CNN-based Multi-model Birdcall Identification on Embedded Devices

Shidong Pan
College of Engineering & Computer Science
Australian National University
Canberra, Australia
Shidong.Pan@anu.edu.au

Dehai Zhao
College of Engineering & Computer Science
Australian National University
Canberra, Australia
Dehai.Zhao@anu.edu.au

Weishan Zhang
College of Computer Science and Technology
China University of Petroleum
Qingdao, China
zhangws@upc.edu.cn

***Abstract*—Bird species identification by their vocalization on embedded devices is an important and challenging task. In this paper, we propose BIRD (Bird Identity Record and Detection) system as a complete IoT solution for birdcall identification in wild field environment, and verify its feasibility by a simulation experiment. The bird call signals are firstly converted to spectral field then input into deep neural network to classify bird species. To improve the classification performance in BIRD system, we introduce a CNN (Convolutional Neural Network)-based multi-model network, which fuses acoustic signals and geographical coordinate information into a unified model. Our work achieves 0.849 accuracy and 0.749 $F1$-score over 397 species on BirdCLEF2021 dataset, outperforming traditional classification models.**

***Index Terms*—Internet of Things, Multi-model, Deep Learning, Convolutional Neural Network, Birdcall Identification, Data Fusion**

I. Introduction

Applying Deep Learning (DL) techniques on biology topics is an emerging research field in recent years, especially for critical issues such as protection of ecosystem biodiversity, preservation of endangered species, monitoring of natural habitat, etc. Birds sit at the top of the food chain, thus naturally, they are regarded as excellent indicators of deteriorating environmental quality and pollution monitoring. Birds are cloaked by feather, which making them visually unrecognizable in nature environment; however their voice are resonant and penetrating enough to be captured. In addition, it is effortless to obtain the coordinates information (latitude, longitude) on Earth by Global Positioning System (GPS). Therefore, in this paper, we focus on the problem of classifying bird species by fusion information on embedded devices.

Generally, DL techniques require strong computational resources, vast storage capacity and the ability to process large amounts of data volume. Thankfully, the emergence of cloud computing provides not only an efficient and economical solution, but also pushes the horizon of a new computing paradigm. With the success of the Internet of Things (IoT), cloud computing and IoT tremendously changed the way that people live, work, and study [1], [2]. However, cloud computing is not efficient enough to support all IoT applications, especially showing its weakness on tasks that process the majority of data at the edge of the system, or scenarios that the Internet connection is unavailable. Therefore, the potential of edge computing has been unveiled gradually, and the combination between cloud and edge computing is becoming a heat topic.

Birds species identification in field is a significant but challenging issue. Firstly, birds play an irreplaceable role in maintaining the natural ecological balance because they can spread pollen and seeds, prey on pests and provide fertilize to coral reef. Also, birds are called "flying sentinels" since they are particularly sensitive to environmental pollution, and researchers can observe hydrological changes in wetlands depending on the migration of migratory birds. To gain the first-hand data, ecologists usually need to take field trips to birds habitats, where generally the working condition is difficult (poor internet quality, limited package space, etc). Therefore, to tackle the above challenging and inspired by the advantages of cloud/edge computing in IoT, we formulate the BIRD system that includes following merits: portable sensor for expedition; excellent performance for birds species identification; real-time; workable without internet or weak signal; and feasible for data collecting and storage.

The BIRD (Bird Identity Record and Detection) system is an elastic system that is compatible with deep learning models and effortlessly to be deployed. The system architecture is conventionally similar to the common IoT configuration strategy, and the core algorithm for birds species identification is our CNN (Convolutional Neural Network)-based multi-model network. We will describe the BIRD system in details at Section III-A and the multi-model network in Section III-E.

In summary, we make the following key contributions:

- We compare the performance and efficiencies for different CNN structures on a real birdcall identification

dataset, and analysis factors that may affect the final results.
- We propose a CNN-based multi-model network for birdcall identification by fusion information between acoustic signals and geographical coordinates, achieving a better performance compared to traditional bird species classification models.
- To best of our knowledge, this paper is the first work to introduce a complete system architecture (BIRD) for real-time birdcall identification at embedded devices, data collecting and storage, as well as identification software updating.

