Remote Sensing in Ecology and Conservation
Open Access
ZSL LET'S WORK FOR WILDLIFE

ORIGINAL RESEARCH

Automated detection of bird roosts using NEXRAD radar data and Convolutional neural networks

Carmen Chilson[1], Katherine Avery[1], Amy McGovern[1,2], Eli Bridge[3,4], Daniel Sheldon[5,6] & Jeffrey Kelly[3,4,7]

[1]School of Computer Science, University of Oklahoma, Norman, Oklahoma
[2]School of Meteorology, University of Oklahoma, Norman, Oklahoma
[3]Oklahoma Biological Survey, University of Oklahoma, Norman,Oklahoma
[4]Department of Biology, University of Oklahoma, Norman, Oklahoma
[5]College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, Massachusetts
[6]Department of Computer Science, Mount Holyoke College, South Hadley
[7]Corix Plains Institute, University of Oklahoma, Norman, Oklahoma

Keywords
Aeroecology, bird roosts, deep learning, machine learning

Correspondence
Amy McGovern, School of Computer Science, 110 W Boyd St, Norman, OK 73019, USA. Tel: +1 405 325 5427;
E-mail: amcgovern@ou.edu

Editor: Ned Horning
Associate Editor: Xuehua Liu

Received: 28 February 2018; Revised: 13 June 2018; Accepted: 6 July 2018

doi: 10.1002/rse2.92

Abstract

Although NEXRAD radars have proven to be an effective tool for detecting airborne animals, detecting biological phenomena in radar images often involves a manual, time- consuming data-extraction process. This paper focuses on applying machine learning to automatically find radar data that snapshots large aggregations of birds (specifically Purple Martins and Tree Swallows) as they depart *en masse* from roosting sites. These aggregations are evident in radar images as rings of elevated reflectivity that appear early in the morning as birds depart from roost sites. Our goal was to develop an algorithm that could determine whether an individual radar image contained at least one Purple Martin or Tree Swallow roost. We use a dataset of known roost locations to train three machine learning algorithms that employed (1) a traditional Artificial Neural Network (ANN), (2) a sophisticated preexisting Convolutional Neural Network (CNN) called Inception-v3, and (3) a shallow CNN built from scratch. The resulting programs were all effective at finding bird roosts, with both the shallow CNN and the Inception-v3 network making correct determinations about 90 per cent of the time with an AUC above .9. To the best of our knowledge, this study is the first to apply neural networks in the analysis of bird roosts in radar imagery, and these analytical tools offer new avenues of research into the ecology and behavior of flying animals, with practical applications to wind farm placement, air traffic administration and wildlife conservation. The NEXRAD radar network offers a tremendous archive of continental-scale data and has the potential to capture entire vertebrate populations. We apply existing machine learning models to a new dataset which constitutes a valuable approach to extracting information from this archive.

Introduction

Monitoring animal populations is a key aspect of biological conservation (Shipley et al. 2017), but effective monitoring is often logically and analytically challenging, especially at over large spatial scales. The NEXRAD radar network offers an unparalleled means of detecting the activities of airborne animals that spans almost all of the continental United States. Unfortunately, identification of biological activity evident in radar data generally requires 'significant computational skills and time investment' (Chilson et al. 2012a), which has resulted in a limited amount of biological research invested in radar-based remote sensing (Bauer et al. 2017).

Considerable effort has gone into automatically detecting birds using radar. Radars adapted specifically for bird detection can identify single small- and medium-sized birds flying across a fixed position radar beam (Zaugg

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/301626

et al. 2008). Similarly, Airport Surveillance Radar (ASR-9) also can automatically detect birds and small groups of birds within a range of about 10 km. (Troxel et al. 2001). NEXRAD radars, cannot detect or track individual birds, but they can detect clusters of 'biological targets' up to 240 km away (DeVault et al. 2013, p. 142). Moreover, NEXRAD data are freely available and easily accessed via Amazon Web Services, with archived data extending as far back as 1991. These radar products have been used to study bird roosts in the past, but these efforts required considerable time dedicated to generating biologically meaningful data without an automation tool (Gauthreaux and Belser 1998; Diehl and Larkin 2005; Stepanian et al. 2016).

Here, we attempt to better enable radar-based bird-monitoring efforts though computer automation. Specifically we demonstrate the feasibility of using machine learning to detect large roosting aggregations of Purple Martins and Tree Swallows. These aggregations are identifiable in radar data owing to distinct circular reflectivity patterns known as 'roost rings' (Kelly and Pletschet 2018). Although these patterns are easy for the human eye to detect, amassing a large dataset of roost locations necessitates the time consuming task of sifting through millions of radar images to find those that contain roosts. Automating this process would enable researchers to focus more of their time on the biological interpretation rather than data collection, so investing time at this fundamental stage is key.

In this paper, we evaluate how well machine learning methods such as Artificial Neural Networks and Convolutional Neural Networks can learn to identify bird roosts in NEXRAD radar images. We used 2D images rendered from selected radar data products as inputs to Neural Networks that we tasked with determining if the image contains a roost. We compared several different network architectures and explain which architectures worked best for this problem. We also performed roost detection of data from before and after the NEXRAD upgrade to dual polarization (see Materials and Methods), which allowed us to compare the utility of different radar data products. Although the machine learning tools we used are frequently employed to characterize photographic images, we know of no other efforts to apply them to identify bird roost in radar imagery.

Materials and Methods

NEXRAD radar

The NEXRAD radar network comprises 151 Doppler weather surveillance radars. These radars complete a series of rotational scans every 5–10 min. They scan the atmosphere at different tilts or elevations, however for this paper we only focus on the lowest elevation scan (i.e. 0.5°). Radar data used for this research came from the level 2 NEXRAD radar archive, which is publicly accessible via Amazon Web Services.[1] This dataset extends back to the mid-1990s and contains data from single-polarization Doppler radars (which we refer to as legacy radar in this paper) and dual-polarization Doppler radar. Data from legacy radars include three products: (1) Reflectivity, which is a measure of the reflected energy from objects within a given air volume; (2) Doppler Radial Velocity, which indicates the relative movement of objects toward or away from the radar; and (3) Spectrum Width, which is variability of the mean radial velocity.

During 2012 and 2013, all NEXRAD radars were upgraded to dual-polarimetry, which means that they now transmit and receive radio pulses that are polarized in the vertical and horizontal orientations. This capability allows for better assessment of the shapes of objects detected by radar, and dual-pol radars offer three new data products in addition to the legacy products: (1) Differential Reflectivity (ZDR), which indicates the difference in reflectivity between the horizontal and vertical pulses; (2) Differential Phase (φ_{DP}), which is a measure of the difference between horizontal and vertical pulse phase shifts; and (3) Correlation Coefficient (ρ_{HV}), which assesses the similarity between the behaviors of the horizontally and vertically polarized pulses within a pulse volume.

