Distinguishing Weather Phenomena from Bird Migration Patterns in Radar Imagery

Aruni RoyChowdhury Daniel Sheldon Subhransu Maji Erik Learned-Miller
University of Massachusetts, Amherst
{arunirc,sheldon,smaji,elm}@cs.umass.edu

Abstract

Data archived by the United States radar network for weather surveillance is useful in studying ecological phenomena such as the migration patterns of birds. However, all such methods require a manual screening stage from domain experts to eliminate radar signatures of weather phenomena, since the radar beam picks up both biological and non-biological targets. Automating this screening step would be of significant help to the large-scale study of ecological phenomenon from radar data. We apply several techniques to this novel task, comparing the performance of Convolutional Neural Networks (CNNs) models against a baseline of the Fisher Vector model on SIFT descriptors. We compare the performance of deeper and shallower network architectures, deep texture models versus the regular CNN model and the effect of fine-tuning ImageNet pre-trained networks on radar imagery. Fine-tuning the networks on the radar imagery provides a significant boost, and we achieve an accuracy of 94.4% on a dataset of 13,194 radar scans, 3,799 of which contained rain.

1. Introduction

In order to get an understanding of ecological phenomena, such as bird migration patterns, it is necessary to acquire and study large scale datasets describing such phenomena. There are three main ways of acquiring such data [8]: by deploying novel sensors, using data collected from citizen science volunteers or by repurposing existing sources of data.

In the first case, given the millions of individual birds in flight during migration, movement-tracking sensors can realistically be placed on only a small fraction of the total number of birds whose movements we intend to model. These would not be fully representative of the population and thus this method of acquiring data is not practical.

Using data from volunteer bird watchers combined with machine learning is a possible alternative, as explored by Fink et al. [11, 10]. However, as with most humans-in-the-loop systems, there are challenges such as the quality of the data and the wide variation in the competence level of the volunteers [16].

The third alternative is to use existing sources of data and extract relevant information from it. This approach has recently been used to reconstruct the velocities of migrating birds using data recorded by weather radars by Sheldon et al. [21]. An extended version of the above work can be found in Farnsworth et al. [8]. They acquired data archived from the early 1990 onwards of the Weather Surveillance radar – 1988 Doppler (WSR-88D) network, consisting of 159 radar stations, covering most of the United States. Although these were intended originally to record weather phenomena [6, 26], the radar stations are also able to detect biological airborne objects such as birds, bats and insects [18].

Among various limiting factors in analysing this abundant source of available data is the fact that these radar scans pick up weather phenomenon such as precipitation along with migrating birds. The presence of non-biological activity like rain would naturally distort the actual readings of bird migration patterns. It requires manual screening by domain experts to identify which scans have the presence of rain. This is not practical when dealing with massive amounts of data – Farnsworth et al. [8] estimate that a single night at peak migration time will generate approximately 15,000 scans across the United States. Thus, *automatically being able to identify the presence of rain would be an important pre-processing step in the further accurate analysis of such data*. Indeed, Sheldon et al. [21] mention screening 351 scans to eliminate those cases that had presence of non-biological targets within a radius of 37.5 km of the station, before proceeding with their automated processing of velocity data from the scans. Our proposed automated system would enable such models to work accurately on much larger amounts of data without being restricted by the time and cost associated with human annotators.

