Fine Grained Video Classification for Endangered Bird Species Protection

Non-Thesis MS Final Report

Chenyu Wang

1. Introduction

1.1 Background

This project is about detecting eagles in videos. Eagles are endangered species at the brim of extinction since 1980s. With the bans of harmful pesticides, the number of eagles keep increasing. However, recent studies on golden eagles' activities in the vicinity of wind turbines have shown significant number of turbine blade collisions with eagles as the major cause of eagles' mortality. [1]

This project is a part of a larger research project to build an eagle detection and deterrent system on wind turbine toward reducing eagles' mortality. [2] The critical component of this study is a computer vision system for eagle detection in videos. The key requirement are that the system should work in real time and detect eagles at a far distance from the camera (i.e. in low resolution).

There are three different bird species in my dataset - falcon, eagle and seagull. The reason for involving only these three species is based on the real world situation. Wind turbines are always installed near coast and mountain hill where falcons and seagulls will be the majority. So my model will classify the minority eagles out of other bird species during the immigration season and protecting them by using the deterrent system.

1.2 Brief Approach

Our approach represents a unified deep-learning architecture for eagle detection. Given videos, our goal is to detect eagle species at far distance from the camera, using both appearance and bird motion cues, so as to meet the recall-precision rates set by the user. Detecting eagle is a challenging task because of the following reasons. Frist, an eagle flies fast and high in the sky which means that we need a lens with wide angle such that captures their movement. However, a camera with wide angle produces a low resolution and low quality video and the detailed appearance of bird is compromised. Second, current neural network typically take as input low resolution images. This is because a higher resolution image will require larger filters and deeper networks which is turn hard to train [3]. So it is not clear whether the low resolution will cause challenge for fine-grained classification task. Last but not the least, there is not a large training database like PASCAL, MNIST or UCF101 [4] available for my research project.

In order to address these challenges, we developed the following approach:

1. A deep, recurrent neural network, called Long-Short-Term Memory (LSTM) for processing a sequence (or multiple sequences) of video frames and detecting eagle appearances in the frames. LSTMs have been demonstrated to achieve state-of-the-art results in both video and audio interpretation [5].
2. Connecting a traditional neural network CNN with LSTM to form a new neural network architecture called Long-Term Recurrent Convolutional Network (LRCN). And compare the new LRCN with traditional CNN.

In detail, LSTM, as well as LRCN, is designed to integrate both color and texture visual cues of bird species, and account for an eagle-specific wing motion pattern. As our results demonstrate LSTMs is able to robustly discriminate between eagles and other birds (and other flying objects) in video. Importantly, in some cases, high recall and low precision of detection (i.e., the large number of detections not missing true appearances of an eagle in the video, but with the low true positive rate) may be of interest when the activation of bird deterrents is not expensive. In other cases, high precision of detection at the cost of missing a few eagle appearances may be of interest. Therefore, we also tried a flexible LSTM design which adjusts to specific recall-precision rates of eagle detection per user's requirements.

Overall, the key contribution of our work is to show that a robust fine-grained object detection can be done in low resolution videos using a deep recurrent neural network. Another contribution is evaluating and benchmarking the approach on a new dataset. Because this is to the best of our knowledge, the first model and dataset for fine-grained bird detection in videos.

The rest of the thesis is organized as follows: In section 2, I will give the detailed information on the architecture of the neural networks we used in our research. In section 3, the dataset and the pruning process will be presented. And I will also discuss the challenges of processing data in detail. Finally, in section 4, the results and evaluation are discussed in detail.

2. Approach

In this project, we use Long-Term Recurrent Convolutional Network (LRCN), which is a combination of Convolutional Neural Network (CNN) and (Long-Short Term Memory) LSTM [6]. As Figure 1 shows, the input video frames are first input to CNN and then LSTM fuses CNN's outputs for every frame and predicts the class of the video.