Spectrum Width and Differential Phase (φ_{DP}) are generally not considered to be useful for detecting bird roosts, so we did not use them as model inputs. Hence used reflectivity, radial velocity, differential reflectivity and correlation coefficient as model inputs. Reflectivity, or echo intensity, provides an overall view of airborne objects and has been shown to be useful for detecting bird roosts as well as calculating the density of birds (Diehl and Larkin 2005). Radial velocity is often useful for identifying birds as it can reveal when airborne objects move in opposition to air currents (Gauthreaux and Belser 1998). Differential Reflectivity (ZDR) often reveals distinctive asymmetries in the reflectivity of biological targets (Stepanian and Horton 2015). Finally, correlation coefficient (ρ_{HV}) is typically lower for biological echoes as opposed meteorological echoes (Van Den Broeke 2013), and it has also been used to determine the orientation of flying birds (Stepanian and Horton 2015).

Roost data

Swallow roosting occurs in the late summer months, and roosts are most apparent in radar data early in the morning, from 20 min before to 40 min after sunrise, when the

 © 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

birds typically leave the roosting site. Kelly and Pletschet have manually identified and mapped hundreds of bird roosts through exhaustive searches of years of radar data from 64 different NEXRAD Radars (Kelly and Petschet 2018). Their search protocol involves searching examining radar imagery from one hour before local sunrise until 30 min after local sunrise from June 1 to September 30 (Kelly and Pletschet 2018), an effort that requires examination of 70,000–140,000 radar images per year.

The information used to train our machine-learning system came in part from a subset of the roost data derived by Kelly and Pletschet (2018), which underwent a post processing step to manually identify which particular radar scans had roosts visible. We also used data generated by an interactive web-page[2] that was used to collect labels for previous research projects (Laughlin et al. 2013, 2016). This data is part on an ongoing database that contains labeled radar data with information about the presence and locations of bird roosts. (Laughlin et al. 2014). The final set of manually labeled roost data came from 10 different radars: KAMX, KBRO, KDOX, KGRK, KJAX, KHGX, KLCH, KLIX, KMLB and KMOB and was a mix of legacy and dual-polarimetry data. A distribution of labels by dataset as well as legacy and dual-pol data can be seen in Figure 1. Both of the datasets contained primarily positive labels, with few instances where it was clear that a roost was not found. To increase the number of negative labels, we selected radar scans from 2 to 1 h before sunrise and from 1 to 2 h after sunrise, leaving a 2 h window in between. The noise in our radar images (dust, weather, sun-streaks, etc.) appeared to be similar directly before, during and after the roost is visible, which should ensure that our machine learning algorithms are detecting roosts as opposed to other patterns in the data to make classifications.

Once the desired radar scans were identified, each was acquired from the AWS database and converted from radial coordinates to two-dimensional raster images. We use the Py-Art library to create the images of the radar products (Helmus and Collis 2016). We used only the lowest radar tilt from each scan (0.5° of elevation), which is where most bird activity is evident. The reflectivity, velocity, ρ_{HV} and ZDR radar products were all saved as individual images. These images serve as the input to our machine learning models.

Table 1 shows how many training labels we have as inputs to our model.

Machine learning methods

We employed three general machine learning approaches to work toward optimizing roost detection in terms of accuracy and computation time. The first approach used a relatively simple traditional, feed-forward artificial neural network (ANN) as depicted in Figure 2. The second approach used a sophisticated convolutional neural network–specifically the 'Inception' network created by Szegedy et al. (2016)–as a starting point and modified the last two layers to tailor it for roost identification. The third approach developed a shallow convolutional network from scratch, with only two convolutional layers as depicted in Figures 3 and 4.

We processed each of the four types of radar imagery (reflectivity, radial velocity, differential reflectivity and correlation coefficient) separately using each machine learning approach as shown in Figure 5. When only legacy data were available, only two types of imagery (reflectivity and radial velocity) were used. In the final step, the results from each of the four networks was combined within a dense neural network to generate an aggregate classification for each image (either containing a roost or not). One advantage of training the networks on the radar fields separately instead of together is that is reduces the number of input variables a single network is required to train on. Our design was inspired by an approach that was used successfully to train separate convolution layers in parallel, each on a different image rotation (Dieleman et al. 2015). The results were then fed into dense layers of the network (Dieleman et al. 2015).

To improve the speed and accuracy of our ANN and Shallow CNN we employed batch normalization (Ioffe and Szegedy 2015). For the transfer learning comparison we chose a network that also used batch normalization, the Inception-v3 network (Szegedy et al. 2016). Batch normalization employs an additional step within each node to normalize their outputs over a batch of images, which stabilizes the node outputs and hastens convergence toward useful sets of weights. Batch normalization was first introduced in 2015 and it improved the accuracy of ImageNet classification while simultaneously speeding up learning 14 times (Ioffe and Szegedy 2015). All the networks were trained using binary cross-entropy as the loss function, and we used average accuracy as the metric to see how well training progressed.

Our traditional ANN employed multiple layers of artificial neurons or nodes. In a machine-learning context, a node combines a vector of inputs (which might be pixel values) with a corresponding vector of weights to generate a summation, or output, that gets transformed by an activation function and passed to the next layer in the network. ANNs consist of highly connected layers, and the weighted connections among the nodes change as the network is trained to recognize the input data (Mitchell 1997). For image classification problems, ANNs have an input layer that takes in pixel values for an image, a number of hidden layers that change the weights, and an