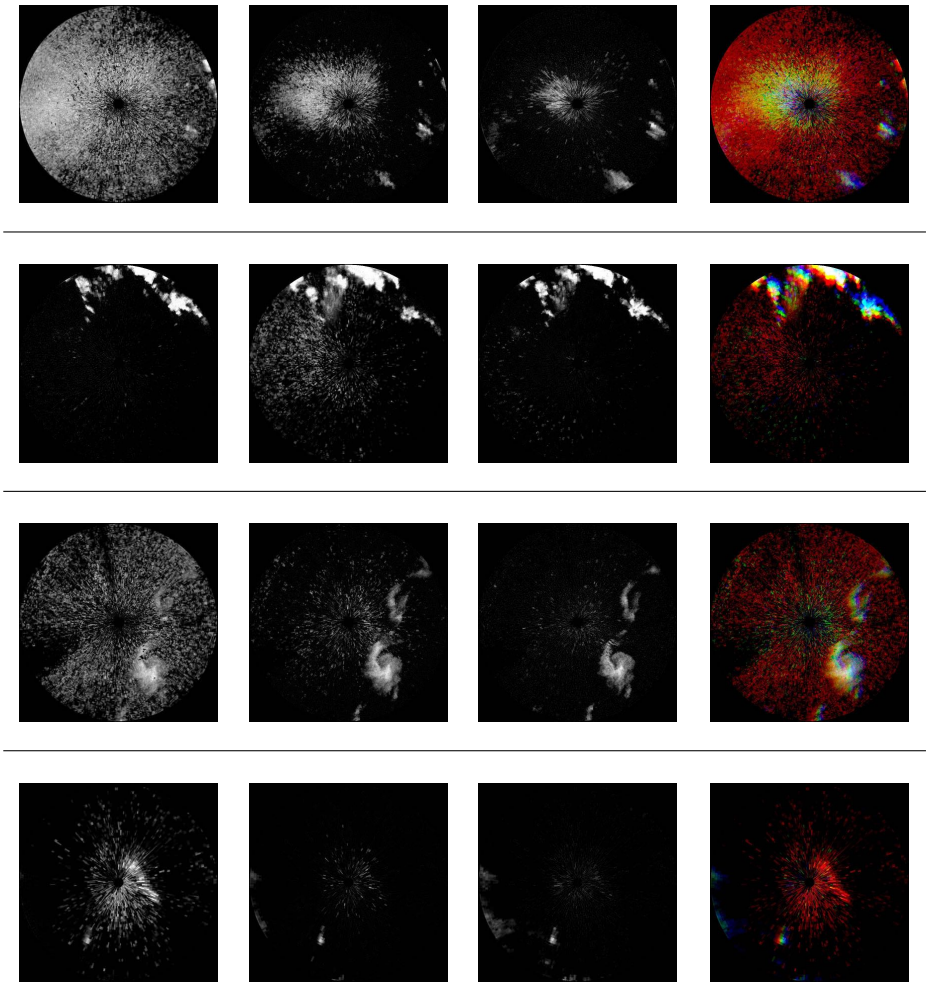

Figure 1. **Rendered scans of reflectivity.** Each row of images represents a different radar scan, using reflectivity data z. The first three columns (a,b,c) correspond to radar sweeps at successively higher elevation angles ρ. The last column (d) shows images with the three elevation sweeps concatenated as the three colour channels of an RGB image – this is the input image to the recognition models. The red channel (a) corresponds to the lowest sweep, green (b) for the next highest and blue (c) for the highest sweep. Bird migration usually occurs at lower elevations and thus is primarily visible in the red (a) and sometimes in the green channels (b). Precipitation usually occurs at all three elevations and thus in all three channels of the RGB image (d), where it is clearly visible as white blobs.

Extracting Biological Signal from Weather Radar

Several research labs in the US are working toward developing more automated systems to provide information about migrating birds using data from a network of weather station radars in the United States [1, 2, 23, 9]. The large volume of data (over 100 million archived radar scans) precludes manual interpretation at scale. Automatically eliminating the presence of precipitation from these scans is a key step towards further analysis of bird migration patterns.

Farnsworth et al. [9] developed an online annotation tool that enables users to efficiently browse through radar images and annotate them with appropriate labels. Two domain experts then used this tool to label approximately 40,000 scans from 13 radar stations as containing non-biological targets such as precipitation. We obtained labeled data from the authors of [9] to train supervised models that can automatically detect the presence of precipitation.

Problem Statement and Contribution
Classifiers of various degrees of sophistication can be used directly on the readings of various data channels recorded in the radar scans – reflectivity, radial velocity, spectrum width, etc. A naive approach would be to treat the data as a vector, without taking advantage of the visual patterns inherent in this data, which humans use to easily distinguish between rain and other effects based on visual inspection alone. We utilize deep convolutional neural networks (CNNs) to exploit the visual nature of this data and learn the feature representation from the data itself. A deep CNN, pre-trained for object recognition on a dataset of a million natural images, ImageNet [7], is fine-tuned on labelled renderings of radar scans to classify them as containing rain or not. We compare this to baseline results using Fisher Vectors [20] and find that fine-tuning deep CNNs on this task gives a significant boost in performance.

2. Radar Scan Data

In this section we give a brief overview of how we render images from the radar scan data. As mentioned in the introductory section, the dataset that we use is a subset of the data acquired from [9]. The reader is referred to [8, 9] for more details.