A. Paper organization

The rest of this paper is organized as follows: Section II briefly introduces the development of related work; Section III covers the definitions and explanations of key methods; Section IV specifically describes the details of experiments and the result; and lastly, in Section V, we present a conclusion and an outlook to the future.

II. Related Work

Recently, with the increment of computing power on portable devices, performing deep learning inference tasks by edge computing in IoT system is attracting growing attention, and because of the limited computing capability, how to execute sophisticated deep learning software on embedded devices become the problem that need to be solved [3]–[7]. [6], [7] both propose a skeleton of IoT deployment and maneuver framework to tackle this issue, but their solution are too viewy to apply on specific topics in the practical perspective. [5] raise the idea of In-situ AI, which is "the first Autonomous and Incremental computing framework and architecture for deep learning based IoT applications." They exploit the computation potential and improved the efficiency of executing Fully Connected Network (FCN) layers on mobile GPU and FPGA (two popular IoT devices). A concomitant merit of applying DL methods in IoT system is that data transporting can be more secure and private. Rather than uploading/downloading the original data, features extracted by Deep Neural Network (DNN) filters in the intermediate layers can be safely delivered to other ends since they are basically non-understandable for attackers [4].

In addition, there is a growing interest for applying DL techniques on sound-related topics. In healthcare field, [8], [9] applies several deep learning models to diagnose vocal disorder in pathology as a binary classification task, achieving decent performance, around 93% and 87% respectively. Apart from that, [10] present a comparison between several DL techniques on environmental sound detection, which expressly classifying captured sounds into "one of fifteen common indoor and outdoor acoustic scenes".

Furthermore, animal voice classification by DL approaches has also been extensively studied in previous literature [11]–[13]. [12], [13] both apply deep CNN models to predict the animal species based on vocalization, because CNN has strong capacity to capture and learn latent spatial information in graphical data. To convert audio data into graphics, [11], [12] convert audio files into spectrograms, whereas [13] to spectrums. Because of higher similarity, the voice identification task becomes more challenging for distinguishing different species for same animal class, such as birds. [14]–[17] briefly review the development of birdcall identification, and we found that most previous researches conducted on small size dataset with comparatively few species (lower than 50), limiting their researches are only meaningful in certain scenario. Moreover, according to [16], applying deep-CNN ($>$ 50 layers) architectures directly on spectrograms to identify birdcall can achieve decent performance.

III. Methodology

A. BIRD System Overview

We firstly introduce the the configuration architecture of the BIRD system, and as it shown in figure 1, there are three main components topologically:

- Portable sensor. An embedded mobile device that can be used by individual researcher to detect bird species by built-in identification software in real-time, without specific pre-knowledge, and it is workable under condition that has weak or no internet connection. Meanwhile, it will record and store raw birdcall data for following researches.
- Local computer. It plays an intersection role between cloud and edge devices. After manually inspection, valid data samples will be upload to central server by here, and researchers can request to update identification software in portable sensors from cloud server.
- Central computer/server. Central computer undertakes the neural network model training and the following updating. Central server is responsible for data storage, processing data uploading requests, and identification software downloading requests. Usually this component can be purchased and supported by cloud services providers such as Google, Microsoft, Baidu, etc.

Limited by the fund, rather than building the whole system, but we run a simulation experiment at Section IV-D to prove the feasibility of our proposal.

B. Spectrograms

The original birds sounds sampled from natural environment can be saved as audio files (.wav, .mp3, .aif, .ogg, etc.), and we're accustomed to use a series of waveform to display signal's amplitude over time as shown in Figure 2. According to [18], [19], from the Fourier analysis perspective, deep CNNs exhibit better ability to capture low-frequency information and faster convergence for lower frequencies information in images. Therefore, by applying Fast Fourier Transform (FFT) algorithm, zigzag waveform graphs can be transformed into spectrograms such as in Figure 2, and apparently high frequency (Fourier) information are converted into low frequency (Fourier) information with clearer patterns and features to be comparatively easier captured.

