BIRD DETECTION NEAR WIND TURBINES FROM HIGH-RESOLUTION VIDEO USING LSTM NETWORKS

Tuan Tu. Trinh[1], Ryota. Yoshihashi[1], Rei. Kawakami[2], Makoto. Iida[3], Takeshi. Naemura[2]

[1]Graduate School of Information Science and Technology, The University of Tokyo

[2]Interfaculty Initiative in Information Studies, The University of Tokyo

[3]Research Center for Advanced Science and Technology, The University of Tokyo

Wind turbines have become one of significant risks causing mortality of wild birds. In order to evaluate this ecological impact, a system that can automatically detect birds draws increased attention from the industry. We propose a bird detection method combining Convolutional Neural Networks (CNNs) and Long Short-term Memory Networks (LSTMs) to leverage rich features extracted from CNNs and the ability of memorizing continuous appearance change of birds in subsequent time frames. Experiments using high-resolution videos captured around wind turbines show that LSTMs combined with CNNs outperform solely using CNNs for recognizing birds, as long as birds are correctly tracked.

Keywords: bird detection, image processing, deep learning, convolutional neural networks, long short-term memory

INTRODUCTION

Wind energy—a low cost and renewable, non-polluting resource—has been being harvested for several decades. However, it has only grown rapidly in the last few years as human became gradually concerned about environmental problems such as air pollution and global warming. Although wind energy has relatively little impact on the environment compared to conventional power plants in terms of air and water pollution, there are many reports of bird mortality caused by collision with turbine blades and loss of nesting and feeding grounds, and these deaths will increase as turbines multiply in near future. Although the scale of ecological impact may not be substantial, wind turbines can affect bird populations, especially of endangered species [1] [2] [3]. To assess the bird ecology and estimate the impact on it caused by wind turbines, an automated bird detecting system draws industrial attention for identifying birds' species, numbers and their flying routes. Such a system may also have a practical use for mitigating the damage by decelerating the blades or making specific sounds to drive birds away.

Image-based detection is one of the promising approaches for bird detection due to the ability of utilizing visual information, which helps recognizing birds among other moving objects and detecting birds in short-range, especially areas that cannot be detected by radar, such as ground surfaces or near wind turbines. Recently, Convolutional Neural Networks (CNNs) have shown remarkable results in recognition tasks in still image data, owing to its ability to extract more hierarchical and data-driven features compared to traditional manually-designed features. However, when CNNs are applied to the images in practice, blades or leaves are still challenging to be distinguished from birds due to their visual similarities. Moreover, birds captured by a static camera are sometimes in very low resolution, and they are even hardly recognized by human eyes in bad weather.

These problems can often be easily solved when the motion information is available. Naturally, several studies have followed the idea and shown that detection performance can be improved by combining motion features with static ones [4] [5] [6]. However, how to utilize motion appropriately to achieve the best performance in detection is still in debate. Most of previous studies incorporate motion information only through handcrafted features where their main ideas are removing background motion and keeping contour of moving objects. In those methods, features for several frames are stacked and constructs a large vector as an input that is fed into classifiers; thus, the input vector size has to be fixed and synchronized. On the other hand, Long Short-term Memory Networks (LSTMs), one kind of recurrent neural networks (RNNs), explicitly handle a sequence of input with ability to learn hidden information from it. It has a memory cell that remembers hidden but important information in the sequence, while it also has a framework that controls how much to learn and how much to forget each time, so that it can handle a long sequence of inputs. They have succeeded in aggregating deep motion features extracted from continuous frames and worked tremendously well on video classification and video transcription tasks.