A *volume scan* consists of a sequence of radar *sweeps* – the radar antenna rotates 360 degrees about the vertical axis at a fixed elevation angle. The discrete portion of the atmosphere sensed at a specific antenna position and range is called a *pulse volume*. The result of a sweep is a set of such pulse volumes, indexed by the triple (r, ϕ, ρ) where r is distance from the radar station, ϕ is the azimuth – the clockwise angle in the horizontal plane between the antenna direction and due north as the antenna rotates, and ρ is the fixed elevation angle of the antenna in the vertical plane when conducting the sweep. A radar samples data by conducting a sequence of volume scans, each of which take 6 to 10 minutes, at a radio frequency of 2.7-3.0 GHz. For a fixed elevation, the data from a radar sweep is aligned to a polar grid (r, ϕ), with a resolution of 250 m and 0.5 degrees, respectively.

The upward angle of a beam and the earth's curvature cause the radar beam to sample points at higher altitudes as it moves further out, as explained in Figure 2. We can see in the top-view plot that after a distance of about 1.5 km from the particular radar station, the beam becomes too high to sense any birds, which usually fly at lower altitudes.

Information is contained in the various data "products" or channels collected from a sweep, the most common of which is *reflectivity*, $z(r, \phi, \rho)$. This is a measure of the total power returned to the radar from targets within each pulse volume. It can indicate the amount of rain or birds present in the atmosphere. Renderings of the scans as images for different elevations are shown in Figure 1. The images of the radar scans are generated by concatenating 3 successive elevation sweeps of reflectivity data into the three channels of an RGB image. The elevation angles are 0.5, 1.5 and 2.5 degrees. The rendered images use reflectivity values sampled at a radius of 37.5 km around the radar station. Along with the fact that presence of birds become rare at higher altitudes (i.e. higher elevation angles of the radar beam), we can observe in Figure 1 that the visual patterns of rain and birds are also distinctive enough for humans to usually be able differentiate between them without specialized training.

We obtained a total of 39,586 rendered scans from 13 radar stations that were labeled to contain rain or not by two domain experts. They would be able to observe the full radar scan data, but would label a scan as containing rain only if rain appeared within a 37.5 km radius of the station. The annotators were experts in radar ornithology, and report having achieved a scan reviewing rate of 2-3 seconds per scan in [9], using a custom-designed web interface [15]. In case of a disagreement in labelling between the two annotators, a conservative approach was adopted and the particular scan was regarded to have contained rain. The agreement rate between the two annotators was 97.27%.

These radar scans are a subset of the data recorded by the 13 WSR-88D stations during the Fall migration months during 2010 and 2011. One scan per hour from a station is taken, beginning from the local sunset time to the local sunrise time. Radar patterns from biological targets may also include diurnal and nocturnal foraging patterns in addition to migrations [14]. Given the times and location of the radar data, which were chosen to minimize the chances of non-migration activity, most of the biological activity is likely to be caused by migration.

3. Image Classification using Deep CNNs

Convolutional neural networks (CNNs) are composed of a hierarchy of units containing a convolutional, pooling (e.g. max or sum) and non-linear layer (e.g., ReLU $\max(0, x)$). In recent years deep CNNs, typically consisting of the order of 10 or so such layers and trained on massive labelled datasets, such as ImageNet [7], have yielded generic features that are applicable in a number of recognition tasks ranging from image classification [17], object detection [12], semantic segmentation [13] to texture recognition [5].

The ability to use a pre-trained CNN as a feature extractor for object recognition is advantageous in our problem, since we do not have sufficient radar data to train a deep network from scratch. However, training linear SVMs on the pre-trained CNN features is possible. We can also fine-tune the pre-trained network (train with a very low learning rate) on the target dataset, which is a straightforward way to adapt an ImageNet-trained CNN, with a source domain of