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

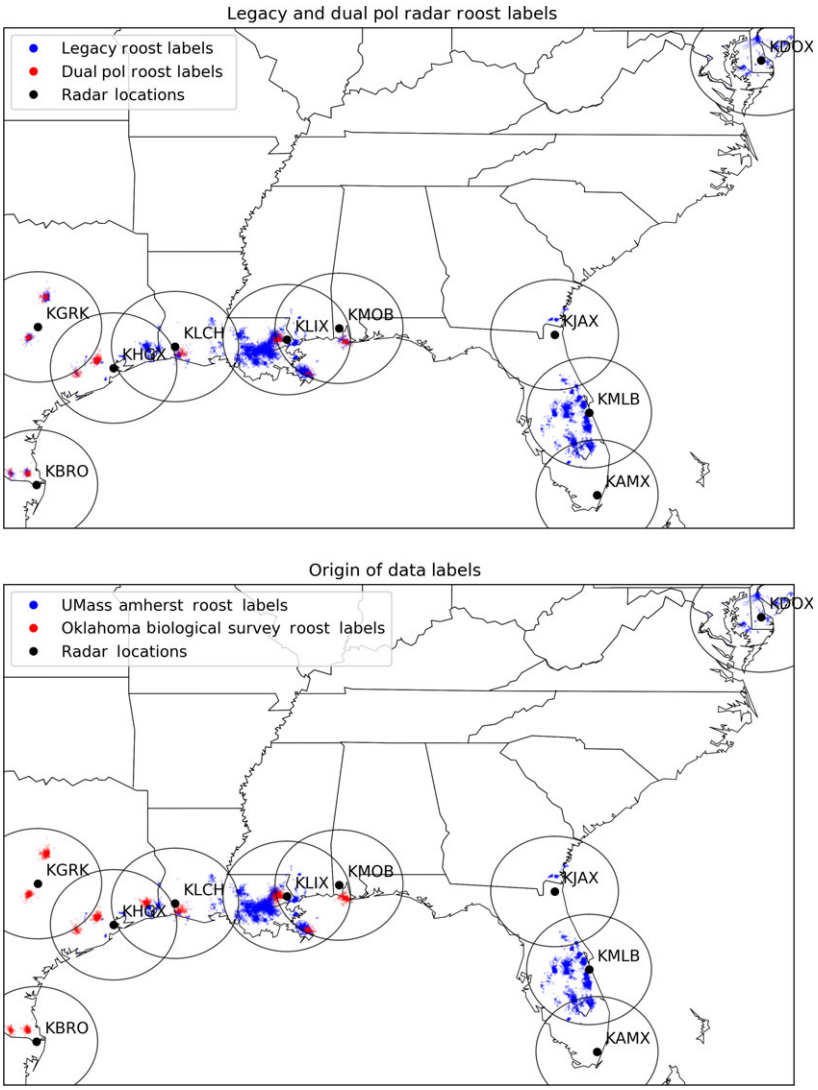


Figure 1. Visual distribution of roost labels form the Oklahoma Biological Survey and UMass Amherst citizen science labels as well as legacy radar and dual-pol radar data. This figure shows where each roost was found. The color is darker if the roost was spotted in the exact same place multiple times. This image also shows the location and 300 km visibility radius of the radars.

Table 1. The distribution of labels. This table lists how many dual-pol radar labels exist within the data.

	Roost	No Roost
Legacy	11,112	19,939
Dual-Pol	1,346	10,806

output layer with a number of nodes equal to the number of classes (Driss et al. 2017). The ANN we used had an input layer, three hidden layers, and an output layer, and it worked on gray-scale images. The input layer accepted the 240 × 240 images and connected them to subsequent layers. The three hidden layers consisted of 128, 64 and 8 nodes. Each of these layers used batch normalization and a rectified-linear-unit activation function. The output layer, which had only two nodes, and used the softmax activation function to generate probabilities for the two possible classifications.

As a second approach, we used Convolutional Neural Networks (CNNs). Convolutional networks use kernels (i.e. arrays or matrices of weighting values) that are applied across an image to identify features that contribute to image classification. In 2012, Deep CNNs achieved record breaking results for classifying the thousands of annotated images that comprise the ImageNet dataset (Krizhevsky et al. 2012). Subsequent work has expanded on CNNs using varying architectures such as VGG16, GoogLeNet, Inception-v3 and ResNet to improve CNN results for classifying images (Simonyan and Zisserman 2014; Ioffe and Szegedy 2015; Szegedy et al. 2015, 2016; He et al. 2016). For this paper we chose to use the Inception-v3 network. Although CNNs produce robust results, they require large amounts of training data to be

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

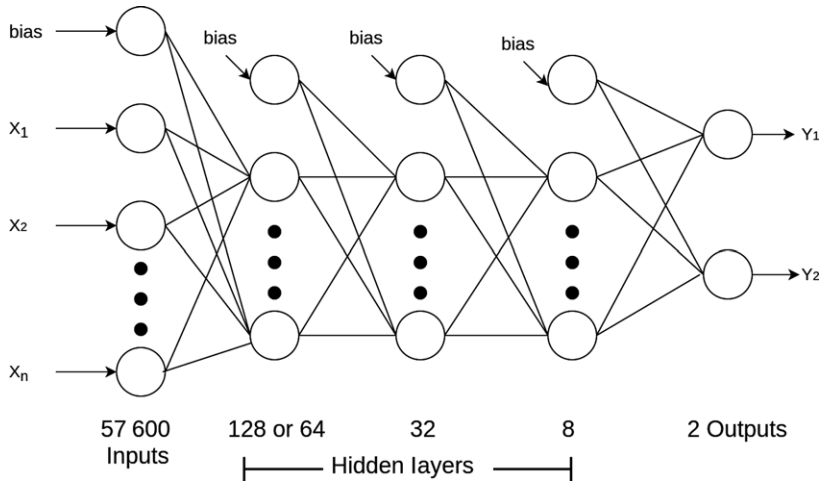


Figure 2. The design of the ANN network used to detect bird roosts. We used a traditional feed- forward classification Neural Network. As input the network takes the flattened 240 × 240 image and the network outputs a classification probability for each label (No Roost, Roost). The ANN is made up of many connected neurons. The inputs to each neuron are multiplied with weights and then summed with the bias node. This value is then passed through an activation function to produce the neuron output.

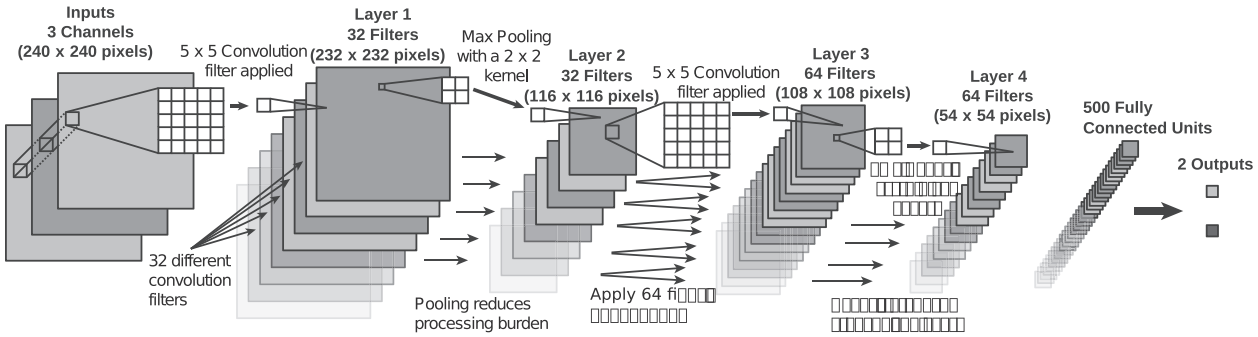


Figure 3. Overview over the shallow CNN architecture employed for bird roost detection. Pixel kernels from the original input layers are processed through a series of convolution steps that apply various filters to the data with alternating pooling steps that down-sample the pixels to reduce the processing burden and adds mild translational invariance. Each convolution step applies a shared set of weights across a moving window with dimensions of 5 × 5 pixels, to extract features from the images. The convolution layers are followed by a fully connected layer which flattens the pixels. The data are multiplied with weights, summed and softmax activation is applied to produce the final outputs.

effective since the network has to modulate millions of parameters (Oquab et al. 2014).

