of transformations applied to each image  $X_k \in C_i, i=[1,2, \dots, 18]$ , the total number of images  $D_n$  in the augmented dataset is based on Equations (1) and (2). The procedure will augment entries from the minority classes to match the quantity of the majority classes without overfitting the oversampled classes and ensure that no image is repeated. Thus, a total of 68,356 different images were generated from 2233 images.

$$Y_n = \frac{\text{Number of Iterations}}{\text{Batch Size of the Images}} \tag{1}$$

$$D_n = Y_n \times F_n \tag{2}$$

For the processed dataset, preliminary training and validation were performed with the same VGG16 model to compare the performance with the imbalanced one. The results are shown in Figure 5 which demonstrates the improvement in both training and validation. It is noteworthy that the validation accuracy is increased to 60% by rectifying the data imbalance problem to a certain extent. Figure 6 shows one sample image from each of the habitats along with habitat type in the processed and augmented dataset.

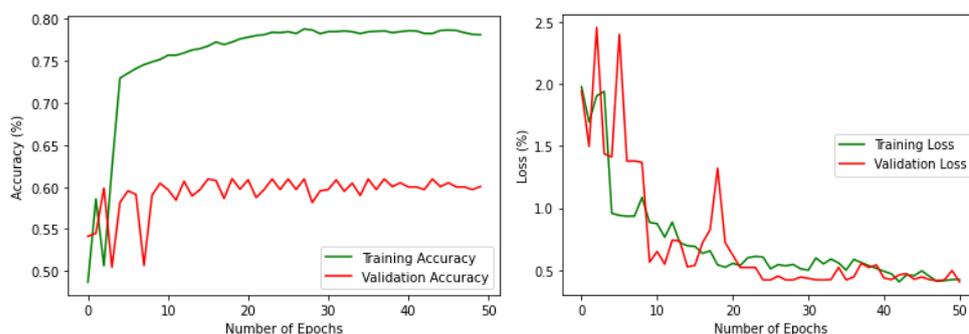
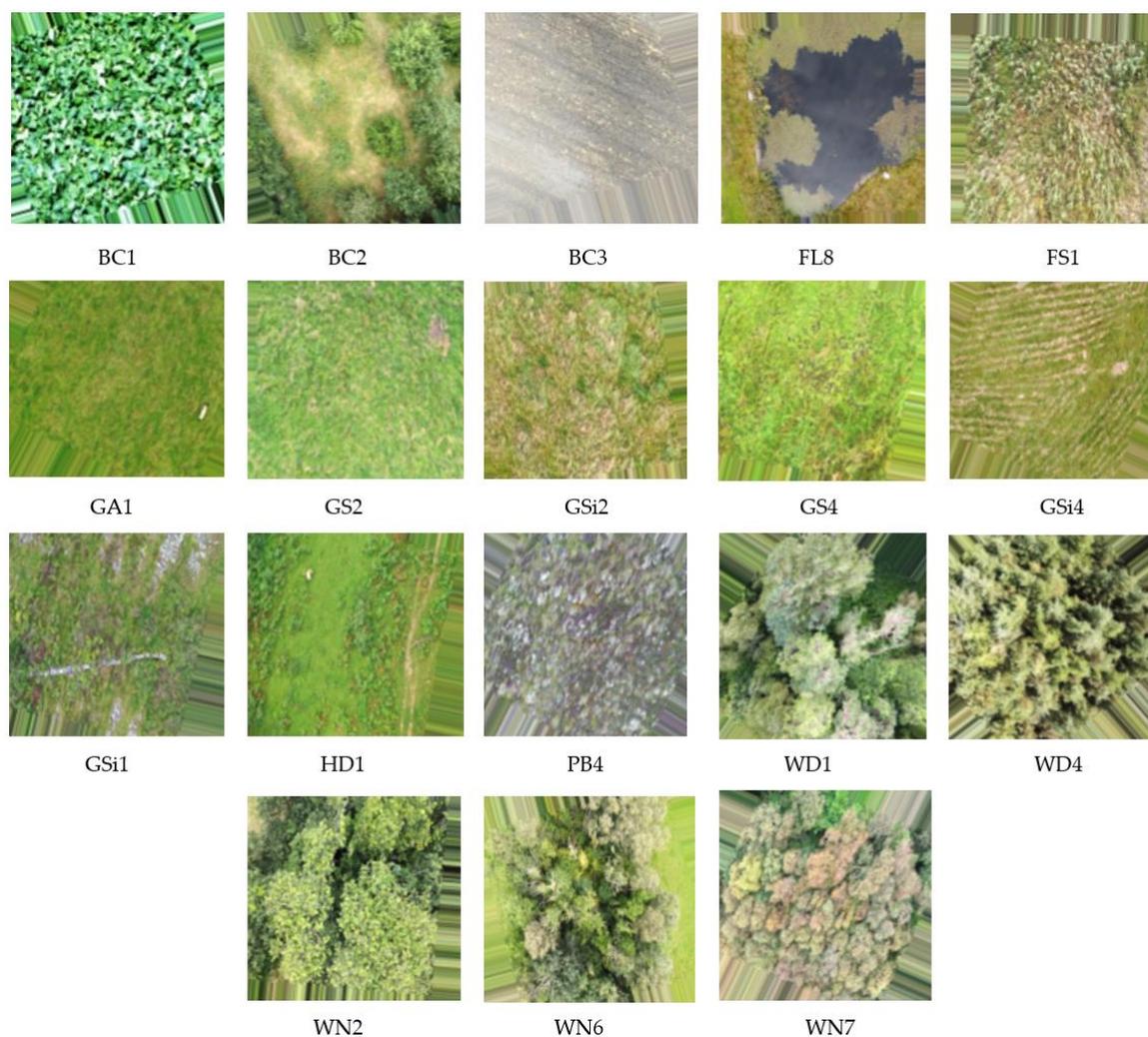


Figure 5. Accuracy and loss, respectively, for the processed data.



**Figure 6.** Sample images of 18 different habitats in the dataset (from Figure 2) following transformation using a combination of rotation, flipping, shifting, and shear.

### 2.3. Model Selection and Tuning

The multi-CNN approach proposed in this work is based on a set of pre-trained CNN models selected by hyperparameter tuning and customized using the transfer learning approach. There are many pre-trained models available for image classification such as VGG [56], Inception [57], Xception [58], Mobilenet [59], Resnet [60], DenseNet [61], SqueezeNet [62], ShuffleNet [63], and many others trained using Imagenet [64], which has more than a million natural images belonging to 1000 different classes. Out of these, 2 comparatively lightweight yet accurate-enough models for classification, VGG16 [56] and ResNet34 [60], were chosen along with MobilenetV2 [58], taking into account that the extended work has to be implemented in embedded boards and eventually in a mobile phone.

CNN models consisting of convolutional, pooling, and fully connected (FC) layers can be efficiently used for image classification [65]. The 2D convolutional layer extracts the local patterns of input features through several feature maps and kernels. This extracted feature vector is then compressed to low resolution by the pooling layer. Pooling helps to decrease the computational cost and over-fitting. Then, these features are given as input to the FC layers. In this work, since the habitat dataset is different compared to the Imagenet database, the final FC layers were removed, and a transfer learning approach was implemented, where pre-trained models are used as the starting point for training and subsequently re-modelled for the specific task. For the specified classification problem, a

feed-forward DNN [66] with various layers and neurons was tried, and finally 3 hidden layers with 512, 256, 128, and 64 neurons respectively were chosen as the end layers of each of the model. A DNN model is suitable for processing nonlinear data, and the performance is better for classification problems. The newly added fully connected dense layers learn the linear and nonlinear relationship between the input features and target, whereas the weights of the convolutional layers of the pre-trained network are kept frozen. In order to prevent overfitting, a dropout layer of 0.2 is added in between the layers. Dropout [67] is a more effective and computationally inexpensive regularizer that prevents overfitting. All layers used rectified linear unit (ReLU), which is the most commonly used non-linear activation function [68]. The final classification layer used a SoftMax activation function [68] with 18 neurons corresponding to the number of classes. All the models are trained for a maximum of 50 epochs with a batch size of 32, where the number of steps in each epoch is fixed using Equation (3). Overtraining of the models during training is avoided by early stopping [69]. Early stopping is a method that allows specifying an arbitrarily large number of training epochs and ceasing training once the model performance stops improving on a holdout validation dataset.

$$\text{Number of Steps in each epoch} = \frac{\text{Number of Images in the dataset}}{\text{Batch Size of the Images}} \quad (3)$$

Since the dimensions of the input images are scaled to  $150 \times 150$  during the pre-processing step, the base models are also modified to make them compatible with the input size. Based on the architecture, the 3 base models with VGG16, ResNet34 and MobileNetV2 produced feature vectors of  $4 \times 4 \times 18$ ,  $5 \times 5 \times 18$ , and  $5 \times 5 \times 18$ , respectively, for every image. This is given as input to the final classification layer. The architecture of each model can be understood from Figure 7.