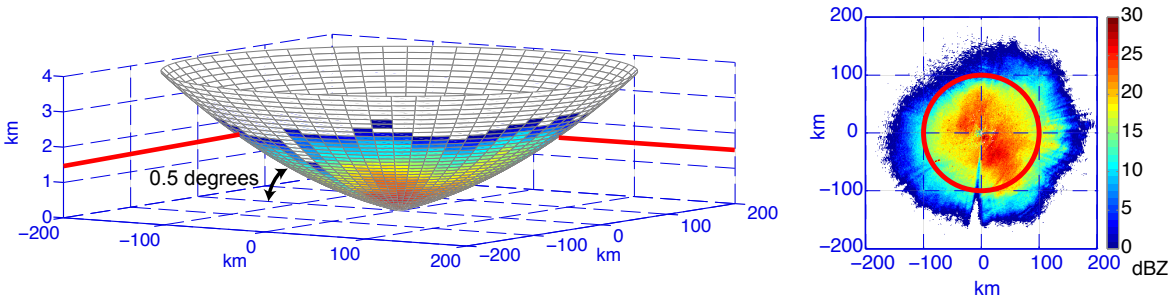

Figure 2. **3-D radar scan volume [8].** *Left:* The figure shows how the beam sent out at a fixed elevation angle of 0.5 degrees rises with increasing distance from the station. *Right:* Figure showing a top down view, with the red circle indicating a radius of 100 km around the radar station. For our experiments, we render images from reflectivity data at a radius of 37.5 km around the station.

natural images, to our target domain of radar imagery.

Since it is not immediately clear whether the presence of rain in radar scans can be regarded as a distinct object or be more accurately described as a sort of "texture", we investigate the performance of recent texture recognition models [5] on this task.

4. Experiments

We report results using a variety of methods on the radar scan dataset – regular CNN, Fisher vector formed using CNN features and a baseline method of Fisher vector using SIFT features. We report results after fine-tuning ImageNet pre-trained models on the radar imagery, using deeper CNN models and the effect of elevation angle on accuracy. We conclude the section with an analysis of the error cases of our best performing model.

4.1. Methods

Our implementation is in MATLAB and the MatConvNet [25] library was used to train the deep networks. The library also provided the ImageNet pre-trained networks for download. The SIFT and Fisher vector implementations were from the VLFeat [24] library.

Dataset

The radar dataset consists of 39,586 rendered scans. Of these images, 26,392 are used for training and 13,194 for testing, in a two-thirds and one-third split. Out of the total 39,586 images, 11,397 were labelled to contain rain, which we consider to be the positive class. The test set has 13,194 radar scans out of which 3,799 contain rain. We use the first three elevation sweeps of reflectivity, z, to generate the images as 3-channel RGB.

Network architecture

Two networks of varying depth are used – VGG-M [4] and VGG "very-deep-16" [22]. Both take in 224×224 3-channel images as inputs and provide 4096-dimensional features in their penultimate fully-connected layer, also referred to as the 'fc7' layer in the network architecture. We use the 'fc7' activations after the ReLU non-linearity. During fine-tuning, a softmax classification loss is used to train the networks.

Network fine-tuning

The learning rate is set to be 0.0001 and is divided by 10 every 10 epochs. The learning rate of the last fully-connected layer (the classification layer) was set to be 10 times that of the global learning rate. Each network is fine-tuned for 30 epochs. The learning rates were determined on a validation set formed by keeping aside a third of the training data.

We now summarize the methods and their acronyms that we have evaluated on this dataset:

- **FV-SIFT:** Fisher vector encoding of SIFT features. The SIFT descriptors are densely computed over the input image with a stride of 4 pixels and a window of size 32×32. The input image of size 224×224 is up-scaled to 448×448. The 128 dimensional SIFT descriptors are then PCA-reduced to 80 dimensions. The number of components in the Gaussian Mixture Model (GMM) for the Fisher vector encoding is set to be $K = 64, 128$. This results in a $2 * K * D$ dimensional encoding, where K is the number of GMM components and D is the dimension of the PCA-SIFT descriptor. Therefore FV-SIFT-64 is 10,240 dimensional while FV-SIFT-128 is 20,480 dimensional. If not specified, when we use "FV-SIFT" in this paper, we are referring to the model with $K = 64$.

- **CNN:** The VGG-M model [4], pre-trained on ImageNet. Feature dimension is 4096.
- **CNN-VD:** The "very-deep-16" model [22], pre-trained on ImageNet. Feature dimension is 4096.
- **FV-CNN:** Fisher vector encoding formed out of CNN (FV-CNN) and CNN-VD (FV-CNN-VD) features [5]. The features from the last fully-convolutional layer of a network provide a set of dense descriptors over a downsampled version of the image. E.g., for the VGG-M network this would be the output of the 'relu5' non-linearity after the 'conv5' layer, which is $13 \times 13 \times 512$ for an input image of size 224×224. Using up-scaled images of size 448×448, we obtain $27 \times 27 \times 512$ dimensional activations at the 'conv5+relu5' layer. These can be regarded as 512-dimensional descriptors over a coarse spatial grid of 27×27. The Fisher vector model is formed by pooling these descriptors. The feature size is 65,536 ($2 * K * D$, with K=64 and D=512).

