Bird Sound Classification using Convolutional Neural Networks

Chih-Yuan Koh[1⋆], Jaw-Yuan Chang[1⋆], Chiang-Lin Tai[1], Da-Yo Huang[1], Han-Hsing Hsieh[2], and Yi-Wen Liu[1]

[1] National Tsing Hua University, Hsinchu 30013, Taiwan
{jimmy133719,aspy15245, taijohnny38, you0112}@gapp.nthu.edu.tw, ywliu@ee.nthu.edu.tw
[2] National Chiao Tung University, Hsinchu 30010, Taiwan
hank0924097907.ee04@g2.nctu.edu.tw

Abstract. Accurate prediction of bird species from audio recordings is beneficial to bird conservation. Thanks to the rapid advance in deep learning, the accuracy of bird species identification from audio recordings has greatly improved in recent years. This year, the BirdCLEF2019[4] task invited participants to design a system that could recognize 659 bird species from 50,000 audio recordings. The challenges in this competition included memory management, the number of bird species for the machine to recognize, and the mismatch in signal-to-noise ratio between the training and the testing sets. To participate in this competition, we adopted two recently popular convolutional neural network architectures — the ResNet[1] and the inception model[13]. The inception model achieved 0.16 classification mean average precision (c-mAP) and ranked the second place among five teams that successfully submitted their predictions.

Keywords: Deep Learning, Inception-v3, Bird sound recognition, BirdCLEF2019

Source code of this project: https://github.com/jimmy133719/BirdCLEF2019

1 Introduction

Public consciousness about environmental conservation and sustainable development has awakened in recent years. Demands for automatic bird call classification have also been rising owing to the key role of birds in the ecosystem. Compared to video-based monitoring, sounds have the advantage of propagation to a long distance without being occluded by objects in between the emitting source (a

⋆ equal contribution

bird in this case) and the recording devices. Therefore, a robust system to identify bird vocalization may become useful for monitoring species diversity at a fixed location as well as detecting bird migration along a route. Realizing the importance of this task, a competition called BirdCLEF has been hosted every year by the LifeCLEF lab[3] since 2014. The goal of the competition is to identify bird species in audio recordings. In the competition this year, participants needed to detect bird calls in every 5 seconds of the soundscape recordings from the Xeno-Canto database[3] and identify the species if bird calls are present.

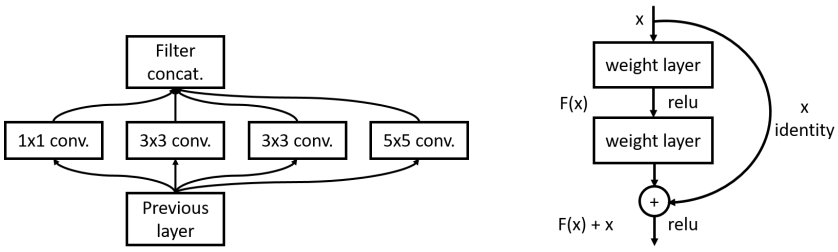
Previous attempts to use machine learning approaches for bird call identification include decision trees, convolutional neural networks (CNN), and recurrent neural networks (RNN). For instance, randomized decision trees were applied [6] and the input consists of features derived from statistics of the spectrograms. By ranking the *feature importance* returned from the decision trees, one can find relevant segments to identify each sound class. Due to the computation load in deriving the statistics from spectrograms, the decision-tree technique might not be most suitable for the current BirdCLEF challenge; its ability to handle more than 600 species remains a concern, too. The RNN-based model was adopted in last year's BirdCLEF challenge. In particular, the bidirectional long short-term memory (LSTM) architecture was applied [8]. It made use of sequential information in bird calls audio. However, because of the gradient vanishing and explosion problems associated with the sigmoid gate function, the model is difficult to reach convergence. Besides, due to the nature of RNN, preprocessing and augmentation are difficult to implement. Therefore, it seems that CNN-based models become the most common approach in bird call recognition. In general, the spectrogram of bird sound audio is regarded as the input and the model would treat the bird-call identification task as an image classification problem. This is intuitive, because features of bird calls unique to each species, such as the pitch and the timbre, can be observed in the spectrograms by experienced human eyes.

To participate in BirdCLEF 2019, we thus decided to apply two modern CNN-based models, ResNet and Inception. The rest of this paper is organized as follows. In Section 2, we briefly review the background of ResNet and Inception. More details concerning model implementation and training are described in Section 3. Experiments and results are described and analyzed in Section 4. In retrospect, Section 5 points out several flaws of our attempts, and conclusions and future directions are given in Section 6.