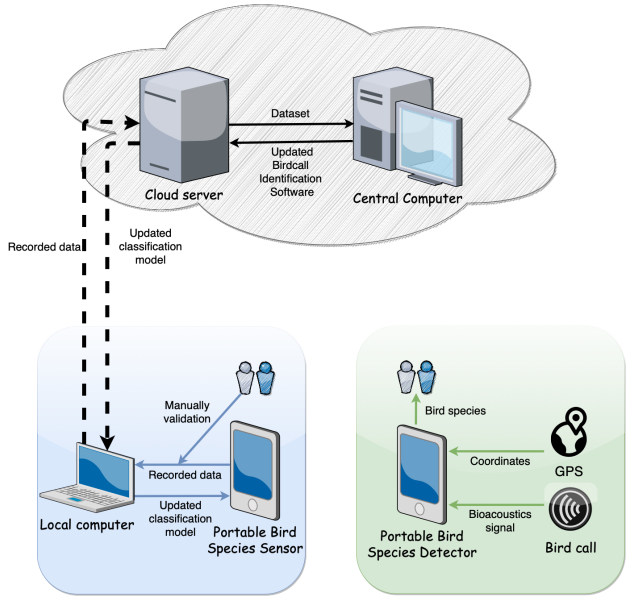


Fig. 1: The configuration architecture of BIRD system. At cloud, the cloud server provides birdcall dataset to the center computer to train the CNN model, and then the central computer returns the model as the birdcall identification software to the cloud server. shows the combination of training model at cloud, and conducting bird species classification tasks on portable devices by edge computing.

Specifically, in spectrograms, the horizontal axis represents time, the vertical axis displays frequency in Hertz (Hz), and the brightness reflects amplitude at every moment. In addition, FFT is an algorithm to compute the Discrete Fourier Transform (DFT) of a sequence, which can be expressed by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad k = 0, \cdots, N-1 \qquad (1)$$

where N is the number of sample, n is the current sample, x_n is the amplitude at time n, k is the current frequency(theoretically from 0 to $N-1$ Hz, but empirically only between 0 to $N/2$), and X_k is the DFT results.

C. CNN

Computer Vision (CV) industry rapidly booms in recent years, and especially the signature Convolutional Neural Network (CNN) model, as a variant of ANN, has been broadly applied on solving traditional CV topics such as images semantic segmentation, object recognition, etc, dominantly outperforming than old-school rule-based methods. With the development of cross-domain between CV and other practical subjects, classic models such as VGG [20], GoogLeNet (inception module) [21] or ResNet [22] have shown their strong capability on tasks in various fields. Based on the previous successes of ResNet on sound-related projects, we will compare several ResNet in terms of performance, efficiency and size on a large dataset to determine which specific ResNet architecture is the optimal choice for BIRD system. Namely, candidates are ResNet18, ResNet50 and ResNeXt50.

The main innovation of ResNet lies in the design of a residual structure which uses skip connection, enabling the network to reach a deep level and improving the performance greatly. The number (18, 50) after ResNet just represent the amount of layers with weights. ResNeXt [23] combines the original ResNet structure and inception module from GoogLeNet, slightly reducing the size of parameters, leading to a better performance compared simply deepening or widening the network. As it is shown in Figure 3, we see that ResNeXt extents the single path into multiple paths with less channels, meanwhile the signature residual shortcut keeps unchanged.

The training of the network is achieved by minimizing the cross-entropy loss [24] computed as:

$$L(f) = -\sum_{i}\sum_{c} y_{ic} * \ln(f_{ic}) \qquad (2)$$

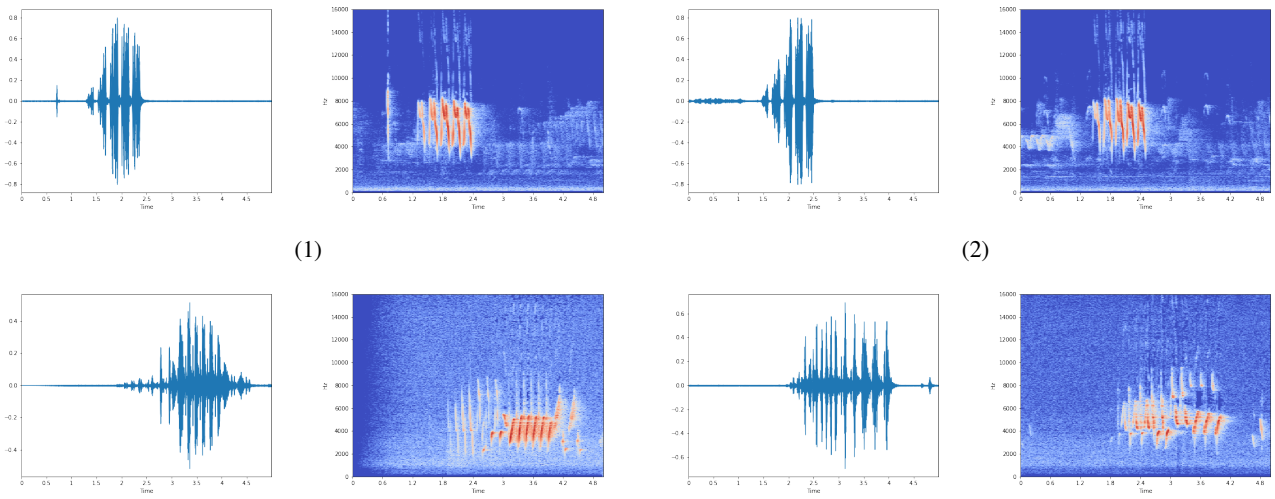
And optimized by Adam optimizer [25].

D. Ont-hot Encoding

The idea of one-hot encoding is converting original data into a binary vector, that is all zeroes except the target digit that is marked with a 1. To maximize the inherit spatial information of geographical coordinates collected from GPS, we are naturally inspired to apply this method to encode them into vectors, so those vectors can be effortlessly feed into our multi-model network . In detail, we apply one-hot encoding to convert the coordinates pairs into a fixed length binary vectors, which uses 1 to represent the birdcall collected location. The most intuitive encoding method is dividing the earth latitude and longitude into x and y intervals respectively, then forming an x-*by*-y grid. Then for each recording clip, the geographical information can be expressed as an $x \times y$ length one-hot vector with a single 1. However, it leads to two critical problems: 1) the vector has a high sparsity which impedes the model to learn; 2) consequently, the lengthy vector will largely increase the parameters in fully connected layer. Therefore, to tackle above disadvantages, we use one x length vector for latitude and another y length one for longitude, then concatenating together as an $x + y$ one-hot vector with two 1 values.

E. CNN-based Multi-model Architecture

Considering the real application scenario of BIRD system, apart from the acoustic signals, there is another important and accessible information that could contribute to birdcall identification: geographic coordinates. The distribution of different bird species is very regular because certain species will only appear in habitable geographic region(s) by their nature. In term of accessibility, GPS is a satellite-based positioning system that does not require any internet connection to obtain current coordinate, and it is widely equipped on embedded devices. Therefore, we propose a multi-model algorithm that based on the data fusion between acoustic and geographic information, aiming to improve the classification performance compare to single-model identification [15]–[17].



(1) (2) (3) (4)

Fig. 2: Left hand side is waveform graphs and right hand side is spectrograms for every pair. (1), (2) are two 5-second clips of class 10 (amewig) and (3), (4) are from class 7 (amepip).

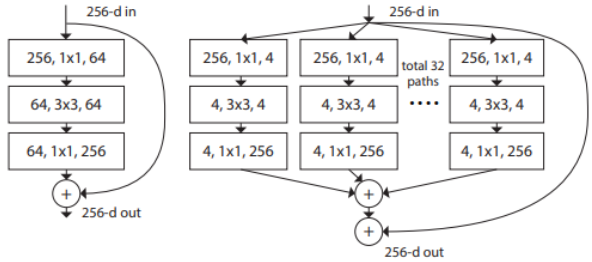


Fig. 3: Original residual structure (left) and ResNeXt structure (right). Obtained from [23].

The architecture of proposed CNN-based multi-model method is shown in Figure 4. Initially, sampled birdcall is transformed into spectrogram, and the coordinates information of the corresponding birdcall is encoded into an one-hot vector. On the strength of CNN structures discussed in previous section, the core structure of ResNet is preserved as the backbone. After that, the output of backbone is concatenated with the one-hot vector, then feeding into a fully connected layer to predict the species eventually.