The features extracted using each of the methods is then used to train a linear SVM classifier, with "rain" being considered as the positive class. For the CNN models, L2-normalization is applied on the features. For the Fisher vector based models, the features are square-root and L2-normalized for added invariance, following standard practice [20, 3]. The SVM hyperparameter, C_{svm} is set to be 1.

Method	w/o ft	w/ ft
FV-SIFT-64	86.4	-
FV-SIFT-128	87.5	-
CNN	86.9	91.5
CNN-VD	87.0	94.4
FV-CNN	89.2	91.0
FV-CNN-VD	88.8	90.6

Table 1. **Comparison of results.** The classification accuracy of the various methods is shown here, before and after fine-tuning the CNNs on radar imagery (the *w/o ft* and *w/ ft* columns, respectively). Fine-tuning significantly improves the performance of the CNN-based methods, showing the advantage of using learned descriptors. The fine-tuned CNN-VD model gives the highest performance.

4.2. Results

The results are summarized in Table 1. Precision-recall curves are shown in Figure 3. We discuss the numbers in details in the following sub-sections.

Baseline model

It is observed that SIFT with Fisher vector performs quite well, at 86.4% when using 64 Gaussian components ($K = 64$). Doubling the number of Gaussians, K, from 64 to 128, in the Fisher vector model improves its accuracy to 87.5%, an increase of about 1%.

Effect of fine-tuning and depth

The deep models, after fine-tuning, significantly outperform the SIFT-based baselines. The performance of the CNN (VGG-M architecture) increases from 86.9% to 91.5%. For the "very-deep 16" architecture CNN-VD, performance increases from 87.0% to **94.4%**, which is the best performance among the methods we compare in our current experiments.

As expected, a deeper network like CNN-VD gives better performance than a shallower architecture such as the VGG-M. This is more apparent after fine-tuning – 91.5% vs. 94.4% as opposed to 86.9% vs. 87.0% when using ImageNet pre-trained networks.

Deep Fisher Vector model

Using pre-trained CNN features (*w/o fine-tuning*), the FV-CNN model gives better performance than the corresponding CNN model (89.2% versus 86.9%). The GMM of the FV-CNN is learned on the descriptors from the radar images, and this unsupervised domain adaptation can provide more informative features to the classifier. However, *after fine-tuning* the CNN on radar images, the FV model using CNN features gives slightly lower performance (91.0%) than the regular CNN (91.5%).

A possible reason for this could be the fact that the Fisher vector model, unlike a CNN, pools features in an orderless manner, which discards all explicit spatial information. However, in radar images, the distance from the radar station has a strong correlation with the altitude of the objects being sampled by the beam (Figure 2). This information may be potentially helpful to distinguish rain from biological phenomena due to the differences in altitude where they are most likely to occur. This distinction can be learned from the data by fine-tuning the CNN. The loss of spatial information in the Fisher vector model may thus be responsible for the slight 0.5% drop in performance compared to the fine-tuned CNN model. Appending positional information as polar coordinates (r, ϕ) to the local descriptors in the Fisher vector model may help in the case of spatial structure in images, as shown in face verification experiments using the FV+SIFT model [19].

Effect of elevation angle

We perform a simple experiment to see if a hybrid of learned descriptors and some "hand-engineering" of features can result in improved performance on our problem. The motivation behind using CNN descriptors is to let the model learn