Because we lacked sufficient data to fully train a deep CNN from scratch, we employed a transfer-learning approach, wherein foundational knowledge learned on one dataset is applied to a new dataset (Oquab et al. 2014). As an initial foundation for our CNN we used Inception-v3 (Szegedy et al. 2016), a network trained on ImageNet (Deng et al. 2009), to make 1000 image classifications. The initial nodes in the Inception-v3 network have learned to identify general features (e.g. edges, shadows and curves) that can be applied generically to other image data (Shin et al. 2016). We initialize our model with the weights learned during the feature extraction part of training the Inception-v3 network in ImageNet. We then replace the classification part of the model (the last two layers) in order to perform 2-way classification instead of 1000-way classification. The last two layers consist of a fully connected layer and a softmax output layer. The weights of all except the very last two layers of the Inception-v3 network are frozen during training, and only the last two layers were fine tuned using the novel radar dataset (Shin et al. 2016). This transfer-learning approach allows us to train deeper CNNs with smaller training datasets as there are fewer weights and parameters that the network is required to optimize. A full description of the Inception-v3 network can be found in Szegedy et al. (2016). The CNNs we employed used full color images for each radar product that were derived from

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

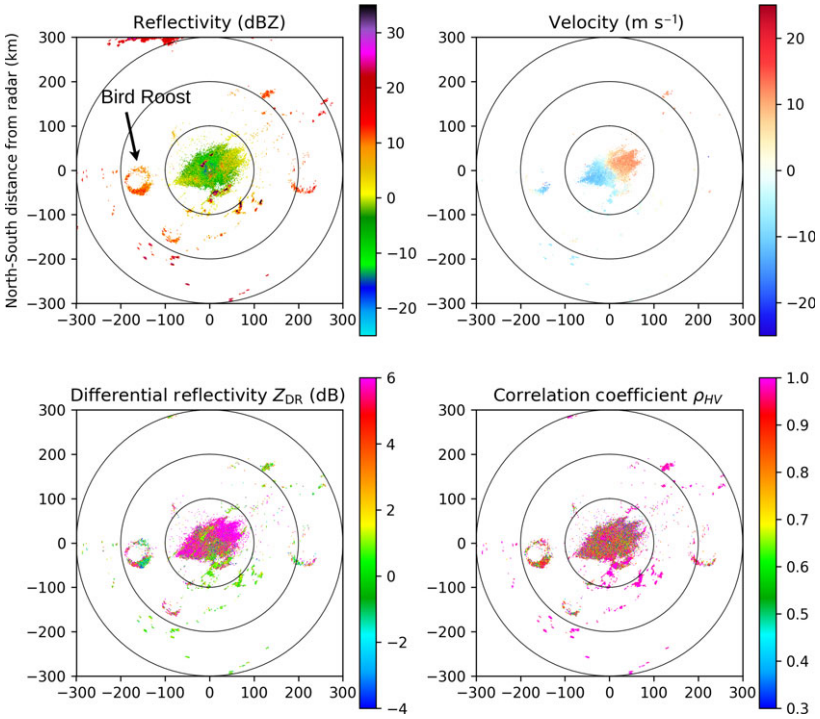


Figure 4. Machine Learning input. This is an example of a radar image that contains a roost. The location of the bird roost is annotated within the image. This is from the KMOB radar from July 4th 2015, 11:19 UTC. Image created using the Py-ART library (Helmus and Collis 2016).

a standard NEXRAD color scheme implemented in Py-ART (Helmus and Collis 2016). Color images were used because the Inception-v3 CNN was originally trained on color image inputs.

Our third approach trained a shallow CNN from scratch. The shallow CNN had only two convolution layers and one fully connected layer in the network. As with the Inception-v3 CNN, the shallow CNN used RGB channel values from each radar product as input. The first convolution layer employed 32 filters, a kernel size of 5, batch normalization and rectified linear units (as an activation function). The second convolution layer has the same setup except with 64 filters. After each convolution layer, we down-sampled the data using max pooling with a pool size of 2 and a stride size of 2.

Metrics

We used four different metrics for evaluating our machine learning results. We evaluated the total accuracy (ACC), the true positive rate (TPR), the true negative rate (TNR) and the area under the receiver operating characteristics (ROC) curve (AUC). ACC, TPR and TNR can all be calculated using the number of true positives (TP), true negatives (TN), false positive (FP) and false negatives (FN).

$$\mathrm{ACC} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}) \quad (1)$$

$$\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \quad (2)$$

$$\mathrm{TNR} = \mathrm{TN}/(\mathrm{FP} + \mathrm{TN}) \quad (3)$$

A ROC curve can be used for visualizing a classifier's performance and the AUC can be used to compare different ROC curves (Fawcett 2006). AUC values range from 0 to 1 where 1 shows a perfect classifier and a score of 0.5 represents random guessing (Fawcett 2006). An AUC value of .9 or above is considered to be a good result. For more details on how to calculate the AUC see Fawcett (2006).

Model training and validation

Establishing a machine-learning classifier typically involves training, validation and testing sets to determine whether the trained models have arrived at a robust solution (Cohen 1995). Because our classification models combined inputs from multiple radar products, our training, validation, and testing was implemented as a two-step process. First the models assigned a classification probability to each radar product separately. Then these classification probabilities served as the inputs to a second classification layer. Ideally these two machine-learning steps would be trained with two different validation sets. However, we did not have a sufficient number of labels to take this approach. Therefore we split the data into three different groups that we will refer to as A, B and C. Group A contained 60% of the data and the remaining 40% were split equally between groups B and