IV. Experiments

A. Dataset and Pre-processing

Our dataset is obtained from the BirdCLEF2021 competition on Kaggle [27] which is an online machine learning competition platform, and the data originally contributed by xeno-canto [28], one of the biggest birds sounds sharing website around the world. There are 62874 birdcall files totally, and every recording file has a rating between 0 to 5 depending on their quality. Generally, low rating (< 4) recordings exist at least one of the following problems: birdcall is too weak to be noticed; background noise is the primary sound source; or the majority of the recording is silence. In application scenario of our sensors, bird calls will have comparatively higher quality, because users only start to record bird calls when they hear them clearly and soundly. Therefore, low rating (< 4) samples are abandoned, and 38226 bird calls recordings in 397 different species remain. The distribution of bird species is shown in Figure 5, which reflects that basically this is a balanced classification task.

As for the geographical coordinates, followed by convention, we use positive/negative values to denote East/West Longitude and North/South Longitude respectively. Since birds are more sensitive to latitude rather than longitude, we set the intervals as $10°$ and $15°$ respectively. In addition, by observing the dataset, it is noticeable that all birdcall data are sampled between $-60°$ to $80°$ latitude, thus we narrow down the range from $(-90, 90)$ to $(-60, 80)$. After applying one-hot encoding discussed in Section III-D, then we gain a 38-digit vector that will be concatenated with the backbone's output.

After pre-processing, those recordings are divided into 5-second clips, converted into waveform graphs, and then waveform graphs are transformed into 422247 128×128 spectrograms by FFT algorithm mentioned in Section III-B in total. We mainly utilize the Librosa [29] python package to process the data.

1) Noise reduction: Noise reduction techniques are widely discussed and applied in previous sound identification or acoustic classification works [11]–[13], [15], [30]. Basically, their motivation for de-noising can be attributed as following: deleting the silence part in recordings as they are not-of-interest; removing the background sound (rain dropping, river flowing, wind blowing, human walking, etc.); discarding samples with outliers. However, is this noise reduction necessary

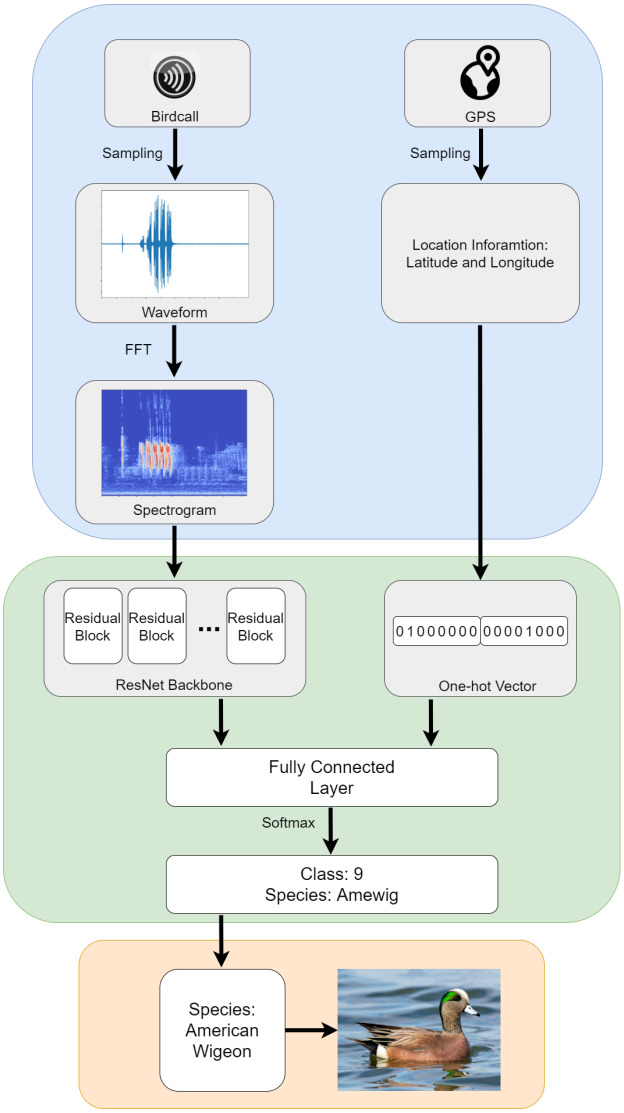


Fig. 4: Proposed CNN-based multi-model network architecture. The bird image is from eBird website [26].

for our proposed BIRD system?