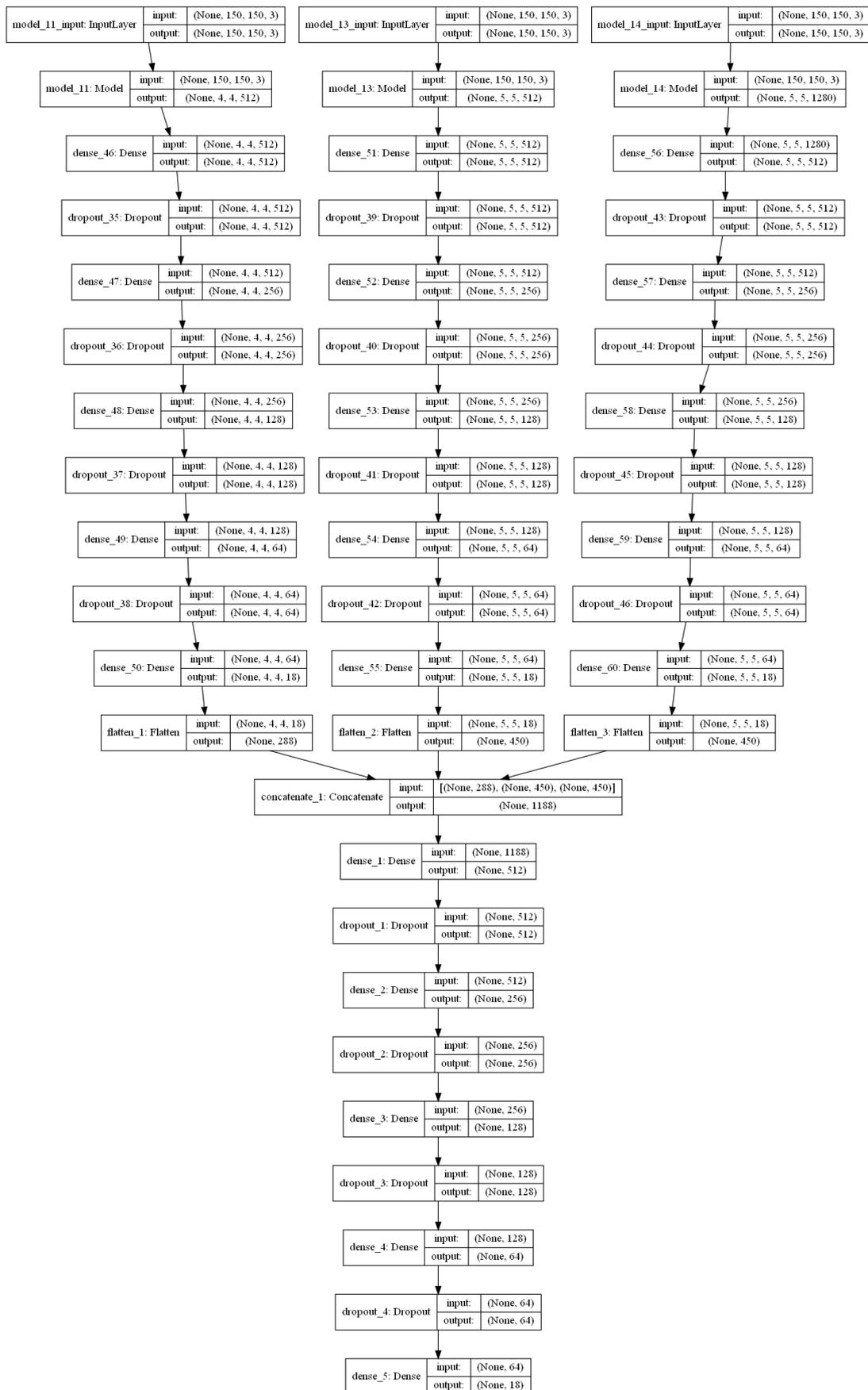


Figure 7. Stacked ensemble model.

#### 2.4. Hyperparameter Tuning

Each model was trained with different optimizers by varying the learning rates and the best-fit hyperparameters were chosen as the ones with the highest validation accuracy. Based on previous work in image classification, the four most suitable optimizers, namely, Adam, RMSProp, SGD, and Nadam were chosen for tuning each model [70–72]. The learning rate was varied from  $10^{-2}$  to  $10^{-5}$  to determine the best fit. Since the problem is image classification, categorical cross-entropy was taken as the loss function for training, due to the fact that one sample can be considered to belong to a specific category with probability 1, and other categories with probability 0. The results of the validation accuracy are given in Table 2. From the Table it can be seen that the Adam optimizer with a learning rate of  $10^{-4}$  and  $10^{-3}$  is the best-fit for the base models with VGG16 and ResNet34 respectively; for the base model with MobilenetV2, RmsProp with a learning rate of  $10^{-3}$  is the best-fit.

**Table 2.** Hyperparameter Tuning Results. Best results achieved for each classifier is shown in bold.

Model	Optimizer	Validation Accuracy for Different Learning Rates			
		$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
Base model with VGG16	<b>ADAM</b>	65.12	68.51	<b>70.05</b>	66.32
	RmsProp	62.51	64.62	63.29	61.52
	SGD	44.07	40.54	45.33	42.06
	NADAM	64.89	66.11	65.24	62.61
Base model with ResNet34	<b>ADAM</b>	69.06	<b>73.89</b>	72.32	71.54
	RmsProp	70.16	72.51	73.01	71.11
	SGD	56.55	58.34	58.22	57.51
	NADAM	61.21	73.51	61.37	60.29
Base model with MobileNetV2	ADAM	65.55	66.02	64.66	62.54
	<b>RmsProp</b>	69.18	<b>75.12</b>	73.51	72.66
	SGD	49.32	53.51	54.34	52.21
	NADAM	57.51	65.07	67.98	65.55
Stacked Ensemble	<b>ADAM</b>	75.54	77.02	<b>78.70</b>	77.37
	RmsProp	76.57	78.23	79.08	75.05
	SGD	63.39	64.21	64.11	65.63
	NADAM	69.54	69.29	65.95	63.49

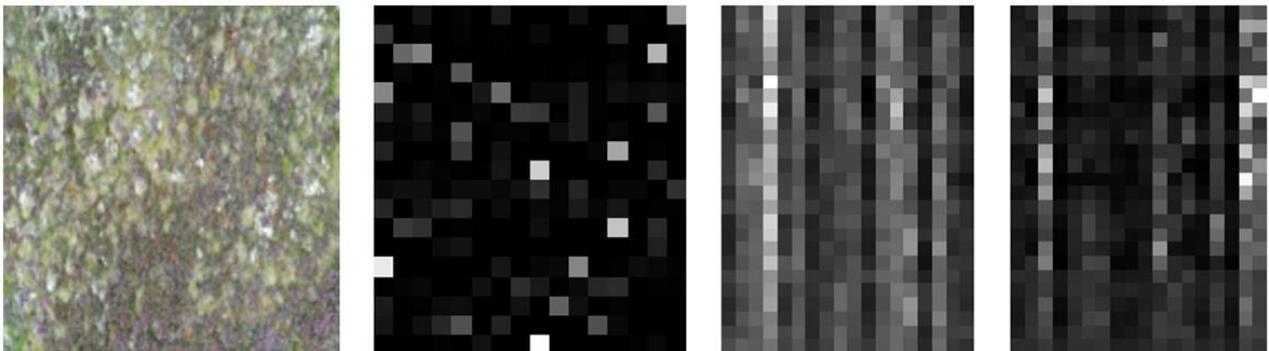
#### 2.5. Stacked Ensemble Model

Stacking the base models reduces the dispersion of the predictions and can make better predictions than any single contributing model [73]. Ideally, a model with low bias and low variance will provide better performance, although in practice it is very challenging and often difficult to achieve. Ensemble models provide a way to reduce the variance of the predictions, leading to improved predictive performance. Ensemble models can be implemented by averaging the individual models, where each model essentially contributes equally to the ensemble prediction, despite the performance of each model. Motivated by this, another approach is a weighted average ensemble, where the models which perform well are required to contribute more, while worse-performing models are required to contribute less. A further generalization of this approach is stacked ensemble which combines the predictions of the sub-models to generate new predictions.