2 Background

In this section, we briefly review the spirits underpinning Inception and ResNet.

[3] https://www.xeno-canto.org/



(a) the inception module (b) the residual block

Fig. 1: Basic concepts of two recently developed techniques in CNN

2.1 Going Deeper with Convolutions

Titled "Going Deeper with Convolutions", the GoogleNet[12] presented a brand new neural network structure which, rather than blindly stacking convolution layers, emphasized on reducing the sparsity of the feature map. It replaced a general convolution layer by what is called the *inception module*; that is, instead of using a large-sized kernel in the convolution layer for feature extraction, smaller kernels were constructed in parallel. The concept is depicted in Fig. 1a.

2.2 Deep residual learning for image recognition

Titled "Deep Residual Learning for Image Recognition"[1], the ResNet mainly addressed the issue of model degradation. Because of the nonlinear activation function, the back-propagation gradients might vanish, which degrades the performance of deep CNNs. Hence, a highway pass between the upper and the lower layers was introduced and the result is known as the *residual block*. It can be expressed in the following general form,

$$y_i = h(x_i) + F(x_i, W_i), \tag{1}$$

where x_i and y_i are the input and the output of the i^{th} block, and F denotes a flexible residual function which is parameterized by the weight array W_i. A common practice is to set $h(x_i)$ as the identity mapping and use the rectified linear unit (ReLU) between the weight function inside F, and Fig. 1b illustrates the main idea of ResNet.

3 Methods

This section describes details concerning the data processing and how the models were trained and tested.

3.1 Preprocessing

Spectrogram extraction Our strategy for the task was to treat bird sound classification as image classification; hence we need to visualize bird sounds. A commonly used technique is the MEL-scale[11] log-amplitude spectrogram, a kind of time-frequency representation that takes human auditory sensitivity with respect to frequency into consideration. Since the occurrence of bird calls could be sparse, we chopped the signal into 1-second chunks instead of extracting a spectrogram for the entire 5-second recording. A band-pass filter with cut-off frequencies of 500Hz and 15kHz was applied, since most bird species vocalize within this frequency range.

.

Table 1: Parameters of spectrograms

Parameter	Value
Sampling rate	44100 Hz
Window length	1024
Hop length	172
Number of Mel filters banks	128
Minimum frequency	500 Hz
Maximum frequency	15000 Hz

Parameters for computing the spectrograms are shown in Table 1. Note that the hop length was determined so that each clip of one second contains exactly 255 frames. With this specification, each a Mel-scale log amplitude spectrogram has a size of 128×256. To decide whether each second contains bird calls, we applied a simple signal-to-noise ratio (SNR) threshold based on the methods described by BirdCLEF participants in previous years[5,10]. Fig. 2 shows a few examples of spectrograms with different SNRs. By inspection, a spectorgram with a higher SNR indeed contains clearer traces that indicate the presence of a bird call. In contrast, a spectrogram with a low SNR may only contain background noise. Based on the SNR, we could set a threshold to include only the spectrograms with a sufficiently high SNR as samples for training the neural networks.

(a) 0.127 (b) 0.021 (c) 0.001

Fig. 2: Extracted spectrograms with different SNR

Table 2: The means for data augmentation applied in this study

Means	**Description**
Gaussian white noise	Additive noise with zero mean and a variance of 0.05
Adjusting brightness	Randomly multiply the whole spectrogram with a value between 1 ± 0.15
Random multiply	Randomly multiply each pixel with a value between 1 ± 0.25
Image blur	Replace each pixel with the mean of its 3×3 kernel neighbors.
Vertical and horizontal roll	Random shift by a value between ± 0.5 of the height and ± 0.15 of the width, respectively.
Random crop	Here we only cropped the height of spectrogram. The width remained the same.
Rotation	The spectrogram was randomly rotated by an angle bewteen ± 10 degrees.
Dropout	Apply a random uniform matrix as a weight of spectrogram and set the weight to zero if that element is less than 0.25; otherwise set it to one.
Blackout	Set the value in randomly chosen consecutive 25 columns to zero.

Data Augmentation We found class imbalance due to two reasons; first, the SNR in some classes (i.e., bird species) is low. Secondly, some classes simply have few recordings in the database. Hence, we randomly applied several augmentation methods (Table 2) to the spectrograms of those classes with insufficient training samples. By inspection, the spectrograms of soundscape recordings in the validation set all seem to contain noise. Therefore, in two of our submissions, we added Gaussian noise to all of spectrograms, and fed both the orginal spectrograms and those with Gaussian noise to the neural-network models.