Briefly, the answer is not. First of all, the BIRD is primarily driven by a CNN, and generally people will manually adding noises on training set to increase the generalization ability and robustness for image-related tasks, such as rotating, adding Gaussian noises, cropping, etc. Therefore, keeping the inherent noises in recordings is beneficial for the achieving decent performance on variant scenarios. Secondly, although it is feasible to de-noise on both training and application stages, to alleviate the computation pressure on embedded devices at edge, unnecessary operations are better to be removed. Above all, there is no noise reduction step in our pipeline.

B. criterion

Confusion matrix is the most common way to evaluate the performance of a prediction or classification model.

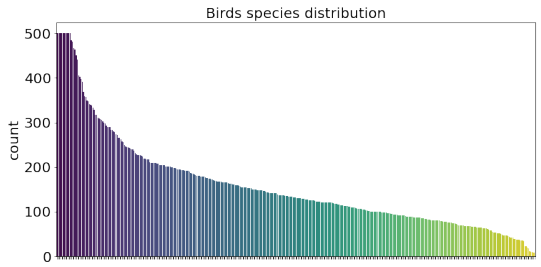


Fig. 5: The birds species distribution in the dataset after pre-processing. Species names on x-axis are omitted for better visualization.

For multi-class task, the recall and precision are computed slightly different from binary prediction. Assuming there are n classes totally (C_1 to C_n), and for C_k, we can get:

$$Precision_k = \frac{TP_k}{TP_k + \sum_{m(m\neq k)}^{n} FP_{mk}} \tag{3}$$

$$Recall_k = \frac{TP_k}{TP_k + \sum_{m(m\neq k)}^{n} FP_{km}} \tag{4}$$

And the global accuracy can be obtained by:

$$Accuracy = \frac{\sum_{m=1}^{n} TP_m}{\sum_{m=1}^{n} TP_m + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}(FP_{ij})} \tag{5}$$

In addition, to evaluate the general predicting ability of our model over all classes, $F1$-Score is also introduced as a metric to assess the performance:

$$F1\text{- Score} = \frac{1}{n}\sum_{m=1}^{n}\frac{2 \times Precision_m \times Recall_m}{Precision_m + Recall_m} \tag{6}$$

Above $F1$-Score is technically the macro-$F1$-Score, because this is a balanced classification task. Apart from that, we also introduce top-3 and top-5 accuracy as indicators to reflect the performance of a model. Top-k accuracy means that if any of the top k highest probability answers matches the target label, then it will be considered as a classification correct. Thus, Equation 5 can be interpreted as the formula of top-1 accuracy. Nowadays top-k ($k > 1$) accuracy is gradually becoming a conventional measurement for classification task, especially in CV field.

C. Results comparison

In this section, we firstly test and evaluate the performance over different CNN structures on a real dataset mentioned in Section IV-A. Considering the limitation of computational resource and the efficiency requirement in real application scenario, we deliberately restrict the training phase in a fixed 10 epochs rather than until the weights of network completely converge, to simulate the real condition. Generally speaking, a deeper network is expected to have a better performance because of more trainable weights, and consequently, it leads to a larger model size and longer training period to achieve

	ResNet18	ResNet50	ResNeXt50	Ours
Performance				
Top-1 Acc	0.834	0.781	0.754	**0.849**
Top-3 Acc	0.868	0.825	0.807	**0.882**
Top-5 Acc	0.899	0.863	0.846	**0.913**
$F1$-score	0.673	0.557	0.547	**0.749**
Efficiency				
Runtime (s / epoch)	**849.1**	1828.69	1633.9	1521.9
Amount of trainable parameters	**11,380,173**	24,321,485	23,793,357	11,440,899
Model size (MB)	**64.11**	186.55	209.10	65.12

TABLE I: "Acc" is the abbreviation of Accuracy. Runtime data are collected from the experiments conducted on the Desktop in Table II.

Device	Nano	Desktop
OS	Ubuntu 18.04 LTS 64-bit	Window 10 Pro 64-bit
CPU	ARMv8 Processor rev 1 (v8l) × 4	Intel i7-8700@ 3.20GHz
GPU	NVIDIA Tegra X1 (nvgpu)/integrated	NVIDIA GeForce RTX 2070
Pytorch Ver.	1.6.0	1.6.0
CuDA Ver.	10.2	10.1

TABLE II: Hardware and software settings of the simulation experiment.