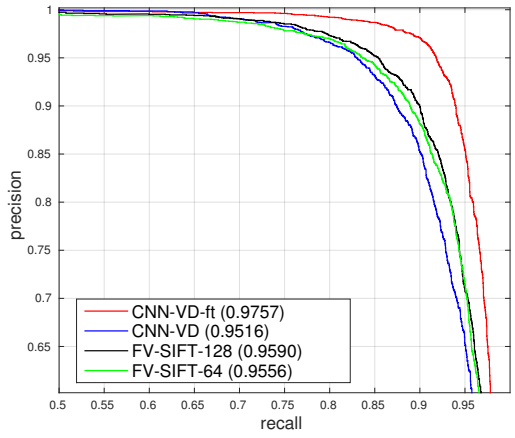

Figure 3. **Precision-Recall curve** (zoomed in)**:** Comparing the performance of the best deep-learning model (CNN-VD) and the hand-crafted descriptor method (FV-SIFT) at classifying rain. We can observe the effect of fine-tuning on the CNN model (CNN-VD-ft) and the effect of doubling the number of Gaussians in the Fisher vector model (FV-SIFT-64 and FV-SIFT-128). The numbers in parenthesis next to the method names are *average precision (AP)* values.

the most suitable feature representation from the data itself, without resorting to the earlier era in computer vision of feeding hand-crafted descriptors as features to classifiers. However, sometimes explicitly using domain knowledge can help.

In our specific case, it is obvious from looking at the rendered scans in Figure 1 that the presence of rain is most prominent in the 3^{rd} elevation sweep. At that altitude, the presence of birds is scarce and all activity is usually due to weather phenomena. The results are shown in Table 2 below.

Method	sweep	w/o ft	w/ ft
CNN	1,2,3	86.9	91.5
	3	86.6	91.0
FV-SIFT	1,2,3	86.4	-
	3	88.1	-

Table 2. **Effect of a single elevation.** Results on using only the highest elevation angle (*sweep* '3') to render the images, compared against using 3 elevation angles (*sweep* '1,2,3') as done in the previous set of experiments. The results before and after fine-tuning are shown under the columns *w/o ft* and *w/ ft* respectively.

With the CNN model, we see negligible differences in performance when using a single elevation sweep versus using the information from all three sweeps. The difference in using pre-trained CNN is 0.3% and after fine-tuning the networks the difference is 0.5%. We hypothesize that this could be due to useful information being contained in the lower elevation sweeps as well, which supports the idea of using all the channels of data and then letting the model figure out from the data what is most discriminative.

However, with a shallow model like FV-SIFT, the intuition about the 3^{rd} elevation sweep being most discriminative holds true – we get 88.1% accuracy when using only the highest elevation sweep as opposed to 86.4% when using all the three elevation sweeps to form the radar scan image. The FV-SIFT model achieves higher performance in 'sweep 3' setting because any signal at that elevation can mostly be attributed to precipitation, without having to find complex interactions between features when using the additional information from multiple elevation sweeps.

We further note a difference in the ways a CNN and a SIFT-based model would handle multiple channels. By construction, the CNN models we use take in 3-channel images as input. So even without fine-tuning, the CNN model has some distinction between different channels in the input image. We use the simplest setting of the FV-SIFT model, where the 3-channel image is converted to grayscale before the dense SIFT descriptors are extracted. The information of the 3 elevation sweeps may get "averaged" together and become less discriminatory, as opposed to simply using a single elevation sweep as a single channel image to the FV-SIFT model. Forming different Fisher vector models for each elevation sweep separately and then using a linear SVM classifier may boost performance in this scenario, however this would also increase the feature dimension of the Fisher vector by 3 (30,720).

Runtime Comparison

The relative evaluation runtime on our system gives some idea of comparative expense of using learned descriptors versus hand-crafted descriptors, shown in Table 3. Our system is a Dell workstation with an NVIDIA Tesla K40 GPU and an Intel Xeon CPU with 14 cores. We use an older version of MatConvNet (beta-9) .

Method	CPU	GPU
CNN	36	124
CNN-VD	13	43
FV-SIFT	10	-

Table 3. **Evaluation runtime.** The average evaluation time as *images/second* is shown here for deep and shallow methods.

Thus we can see that even on a CPU, a moderately-sized CNN architecture like VGG-M is faster than traditional approaches using hand-engineered features like SIFT, while giving superior performance. Using a deeper architecture like CNN-VD results in higher accuracy at speeds comparable to the FV-SIFT model.

4.3. Error Analysis

The top false positive and false negative images using the CNN-VD model are shown in Figure 4. A false positive is when the classifier predicts "rain" for an image with ground truth "not rain". Similarly, a false negative is when the classifier predicts "not rain" when ground truth is "rain".