Normalization In two of our submissions, normalization was applied. The mean and the variance for transformation was calculated from the entire training dataset.

Output format By observing the training data, we find out that bird vocalization could be quite sporadic. Also, in the validation data, audibly different bird calls rarely occur at the same time. Hence, we determined to decode the output of the neural network into a one-hot vector.

3.2 Network architecture and configuration

ResNet In one attempt, we adopted the ResNet-18[1] classifier for bird-call identification. The optimizer was stochastic gradient descent (SGD) with momentum and the batch size was set to 64. Weight initialization was applied, which sampled from normal distribution $\mathcal{N}(0, 0.02)$ for convolution layers and $\mathcal{N}(1, 0.02)$ for batch normalization2d layers. the number of channels in ResNet has been reduced to 1 (because spectrograms do not have RGB colors); as the number of input features decreased, the model complexity can also be lowered so as to increase the training efficiency within a limited amount of time.

Inception model On the basis of Inception v3[13], we substituted two connected convolution layers with kernel size $(1, 7)$ and $(7, 1)$ for the convolution layer with kernel size $(7, 7)$. Meanwhile, an additional activation function has been added between the two small convolution layers. The optimizer was changed to Adam and the batch size was set to 64. The same weight initialization that was applied to ResNet has also been applied here. The number of channels was decreased before the data flows into Inception module 5c. Other details about modification of the Inception model can be found in our github contribution.[4].

4 Results

Table 3: Evaluation of Inception and ResNet via validation data

model	Inception	ResNet18	ResNet34
cmAP	0.23	0.13	0.11
rmAP	0.39	0.05	0.01

Table 3 shows the best performance that we obtained from the two models on validation data. For some reasons, inception obviously was better than ResNet. For more details about hyperparameter and corresponding validation result, please see the appendix. The official main score, cmAP, was defined by the competition organizers as follows,

$$\text{cmAP} = \frac{\sum_{c=1}^{C} \text{AveP}(c)}{C}, \tag{2}$$

and

$$\text{AveP}(c) = \frac{\sum_{k=1}^{n} P(k) \times \text{rel}(k)}{n_{\text{rel}}(c)}. \tag{3}$$

[4] https://github.com/jimmy133719/BirdCLEF2019

In Eq. (3), C denotes the number of species in the ground truth, and AveP(c) is the average precision for a given species c, which is defined in Eq. (4) — k denotes the rank of an item in the list of the predicted segments containing c, n denotes the total number of predicted segments containing c, $P(k)$ denotes the precision at cut-off k in the list, rel(k) is an indicator function that equals 1 if the segment at rank k is labeled as containing c in the ground truth (and zero otherwise), and n_{rel} is the total number of relevant segments for c.

Table 4: Evaluation of each run

run	1	2	3	4
cmAP	0.140	0.149	0.160	0.154
rmAP	0.110	0.117	0.164	0.184

Due to the comparatively better performance of the inception model, we adopted it in all of our submissions. Table 4 shows the official evaluation of each run. In the first two runs, Gaussian noise was not added to the whole spectrograms. The only difference between them was in the threshold of SNR with 0.005 and 0.001, respectively. In the third and forth runs, we added Gaussian noise to all spectrograms and normalized all of the spectrograms before training. The difference between them was the epochs we chose. The official results surprised us since the first two runs actually performed better when evaluated in the validation set. Otherwise, compared to other teams, our rmAP peculiarly was not much higher than cmAP. This is not surprising, probably because we assigned five classes to most of the five-second segments that contains bird sound, which increases the denominator in the equation of rmAP.

5 Discussion

The audio in the testing set contains various kinds of environmental noise (such as the sounds of insects), but we ran out of time in designing a generalized method to deal with it. The noise results in difficulties to extract the correct feature from spectrograms. The SNR threshold we applied to determine the bird calls' presence also has some concerns, since we might have included a spectrogram with intense noise instead of targets, and those spectrograms contain only noise would be treated as training data. In data augmentation, though most of the means in Table 2 could be useful in image processing, but in retrospect they are not always suitable to spectrograms. In particular, the two axes of spectrogram have different meanings (time vs. frequency), so a similiar shape occuring at different locations may correspond to features that are unique to different bird species. Hence, augmentation such as rotation might actually have been harmful during the training stage.