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

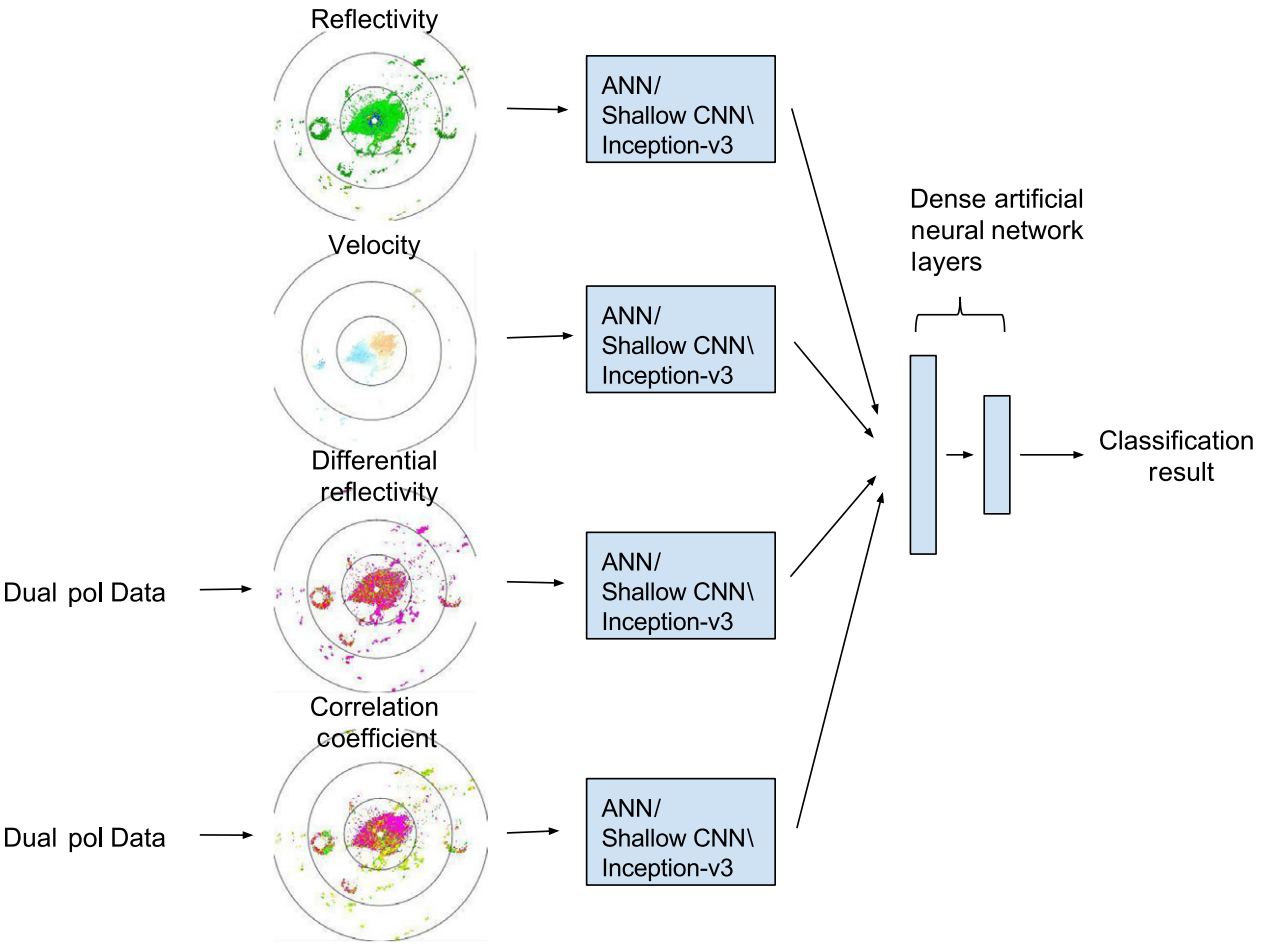


Figure 5. Design of the machine learning classification system for dual polarization data and legacy data.

C. When training the models to detect bird roosts from image data for a single radar product we use A as our training set, B as our validation set, and C as our testing set. For the second stage of learning, we input the probabilities of the different radar products into a hidden layer for the aggregate classifier. At this stage of the problem we use B as our training set, A as the validation set, and C as the test set. We swap A and B in order to get a second validation set as well as give the network data to train on that will be as similar to the test set as possible. C is consistently used as the test set throughout. In each of these cases the entire dataset is used either in train, test, or validation.

To assess confidence in our final classifications we used a K-fold cross-validation, wherein the data were partitioned into k subsets (folds), with training performed on $k-2$ folds, validation performed on 1 fold and testing performed on the remaining fold (Kohavi 1995). In our case, we put 3 folds into set A, 1 fold into set B and 1 fold into set C. The training is repeated k times, where each fold is used as the testing and validation fold exactly once. K-fold allows us to evaluate every labeled datum. We used 5-fold cross-validation to train and evaluate our models. We chose a small number for k because convolutional neural networks are computationally expensive to train.

For each of our metrics we calculated the confidence interval using the bootstrapping percentile method. The percentile method calculates the chosen metric (e.g. loss or accuracy) on randomly selected samples of the data iteratively (Efron and Tibshirani 1986). Then for a 95% confidence interval we take the upper and lower 2.5% points of distribution (Efron and Tibshirani 1986). This is a range that 95% of the bootstrapped samples fall within. The upper and lower bound of the distribution become the confidence interval for the performance metric (Efron and Tibshirani 1986).

Each testing fold was evaluated using its corresponding network. The results from each of the testing folds were then combined. To compute the confidence intervals for

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

ACC and AUC, we randomly selected one thousand samples with re-sampling from the combined testing results. For TPR we select one thousand samples from the roost data and for TNR we select one thousand samples from the no roost data. We repeat this process for one thousand iterations on each of our metrics in order to compute the confidence intervals.

Results

Of the three different machine learning networks we trained, the shallow CNN and Inception-v3 aggregate classifiers produced the best results with an accuracy of 90%. The Inception-v3 aggregate classifier has the highest true positive rate, and the shallow CNN and Inception-v3 Dual-Pol aggregate classifier has the highest true negative rate. These results are averages from five runs of each of the networks. The full results for each network are included in Table 2 and shown visually in Figure 6.

We predicted that both the Inception-v3 network and the shallow CNN would outperform the traditional ANN since CNNs are designed to exploit spatial context and have been shown to be a superior method for image classification. We also assumed that the Inception-v3 network would outperform the shallow CNN since it had more layers and was pre-trained on ImageNet. The ANN accuracy for three of the four radar products was higher than Inception-v3 network even though we expected the Inception-v3 to outperform the ANN. It is worth noting that although the ANN accuracies are high, they are biased toward a classification of 'no roost'. The Inception-v3 network has slightly lower accuracies than the ANN, however unlike the ANN the Inception-v3 network results are not as biased toward a single class of data.

The Inception-v3 network performed worse on individual radar product than expected. We believe that transfer learning would have achieved a higher accuracy and AUC if the Inception-v3 network was initially trained on a large set of radar data. Typical photographic images are differ from radar images and may require different convolutional filters that may not necessary translate to radar data. Photographic images contain shadow, light, objects in the foreground and background, lines, edges, etc. It may also have helped if we trained the lower layers of the Inception-v3 network instead of relying on ImageNet to find useful features for radar data. Another reason the Inception-v3 network may not have performed as well as expected is that we did not have enough radar data, especially dual-polarimetric radar data, to effectively train this network. The Inception-v3 network learns to utilize a wide range of image properties in the features, and it may take more training data to fully utilize this information.

The shallow CNN produced the highest accuracy and true negative rate for the reflectivity and correlation coefficient radar products. This network's True Positive Rate

Table 2. Machine learning results for the ANN, Inception-v3 Net and Shallow CNN.

	ACC	TPR	TNR	AUC
ANN				
Reflectivity	0.814–0.860	0.651–0.709	0.877–0.917	0.853–0.901
Velocity	0.522–0.585	0.963–0.983	0.361–0.420	0.850–0.898
Differential Reflectivity	0.872–0.909	0.421–0.484	0.931–0.959	0.794–0.881
Correlation Coefficient	0.780–0.830	0.573–0.634	0.804–0.850	0.761–0.851
Legacy Final	0.778–0.828	0.769–0.820	0.782–0.832	0.866–0.910
Dual-Pol Final	0.739–0.794	0.396–0.458	0.783–0.833	0.542–0.684
Inception				
Reflectivity	0.757–0.808	0.803–0.848	0.728–0.779	0.839–0.884
Velocity	0.770–0.819	0.789–0.837	0.757–0.807	0.851–0.894
Differential reflectivity	0.746–0.796	0.819–0.865	0.732–0.787	0.847–0.905
Correlation coefficient	0.712–0.766	0.774–0.824	0.705–0.760	0.805–0.874
Legacy aggregate	0.848–0.889	0.857–0.896	0.841–0.885	0.929–0.956
Dual-Pol Aggregate	0.906–0.938	0.923–0.952	0.904–0.937	0.971–0.990
Shallow CNN				
Reflectivity	0.873–0.912	0.785–0.832	0.920–0.950	0.937–0.964
Velocity	0.636–0.692	0.000–0.002	0.999–1.00	0.684–0.754
Differential reflectivity	0.882–0.919	0.727–0.781	0.903–0.936	0.912–0.953
Correlation coefficient	0.901–0.935	0.697–0.751	0.927–0.955	0.910–0.956
Legacy aggregate	0.875–0.913	0.797–0.844	0.915–0.947	0.930–0.961
Dual-Pol aggregate	0.905–0.938	0.813–0.857	0.916–0.948	0.931–0.970

TPR, true positive rate; TNR, true negative rate.