In the case of *false positives*, all the images have some faint blue-green or whitish blobs, usually around the edges. High magnitudes in G and B colour channels signify objects at higher altitudes. If an object is present all lower and higher altitudes it would be rendered as a white patch in the image. This is usually the case with rain, as opposed to birds which are restricted to lower altitudes. Thus, even though these images have been labeled as without rain, the classifier mistakes them to be rain as they have portions very similar in appearance to rain.

There is also a high possibility of there being labeling errors among these images. If the lowest sweep looked clear, the labelers usually would not examine the second and third sweeps, so they could easily miss rain that only appeared at higher elevations or if there just wasn't anything in the lowest elevation to suggest that rain might be occurring.

For *false negatives*, the top two images show signs of mixed rain along with migration, so it is possible that the classifier got confused. In particular, the first image shows majority of activity in the R channel. The white blobs, possibly indicating rain at all three elevation angles, is present in the center of the scan. It is possible that the CNN has learned that rain, being at high altitudes, would appear as white blobs with fairly sharp boundaries (see examples of such features in Figure 1) at the edges of images where the beam samples points at higher altitudes.

The bottom three images are the most straightforward to explain – they have very little activity outside the R channel (i.e. the lowest altitude sweep). The green blobs, indicating activity in the 2^{nd} sweep elevation, are mostly faint or present at the edges. This could lead to them being missed in some cases by the CNN. It is also to be noted, the annotators had the additional context of looking at the full area of the radar scan. Heavy presence of rain outside the 37.5 km radius could clue them onto the faint presence of rain within that radius, which is missed by the classifier as it is looking at images only within the 37.5 km radius.

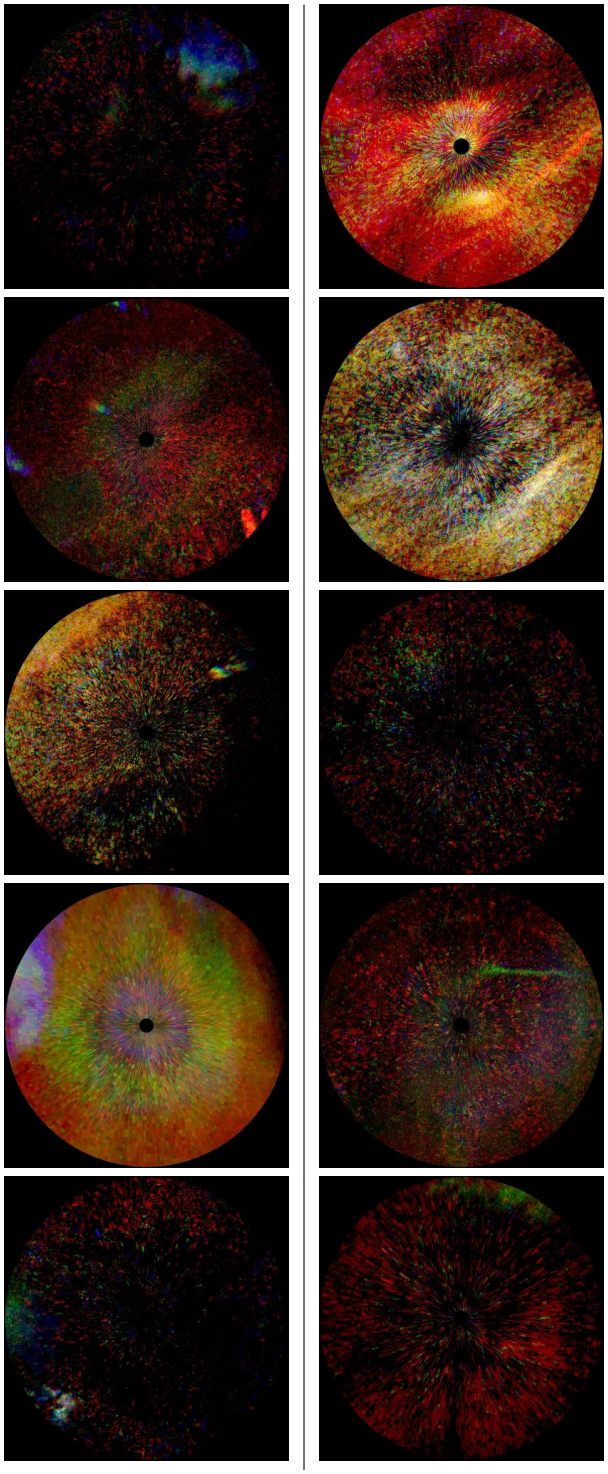

Figure 4. **Visualizing error cases.** *Left:* false positives (classifier="rain", groundtruth="not rain"). *Right:* false negatives (classifier="not rain", groundtruth="rain").