The proposed stacked ensemble learning process consists of two-level stacking. In the first level, 3 base models are trained to obtain the initially predicted features. In the second level, a feed-forward DNN model is used as the meta-learner trained with the

combination of predicted features of the first level to obtain the final classification result. The feed-forward DNN of the meta-learner is also designed with 3 hidden layers and a final classification layer similar to the base-learners. The optimizer and learning rate are tuned similar to the base-learners to derive the best fit by considering validation accuracy as reference. The number of nodes and all other parameters is set in the same way as that of base-learners. Based on the results presented in Table 2, Adam optimizer with a learning rate of  $10^{-4}$  demonstrated better performance. The architecture of the stacked ensemble model is shown in Figure 7. Increasing the number of levels of the stacking will improve the performance of the model up to a certain limit which must be found experimentally, but it also increases the execution time. Also, since each level demands different datasets, for this particular work, the number of stacking levels is confined to 2 because of the limitations of the dataset mentioned in Section 2.2.

The pre-processed data was split into training, validation, and test data with a ratio of 7:2:1 to ensure different datasets as input to different levels. While training the base-learners, the hidden layers were not flattened as the input is a 2D image, which has significant importance for its spatial coordinates corresponding to brightness, edges, and corners. The predicted features for a sample image are shown in Figure 8. The feature vectors of size  $4 \times 4 \times 18$  and  $5 \times 5 \times 18$  respectively (refer to the previous Section 2.3) are reshaped to 2D vectors of size (16,18) and (25,18), respectively, for visualization. The base models are trained with training data and the model is saved.

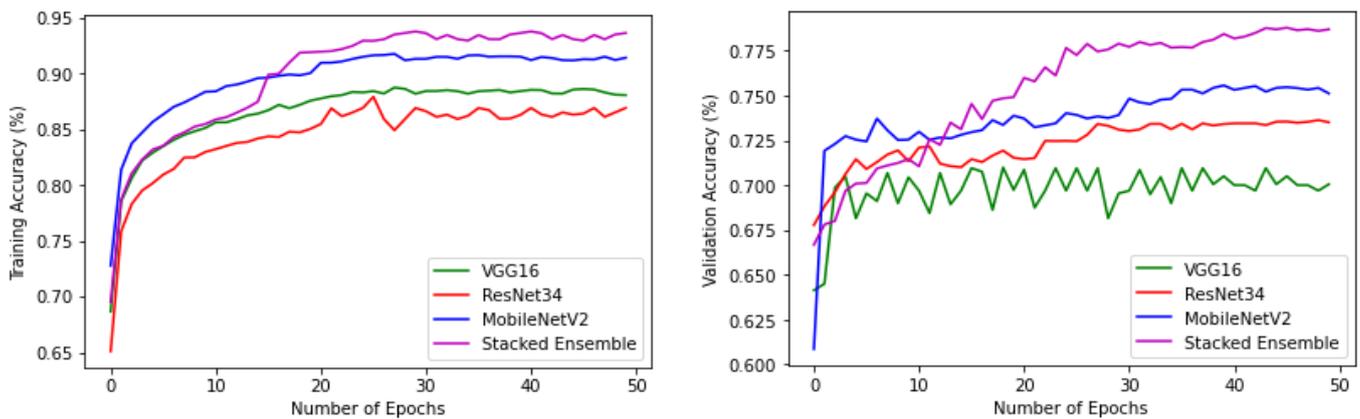


**Figure 8.** Sample Image from PB4 and its predicted features for base-learners with VGG16, Resnet34 and MobilenetV2 respectively.

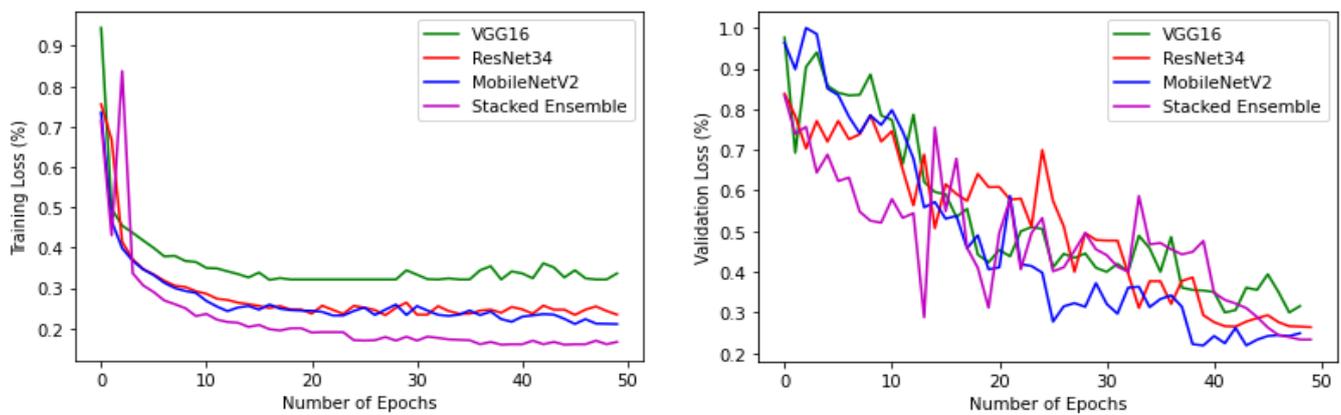
For training the meta-learner, validation data was fed to the already trained base models to obtain the preliminary predictions. Then these initial predictions were concatenated to get a single dataset which was then used to train the meta-learner. The weights of the meta-learner were optimized based on target classes. After training the base models and meta-learner, the test data was given for performance evaluation of the stacked ensemble model. The test data was first given to the base models to obtain the preliminary classification probabilities, which were then combined and fed to the meta learner to get final predictions.

### 3. Results

The experiments were conducted using a core i7, GTX 1650 Graphics Processing Unit (GPU) with 16 GB RAM. The weights of each model were saved when it achieved maximum validation accuracy, preventing it from overfitting. The variation of accuracy and loss with respect to the number of epochs for training and validation are shown in Figures 9 and 10. For all models, the training step was carried out to 50 epochs. It can be seen from the results that the highest accuracy and lowest loss for both training and validation were obtained for the stacked ensemble model. The accuracy was improved considerably for the meta-learner.



**Figure 9.** Training and validation accuracy of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively.



**Figure 10.** Training and validation loss of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively.

The performance metrics used in this study to evaluate the models are accuracy, precision, recall,  $F1$ -score, and area under the receiver operating characteristic curve (AUC). Precision is the percentage of samples classified in a class that were actually members of that class and recall is the percentage of samples of a class that were classified as that class by the model.  $F1$ -score, the summary of recall and precision, is the weighted average between these two values. Accuracy is the percentage of correct predictions across the whole dataset. They are defined using Equations (4)–(7).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + TP + FN} \quad (7)$$

where  $TP$  is the number of true positives,  $TN$  is the number of true negatives,  $FP$  is the number of false positives, and  $FN$  is the number of false negatives. Values close to 1 indicate good performance. These values are calculated from the confusion matrix which describes the complete performance of the model.

The confusion matrix of the base-learners along with the ensemble model is shown in Figure 11. Table 3 describes class-wise accuracy values of the performance metrics. It is calculated by taking the average of values lying across the main diagonal of the confusion matrix, which is the ratio of total correct predictions to total predictions made. Table 4 depicts the comparison of results achieved using various base-learners in combination with the stacked ensemble model.