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

was lower than the True Negative Rate for every radar field, which means it was biased toward assuming the radar images don't contain roosts. The CNN and Inception-v3 legacy accuracies, AUCs and True Positive Rates were not statistically significantly different from each other, however the Inception-v3 had a higher True

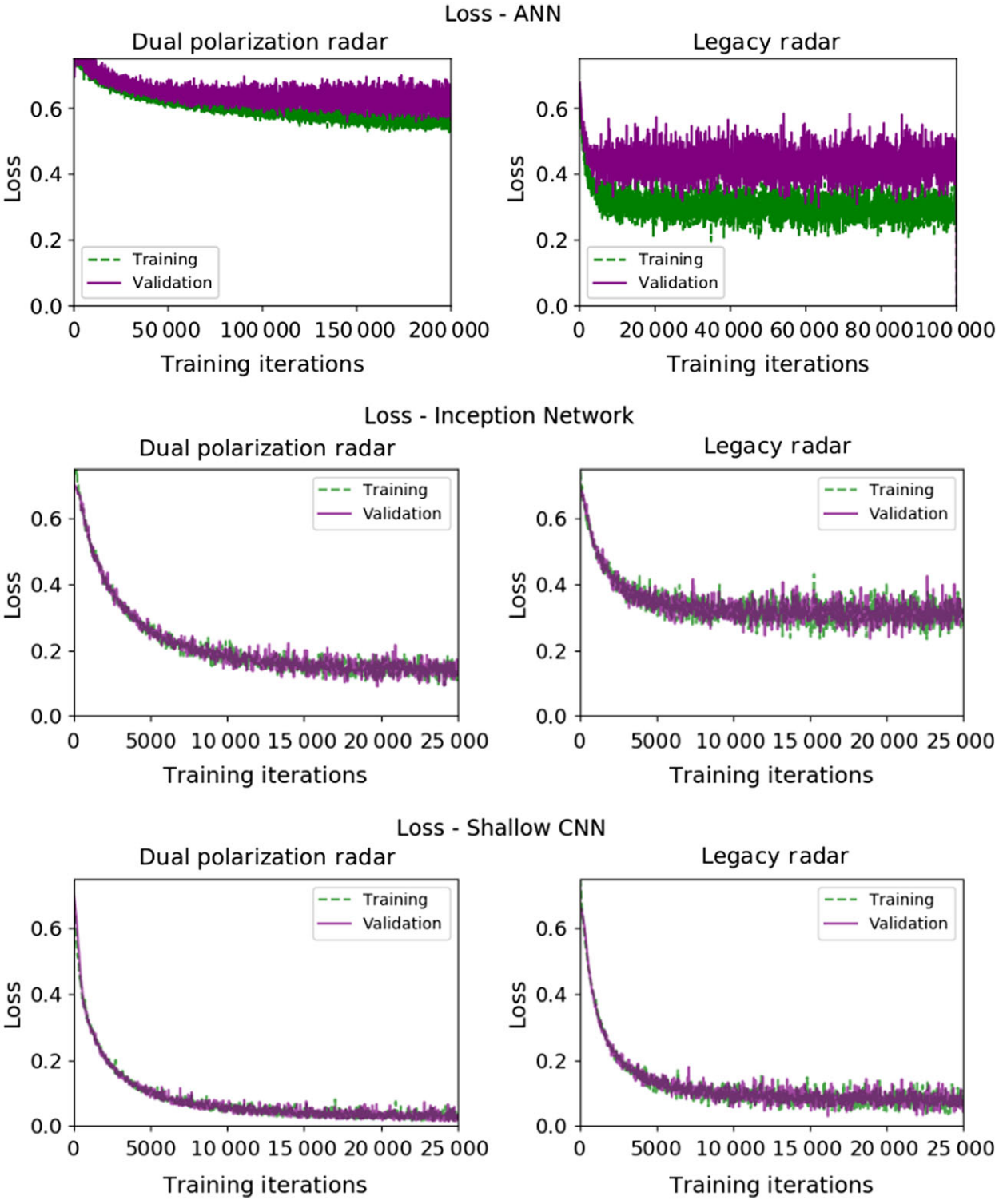


Figure 6. Learning Curves for the aggregate classifiers given probabilities as inputs.

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

Positive Rate. For the Inception-v3 and shallow CNN Dual-Pol aggregate classifier the Inception-v3 has a higher AUC and TPR, however the other two metrics are not statistically different for these two networks.

The ROC curves for all of the networks can be found in Figure 7. Of the single radar product networks, the Shallow CNN had the largest Area Under Curve (AUC) with the exception of Velocity. The ANN had the lowest AUCs for all of the radar products except for Velocity, which match the results of the Inception-v3 network. Although the Inception-v3 network overall performed better than the ANN and worse than the Shallow CNN, it was able to outperform the shallow CNN on the velocity AUC. The ANN's aggregate legacy results show no improvement over the single radar products results, and the ANN's aggregate Dual-Pol results are worse than the single radar product results. Both of these results are surprising. The Inception-v3 and Shallow CNN aggregate