5. Conclusion

Human annotators perform reasonably well at the task of distinguishing weather patterns from biological targets. The two expert annotators had a labeling agreement rate of about 98% on nearly 40,000 scans.

Even without domain knowledge it is possible to visually differentiate between rain and other phenomena in many cases, as can be seen from the sample radar scans in Figure 1. Thus, it seems justified to expect computer vision based approaches to perform quite well on this task. We

set a good baseline using a simple SIFT+FV model on this data. The CNN-based models surpass this baseline after fine-tuning, while performing quite fast even without using a GPU at evaluation-time.

Acknowledgement

We would like to thank NVIDIA for their generous donation of the Tesla K40 GPU used in these experiments. ARC would also like to thank Kevin Winner and Garrett Bernstein for their help with the radar data.

References

[1] J. Buler and R. Diehl. Quantifying Bird Density During Migratory Stopover Using Weather Surveillance Radar. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2741–2751, 2009.

[2] J. J. Buler and D. K. Dawson. Radar analysis of fall bird migration stopover sites in the northeastern US. *The Condor*, 116(3):357–370, 2014.

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC*, 2014.

[5] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and description. In *Proc. CVPR*, 2015.

[6] T. D. Crum and R. L. Alberty. The WSR-88D and the WSR-88D operational support facility. *Bulletin of the American Meteorological Society*, 74(9):1669–1687, 1993.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

[8] A. Farnsworth, D. Sheldon, J. Geevarghese, J. Irvine, B. Van Doren, K. Webb, T. G. Dietterich, and S. Kelling. Reconstructing velocities of migrating birds from weather radar–a case study in computational sustainability. 2014.

[9] A. Farnsworth, B. M. Van Doren, W. M. Hochachka, D. Sheldon, K. Winner, J. Irvine, J. Geevarghese, and S. Kelling. A characterization of autumn nocturnal migration detected by weather surveillance radars in the northeastern US. *Ecological Applications*, pages n/a–n/a, 2016.

[10] D. Fink, T. Damoulas, and J. Dave. Adaptive spatio-temporal exploratory models: Hemisphere-wide species distributions from massively crowdsourced ebird data. aaai 2013, washington, usa. 2013.

[11] D. Fink, W. M. Hochachka, B. Zuckerberg, D. W. Winkler, B. Shaby, M. A. Munson, G. Hooker, M. Riedewald, D. Sheldon, and S. Kelling. Spatiotemporal exploratory models for broad-scale survey data. *Ecological Applications*, 20(8):2131–2147, 2010.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. ECCV*, 2014.

[14] J. W. Horn and T. H. Kunz. Analyzing nexrad doppler radar images to assess nightly dispersal patterns and population trends in brazilian free-tailed bats (tadarida brasiliensis). *Integrative and Comparative Biology*, 48(1):24–39, 2008.

[15] J. Irving. Scanlabeler. http://web.engr.oregonstate.edu/~irvine/scanlabeler. Accessed: 2016-04-18.

[16] S. Kelling, J. Gerbracht, D. Fink, C. Lagoze, W.-K. Wong, J. Yu, T. Damoulas, and C. Gomes. A human/computer learning network to improve biodiversity conservation and research. *AI magazine*, 34(1):10, 2012.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[18] D. Lack and G. Varley. Detection of birds by radar. *Nature*, 156:446, 1945.

[19] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proc. CVPR*, 2014.

[20] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[21] D. R. Sheldon, A. Farnsworth, J. Irvine, B. Van Doren, K. F. Webb, T. G. Dietterich, and S. Kelling. Approximate bayesian inference for reconstructing velocities of migrating birds from weather radar. In *AAAI*, 2013.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[23] P. M. Stepanian and K. G. Horton. Extracting migrant flight orientation profiles using polarimetric radar. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(12):6518–6528, 2015.

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[25] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for MATLAB. *CoRR*, abs/1412.4564, 2014.

[26] R. D. D. Zrnic and R. J. Doviak. Doppler radar and weather observations, 1992.