In this paper, we propose a detection method combining CNNs and LSTMs, which can leverage rich features extracted from CNNs and can learn long-term dependencies processed from continuous information. Applying LSTMs for detection is different from video classification or video transcription, since the input sequence for LSTMs is undefined. We have to search the target object that may change its appearance in subsequent

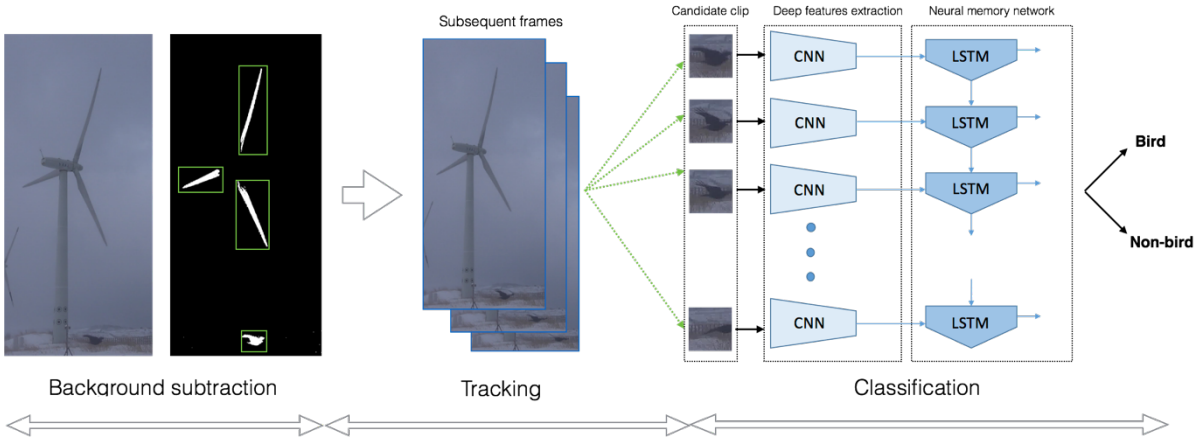


Figure 1 Overview of proposed system

frames and determine the input sequence with adequate length at reasonable position. To handle this, we introduce background subtraction and a general tracker [7]; thus, the method can first find candidate moving objects and then track them for several frames to construct an input sequence for CNNs and LSTMs. CNNs extract deep features from the sequence, and LSTMs classify whether the sequence is of birds or not. We also constructed our own video data taken near the wind farms in Hokkaido, Japan, to evaluate our proposed method. We experimentally verified the improvement of our method's performance compared to CNNs used solely frame by frame.

Long Short-term Memory (LSTMs) Since the breakthrough of CNNs in ImageNet Large-scale Visual Recognition Challenge 2012 [8], handcrafted features have been gradually replaced by hierarchical deep features extracted by CNNs. In recognition and detection tasks using video data, the motion should be an important hint beside the static features extracted from each frame. There are many approaches to take advantage of motion information in videos; one approach is extending CNNs to 3D-CNNs so that it can handle additional temporal dimension, and another introduces extra stream consists of CNNs to process optical flow images for utilizing the motion [9] [10]. However, one of the recent remarkable approaches is to combine CNNs and LSTMs, which is found to be useful in human action recognition [11]. LSTMs [12] is a special kind of recurrent neural networks (RNNs). RNNs are invented so that neural networks are allowed to be influenced by the past inputs. LSTMs are improved version of RNNs so that they have an ability to choose learning from recent previous information or the information in further past. The ability is important since information in the past may be noisy and falsify the current result, but there are also cases where we need a lot of context information. Conventional RNNs cannot learn long dependencies [13], while LSTMs overcome the weakness. LSTMs are recently widely applied for problems such as speech recognition, auto translation, and video captioning, and have provided successful results. Many visual tasks require the neural networks to remember previous information to understand the present frame, and LSTMs fulfill the requirement. The details of LSTMs cell is illustrated in Figure 2. An LSTM cell contains four main elements: input gate, output gate, forget gate and a cell with recurrent connection. The input gate controls which part of incoming signal should be blocked, and which part should be allowed to alter the state of the cell. Meanwhile, the output gate controls which part of output signal can have effect on next neurons. The forget gate modulates the recurrent connection to decides which information the cell should remember and forget. In addition, the weight of recurrent connection is set to 1 in order to prevent gradients from vanishing or exploding.