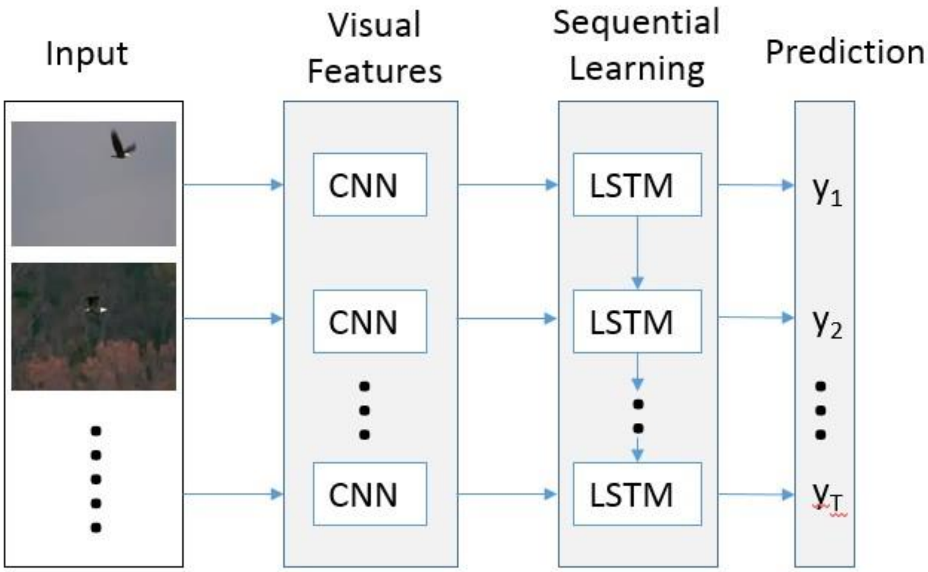


Figure 1. The Sequence Learning box of the LRCN model. (Left: the input video frames; Middle: CNN's output comes to LSTM; Right: y means label of the prediction)

2.1 Convolutional Neural Network (CNN)

CNN is the first module of our approach. It takes as input each video frame at a time. CNN is a neural network which is known very effective for image recognition and classification. So the implementation on our project is reasonable. The key component of CNN is called convolutional filter which is used to process the input images in kernel layers. The quality and quantity of filter will impact the final result of our prediction. Figure 2 shows the basic filters on the first layer after training. The filter will be scanned through the whole image and extract the matched pattern as Figure 3 shows.

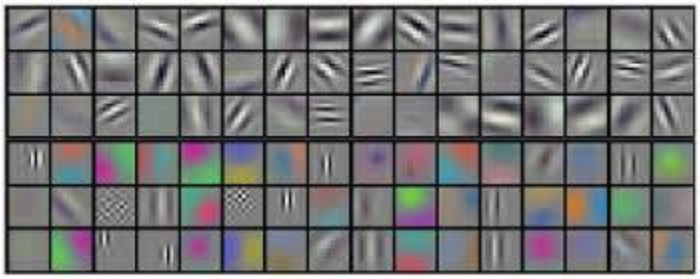


Figure2. The basic filter in convolutional layer

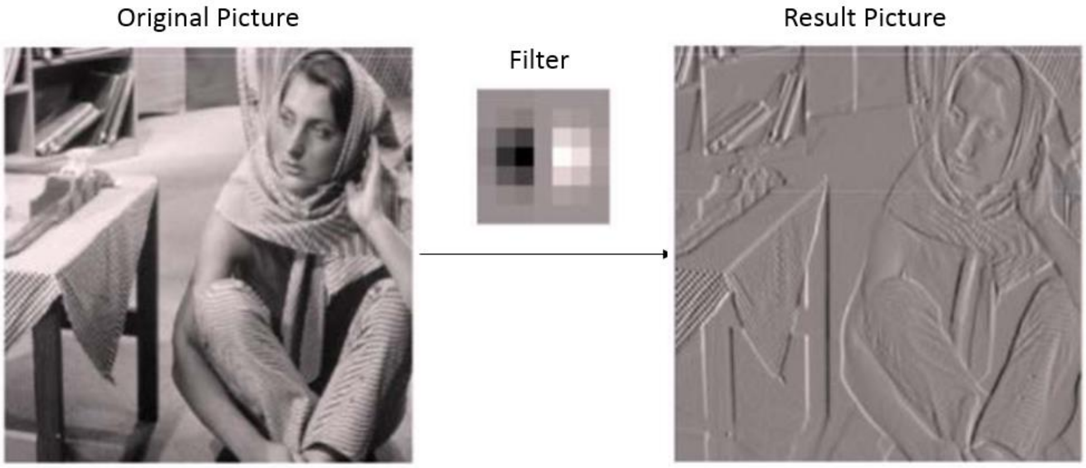


Figure 3. The original image (left) and the image after first convolutional layer (right)

After convolutional layer, the pooling layer will only leave the neural activation with the maximum value and ignore the rest, which effectively shrinks the image size for the further layers. This strategy works fine in images, because of the fact that nearby pixels are likely to have similar value. With more convolutional layers and pooling layers, the filters at deep layers will extract high level features based on the features extracted in lower layers like Figure 4 shows.