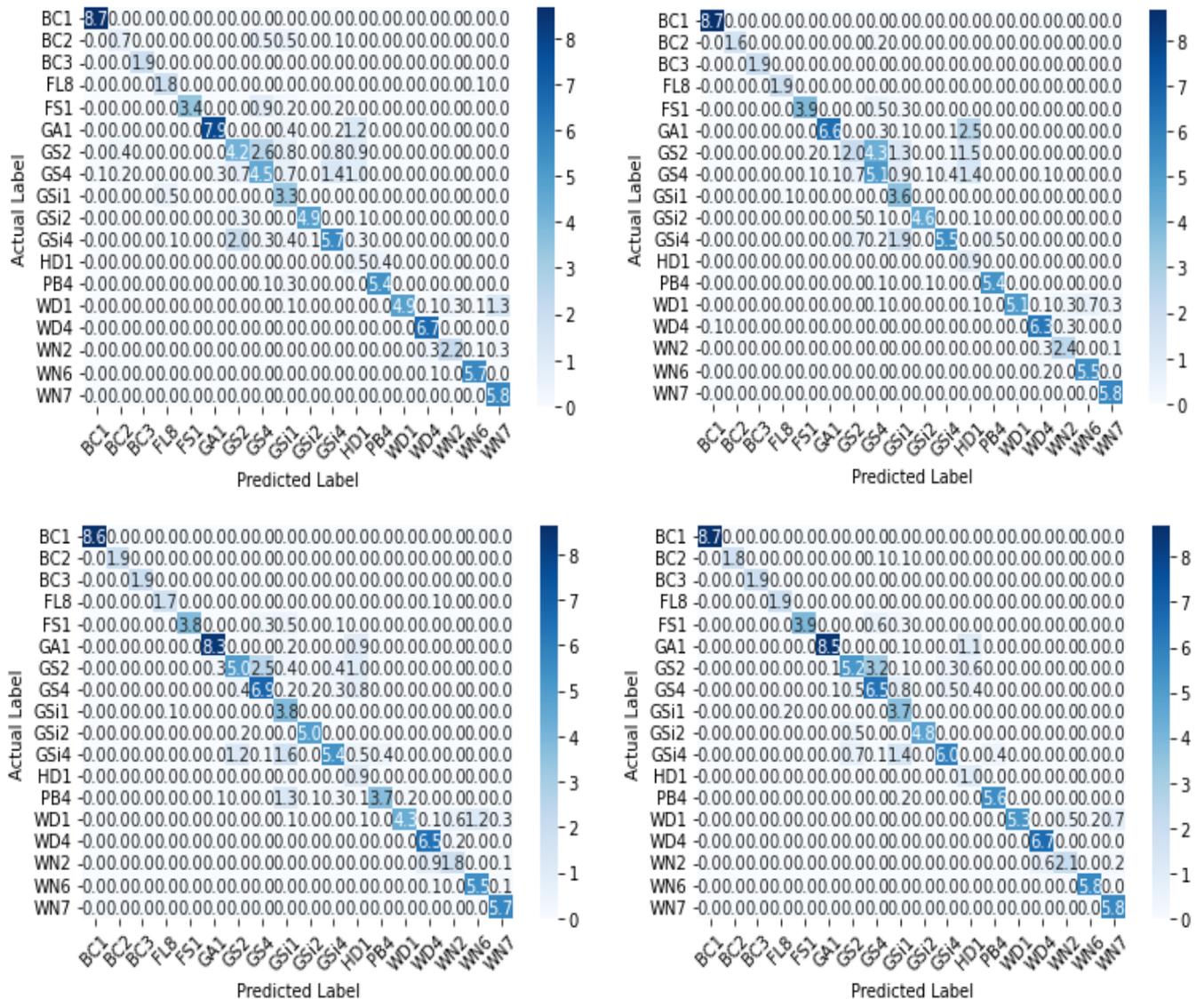


Figure 11. Confusion matrix (in %) of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively.

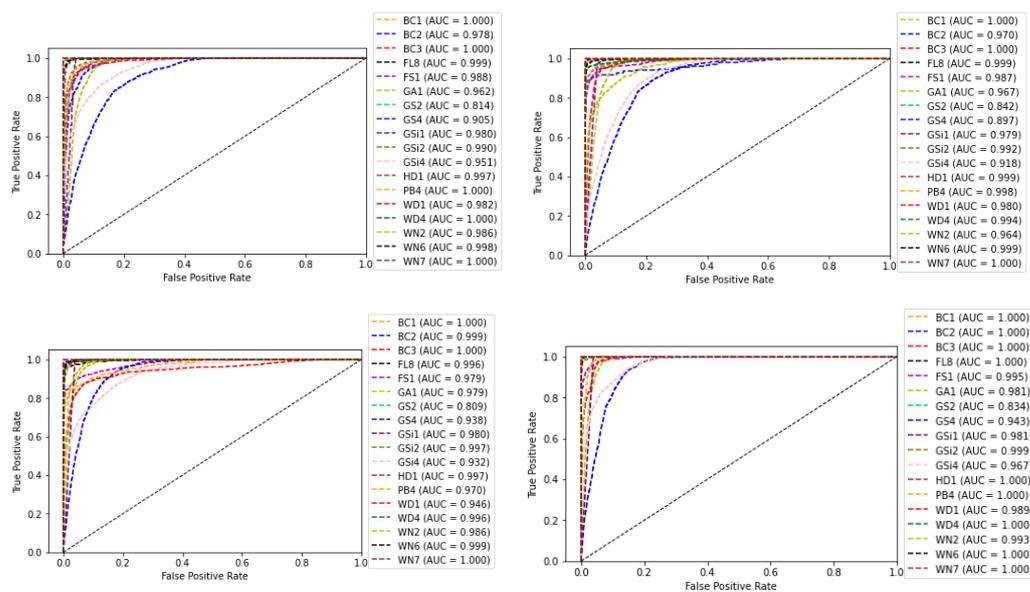
The receiver operating characteristic (ROC) curve, which is a plot between false positive rate (FPR) and true positive rate (TPR) corresponding to the four classifiers, is shown in Figure 12. The area under the ROC curve (AUC) for each class is also shown along with the plots. AUC measures the entire two-dimensional area underneath the ROC curve from (0,0) to (1,1). The value ranges from 0 to 1, representing the probability that the model ranks a random positive sample more highly than a random negative sample.

**Table 3.** Class-wise accuracy of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models.

Class	Model			
	VGG16	ResNet34	MobileNetV2	Ensemble
BC1	1	1	1	1
BC2	0.371	0.85	0.968	0.925
BC3	1	1	0.993	1
FL8	0.946	0.981	0.887	0.993
FS1	0.715	0.81	0.798	0.818
GA1	0.819	0.688	0.871	0.881
GS2	0.433	0.208	0.521	0.543
GS4	0.511	0.576	0.779	0.727
GSi1	0.856	0.948	0.979	0.948
GSi2	0.923	0.875	0.945	0.912
GSi4	0.637	0.623	0.575	0.699
HD1	0.531	0.962	0.968	0.993
PB4	0.934	0.931	0.646	0.962
WD1	0.724	0.755	0.647	0.778
WD4	1	0.933	0.965	0.993
WN2	0.756	0.841	0.641	0.710
WN6	0.987	0.956	0.954	0.993
WN7	1	0.998	1	1

**Table 4.** Comparison of results achieved for various base-learners in combination with proposed classifiers. Best results achieved using the Stacked Ensemble model is indicated in bold.

Model	Performance Metrics			
	Precision	Recall	F-Measure	Accuracy
VGG16	0.8019	0.7647	0.7584	0.7861
Resnet34	0.8489	0.8451	0.8273	0.8300
MobilenetV2	0.8443	0.8611	0.82	0.8414
<b>Stacked Ensemble</b>	<b>0.8944</b>	<b>0.9016</b>	<b>0.8842</b>	<b>0.8844</b>



**Figure 12.** ROC curve and AUC values of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively.

The number of parameters and execution time for 1 epoch along with classification accuracy are compared in Table 5 for each of the models. This will provide valuable insights for developing efficient edge algorithms in the future where these values will be a major concern in the model design. It can be seen that the base-learner with MobilenetV2 had the lowest execution time among all the models and the highest accuracy among the base models. The base-learner with Resnet34 had the highest execution time among all the models but its accuracy was even lower than the stacked ensemble one. The second highest execution time was for the stacked ensemble model which is obvious from the number of levels to train. However, when compared with its high accuracy the execution time is satisfactory.