METHOD

Our proposed detection method consists of three steps combining background subtraction, tracking, and object classification, as described in Figure 1. First, it extracts candidate areas that might contain birds using background subtraction. Second, it tracks these candidate objects by a robust tracker in order to acquire temporal as well as static information. Finally, we feed the extracted image sequences into CNN-LSTM networks to filter birds from other background objects such as turbine blades, trees and cloud.

Background subtraction Background subtraction is the most fundamental method for extracting moving objects from a fixed background for further processing and worked well in our scenes since the background does not change rapidly. Gaussian Mixture Model is used to represent background and the model was trained using the last 100 frames in this paper. Each frame will be compared to this background model and individual pixels are categorized either into background or foreground.

Discriminative Scale Space Tracker (DSST) In order to define the input sequence for the subsequent process and obtain the motion information, we used the Discriminative Scale Space Tracker (DSST) [7]. As our dataset contains scenes that birds frequently appear and disappear and they have large scale variations, we choose DSST, which can accurately estimate both scale and translation of the moving object. DSST is a filter-based tracker, which learns separate discriminative correlation filters for translation and scale estimation based on a scale pyramid representation. It yielded the best performance in Visual Object Tracking Challenge in 2014 among 38 participating tracking methods. Since our dataset contains extremely small birds and they cannot be tracked properly in long term, we tracked every objects for 5 frames in our experiment.

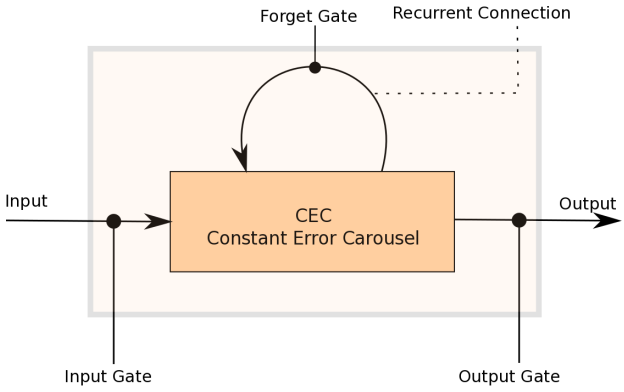


Figure 2 LSTMs cell

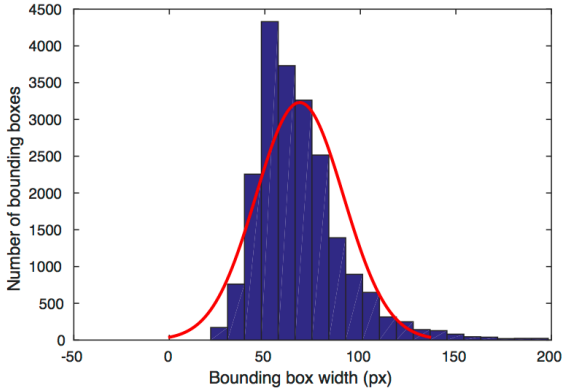


Figure 3 Distribution of bird's size in dataset

CNN-LSTM Network We follow Donahue et al.'s LRCN networks [11] designed for human action recognition which have shown significant improvement in classifying 100 classes of actions compared to the baseline that solely using CNNs. The LSTMs was redesigned to output the probabilities of 2 classes: bird and non-bird. As shown in Figure 1, the candidate clip will be passed through CNNs and transformed to a sequence of fixed-length vectors, these outputs are then passed into LSTM networks to generate the final results. CNNs will be pre-trained using first frames and the parameters of CNNs will be kept unchanged when we fine-tune the entire CNN-LSTM networks using clip data.