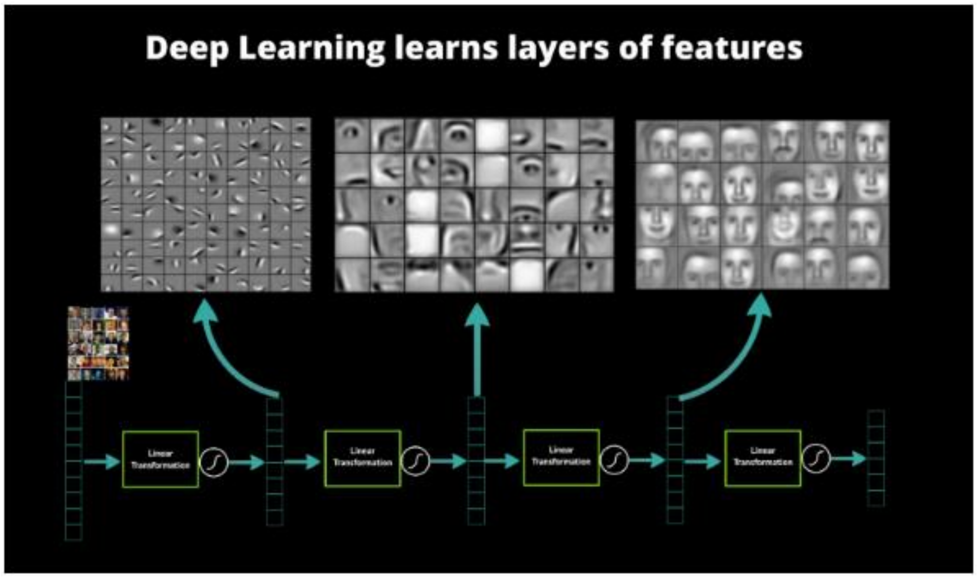


Figure4. Filters at higher layers will extract more meaningful features (left to right)

Figure 5 shows the detail architecture of our CNN network, implemented in CaffeNet [7]. Our CNN has have five convolutional layers before the fully connected layer. Traditionally, there will be a Softmax layer after the fully connected layer to calculate scores of each label the image belongs to. While in our project, we move the Softmax layer to the end of LSTM.

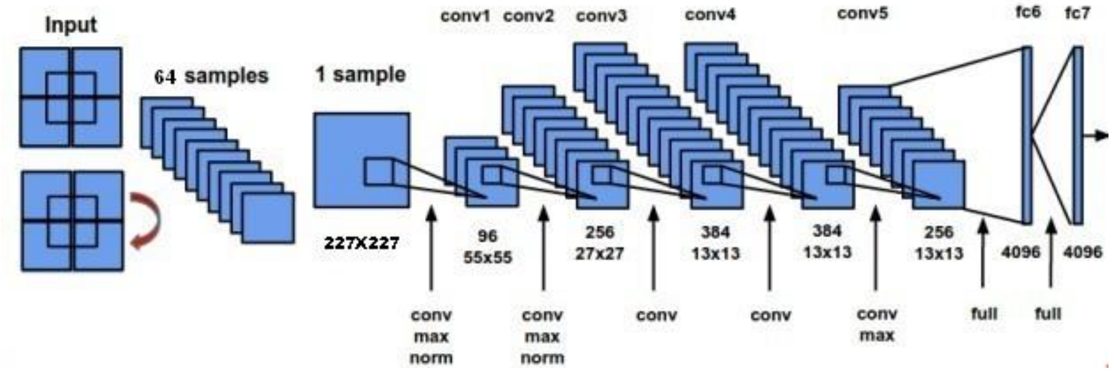


Figure 5. The architecture of CaffeNet. (Left to right is the data flow direction. The number beneath each layer is the data size and the matrix size)

2.2 Long Short-Term Memory network (LSTM)

CNN classifies images based on their appearance features. This is limited for our purposes. First, we need high resolution images for training and testing. For birds, one needs a lens long enough to take clear images about the appearance of birds before it's getting too close to the wind turbine. Second, falcon and eagle looks similar even when flying. It will be a big challenge for CNN network in detection.

Therefore, we use LSTM to overcome these above limitations from CNN network. LSTM units have hidden state augmented with nonlinear mechanisms to allow state to propagate without modification, be updated, or be reset, using learned gating functions. [8]

The appealing features of LSTM are twofold. First, they are able to connect previous information to the present task, such as using previous video frames will inform the understanding of the present frame. That means, we do not rely on the detailed images with clear bird appearance in classification the birds. LSTM allow us to use the motion features between different frames for classification. Second, LSTM is not constrained by a fixed-size input and fixed-size output. We can have videos with variant length for one prediction. As recent research, LSTMs are also demonstrated to be capable of large-scale learning of speech recognition [9] and language translation models [11], [10].