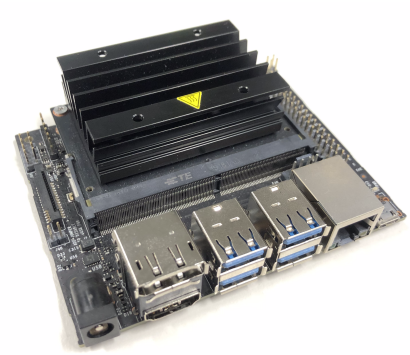


Fig. 6: Jetson Nano.

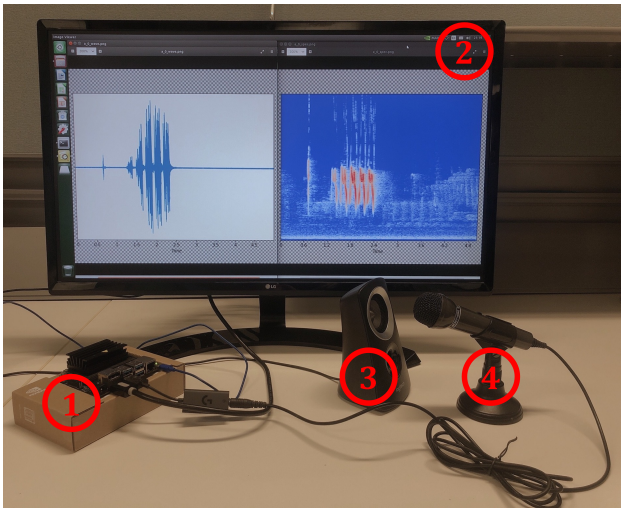


Fig. 7: Simulation experiment overview. Device 1 is the Nano, device 2 is the displaying device, device 3 is the mimic birdcall, and device 4 is the sound collection device.

desired performance. Thus, there is a trade-off between better performance and higher efficiency.

As shown in Table I, it clearly illustrates that ResNet18 has the best performance on all indicators under the limited training condition, especially the top-1 and top-5 accuracy achieves 83.4% and 89.9% respectively, which is surprising performance on a a 397 classification task. Also, ResNet18 model has the smallest size and spends the shortest training time, and it actually reaches convergence around epoch 8. Above all, we believe that ResNet18 is the optimal option to be the backbone of our multi-model network.

Then, comparing our proposed CNN-based multi-model network with the original ResNet18, in term of efficiency, although the training runtime almost double, the amount of parameters and model size basically keep approximately at the same level. As for the performance, our work surpasses ResNet18 on all indicators, and especially the $F1$-score greatly improves from 0.673 to 0.749, reflecting a much better classification ability across all bird species.

D. Simulation Experiment

In this part, to prove the feasibility of our proposed CNN-based multi-model network on embedded devices, especially the real-time processing capacity, it is necessary to conduct a simulation experiment. Therefore, we choose Jetson Nano (Figure 6), which is a popular embedded device presented by NVIDIA, to act as the portable sensor, and a desktop equipped with GPU to play the role of central computer in Figure 1. The specific settings are shown in Table II.

As shown in Figure 7, we use common accessories to simulate input and output devices, to build up the simulation experiment environment. After training on "central computer", we firstly transfer the network to the Jetson Nano, and then run the testing set on our proposed network in it. The experiment shows that this embedded device can convert four 5-second clips into corresponding spectrograms or predict 40 spectrograms per second via our CNN-based multi-model network. Above all, the feasibility of our proposed CNN-based multi-model network on embedded devices is verified.

V. Conclusion and future work

In conclude, we present a novel CNN-based multi-model network to identify birdcall on a real large dataset, and our results show that this method achieves improved performance on classification task compared to previous models. In addition, we also propose BIRD system as a complete IoT solution for applying real-time birdcall identification model on embedded devices.

Potential future work includes: exploring the affect of different parameters on generating spectrograms to the CNN performance; exploring the possibility to introduce attention mechanism into out current model; and for utility's sake, encapsulating the BIRD system as an mobile APP.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE internet of things journal 3 (5) (2016) 637–646.

[2] W. Shi, S. Dustdar, The promise of edge computing, Computer 49 (5) (2016) 78–81.