**Table 5.** Comparison of execution time and accuracy for various models. Best results are indicated in bold.

Model	Number of Parameters		Execution Time in Minutes/Epoch		Accuracy
	Total	Non-Trainable	Trainable		
VGG16	18,662,906	14,714,688	3,948,216	3.41	0.7861
Resnet34	27,684,411	21,302,473	6,381,938	5.57	0.8300
MobilenetV2	18,086,322	12,257,984	5,828,338	<b>1.04</b>	0.8414
Ensemble	40,759,762	39,977,279	782,483	5.50	<b>0.8844</b>

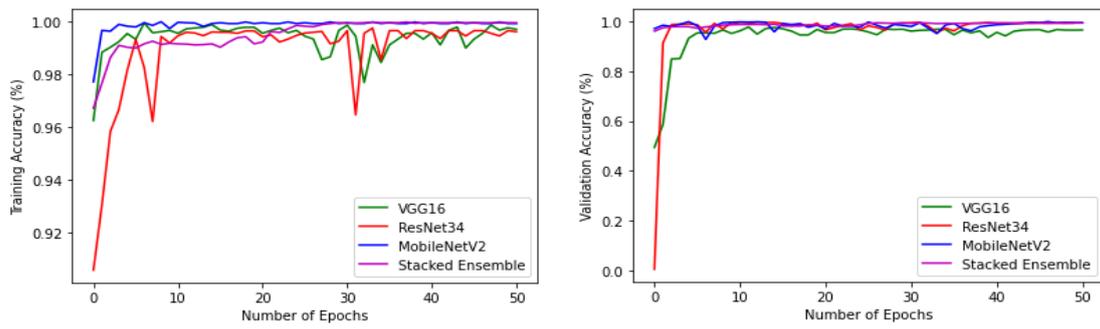
Table 6 displays the results of major state-of-the-art methods published not long ago, along with the proposed best performing stacked ensemble model. Most of the existing methods deal with marine and land habitats and none of them addresses the problem of farmland habitat classification. The dataset, number of samples, and validation techniques used by the various state-of-the-art methods are different. Therefore, a fair comparison of the results is not possible. Though the habitat types are different, it is noteworthy that the proposed method has proven its effectiveness.

**Table 6.** Comparison of results with other recent state-of-the-art methods.

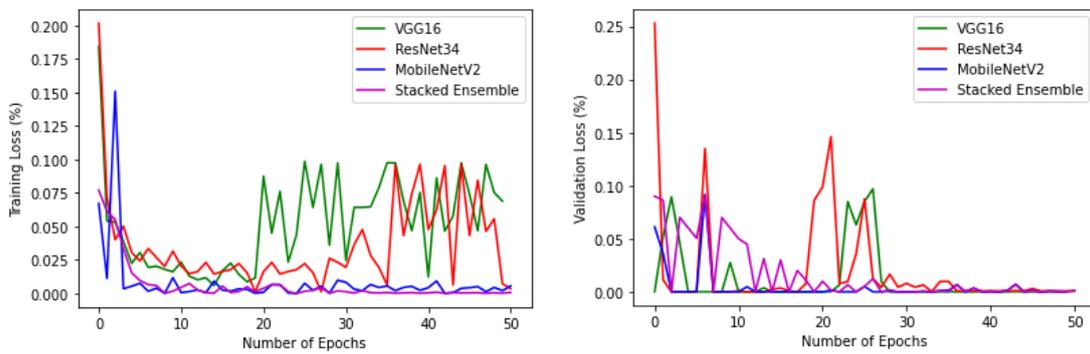
Study	Habitat Type	Dataset	Classes	Classifier	Model	Accuracy
A.Diegues et al., 2018 [45]	marine	Self-collected	2	CNN	VGG16	85.10
T.Liu et al., 2018 [37]	wetland	Self-Collected	7	CNN	-	76.90
A. Gómez-Ríos et al., 2019 [43]	marine	EILAT & RSMAS	8, 14	CNN	ResNet50 & ResNet151	98.90 <sup>1</sup>
M. Yasir et al., 2020 [38]	marine	MLC	9	MLP	DenseNet169	87.40
<b>This study *</b>	<b>farmland</b>	<b>Self-collected</b>	<b>18</b>	<b>CNN</b>	<b>Stacked Ensemble</b>	<b>88.44</b>

<sup>1</sup> Combined accuracy for all datasets; \* Results achieved using proposed method is indicated in bold.

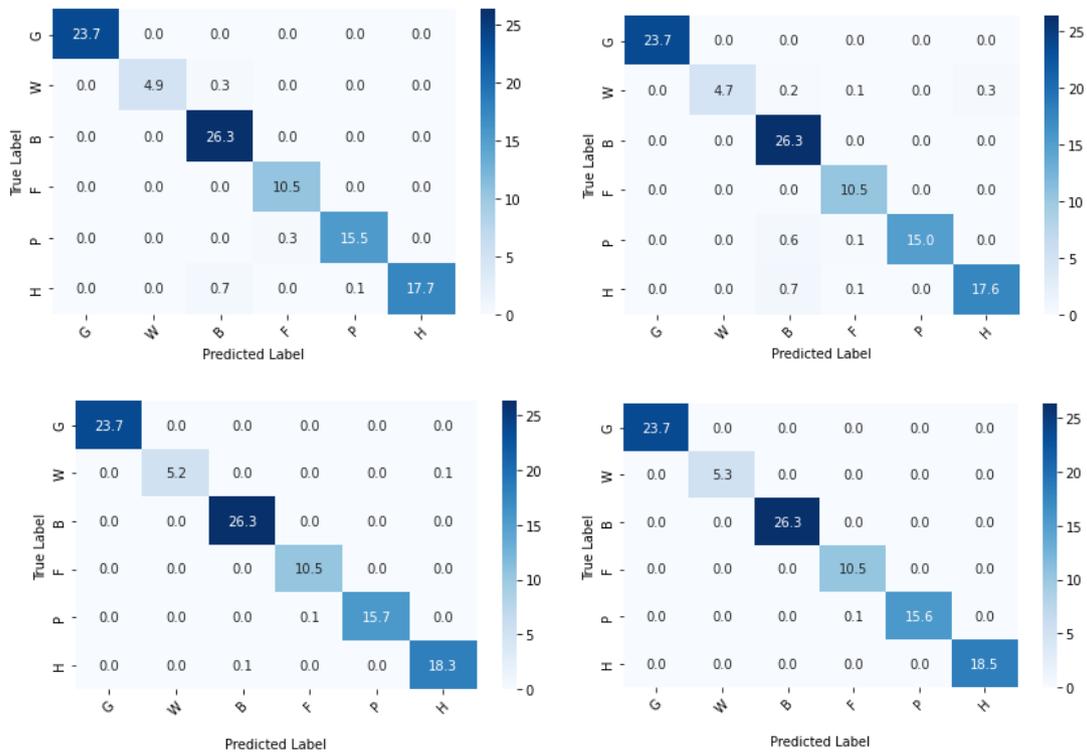
Another experiment was conducted, by choosing a single subclass from each category and applying the same algorithms and models described in Section 2 to the reduced dataset to classify the samples into six broader classes, as described in Table 1. The variation of accuracy and loss with respect to the number of epochs for training and validation are shown in Figures 13 and 14 respectively. The confusion matrix of the base-learners along with the ensemble model is shown in Figure 15. It can be seen that all the models show very good results. The AUC approached 1 with nearly 0 misclassifications for the stacked ensemble model (Figure 16). Figure 17 shows some of the sample classification results along with the classification accuracy for the second experiment in which the habitats are classified into each of the broad categories.



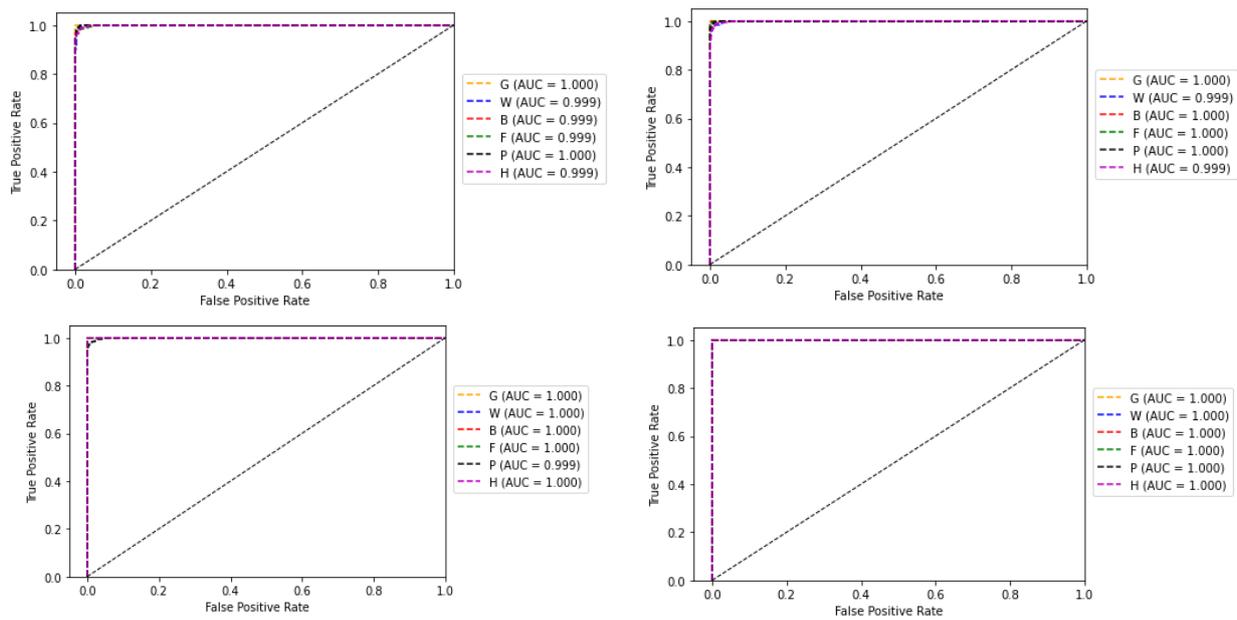
**Figure 13.** Training and validation accuracy of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively, for the reduced dataset.