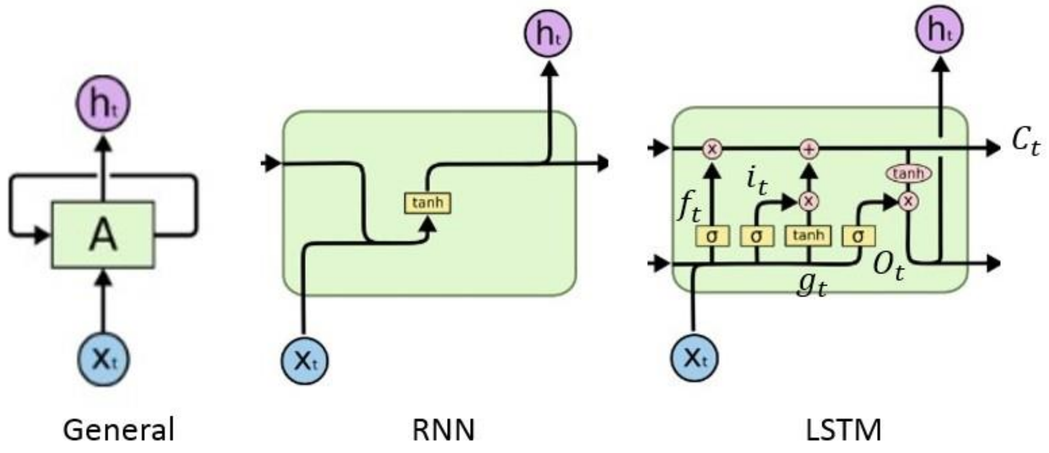


Figure 6. Left is the general form of network with self-loop. With different structure of A, it can split into RNN (Mid) and LSTM (Right)

In the right side of Figure6, it shows the detailed structure of the LSTM network we used in our approach. The yellow square represents network layer. The red cycle is point wise operation. And the arrow line means vector transfer.

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged. But the point wise multiplication operation works as a forget gate and filtering the information going through.

Letting $\sigma(x)=(1+e^{-x})^{-1}$ be the sigmoid non-linearity which squashes real-valued inputs to a [0, 1] range, and letting $\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=2\sigma(2x)-1$ be the hyperbolic tangent non-linearity, similarly squashing its inputs to a [-1; 1] range, the LSTM updates for time step t given inputs $x_t,\ h_{t-1},\ c_{t-1}$ are: [6]

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)\\
g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

When we stack several LSTMs on the top of each other, it will give us additional depth. The advantages of using LSTMs for sequential data in vision problems are the straightforward of LSTMs to fine-tune end-to-end. Besides, LSTMs can be apply to flexible length inputs or outputs, as I mentioned before which allows simple modeling for sequential data of varying lengths, such as text or video. Based on our motion, LSTM seems more suitable for our eagle flying detection. But the answer is still unknown because fine-grain classification is a challenge problem not only for LSTM but also for CNN.

We next describe a unified framework to combine LSTMs with deep convolutional networks to form end-to-end trainable networks capable of complex visual and sequence prediction tasks.

2.3 Long-Term Recurrent Convolutional Network (LRCN)

In our project, LRCN combines high-level hierarchical visual features with the model that can learn to recognize dynamic tasks involving sequential visual data, or otherwise to achieve higher prediction performance. LRCN works by passing each visual input x_t (an image in isolation, or a frame from a video) through a feature transformation φV (.) with parameters V, usually a CNN, to produce a fixed-length vector representation φV (xt). The outputs of φV are then passed into a recurrent sequence learning module. [6]

In its most general form, a recurrent model has parameters W, and use an input x_t and a previous time step hidden state h_{t-1} to calculate an output z_t and updated hidden state $h_{t.}$ Therefore, the whole model must be run sequentially (i.e., from top to bottom, in the *Sequence Learning* box of Figure 1), by computing in order: $h_1 = f_W (x_1; h_0) = f_W (x_1; 0)$, then $h2 = f_W (x_2; h_1)$, etc., up to h_T .