EXPERIMENT

We implemented a bird detection method as described above and trained CNN-LSTM networks as a classifier to discriminate birds from other objects. We also implemented single CNN networks as a baseline and compared the performance to our proposed method.

Dataset We used a fixed point camera to record video data near a wind turbine for three days. The recording frame rate was set to 30fps and we recorded totally 23 hours of video. Birds are annotated using bounding boxes every 10 frames and the size of annotated bounding boxes distributed from 20 pixels to 200 pixels. Several examples of birds labeled in our data and size distribution of birds are shown in Figure 3. Only 17,222 frames that contain at least one bird were used in our experiments. We used 15,000 frames for training and 2,200 frames for testing.

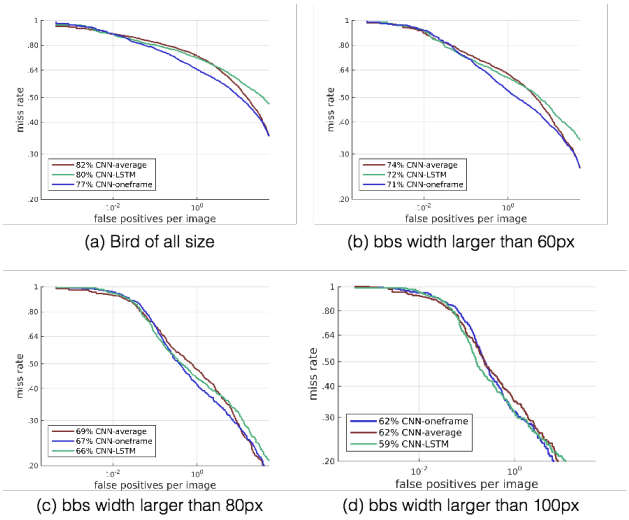
Training Procedure We extracted 18,385 birds from training set and used these as positive samples to train the models. The negative samples were collected using background subtraction method and we selected totally 73,900 areas that have no overlap to the ground-truth of birds. Each sample was tracked for 5 frames to generate image sequences which will be used to train the CNN-LSTM model. The CNN contains 5 convolutional layers and 2 fully connected layers. In order to classify images of different sizes by one classifier, we resized all the input to the same size of 256 × 256. We also designed the CNNs to take 256 x 256 images as input and output is a 4096 dimensional vector at the second fully connected layer. In the single CNN model, these vectors will be passed to soft-max layer to generate final score. In CNN-LSTM model, these vector representations will be subsequently fed to LSTMs with 256 hidden units to verify whether the inputted image sequence is bird or not. Both CNN and CNN-LSTM model was trained by the backpropagation algorithm until the value of loss function converges. In this experiment, we trained CNN and CNN-LSTM respectively for 2000 iterations and 20000 iterations.

Evaluation We used 2,200 frames to evaluate the performance of our entire system. Each detector performance was visualized by the *Miss rate - false positives per image* graph. The performance was also quantified by *log-average miss rate*, which was computed by averaging miss rate at nine *false positives per image* rates evenly sampled in log-space from 10-2to 100. We supposed a detected bounding box *dt* form a match with a ground-truth bounding box *gt* if the overlap area defined as follows is greater than the threshold *0.5*, and each detected bounding box can be matched at most once.

$$\text{Overlap} = \frac{\text{area}(dt \cap gt)}{\text{area}(dt)} > 0.5 \quad (1)$$

Basically, the non-maximal suppression for merging nearby detections should be performed at this step, but we skipped it since we already did the blob analysis when doing background subtraction.