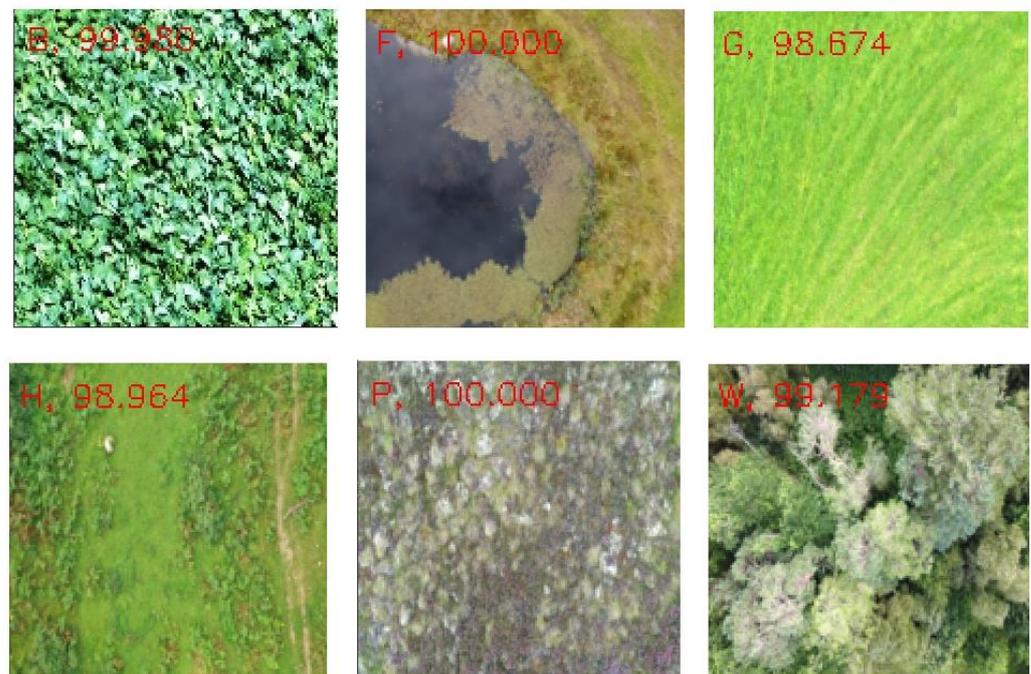
**Figure 14.** Training and validation loss of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively, for the reduced dataset.



**Figure 15.** Confusion matrix (%) of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively, for the reduced dataset.



**Figure 16.** ROC curve and AUC values of base-learners with VGG16, Resnet34, MobilenetV2, and Ensemble models, respectively, for the reduced dataset.



**Figure 17.** Classification accuracy (%) of sample images.

#### 4. Discussion

From the results, it was verified that the performance of the stacked ensemble model outperforms the base-learners. The overall prediction accuracy achieved by the stacked ensemble model was 88.44%. Some classes were predicted well, whereas the prediction of other classes was not so good, which ultimately affects the overall performance of the model (Table 3). For any deep learning model, during the training or the validation phase, the model can overfit as it tries to represent as much as possible the underlying characteristics of training data. This process deeply downgrades the model's ability to perform accurate predictions on new data. Although techniques to prevent the model overfitting (such

as using three different datasets at the three levels of training, data augmentation and transformation, dropout, and early stopping), deep learning models produce better results only when given a vast number of samples in the training phase. Therefore, the main reason for this is the imbalance of the self-collected dataset that was used for the work, which is clear from Figure 3. The other reason is that the method needs to classify many subclasses that belong to different broad categories which are often confusing. However, this could be overcome by the collection and addition of more images from under-represented classes, and further development of this preliminary approach.

As seen from Table 1, the broader classes have many sub-categories within them. Sub-classes belonging to the same broad category have very similar features for which the models find it difficult to establish distinguishable patterns between them. Heterogeneity of samples within the same class is another reason for misclassification. This is because of the dynamic nature of the farmland habitats which makes the acquisition system capture the same habitat types in slightly varying formats at different time intervals. This was overcome to a certain extent by collecting the data in the same season and almost at the same time throughout the data collection period. In addition, certain samples that showed high variance compared to other samples in the same class were manually removed. However, variations within the habitats still exist.

A lot of misclassifications can be seen in the middle blocks of the confusion matrix. The reason is that the middle row constitutes G: grassland and marsh, comprising six subclasses, which is the most in this experiment, all of which have nearly indistinguishable features. In this category, GSi1 had the highest prediction accuracy as it is dry calcareous and neutral grassland with rocky features, not seen in any other habitat types. Misclassifications can be seen in the bottom blocks of the confusion matrix also which constitutes W: woodland and scrub, which has the second-highest number of sub-classes—five, in which WD1 and WN2 have similar features contributing to prediction errors. Though the habitat mapping problem is a challenging task, from Table 6, it can be understood that the proposed method performed well.

Based on the results provided in Table 5, if the execution is done on a GPU machine, the stacked ensemble model will always be a good choice considering the high accuracy. However, when there is a trade-off between execution time and accuracy, especially for resource-constrained edge devices, MobilenetV2 may be a good choice. The classification accuracy of the stacked ensemble model is around 4% higher than the base-learner with MobilenetV2 whereas its execution time is approximately 6 min lower. However, more experiments need to be performed for better understanding in this regard and other models need to be explored.

In the second experiment when the number of classes was reduced, accuracy approached approximately 100% with no misclassifications. This can be verified from the associated results shown from Figures 13–17. From this experiment, it is clear that the imbalance of the data, similar features between subclasses, and varying features within the same classes decrease the prediction accuracy of the habitat mapping process.

## 5. Conclusions

We investigated an approach for farmland habitat classification which is an important contribution to improved monitoring and preservation of ecosystems. High-resolution self-collected drone images were used collected over 18 different habitat types that were grouped into six broad categories. A stacked ensemble model was proposed which reduced the bias and variance and thus improved the prediction accuracy. The average accuracy over all the classes reported was 88.31% using this approach. Although an effective farmland habitat classification technique is proposed, there is room for further work. The data set used in this work is an imbalanced one with only 2233 samples across a comparatively large number of classes. More samples need to be collected that should be balanced across all possible habitat types to improve the accuracy of the deep learning technique. As an extension of this work, we aim to collect more samples and label them to make

the dataset publicly available so that the entire research community can benefit. Other state-of-the-art pre-trained CNN architectures could be tried and various combinations of pre-trained CNNs could be explored in the future to understand the best possible model that will adapt the minute variations of the features among different habitats. In this work, tuning was conducted only for identifying the best optimizer and learning rate. Extensive hyper-parameter tuning could be conducted to develop the best fit model for particular applications. Different mechanisms to develop resource-efficient, deep learning models for mobile edge devices and other IoT devices used in smart agriculture have to be investigated. Based on the requirements of the execution environment (e.g., mobile phone or UAV capturing farmland images), a suitable compressed deep learning model can be generated.

**Author Contributions:** L.A. and S.D. contributed to the design and conception of the study; L.A. performed the experiments and analysed the data; M.Z. and R.M. contributed to the draft of the manuscript; L.A. wrote the paper; S.D. supervised the project and acquired the research funds; J.A.F. and P.M. managed the data collection using drones and labelled the images. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Science Foundation Ireland and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland VistaMilk research centre under the grant 16/RC/3835. JAF and PM were supported by the SmartAgriHubs project, which received funding from the European Union’s Horizon 2020 Research and Innovation Programme under grant agreement No. 818182.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset will be made available upon request.