To predict a distribution $P(y_t)$ over outcomes y_t belongs to C (where C is a discrete, finite set of outcomes) at time step t, the outputs z_t of the sequential model are passed through a linear prediction layer $\mathbb{z} = W_z z_t + b_z$, where W_z the weight of Z_t and b_z the bias are learned parameters. Finally, the predicted distribution P (y_t) is computed by taking the softmax of $\mathbb{z}$:

$$P\left(y_t = c\right) = \frac{\exp(\mathbb{Z}, c)}{\sum_{c' \in C} \exp(\mathbb{Z}, c')}$$

The CNN base of LRCN in our eagle recognition model is a hybrid of the CaffeNet [12] reference model and the network is pre-trained on the 1.2M image ILSVRC-2012 [13] classification training subset of the ImageNet [14] dataset, giving the network a strong initialization to facilitate faster training and avoid overfitting to the relatively small eagle recognition datasets. Based on the pre-trained model, I fine-tuned the new CNN model and applied the high level features extracted by fc6 as an input of LSTMs to make the whole LRCN eagle recognition model for my approach.

3. Dataset

For this project, we prepared a new dataset of videos and the optical flow images. There are 20 videos of eagle, 30 videos of falcon and 18 videos of seagull being uses as training data in my network. For average, each video has 160 frames.

The dataset is gathered manually online by myself. They are videos from YouTube, National Geography, BBC Nature and Wildscreen Arkive website. Some of these videos can last more than 1 hour and containing many unrelated information. So the first thing is to find out the sections I can use for training and peel off the unrelated parts. To fulfill the real world purpose, the selected video sections are always birds flying in the sky. Then all the videos are cut to approximate 5 seconds long. Because the CNN and LSTM network won't take video as input directly. So I have to extract each frame out of the video. Then all the frames are resized to 320*240 pixels for fitting as the input of our network.

Besides the original image extracted from videos, I also calculated their optical flow images by using MATLAB. The difference between original image and flow image is showed in Figure7. Basically, the optical flow function in MATLAB paints the color onto the image according to motion. The larger the motion of object, the more colorful the moving part will be in the image. Ideally, it can give us more information on the flying pattern of different birds, such as the flapping frequency. I also compared the model trained by original images with the model trained by flow images in the result section.

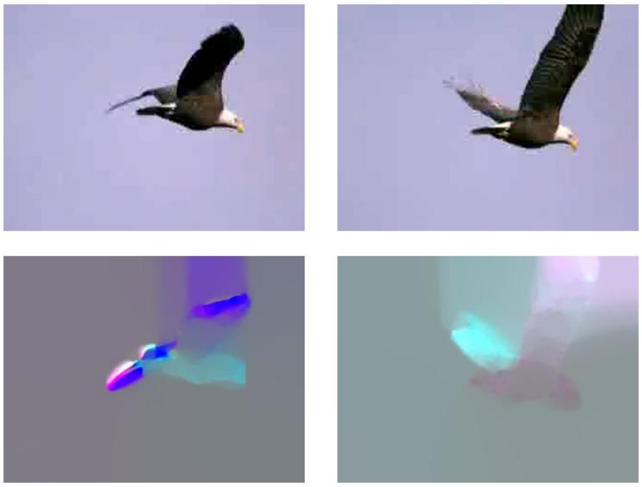
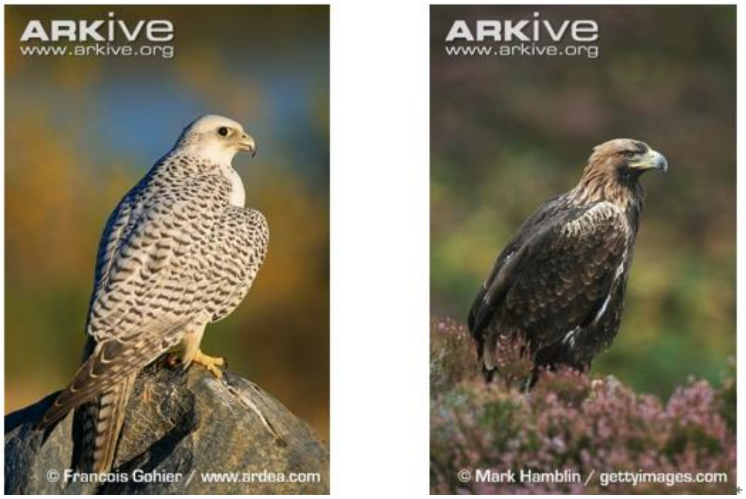


Figure7. Original image (top) and calculated flow image (bottom)

For the training part, I randomly select 90% of the total videos as training data and the remaining 10% as the testing data. I also saved several unused videos as the final evaluation data for testing final performance of my model. To minimize the impact of small training data, I also use the 10-fold technology and mirroring of the frame data to extend the training data group.