[3] M. Verhelst, B. Moons, Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to iot and edge devices, IEEE Solid-State Circuits Magazine 9 (4) (2017) 55–65. doi:10.1109/MSSC.2017.2745818.

[4] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (iot) security, IEEE Communications Surveys Tutorials 22 (3) (2020) 1646–1685. doi:10.1109/COMST.2020.2988293.

[5] M. Song, K. Zhong, J. Zhang, Y. Hu, D. Liu, W. Zhang, J. Wang, T. Li, In-situ ai: Towards autonomous and incremental deep learning for iot systems, in: 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2018, pp. 92–103. doi:10.1109/HPCA.2018.00018.

[6] J. Tang, D. Sun, S. Liu, J.-L. Gaudiot, Enabling deep learning on iot devices, Computer 50 (10) (2017) 92–96. doi:10.1109/MC.2017.3641648.

[7] H. Li, K. Ota, M. Dong, Learning iot in edge: Deep learning for the internet of things with edge computing, IEEE Network 32 (1) (2018) 96–101. doi:10.1109/MNET.2018.1700202.

[8] M. Pishgar, F. Karim, S. Majumdar, H. Darabi, Pathological voice classification using mel-cepstrum vectors and support vector machine, arXiv preprint arXiv:1812.07729 (2018).

[9] S.-H. Fang, Y. Tsao, M.-J. Hsiao, J.-Y. Chen, Y.-H. Lai, F.-C. Lin, C.-T. Wang, Detection of pathological voice using cepstrum vectors: A deep learning approach, Journal of Voice 33 (5) (2019) 634–641.

[10] J. Li, W. Dai, F. Metze, S. Qu, S. Das, A comparison of deep learning methods for environmental sound detection, in: 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP), IEEE, 2017, pp. 126–130.

[11] S. Balemarthy, A. Sajjanhar, J. X. Zheng, Our practice of using machine learning to recognize species by voice, arXiv preprint arXiv:1810.09078 (2018).

[12] T. Oikarinen, K. Srinivasan, O. Meisner, J. B. Hyman, S. Parmar, A. Fanucci-Kiss, R. Desimone, R. Landman, G. Feng, Deep convolutional network for animal sound classification and source attribution using dual audio recordings, The Journal of the Acoustical Society of America 145 (2) (2019) 654–662.

[13] W. Xu, X. Zhang, L. Yao, W. Xue, B. Wei, A multi-view cnn-based acoustic classification system for automatic animal species identification, Ad Hoc Networks 102 (2020) 102115.

[14] J. A. Mortimer, T. C. Greene, Investigating bird call identification uncertainty using data from processed audio recordings, New Zealand journal of ecology 41 (1) (2017) 126–133.

[15] N. Priyadarshani, S. Marsland, I. Castro, Automated birdsong recognition in complex acoustic environments: a review, Journal of Avian Biology 49 (5) (2018) jav–01447.

[16] M. Sankupellay, D. Konovalov, Bird call recognition using deep convolutional neural network, resnet-50, in: Proceedings of ACOUSTICS, Vol. 7, 2018.

[17] R. Mohanty, B. K. Mallik, S. S. Solanki, Automatic bird species recognition system using neural network based on spike, Applied Acoustics 161 (2020) 107177.

[18] Z.-Q. J. Xu, Y. Zhang, Y. Xiao, Training behavior of deep neural network in frequency domain, in: International Conference on Neural Information Processing, Springer, 2019, pp. 264–274.

[19] H. Wang, X. Wu, Z. Huang, E. P. Xing, High-frequency component helps explain the generalization of convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8684–8694.

[20] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[23] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[24] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[25] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[26] ebird, `https://www.ebird.org/`, accessed: 2021-05-15.

[27] Kaggle, `https://www.kaggle.com/`, accessed: 2021-04-30.

[28] xeno-canto, `https://www.xeno-canto.org/`, accessed: 2021-04-30.

[29] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python, in: Proceedings of the 14th python in science conference, Vol. 8, Citeseer, 2015, pp. 18–25.

[30] J. Luque, D. F. Larios, E. Personal, J. Barbancho, C. León, Evaluation of mpeg-7-based audio descriptors for animal voice recognition over wireless acoustic sensor networks, Sensors 16 (5) (2016) 717.