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

## References

1. Xie, Y.; Sha, Z.; Yu, M. Remote Sensing Imagery in Vegetation Mapping: A review. *J. Plant Ecol.* **2008**, *1*, 9–23. [CrossRef]
2. Adamo, M.; Tomaselli, V.; Tarantino, C.; Vicario, S.; Veronico, G.; Lucas, R.; Blonda, P. Knowledge-Based Classification of Grassland Ecosystem Based on Multi-Temporal WorldView-2 Data and FAO-LCCS Taxonomy. *Remote Sens.* **2020**, *12*, 1447. [CrossRef]
3. Lapini, A.; Pettinato, S.; Santi, E.; Paloscia, S.; Fontanelli, G.; Garzelli, A. Comparison of Machine Learning Methods Applied to SAR Images for Forest Classification in Mediterranean Areas. *Remote Sens.* **2020**, *12*, 369. [CrossRef]
4. Miller, J.; Franklin, J. Modeling the Distribution of Four Vegetation Alliances using Generalized Linear Models and Classification Trees with Spatial Dependence. *Ecol. Model.* **2002**, *157*, 227–247. [CrossRef]
5. Souza, C.M., Jr.; Siqueira, J.V.; Sales, M.H.; Fonseca, A.V.; Ribeiro, J.G.; Numata, I.; Cochrane, M.A.; Barber, C.P.; Roberts, D.A.; Barlow, J. Ten-Year Landsat Classification of Deforestation and Forest Degradation in the Brazilian Amazon. *Remote Sens.* **2013**, *5*, 5493–5513. [CrossRef]
6. Sinergise. Sentinel Hub. 2014. Available online: <https://www.sentinel-hub.com/> (accessed on 30 January 2021).
7. de Bem, P.P.; de Carvalho, O.A.; Guimaraes, R.F.; Gomes, R.A.T. Change Detection of Deforestation in the Brazilian Amazon using Landsat Data and Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 901. [CrossRef]
8. Bragagnolo, L.; da Silva, R.V.; Grzybowski, J.M.V. Amazon Forest Cover Change Mapping based on Semantic Segmentation by U-Nets. *Ecol. Inform.* **2021**, *62*, 101279. [CrossRef]
9. Isaienkov, K.; Yushchuk, M.; Khramtsov, V.; Seliverstov, O. Deep Learning for Regular Change Detection in Ukrainian Forest Ecosystem with Sentinel-2. *IEEE J. Sel. Top. Appl. Earth Obs.* **2021**, *14*, 364–376. [CrossRef]
10. Lee, S.H.; Han, K.J.; Lee, K.; Lee, K.J.; Oh, K.Y.; Lee, M.J. Classification of Landscape affected by Deforestation using High-Resolution Remote Sensing Data and Deep-Learning Techniques. *Remote Sens.* **2020**, *12*, 3372. [CrossRef]
11. Hay, G.J.; Blaschke, T.; Marceau, D.J.; Bouchard, A. A comparison of Three Image-Object Methods for the Multiscale Analysis of Landscape Structure. *ISPRS J. Photogramm. Remote Sens.* **2003**, *57*, 327–345. [CrossRef]
12. Newsam, S.; Wang, L.; Bhagavathy, S.; Manjunath, B.S. Using Texture to Analyze and Manage Large Collections of Remote Sensed Image and Video Data. *Appl. Opt.* **2004**, *43*, 210–217. [CrossRef] [PubMed]
13. Fransson, J.E.S.; Magnusson, M.; Olsson, H.; Eriksson, L.E.B.; Sandberg, G.; SmithJonforsen, G.; Ulander, L.M.H. Detection of Forest Changes using ALOS PALSAR Satellite Images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–27 July 2007; pp. 2330–2333.

14. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
15. Corbane, C.; Lang, S.; Pipkins, K.; Alleaume, S.; Deshayes, M.; Millán, V.E.G.; Strasser, T.; Borre, J.V.; Toon, S.; Michael, F. Remote Sensing for Mapping Natural Habitats and their Conservation Status—New Opportunities and Challenges. *Int. J. Appl. Earth Obs.* **2015**, *37*, 7–16. [CrossRef]
16. Berry, P.; Smith, A.; Eales, R.; Papadopoulou, L.; Erhard, M.; Meiner, A.; Bastrup-Birk, A.; Ivits, E.; Royo Gelabert, E.; Dige, G. *Mapping and Assessing the Condition of Europe's Ecosystems—Progress and Challenges*, 3rd ed.; Publications Office of the European Union: Luxembourg, 2016; ISBN 978-92-79-55019-5.
17. Copernicus Land Monitoring System. Available online: <https://land.copernicus.eu> (accessed on 26 January 2021).
18. Pieldebo, L.; Taramelli, A.; Schiavon, E.; Valentini, E.; Molina, J.-L.; Nguyen Xuan, A.; González-Aguilera, D. Assessment of Green Infrastructure in Riparian Zones Using Copernicus Programme. *Remote Sens.* **2019**, *11*, 2967. [CrossRef]
19. Taramelli, A.; Lissoni, M.; Pieldebo, L.; Schiavon, E.; Valentini, E.; Nguyen Xuan, A.; González-Aguilera, D. Monitoring Green Infrastructure for Natural Water Retention Using Copernicus Global Land Products. *Remote Sens.* **2019**, *11*, 1583. [CrossRef]
20. ESA—Sentinel Online. Available online: <https://sentinel.esa.int/web/sentinel/home> (accessed on 26 January 2021).
21. EUNIS European Nature Information System. Available online: <https://www.eea.europa.eu/data-and-maps/data/eunishabitat-classification> (accessed on 26 January 2021).
22. Revision of the EUNIS Habitat Classification. Available online: <https://www.eea.europa.eu/themes/biodiversity/anintroduction-to-habitats/underpinning-european-policy-on-nature-conservation-1> (accessed on 26 January 2021).
23. Davies, C.E.; Moss, D. EUNIS Habitat Classification. In *Final Report to the European Topic Centre on Nature Conservation*; European Environment Agency: Copenhagen, Denmark, 1998.
24. Davies, C.E.; Moss, D.; Hill, M.O. *EUNIS Habitat Classification*; European Environment Agency: Copenhagen, Denmark, 2004.
25. Vilà, M.; Pino, J.; Font, X. Regional Assessment of Plant Invasions across Different Habitat Types. *J. Veg. Sci.* **2007**, *18*, 35–42. [CrossRef]
26. Rodwell, J.S.; Evans, D.; Schaminée, J.H.J. Phytosociological Relationships in European Union Policy-related Habitat Classifications. *Rend. Lincei Sci. Fis. Nat.* **2018**, *29*, 237–249. [CrossRef]
27. INSPIRE. *D2.8.III.18 Data Specification on Habitats and Biotopes—Technical Guidelines*; European Commission Joint Research Centre: Brussels, Belgium, 2013; pp. 1–121.
28. Council of Europe. Revised Annex I to Resolution 4 of the Bern Convention on Endangered Natural Habitat Types using the EUNIS Habitat Classification (Adopted by the Standing Committee on 30 November 2018). 2018. Available online: <https://rm.coe.int/16807469e7> (accessed on 1 February 2020).
29. Natura 2000—Environment—European Commission. Available online: <https://ec.europa.eu/environment/nature/natura2000/> (accessed on 2 February 2022).
30. Dengler, J.; Oldeland, J.; Jansen, F.; Chytrý, M.; Ewald, J.; Finckh, M.; Glockler, F.; Lopez-Gonzalez, G.; Peet, R.K.; Schaminée, J.H.J. Vegetation databases for the 21st century. *Biodivers. Ecol.* **2012**, *4*, 15–24. [CrossRef]
31. Chytrý, M.; Hennekens, S.M.; Jiménez-Alfaro, B.; Knollová, I.; Dengler, J.; Jansen, F.; Landucci, F.; Schaminée, J.H.; Acic, S.; Agrillo, E.; et al. European Vegetation Archive (EVA): An Integrated Database of European Vegetation Plots. *Appl. Veg. Sci.* **2016**, *19*, 173–180. [CrossRef]
32. Bruelheide, H.; Dengler, J.; Jiménez-Alfaro, B.; Purschke, O.; Hennekens, S.M.; Chytrý, M.; Pillar, V.D.; Jansen, F.; Kattge, J.; Sandel, B.; et al. sPlot—A New Tool for Global Vegetation Analyses. *J. Veg. Sci.* **2019**, *30*, 161–186. [CrossRef]
33. Bhatnagar, S.; Gill, L.; Ghosh, B. Drone Image Segmentation Using Machine and Deep Learning for Mapping Raised Bog Vegetation Communities. *Remote Sens.* **2020**, *12*, 2602. [CrossRef]
34. de Castro, A.I.; Shi, Y.; Maja, J.M.; Peña, J.M. UAVs for Vegetation Monitoring: Overview and Recent Scientific Contributions. *Remote Sens.* **2021**, *13*, 2139. [CrossRef]
35. Sandino, J.; Gonzalez, F.; Mengersen, K.; Gaston, K.J. UAVs and Machine Learning Revolutionising Invasive Grass and Vegetation Surveys in Remote Arid Lands. *Sensors* **2018**, *18*, 605. [CrossRef] [PubMed]
36. Hamylton, S.M.; Morris, R.H.; Carvalho, R.C.; Order, N.; Barlow, P.; Mills, K.; Wang, L. Evaluating Techniques for Mapping Island Vegetation from Unmanned Aerial Vehicle (UAV) Images: Pixel Classification, Visual Interpretation and Machine Learning Approaches. *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *89*, 102085. [CrossRef]
37. Liu, T.; Abd-Elrahman, A.; Morton, J.; Wilhelm, V.L. Comparing Fully Convolutional Networks, Random Forest, Support Vector Machine, and Patch-based Deep Convolutional Neural Networks for Object-based Wetland Mapping using Images from Small Unmanned Aircraft System. *GIScience Remote Sens.* **2018**, *55*, 243–264. [CrossRef]
38. Yasir, M.; Rahman, A.U.; Gohar, M. Habitat Mapping using Deep Neural Networks. *Multimed. Syst.* **2020**, *4*, 1–12. [CrossRef]
39. Lubis, M.Z.; Anurogo, W.; Hanafi, A.; Kausarian, H.; Taki, H.M.; Antoni, S. Distribution of Benthic Habitat using Landsat-7 Imagery in Shallow Waters of Sekupang, Batam Island, Indonesia. *Biodiversitas* **2018**, *19*, 1117–1122. [CrossRef]
40. Gonçalves, J.; Henriques, R.; Alves, P.; Honrado, J. Evaluating an Unmanned Aerial Vehicle-based Approach for Assessing Habitat Extent and Condition in Finescale Early Successional Mountain Mosaics. *Appl. Veg. Sci.* **2016**, *19*, 132–146. [CrossRef]
41. Wicaksono, P.; Aryaguna, P.A.; Lazuardi, W. Benthic Habitat Mapping Model and Cross Validation using Machine-Learning Classification Algorithms. *Remote Sens.* **2019**, *11*, 1279. [CrossRef]