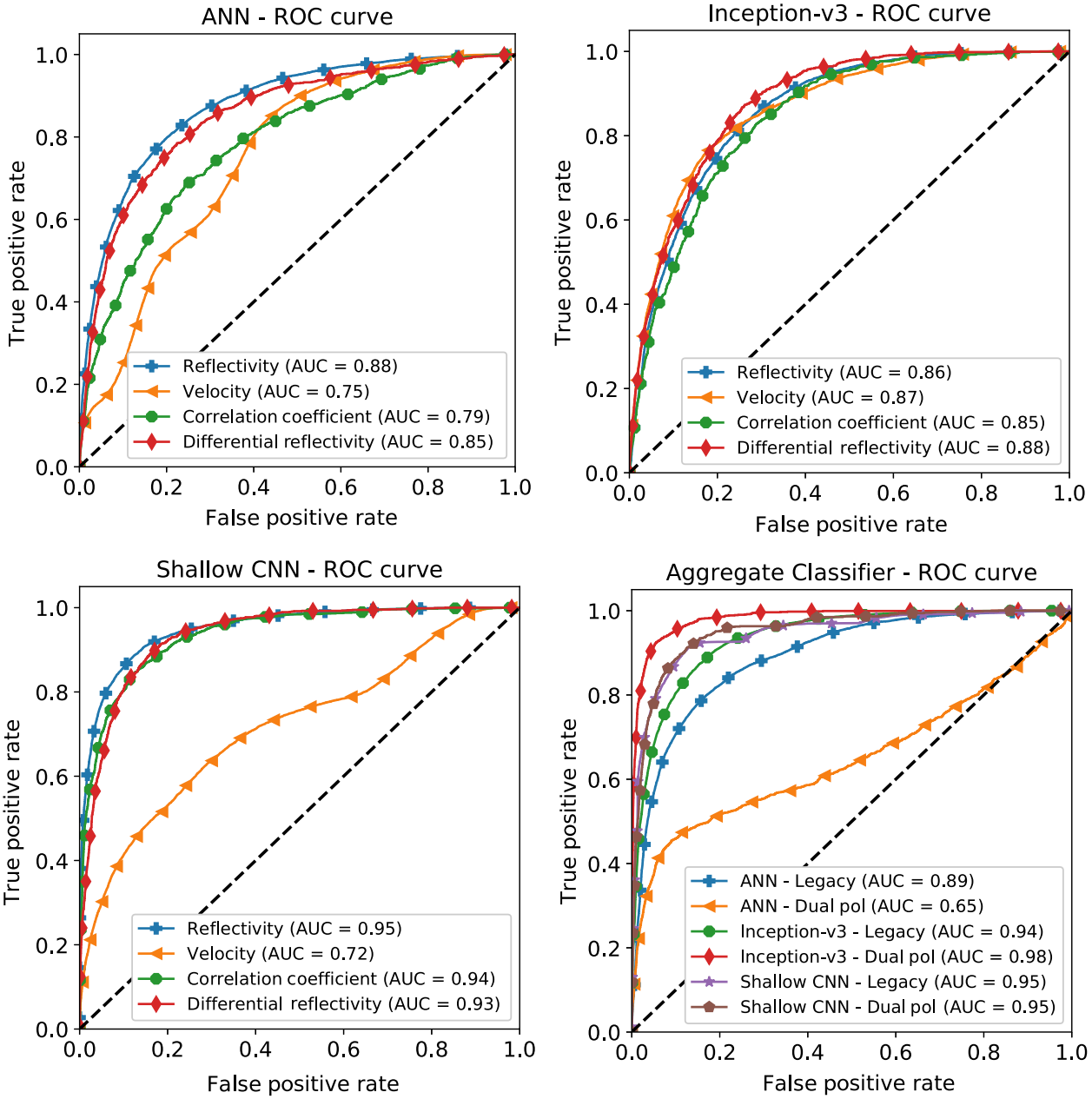


Figure 7. ROC curves for each of machine learning models trained on Reflectivity, Velocity, Correlation Coefficient, Differential Reflectivity and the aggregate results of the combined radar products.

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

networks were both able to outperform the single radar product networks. The Inception-v3 and Shallow CNN perform equally well on legacy data and the Inception-v3 network has the highest AUC of the three networks on the Dual-Pol data.

Discussion

Our classification method vastly reduces the number of images that need to be manually searched through in order to find the bird roosts, especially since most radar images do not contain visible bird roosts. Eliminating the final 10% of false positives from the dataset by hand will be much less time consuming that sifting through all 70,000 radar images a year searching for bird roosts. We were able to reduce the amount of time it takes to process radar image data and we believe these results can be improved in the future with a temporal analysis of the data and more dual polarization labels. Research projects that study pre-migratory roosts using radars (Kelly and Pletschet 2018; Bridge et al. 2016; Gauthreaux and Belser 2003; Chilson et al. 2012b) could benefit from these results.

Future work

In the future, we hope to fully automate the bird roost detection process by preprocessing the radar images. Just as radar data is quality-controlled for weather (Lakshmanan et al. 2014), we could filter out weather from our radar data to eliminate some of the noise from our radar data. In addition, biological reflectivity generally falls within a range of −10 dBZs to 10 dBZs (Koistinen 2000), and by filtering out values outside this range we can eliminate some of the noise. We cannot use a reflectivity filter to fully determine where birds are since light drizzle and insects are often detected in this range as well (Koistinen 2000). Biological scatter will likely have a high differential reflectivity and a low correlation coefficient (Van Den Broeke 2013), and we could use these properties to further filter and clean the data.

We are also not currently taking advantage of the temporal component of the data during learning. The expanding roost rings over sequential radar snapshots are an important roost characteristic used in manual detection of bird roosts. There are several machine learning methods such as Recurrent Neural Networks (RNN) or Long Short Term Memory networks (LSTM) that use temporal data. LSTM networks (Donahue et al. 2015) have been used on sequences of images, for example to re-identifying a person over disjoint cameras (Wu et al. 2016) or to detect the type of activity (run, jump, etc.) a person is performing in a video (Yeung et al. 2018). An LSTM network is one way to potentially increase accuracy using temporal data, although it's worth mentioning that they can require more labeled data since they need to learn a larger amount of parameters. They can also take longer to train, although this should not be an issue for a small network like the shallow CNN.

Our results could be improved with additional Dual Polarization Labels. As stated above, CNNs require lots of data to train properly (Oquab et al. 2014). Our dual polarization radar results were better than our legacy radar results even though we had fewer dual polarization machine learning inputs. Hand classifying roost data is a time consuming process, however, it would be useful for better automated roost detection. One of the advantages of polarimeter radar for weather is that it helps quality control the biology more accurately from the weather data (Zrnic and Ryzhkov 1998). It stands to reason that the same method that is used to remove biology from the radar data can be used to find it as well.

The biggest next step for this project is locating the bird roosts within the radar images instead of only detecting them, which is a very challenging problem. Image segmentation or Regional-CNNs are both potential approaches to this research. There is a trade-off between recognizing and locating objects within an image (Maggiori et al. 2017), so having a network that can detect roosts first will prove useful for the next stage of this research.

Conclusion

We have tested and compared three different machine learning architectures for detecting bird roosts in radar images. Out best model is able to achieve a 0.971–0.990 AUC and 0.906–0.938 ACC for the dual polarization radar data and a 0.875–0.913 AUC and 0.930–0.961 ACC for legacy radar data. This is the first successful attempt to detect bird roosts in radar images that we are aware of. We hope to improve these results in the future by doing some quality control on the radar data, using the temporal data in machine learning, and training on additional labels. The next big step in this project is to build a machine learning model to locate the roosts within the images.

Acknowledgement

The funding from the NSF-DGE-1545261 grant helped make this research possible. We thank Sandra Pletschet for her time spent collecting the roost data and Dr. Phillip Chilson for his advice on the project. Some of the computing for this project was performed at the OU Supercomputing Center for Education & Research (OSCER) at the University of Oklahoma (OU).

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

Notes

[1]aws.amazon.com/public-datasets/nexrad
[2]radar.cs.umass.edu/roost-label/

References

Bauer, S., J. W. Chapman, D. R. Reynolds, J. A. Alves, A. M. Dokter, M. M. Menz, et al., et al. 2017. From agricultural benefits to aviation safety: realizing the potential of continent-wide radar networks. *Bioscience* **67**(10), 912–918.

Bridge, E. S., S. M. Pletschet, T. Fagin, P. B. Chilson, K. G. Horton, K. R. Broadfoot, et al. 2016. Persistence and habitat associations of purple martin roosts quantified via weather surveillance radar. *Landscape Ecol.* **31**(1), 43–53.