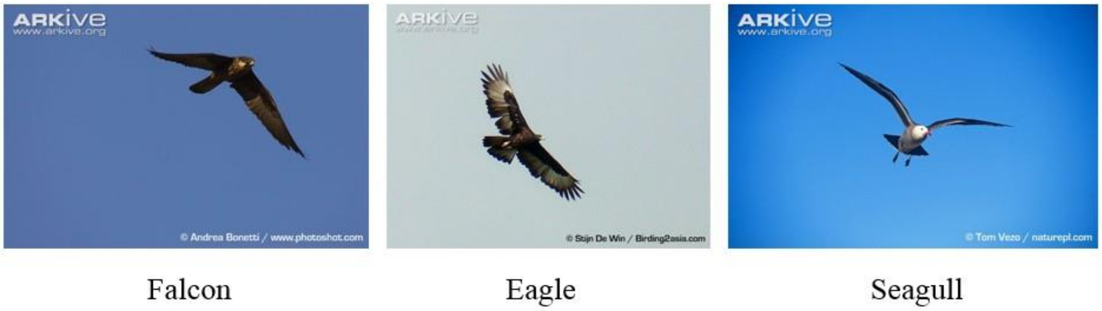
© Francois Gohier / www.ardea.com

© Mark Hamblin / gettyimages.com

Figure8. Left is standing Gyr Falcon. Right is standing Golden Eagle

There are four main challenges that impact the performance of the model.

First challenge is for the CNN model. Because falcon and eagle have very similar appearance. With large number of clear images like Figure8 and 9, we can make a high accuracy CNN model for classification. But it will not work well in our application because the data itself is not general. In our situation, we cannot easily get such high resolution and clear image of our target because of the camera. In my model, the input data is always far away in the sky and the resolution is low. That means in the testing videos, we may not capture any appearance information of the birds, but the flying pose as Figure 8 shows. The still image of birds' flying patterns won't give too much cue because of similarity. It likely that only using CNN model won't make correct classification because it lacks high detailed features. For LRCN model, we can alleviate this limitation using the time serial information as an important feature.



© Andrea Bonetti / www.photoshot.com

© Stijn De Win / Birding2asia.com

© Tom Vezo / naturepl.com

Falcon

Eagle

Seagull

Figure9. Flying pattern of three birds

Secondly, the dynamic background will also become a challenge for our model. Ideally, the background of birds should be homogeneous, as Figure 9 shows, which will help emphasizing the targets and eliminating distractions. While, in real situation, the background can be tree branches, clouds and ocean waves, etc. The importance of a pure background is also caused by the low resolution of our video data. Lacking of high level detailed features means the difference of background in each frames probably will be learned as features to classify the bird species. It will mostly cause the overfitting problem, such that the model are making classification by ‘shortcuts’ that is the difference of background.

Third, the clip speed of the videos are not consistent. Some of the falcon videos are slowed down originally. As the flapping frequency is very important feature for LSTM. The slowed down video will generate large number of misleading information. If I don’t adjust the speed accordingly, falcon will have a similar flying pattern like eagle. Because falcon originally has almost two time faster flapping frequency than eagle. While the video speed adjustment has no ground truth. I can only really on personal estimation.

Last but not the least, the videos capture by the camera on the wind turbine may vary from the ideal training video. In the training data, all the videos are captured under perfect condition, such as with stable base, nice telephoto lens. They are all different from the condition in our situation. Installed 50 meters above ground with strong wind blowing, the camera cannot easily take stable video like we did on the ground. For a light weight design, the camera won’t necessary have a telephoto lens on it which will make our target looks even smaller than it’s presented in training data. The rotation of the wind turbine will also cause unexpected irregular movement of the installed camera which never covered in the training data.

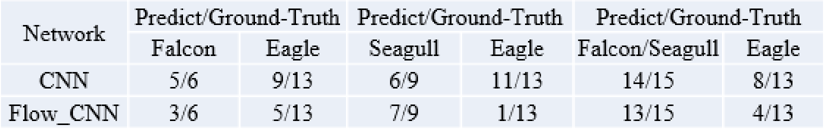
Even though there are big challenges of our data, the LRCN model still shows its advantage when comparing the single CNN model. We will discuss the result in the next section.

4. Results

From our empirical studies, the most influential hyperparameters include the number of hidden units in LSTM and the layer of CNN as an input to LSTM. We are using 1024 LSTM hidden unites our performance has 2.1% boost in accuracy in comparison to LSTM with 256 hidden units. And when using fc6 layer of CNN as input to LSTM, we get a better performance of using fc7 layer in CNN. Therefore in the test I use 1024 hidden units in LSTM and fc6 layer of CNN as input to LSTM.