We deduce a simple reason why Inception-v3[13] outperformed ResNet[1] could be because of the number of parameters. The more parameters a model

has, the more accurately the model can representation the mapping between the input and the output. In our experiment, the parameter size of ResNet-18 is 9.12 MB, the parameter size of ResNet-34 is 10.41 MB, and the parameter size of Inception-v3 is 92.3 MB.

6 Conclusion and Future work

Our work is based on the baseline of BirdCLEF last year. The main difference is that we change the model to Inception-v3. Future work will focus on improving the preprocessing. We need to enhance clear bird sound features on the spectrograms of soundscape recordings so that it can be similar to the training set. In the current approach, we added Gaussian white noise as a mean of data augmentation. We would like to change it to the noise from the recording environments. Moreover, since the current preprocessing mainly focuses on the magnitude spectrogram, it might beneficial to learn additionally information from the phase spectrogram, especially when multiple recording channels are available.

On the part of model, due to the shift-invariance and the parameter-sharing property, CNN may be in trouble distinguishing spectrograms that contain features of similar shapes but occuring in different frequencies. A possible way to mend this would be tiled convolution. Although it is still a CNN model, a tiled convolution model [9] has locally confined receptive field; that is, the parameter sharing of tiled convolution is not global. Employing attention mechanisms can also be recommended, since not all the neurons would end up equally important as the signal comes out from a specific range of frequency [7]. Yet another neural architecture, the SENet[2], is worth trying because the resolution of spectrograms can be increased in order to preserve more details.

7 Acknowledgement

This research is supported by Airoha Technology Corp. We are also grateful to LifeCLEF2019 team for holding this challenge.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 770–778 (2016)
2. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 7132–7141 (2018)
3. Joly, A., Goëau, H., Botella, Christophe Kahl, S., Servajean, Maximillien Glotin, H., Bonnet, P., Vellinga, W.P., Planqué, R., Stöter, F.R., Müller, H.: Overview of lifeclef 2019: Identification of amazonian plants, south & north american birds, and niche prediction. In: Proceedings of CLEF 2019 (2019)
4. Kahl, S., Stöter, F.R., Glotin, H., Planque, R., Vellinga, W.P., Joly, A.: Overview of birdclef 2019: Large-scale bird recognition in soundscapes. In: CLEF working notes 2019 (2019)

5. Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M., Eibl, M.: Large-scale bird sound classification using convolutional neural networks. In: CLEF (Working Notes) (2017)
6. Lasseck, M.: Bird song classification in field recordings: winning solution for NIPS4B 2013 competition. In: Proc. Int. Symp. Neural Information Scaled for Bioacoustics, sabiod.org/nips4b, joint to NIPS, Nevada. pp. 176–181 (2013)
7. Liao, H.W., Huang, J.Y., Lan, S.S., Lee, T.H., Liu, Y.W., Bai, M.R.: Sound event classification by a deep neural network with attention and minimum variance distortionless response enhancement. In: IEEE DCASE Challenge Technical Reports (2018)
8. Müller, L., Marti, M.: Bird sound classification using a bidirectional LSTM. In: Working Notes of CLEF 2018 (Cross Language Evaluation Forum) (2018)
9. Ngiam, J., Chen, Z., Chia, D., Koh, P.W., Le, Q.V., Ng, A.Y.: Tiled convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1279–1287 (2010)
10. Sprengel, E., Jaggi, M., Kilcher, Y., Hofmann, T.: Audio based bird species identification using deep learning techniques. LifeCLEF 2016 pp. 547–559 (2016)
11. Stevens, S.S., Volkmann, J., Newman, E.B.: A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. **8**(3), 185–190 (1937)
12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 1–9 (2015)
13. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)

A Hyperparameter sheet and the corresponding validation results

Table 5: Hyperparameter sheet

model	learning rate	L2 lambda	optimizer	cmAP	rmAP
Resnet-18[1]	0.1	0	SGD	0.13	0.05
Resnet-18[1]	0.01	0	SGD	0.01	0.01
Resnet-18[1]	0.001	0	SGD	0	0
Resnet-34[1]	0.01	0	SGD	0	0
Resnet-34[1]	0.001	0	SGD	0.11	0.01
Resnet-34[1]	0.01	0.001	SGD	0.05	0.04
Resnet-34[1]	0.001	0.001	SGD	0.11	0.01
Inception-v3[13]	0.001	0	Adam	0.23	0.39

The momentum of optimizer SGD is set to 0.9. The β_1 and β_2 for optimizer Adam are set to 0.9 and 0.999, respectively.