Chilson, P. B., E. Bridge, W. F. Frick, J. W. Chapman, and J. F. Kelly. (2012a). Radar aeroecology: exploring the movements of aerial fauna through radio-wave remote sensing.

Chilson, P. B., W. F. Frick, J. F. Kelly, K. W. Howard, R. P. Larkin, R. H. Diehl, et al. 2012b. Partly cloudy with a chance of migration: weather, radars, and aeroecology. *Bull. Am. Meteor. Soc.* **93**(5), 669–686.

Cohen, P. R. 1995. *Empirical methods for artificial intelligence* Vol. **139**. MIT press Cambridge, MA.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: a large- scale hierarchical image database. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 248–255.

DeVault, T. L., B. F. Blackwell, and J. L. Belant. 2013. *Wildlife in airport environments: preventing animal–aircraft collisions through science-based management*. JHU Press.

Diehl, R. H., R. P. Larkin. 2005. Introduction to the wsr-88d (nexrad) for ornithological research.

Dieleman, S., K. W. Willett, and J. Dambre. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. R. Astron. Soc.* **450**(2), 1441–1459.

Donahue, J., L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, et al. 2015. Long-term recurrent convolutional networks for visual recognition and description. *Proc. Conf. Comput. Robot Vis.* **39**(4): 677–691.

Driss, S. B., M. Soua, R. Kachouri, and M. Akil. 2017. A comparison study between mlp and convolutional neural network models for character recognition. SPIE Conference on Real-Time Image and Video Processing, Vol. 10223.

Efron, B., and R. Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Stat. Sci.* **1**, 54–75.

Fawcett, T. 2006. An introduction to roc analysis. *Pattern Recogn. Lett.* **27**(8), 861–874.

Gauthreaux, S. A. Jr, and C. G. Belser. 1998. Displays of bird movements on the wsr-88d:atterns and quantification. *Weather Forecasting* **13**(2), 453–464.

Gauthreaux, S. A. Jr, and C. G. Belser. 2003. Radar ornithology and biological conservation. *Auk* **120**(2), 266–277.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. Pp. 770–778 *in* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV. https://ieeexplore.ieee.org/document/7780459/

Helmus, J., and S. Collis. 2016. The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language. *J. Open Res. Softw.* **4**(1), e25.

Ioffe, S., and C. Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. Pp. 448–456 *in* F. Bach and D. Blei, eds. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15), Vol. 37. https://dl.acm.org/citation.cfm?id=3045118.3045167

Kelly, J. F., and S. M. Pletschet. 2018. Accuracy of swallow roost locations assigned using weather surveillance radar. *Remote Sens. Ecol. Conserv.* **4**, 166–172.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Pp. 1137–1143 *in* Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 (IJCAI'95), Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. https://dl.acm.org/citation.cfm?id=1643047

Koistinen, J. 2000. Bird migration patterns on weather radars. *Phys. Chem. Earth Part B* **25**, 1185–1193.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* **1**, 1097–1105.

Lakshmanan, V., C. Karstens, J. Krause, and L. Tang. 2014. Quality control of weather radar data using polarimetric variables. *J. Atmos. Ocean Tech.* **31**(6), 1234–1249.

Laughlin, A. J., C. M. Taylor, D. W. Bradley, D. Leclair, R. C. Clark, R. D. Dawson, et al., et al. 2013. Integrating information from geolocators, weather radar, and citizen science to uncover a key stopover area of an aerial insectivore. *Auk* **130**(2), 230–239.

Laughlin, A. J., D. R. Sheldon, D. W. Winkler, and C. M. Taylor. 2014. Behavioral drivers of communal roosting in a songbird: a combined theoretical and empirical approach. *Behav. Ecol.* **25**(4), 734–743.

Laughlin, A. J., D. R. Sheldon, D. W. Winkler, and C. M. Taylor. 2016. Quantifying non-breeding season occupancy patterns and the timing and drivers of autumn migration for a migratory songbird using doppler radar. *Ecography* **39**(10), 1017–1024.

Maggiori, E., Y. Tarabalka, G. Charpiat, and P. Alliez. 2017. High-resolution image classification with convolutional networks. Pp. 5157–5160 *in* 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX. https://ieeexplore.ieee.org/document/8128163/

 © 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.

Mitchell, T. M. 1997. *Machine Learning*, 1st ed. McGraw-Hill, Inc. New York, NY.

Oquab, M., L. Bottou, I. Laptev, and J. Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. Pp. 1717–1724 *in* 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH. https://ieeexplore.ieee.org/document/6909618/

Shin, H.-C., H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, et al. 2016. Deep convolutional neural networks for computer-aided detec- tion: Cnn architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**(5), 1285–1298.

Shipley, J. R., J. F. Kelly, and W. F. Frick. 2017. Toward integrating citizen science and radar data for migrant bird conservation. *Remote Sens. Ecol. Conserv.* **4**, 127–136.

Simonyan, K., and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Stepanian, P. M., and K. G. Horton. 2015. Extracting migrant flight orientation profiles using polarimetric radar. *IEEE Trans. Geosci. Remote Sens.* **53**(12), 6518–6528.

Stepanian, P. M., K. G. Horton, V. M. Melnikov, D. S. Zrnić, and S. A. Gauthreaux. 2016. Dual-polarization radar products for biological applications. *Ecosphere* **7**(11), e01539.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., et al. 2015. Going deeper with convolutions. Cvpr.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. Pp. 2818–2826 *in* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV. https://ieeexplore.ieee.org/document/7780677/

Troxel, S., M. Isaminger, B. Karl, M. Weber, and A. Levy. 2001. Designing a terminal area bird detection and monitoring system based on asr-9 data.

Van Den Broeke, M. S. 2013. Polarimetric radar observations of biological scatterers in hurricanes irene (2011) and sandy (2012). *J. Atmos. Ocean Tech.* **30**(12), 2754–2767.

Wu, L., C. Shen, and van den Hengel A.. 2016. Convolutional LSTM networks for video- based person re-identification. arXiv preprint arXiv:1606.01609.

Yeung, S., O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. 2018. Every moment counts: dense detailed labeling of actions in complex videos. *Int. J. Comput. Vision* **126**, 375.

Zaugg, S., G. Saporta, E. Van Loon, H. Schmaljohann, and F. Liechti. 2008. Automatic identification of bird targets with radar via patterns produced by wing flapping. *J. R. Soc. Interface* **5**(26), 1041–1053.

Zrnic, D. S., and A. V. Ryzhkov. 1998. Observations of insects and birds with a polari- metric radar. *IEEE Trans. Geosci. Remote Sens.* **36**(2), 661–668.

Supporting Information

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

Figure S1. Learning Curve for training the ANN.
Figure S2. Learning Curve for retraining the Inception-v3 network.
Figure S3. Learning Curve for training the shallow CNN network.

© 2018 The Authors. *Remote Sensing in Ecology and Conservation* published by John Wiley & Sons Ltd on behalf of Zoological Society of London.