To discuss the robustness of our architecture, we use three different data groups for the training which are falcon_eagle, seagull_eagle and falcon_seagull_eagle. Based on the various of our approach, we trained four different models for each dataset group.

4.1 CNN/ Flow_CNN



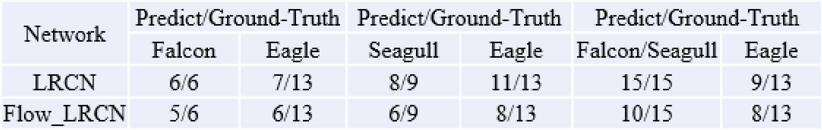
Network	Predict/Ground-Truth		Predict/Ground-Truth		Predict/Ground-Truth	
	Falcon	Eagle	Seagull	Eagle	Falcon/Seagull	Eagle
CNN	5/6	9/13	6/9	11/13	14/15	8/13
Flow_CNN	3/6	5/13	7/9	1/13	13/15	4/13

Table 1. CNN/Flow_CNN results on falcon/eagle, seagull/eagle and falcon/seagull/eagle data groups

The fraction means correct prediction out of ground-truth

As Table 1 shows, the average accuracy of using CNN is 78%. Comparing with the Flow_CNN which gives 50% accuracy, the original images as input offer more reliable features for classification than using optical flow images. Because CNN takes only one image each time for making prediction, the sequential flow information is ignored. And the limited appearance in flow image leads to the random guess result in the output.

4.2 LRCN/Flow_LRCN



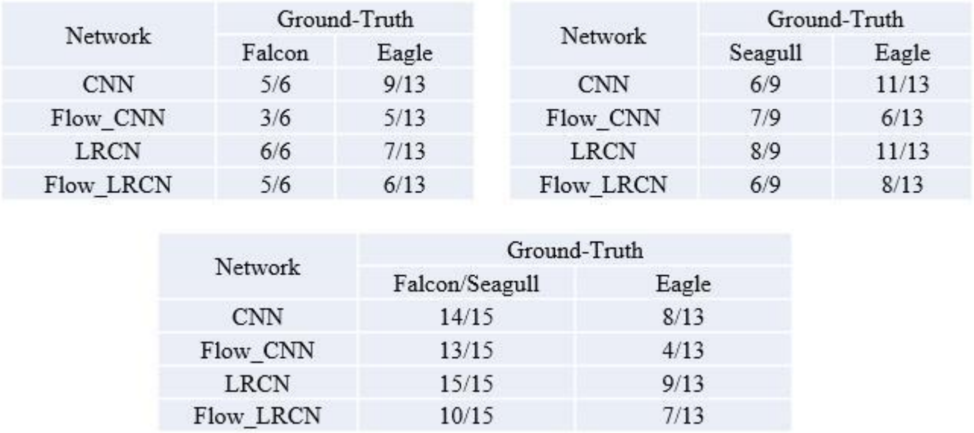
Network	Predict/Ground-Truth		Predict/Ground-Truth		Predict/Ground-Truth	
	Falcon	Eagle	Seagull	Eagle	Falcon/Seagull	Eagle
LRCN	6/6	7/13	8/9	11/13	15/15	9/13
Flow_LRCN	5/6	6/13	6/9	8/13	10/15	8/13

Table 2. LRCN/Flow_LRCN results on falcon/eagle, seagull/eagle and falcon/seagull/eagle data groups

The fraction means correct prediction out of ground-truth

The average accuracy of using LRCN is 84% which is 6% higher than only using CNN for classification. And the Flow_LRCN also yields 65% of accuracy. Unfortunately, the Flow_LRCN using optical flow images did not give higher accuracy comparing with LRCN using the original frames data. The none-homogenous background must has large impact on the classification.

4.3 Overall Comparison



Network	Ground-Truth	
	Falcon	Eagle
CNN	5/6	9/13
Flow_CNN	3/6	5/13
LRCN	6/6	7/13
Flow_LRCN	5/6	6/13

Network	Ground-Truth	
	Seagull	Eagle
CNN	6/9	11/13
Flow_CNN	7/9	6/13
LRCN	8/9	11/13
Flow_LRCN	6/9	8/13

Network	Ground-Truth	
	Falcon/Seagull	Eagle
CNN	14/15	8/13
Flow_CNN	13/15	4/13
LRCN	15/15	9/13
Flow_LRCN	10/15	7/13

Table 3 . Final result of using three models-Falcon/eagle (top left), Seagull/eagle(top right), Falcon/seagull/eagle (bottom). The fraction means correct prediction out of ground-truth

From Table3, the overall performance of using LRCN is higher than using single CNN network in most cases. Especially, in the three species model, the LRCN improves accuracy by 6.67% on predicting falcon/seagull and 7.69% on predicting eagle than using single CNN.

