Report: Bioacoustics Classifiers For bird detection in audio Recordings.

Herman Franclin Tesso Tassang
AI Master Student
African Institut of Mathematical Science
Muizenberg, South Africa
email: herman@aims.ac.za

***Abstract*—Extracting species calls from passive acoustic recordings is a commonly used methods in ecological analysis for studying species within their habitats.This study of animal vocalisation is among other one of the mostly used approach in ecology science for biodiversity preservation purpose, and the process involves handle large volumes of data from passive recorder which can be time consuming for manual extraction process.Deep learning Neural Network have been shown to offer relatively high performance across a range of acoustic classification applications.We present in this paper a well known approaches to detect the presence of bird calls in audio recordings using convolutional Neural networks on Mel spectrograms.**

***Index Terms*—bird calls, convolutional Neural Networks, Mel Spectrograms, acoustic recordings**

I. Introduction

Detecting the presence of bird calls, in audio recordings can serve as a basic step for wildlife and biodiversity monitoring.Our task was to build and train algorithms that predict weather a given recording contains any bird vocalization deploying convolutional Neural networks regardless of the species. In the following, we first describe the data we used for the task before the next section goes into depth regarding the approach used to tackle the problem going from prepossessing to post-processing (linking model prediction to facilitate manual labeling).The last section provides and overview of results obtained jointed by a short conclusion.

II. Data

A. Data Source

The data used (audio) was recorded at different locations of **Intaka Island Reserve, Century City, Capetown** (fig. 1) between 9:00AM-11:30AM by around 26 students , and the device used to record those acoustic data was mainly ***Raspbery Pi ARU*** (fig. 2) recording units.

The audio covers a wide distribution of IntaKa Island locations and includes weather noise, traffic noise and human speech.

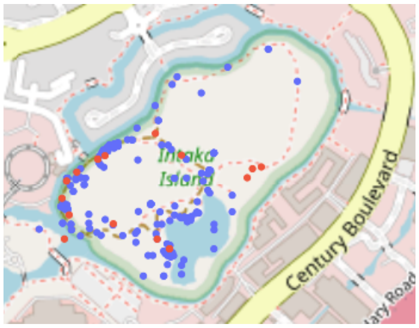


Fig. 1: geolocation of data source (blue=bird heard; red=no bird heard

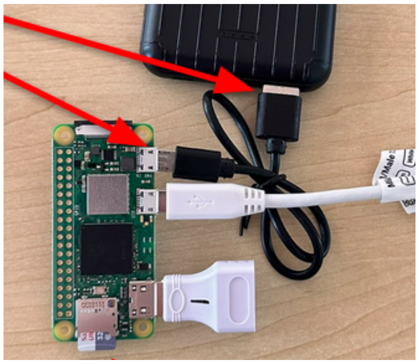


Fig. 2: Raspberry Pi zero ARU setup

B. Data Structure

Each audio file was recorded at a sampling rate of 44100Hz.The data was split into training and test data by specifying different recording locations for test data and train data. Each training example come with an individual manual annotation's if birds are present anywhere in audio label with **'1'**, or no bird present at all, label with **'0'** by inspecting spectrograms and listening to audio using *Sonic Visualizer*.Most of the files are 30s long, but there are exceptions with a duration of up to 10 min.The train data is unbalanced in the sense that bird absence events represent only 27% of the entire training data against 73% of bird presence events.

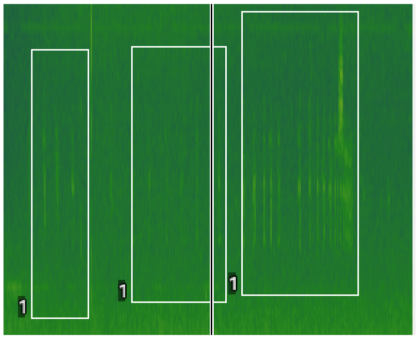


Fig. 3: an annotation example

III. Approach

Our approach to bird vocalization detection challenge deploys feed forwards CNN train on Mel-scaled log-magnitude spectrograms.We build and train two principally different Networks architectures after performing several preprocessing step to address various challenges present by the raw audio data to get them ready for the training.