42. Abrams, J.F.; Vashishtha, A.; Wong, S.T.; Nguyen, A.; Mohamed, A.; Wieser, S.; Kuijper, A.; Wilting, A.; Mukhopadhyay, A. Habitat-Net: Segmentation of Habitat Images using Deep Learning. *Ecol. Inform.* **2019**, *51*, 121–128. [[CrossRef](#)]
43. Gómez-Ríos, A.; Tabik, S.; Luengo, J.; Shihavuddin, A.; Krawczyk, B.; Herrera, F. Towards Highly Accurate Coral Texture Images Classification using Deep Convolutional Neural Networks and Data Augmentation. *Expert Syst. Appl.* **2019**, *118*, 315–328. [[CrossRef](#)]
44. Perez-Carabaza, S.; Boydell, O.; O’Connell, J. Monitoring Threatened Irish Habitats Using Multi-Temporal Multi-Spectral Aerial Imagery and Convolutional Neural Networks. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 2556–2559.
45. Diegues, A.; Pinto, J.; Ribeiro, P. Automatic Habitat Mapping using Convolutional Neural Networks. In Proceedings of the IEEE OES Autonomous Underwater Vehicle Symposium (AUV), Porto, Portugal, 1–6 November 2018.
46. Le Quilleuc, A.; Collin, A.; Jasinski, M.F.; Devillers, R. Very High-Resolution Satellite-Derived Bathymetry and Habitat Mapping Using Pleiades-1 and ICESat-2. *Remote Sens.* **2022**, *14*, 133. [[CrossRef](#)]
47. Fogliani, F.; Grande, V.; Marchese, F.; Bracchi, V.A.; Prampolini, M.; Angeletti, L.; Castellan, G.; Chimienti, G.; Hansen, I.M.; Gudmundsen, M.; et al. Application of Hyperspectral Imaging to Underwater Habitat Mapping, Southern Adriatic Sea. *Sensors* **2019**, *19*, 2261. [[CrossRef](#)] [[PubMed](#)]
48. Tarantino, C.; Forte, L.; Blonda, P.; Vicario, S.; Tomaselli, V.; Beierkuhnlein, C.; Adamo, M. Intra-Annual Sentinel-2 Time-Series Supporting Grassland Habitat Discrimination. *Remote Sens.* **2021**, *13*, 277. [[CrossRef](#)]
49. Eugenio, F.; Marcello, J.; Martin, J.; Rodríguez-Esparragón, D. Benthic Habitat Mapping Using Multispectral High-Resolution Imagery: Evaluation of Shallow Water Atmospheric Correction Techniques. *Sensors* **2017**, *17*, 2639. [[CrossRef](#)] [[PubMed](#)]
50. Shafique, M.; Sullivan, C.A.; Finn, J.A.; Green, S.; Meredith, D.; Moran, J. Assessing the Distribution and Extent of High Nature Value Farmland in the Republic of Ireland. *Ecol. Indic.* **2020**, *108*, 105700.
51. O’Rourke, E.; Finn, J.A. *Farming for Nature: The Role of Results-Based Payments*, 1st ed.; Teagasc and National Parks and Wildlife Service (NPWS): Dublin, Ireland, 2020; ISBN 978-1-84170-663-4.
52. Fossitt, J.A. *A Guide to Habitats in Ireland*, 1st ed.; Heritage Council: Kilkenny, Ireland, 2000; ISSN 1393-68 08.
53. Sheridan, H.; Keogh, B.; Anderson, A.; Carnus, T.; McMahon, B.J.; Green, S.; Purvis, G. Farmland Habitat Diversity in Ireland. *Land Use Policy* **2017**, *63*, 206–213. [[CrossRef](#)]
54. Smith, G.F.; O’Donoghue, P.; O’Hora, K.; Delaney, E. *Best Practice Guidance for Habitat Survey and Mapping*, 1st ed.; Heritage Council: Kilkenny, Ireland, 2011.
55. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed.; Cambridge University Press: New York, NY, USA, 1992; pp. 123–128. ISBN 0-521-43108-5.
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
57. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 12 August 2016; pp. 1–12.
58. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
59. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
60. 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; pp. 770–778.
61. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
62. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017; pp. 1–13.
63. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856.
64. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
65. Yadav, S.S.; Jadhav, S.M. Deep Convolutional Neural Network based Medical Image Classification for Disease Diagnosis. *J. Big Data* **2019**, *6*, 113. [[CrossRef](#)]
66. Bahri, Y.; Kadmon, J.; Pennington, J. Statistical Mechanics of Deep Learning. *Annu. Rev. Condens. Matter Phys.* **2020**, *11*, 501–528. [[CrossRef](#)]
67. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
68. Abraham, B.; Nair, M.S. Computer-aided Detection of COVID-19 from X-ray Images using Multi-CNN and Bayesnet Classifier. *Biocybern. Biomed. Eng.* **2020**, *40*, 1436–1445. [[CrossRef](#)] [[PubMed](#)]

69. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [[CrossRef](#)]
70. Kandel, I.; Castelli, M.; Popovič, A. Comparative Study of First Order Optimizers for Image Classification Using Convolutional Neural Networks on Histopathology Images. *J. Imaging* **2020**, *6*, 92. [[CrossRef](#)] [[PubMed](#)]
71. Blume, S.; Benedens, T.; Schramm, D. Hyperparameter Optimization Techniques for Designing Software Sensors Based on Artificial Neural Networks. *Sensors* **2021**, *21*, 8435. [[CrossRef](#)] [[PubMed](#)]
72. Kim, K.-S.; Choi, Y.-S. HyAdamC: A New Adam-Based Hybrid Optimization Algorithm for Convolution Neural Networks. *Sensors* **2021**, *21*, 4054. [[CrossRef](#)]
73. Ganaie, M.A.; Hu, M.; Tanveer, M.; Suganthan, P.N. Ensemble Deep Learning: A Review. *arXiv* **2021**, arXiv:2104.02395.