Result The result of detection is shown in Figure 4. For comparison, we evaluated the system with three classifiers, single CNN using first frame only (CNN-single), single CNN whose output is averaged on the tracked 5 frames (CNN-average), and the proposed CNN-LSTM classifier. The first two cases in Figure 4 show evaluation on subsets that consist of birds of all size or birds larger than 60 pixels in size. The graphs and average miss rate indicated that CNN-LSTM classifier was better than CNN-average but still had worse performance compared to CNN-single. Theoretically, CNN-average should have at least the same performance as CNN-single with more information from the subsequent frames, if the tracker worked properly. This possible reason is that the tracker failed to track birds with tiny size. After failure of tracking, the subsequent frames contain birds no longer, which affected the entire score when averaged. CNN-LSTM also used subsequent frames extracted from the tracker, therefore the improper tracking may also explain that CNN-LSTM had worse score than CNN-single. In contrast, CNN-LSTM outperformed the other methods on the subset of birds with size bigger than 80 pixels and 100 pixels. We also noticed that with birds bigger than 100 pixels, CNN-



(a) Bird of all size (b) bbs width larger than 60px

(c) bbs width larger than 80px (d) bbs width larger than 100px

Figure 4 Detection results

average had the same performance with CNN-single. Since the size of birds was bigger, the tracker worked properly and gave CNN-average as well as CNN-LSTM additional trustable information from the subsequent frames. While CNN-average achieved only the same result as CNN-single even with additional information, CNN-LSTM outperformed these two methods. This experimental result matched our hypothesis that although being effective in learning static features, since CNNs do not consider motion, they may have difficulty classifying low resolution image or hard-negative samples. Contrarily, CNN-LSTM can learn both static features as well as the motion of birds such as wing flapping or gliding to distinguish them from other objects.

CONCLUSION

In this paper, we introduced a practical image-based method to detect birds from high-resolution video, utilizing both static features and motion features from the frames. We also constructed our own dataset and conducted experiments of bird detection to compare the performance of proposed method to the conventional methods. Our system was shown to be effective when the tracker works properly, although it still has disadvantages in detecting tiny birds. We can still boost the performance by using more proper tracker or fine-tuning the parameters of CNN-LSTMs. Our system could be one of potential solutions for bird strikes problem, minimizing the impact of wind energy to bird ecology.

ACKNOWLEDGEMENT

A part of this work is entrusted by the Ministry of the Environment, Japan (MOEJ), the project of which is to examine effective measures for preventing birds, especially sea-eagles, from colliding with wind turbines. This work was also supported by JSPS KAKENHI Grant Number JP16K16083 and Grant-in-Aid for JSPS Fellows JP16J04552.

REFERENCE

[1] K. Smallwood, L. Rugge and M. L. Morrison, "Influence of behavior on bird mortality in wind energy developments," *The Journal of Wildlife Management,* vol. **73**, no. 7, pp. 1082-1098, 2009.

[2] A. Drewitt and R. Langston, "Collision Effects of Wind-power Generators and Other Obstacles on Birds," *Annals of the New York Aacdemy of Sciences,* vol. **1134**, no. 1, pp. 233-266.

[3] A. Drewitt and R. Langston, "Assessing the Impacts of Wind Farms on Birds," *IBIS The International Journal of Avian Science,* vol. **148**, pp. 29-42, 2006.

[4] P. Viola, M. J. Jones and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *International Journal of Computer Vision,* vol. **63**, no. 2, pp. 153-161, 2005.

[5] D. Park, L. C. Zitnick, D. Ramanan and P. Dollar, "Exploring Weak Stabilization for Motion Feature Extraction," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*.

[6] N. Dalal, B. Triggs and C. Schmid, "Human Detection Using Oriented Histograms of Flow and Appearance," in *ECCV*, 2006.

[7] M. Danelljan, G. Hager, F. Khan and M. Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," in *BMVC*, 2014.

[8] A. Krizhevsky, I. Suskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014.

[10] S. Ji, W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. **35**, no. 1, pp. 221-231, 2013.

[11] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venupogalan, K. Saenko and T. Darrel, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," in *CVPR*, 2015.

[12] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation,* vol. **9**, no. 8, pp. 1735-1780, 1997.

[13] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks,* vol. **5**, no. 2, pp. 157-166, 1994.