A. Preprocessing

Each audio recording was downsampled to 22000Hz so that the Nyquist rate (11000) was higher than the maximum frequency (9000Hz) of bird calls. To construct the fixed-length inputs required CNNs, we divided each 30s recording into segments with window length of 3s chosen as a median of different vocalisations duration.On the basis of preliminary investigations, we converted each audio segment into a mel-scale spectrogram using a smoothly Hann window of length 1024 samples, a hop size of 256 (25% of the window size) and 128 mels frequency bins resulting in spectrogram images of size 128 × 258 pixels compute using Librosa library.We scaled the magnitude logarithmically and operated a MinMax-normalization to Mel-spectrograms to scale the pixels values within the range [0,1].

After processing, our training dataset consisted of 6497 Mel-spectrogram images, and like we mentioned earlier, the vast majority of spectrograms (or audio segments) do not contain any bird calls.To address this issues, we applied a simple augmentation technique to increase the number of bird absence events examples so that we avoid a large class imbalance.

1) Augmentaion: Technique we used is a simple Time-frequency domain augmentation named Spectrogram-shifting , where the spectrogram of an audio signal is shifted by a certain amount by adding a constant offset to the time or frequency axis.

After augmenting the original spectrograms, we obtained 10234 spectrograms examples (5114 bird absence, 5120 bird presence) with their corresponding label that we One-hot encode for the training .Spectrogram images obtained is to be used as an input image to a CNN.

B. Neural Networks Architecture

We considered to kinds of CNN architecture both having as inputs mel spectrogram images constructed from the preprocessed amplitudes.However the images need to be reshape to match the inputs of each model.

1. **2D CNN :**The 2D CNN architecture (see fig. 4a) we used is design to takes an input image with dimension $128 \times 258 \times 1$ (grayscale images), the first two dimensions representing the shape of one spectrogram and the last one representing the depth of the image. It is consisted of 3 convolutional layers each followed by max pooling (to reduce spatial dimensions and extra dominants features) ; one fully connect layers consisted of 20 nodes (units) and an output layer consisted of 2 nodes(units) using "Softmax" activation function to perform the binary classification task.
2. **Transfer Learning:** We build a transfer learning model (see fig. 4b) using the convolutional Neural Network architecture of ResNet50V2 model that has been pre-trained on an ImageNet dataset for image classification.The spectrogram image are adapted as a standard RGB image with tree channels (128 × 258 × 3) and are passed through the pre-trained layers of ResNet50V2 for feature extraction; On top of ResNet50V2, we add one fully connect layer(consisted of 20 nodes) followed by an output layer (2 units , with "Softmax" activation function) to adapt the model to our classification task.

C. Training

Training is done by stochastic gradient descent on mini-batches of 32 examples, using the ADAM update rule with a learning rate of 0.001 (for 2D CNN model) and 0.0001 (for Transfer learning model).Each model was trained for 5 epoch. Model hyperparameters (filters in each layers, number of Dense and convolutional layers, nodes in each Dense layers, kernel size,...) was chosen based on repeated process of turning→runing→inspection. finally, the model has been trained on 50% of the initial training set because of the limited computational resource (RAM).

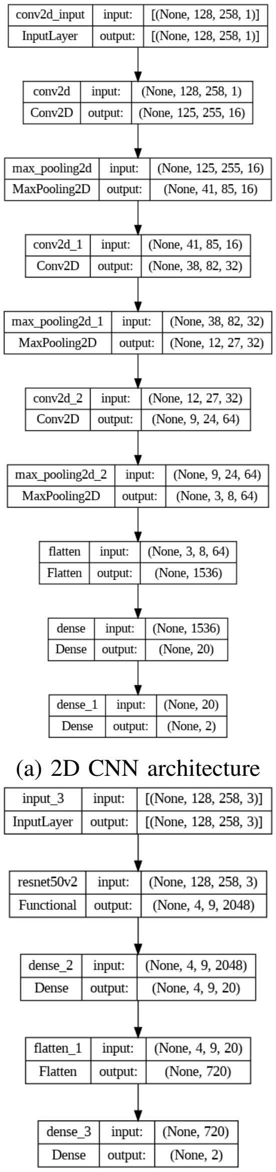
D. Predicting and model evaluation

After training, we preprocessed the testing audio file the same way we did with the training audio file but without any augmentation; made predictions on the test data(spectogram images obtained after preprocessing), computed the area under the ROC curve (AUC) and evaluated the models based on:

- The accuracy
- Recall (sensitivity)
- specificity
- precision

1) Post-processing:Visualize predictions: We also assess the performance of our models by predicting on an entire audio file and visually inspecting the predictions as a bounding boxes using Sonic visualizer.The process require some processing of the audio file, but to over-simplified, we followed the steps below for a given testing audio file:

- predict on the entire file in a sliding window manner assuming a 3s second input window.


conv2d_input | InputLayer | input: [(None, 128, 258, 1)] | output: [(None, 128, 258, 1)]
conv2d | Conv2D | input: (None, 128, 258, 1) | output: (None, 125, 255, 16)
max_pooling2d | MaxPooling2D | input: (None, 125, 255, 16) | output: (None, 41, 85, 16)
conv2d_1 | Conv2D | input: (None, 41, 85, 16) | output: (None, 38, 82, 32)
max_pooling2d_1 | MaxPooling2D | input: (None, 38, 82, 32) | output: (None, 12, 27, 32)
conv2d_2 | Conv2D | input: (None, 12, 27, 32) | output: (None, 9, 24, 64)
max_pooling2d_2 | MaxPooling2D | input: (None, 9, 24, 64) | output: (None, 3, 8, 64)
flatten | Flatten | input: (None, 3, 8, 64) | output: (None, 1536)
dense | Dense | input: (None, 1536) | output: (None, 20)
dense_1 | Dense | input: (None, 20) | output: (None, 2)
input_3 | InputLayer | input: [(None, 128, 258, 3)] | output: [(None, 128, 258, 3)]
resnet50v2 | Functional | input: (None, 128, 258, 3) | output: (None, 4, 9, 2048)
dense_2 | Dense | input: (None, 4, 9, 2048) | output: (None, 4, 9, 20)
flatten_1 | Flatten | input: (None, 4, 9, 20) | output: (None, 720)
dense_3 | Dense | input: (None, 720) | output: (None, 2)

(a) 2D CNN architecture

(b) Transfer learning architecture

Fig. 4: Side by side images

- store the predictions in a dataframe and get the index position where there was a bird presence detected.
- group consecutive detection together.
- generates an ".svl " file which can be input into Sonic Visualizer.
- visually inspect the prediction (bounding boxes) into Sonic visualizer to assess the performance of the model over the audio file.

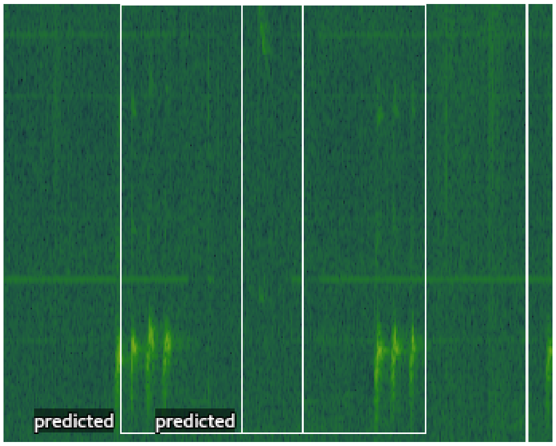
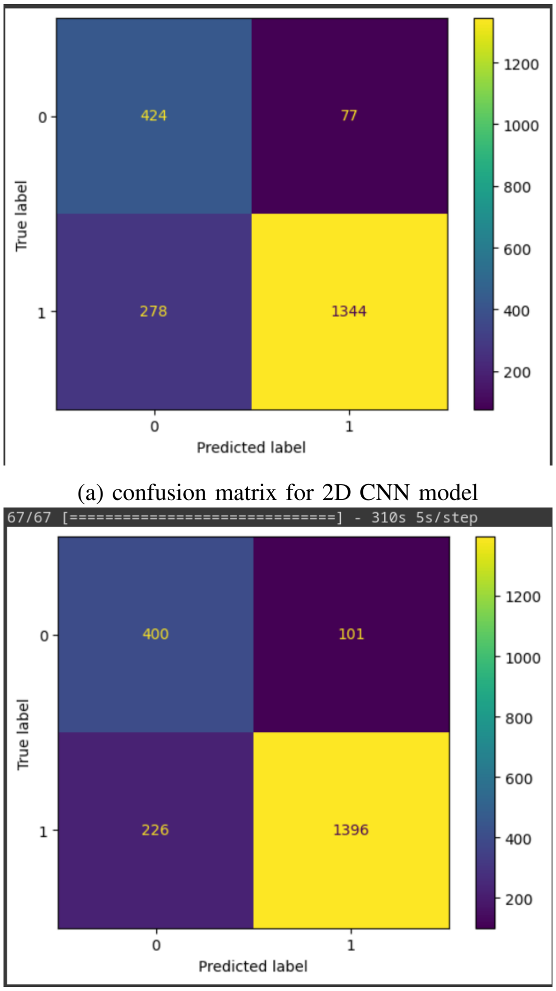