Surprisingly, the CNN model still gives as high accuracy as LRCN model on Falcon/eagle and Seagull/eagle data groups. It means the CNN model is well designed enough to handle fine-grained bird classification by little appearance images. While, with the including of more negative data (falcon/seagull) the CNN model shows its disadvantage comparing with the LRCN model. The mixed high level features of both falcon and seagull become a bottleneck of the model which is hard to improve. While, with the help of LSTM, the LRCN model can successfully overcome the drawback and make correct classification by the movement pattern between video frames.

For optical flow images, the accuracy of falcon/eagle dataset is lower than the models using original image as input data. There are number of reasons. Mostly, the camera is moving during the recording process which means the eagle maybe not the only moving object in the frames. When the background moves, the calculated optical flow image will give large weight to the background. This will cause noise for our classification because the background should not become the main feature during LRCN training. Inside the falcon testing data, it contains a lot of trees as the background. The low accuracy of flow falcon/eagle images is mainly caused by that reason. On the other side, the seagull/falcon flow image dataset yields higher or as much accuracy as original image dataset for both model. It's caused by the homogenous background of the seagull dataset which largely reduced noise to the model.

By the comparison of different birds' data groups and flow/original images, LRCN shows its advantage on most of the cases than single CNN model. While, the optical flow data is not so reliable with nonhomogeneous background.

4.4 Running time

The network training process is one of the most time-consuming part of our project. The training was done on the campus server: gpu-bart.eecs.oregonstate.edu. The graphic cards installed on the server are two NVIDIA K80, each with 24GB memory and 4992 CUDA cores.

We fine-tuned the CaffeNet from Zeiler&Fergus [15]. There are totally 7000 images in the training group and 1000 images in the testing group. It took 5 hours and 4 GB of memory for fine-tuning one model with 50000 iterations. After the training of CNN, we fused the CNN with LSTM and started training the LRCN model. It took 3 days and 11 GB of memory for training each model with 30000 iterations. Finally, it will take 2 hours to run the extra evaluation image group. It approximately took one week to finish one LRCN model using one data group. There are three data groups and two LRCN models I trained in our project in total.

5. Summery

Overall, the key contribution of our work is to show that a robust fine-grained object detection can be done in low resolution videos using a deep recurrent neural network. Another contribution is evaluating and benchmarking the approach on a new dataset. Because this is to the best of our knowledge, the first model and dataset for fine-grained bird detection in videos.

While, there is still various way to improve this project. When analyzing the output, I found that the larger training data group on one class will yield a higher accuracy on this class. In falcon/eagle data group, I have 30 videos of falcon and 20 videos of eagle as training data. The accuracy of predicting falcon is higher than accuracy of predicting eagle. Same thing happened in seagull/eagle data group. To reduce this impact, the next step will be collecting more dataset for training and give same number of training data for each class. To better discuss the using of optical flow image as training data, it would be necessary to collect more videos with homogenous background for training, as well.

References

[1] Vasilakis, Dp ; Whitfield, Dp ; Schindler, S ; Poirazidis, KS ; Kati, V "Reconciling endangered species conservation with wind farm development", 2016

[2] Karin Sinclair, Elise DeGeorge "Wind Energy Industry Eagle Detection and Deterrents: Research Gaps and Solutions Workshop Summary Report", National Renewable Energy laboratory, 2016

[3] Dong, C., Loy, C., He, K., & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 38*(2), 295-307.

[4] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," CRCV-TR-12-01, Tech. Rep., 2012.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in Neural Computation. MIT Press, 1997.

[6] Jeff Donahue, Lisa Anna, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell "Long-term Recurrent Convolutional Networks for Visual Recognition and Description", in arXiv.org, 2016

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in ACM MM, 2014.

[8] Yao, K., Cohn, T., Vylomova, K., Duh, K., & Dyer, C. (2015). Depth-Gated LSTM.

[9] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in ICML, 2014.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in NIPS, 2014.

[11] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in SSST Workshop, 2014.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *IJCV*, vol. 115, no. 3, 2015.

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

[15] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In ECCV. 2014.