Fig. 5: e.g prediction visualize as bounding box.

IV. Results

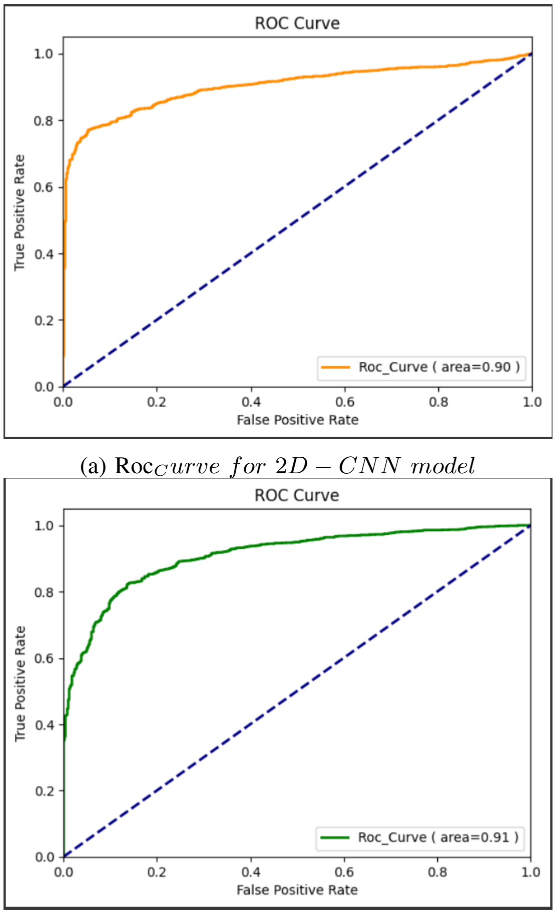
The entire test data after required processing consisted of 2123 labeled spectrogram examples.After predictions, we obtained the following results:



(a) confusion matrix for 2D CNN model

```
67/67 [==============================] - 310s 5s/step
```

(b) confusion matrix for transfer learning model



(a) Roc$_{C}urve\ for\ 2D-CNN\ model$

(b) Roc$_{C}urve\ for\ transfer\ learning\ model$

TABLE I: summary of average performance across 2123 segments of test recordings file.

Evaluation metrics	**2D-CNN**	**Transfer learning**
AUC	0.90	0.91
Accuracy	83%	85%
Precision	94,58%	93,25%
Recall	82,86%	86,06%
Specificity	84,63%	79,84%

1) Discussion: The best performance approach was achieve with the transfer learning model regarding both the *Accuracy* and the *sensitivity*(recall) , however, the 2D-CNN improve over the later when considering *Precision* and *Specificity*. The whole performance of the two model on the testing audio files is not that bad, but it is not good enough.We can probably achieve a better performance by reviewing some details of the audio preprocessing given that this step can have a crucial impact on the result.In addition, the choice of frequency for downsampling the audio signal to 22000Hz can be responsible for a significant degradation of bird detection performance which can potentially lowering AUC.

V. Conclusion

Our work was about, build a classification algorithm to identify bird calls in audio recordings.We have presented two deep learning approaches (2-D CNN and Transfer learning) all of which use Neural Networks on spectrogram.Despite using different Network architectures, they perform nearly similarly.The overall performance of both model can be improved by applying a rigorous preprocessing of the training audio files.

Acknowledgment

We thank African Institut of Mathematical Science-South Africa for logistical assistance during the process. We also thank the following local member of ”*machine learning for ecology research group*” in AIMS-South Africa who oriented and provided us with the core knowledge in using deep learning in ecological science: Emmanuel Dufourq, Lorène Jeantet, Matthew Van. D.B.