ELEC5305 Project Report

Zhuochen Tang (ztan6808@sydney.edu.au)

School of Electrical and Computer Engineering

The University of Sydney, NSW, Australia

A preliminary pipeline for the bird call classification task has been implemented ,aimed at familiarizing with the entire process, identifying potential challenges, and establishing a foundational workflow for future enhancements.

The dataset used comprises audio recordings from Xeno-canto, featuring 10 commonly found bird species in Britain, listed in the table below:

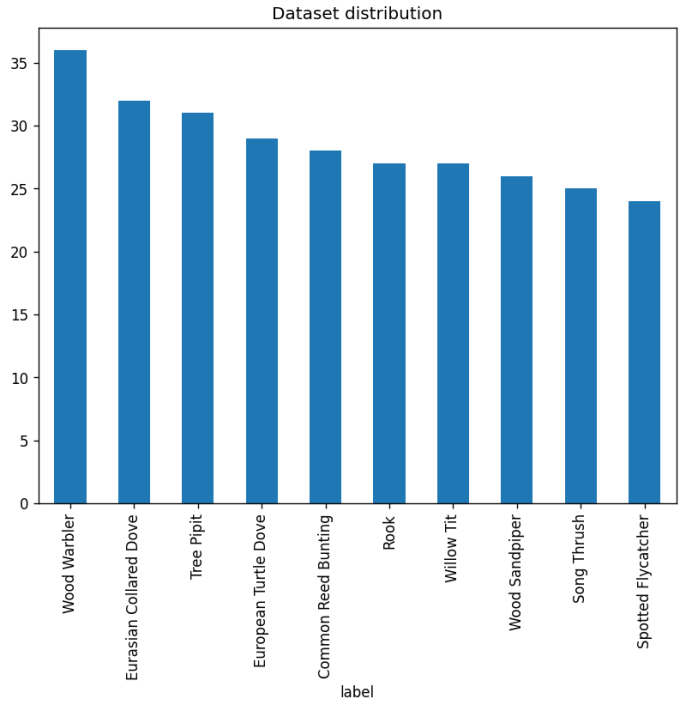

Fig. 1. Distribution of Dataset

Each audio sample consists of a 5-second segment extracted from the original dataset to standardize the length for further processing. In total, there are 285 samples, which are divided into a training set and a test set in an 8:2 ratio, resulting in 228 samples in the training set and 57 samples in the test set.

The audio feature used in the current process is the Mel-frequency cepstral coefficients (MFCCs), which are derived from the Mel spectrum after applying a discrete cosine transform (DCT). While the Mel spectrum displays the frequency distribution, MFCC represents the spectral envelope, making it more suitable for further processing with computer vision techniques. By reducing to a smaller set of cepstral coefficients, MFCCs provide a compact

representation that filters out unimportant details like noise. Based on the Mel frequency scale, which mimics the non-linear human perception of frequency, MFCCs are particularly well-suited for analyzing bird calls within specific frequency ranges. MFCCs are widely used in audio classification tasks, as discussed previously. An example of the extracted data features is shown below:

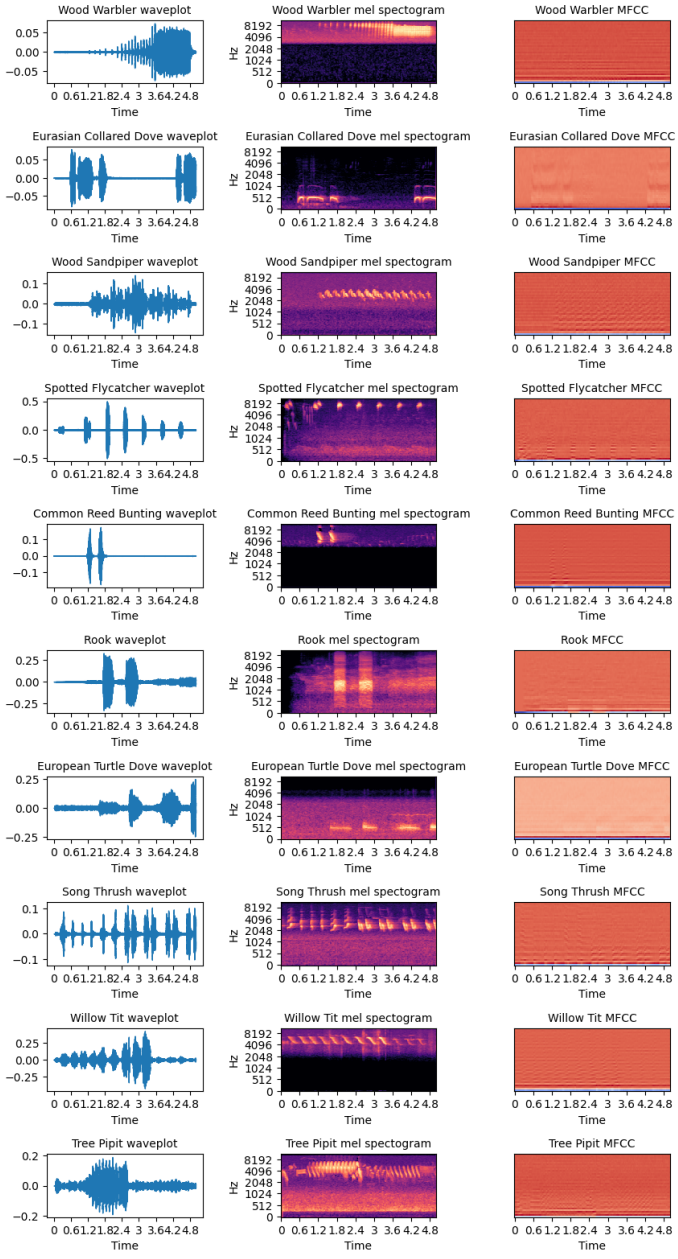

Fig. 2. Features Extracted(waveplot, mel spectrum, MFCC)

Each extracted MFCC is a 2D array of size 40*216, which can be treated as a single-channel image. Given this structure, a convolutional neural network (CNN)—a classic model for image classification in computer vision—is applied to process the data. After experimenting with different layer combinations, the following model has proven effective for this task:

TABLE I
MODEL SUMMARY

Layer (type)	**Output Shape**	**Parameters**
conv2d_4 (Conv2D)	(None, 39, 215, 16)	80
max_pooling2d_4 (MaxPooling2D)	(None, 19, 107, 16)	0
dropout_4 (Dropout)	(None, 19, 107, 16)	0
conv2d_5 (Conv2D)	(None, 18, 106, 32)	2,080
max_pooling2d_5 (MaxPooling2D)	(None, 9, 53, 32)	0
dropout_5 (Dropout)	(None, 9, 53, 32)	0
conv2d_6 (Conv2D)	(None, 8, 52, 64)	8,256
max_pooling2d_6 (MaxPooling2D)	(None, 4, 26, 64)	0
dropout_6 (Dropout)	(None, 4, 26, 64)	0
conv2d_7 (Conv2D)	(None, 3, 25, 128)	32,896
max_pooling2d_7 (MaxPooling2D)	(None, 1, 12, 128)	0
dropout_7 (Dropout)	(None, 1, 12, 128)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290
Total parameters:	44,602 (174.23 KB)	
Trainable parameters:	44,602 (174.23 KB)	
Non-trainable parameters:	0 (0.00 B)	

The designed model is relatively lightweight, with fewer parameters to train compared to larger networks like ResNet-50 [1], which contains millions of parameters. This choice is due to the relatively small dataset used in this project, where large networks could lead to overfitting, despite their potential for higher performance. The model consists of 4 convolutional layers to extract high-level features, with dropout layers added to enhance network stability. It was trained with a learning rate of 0.001, using categorical cross-entropy loss and the Adam optimizer, over 50 epochs. The training results are as follows:

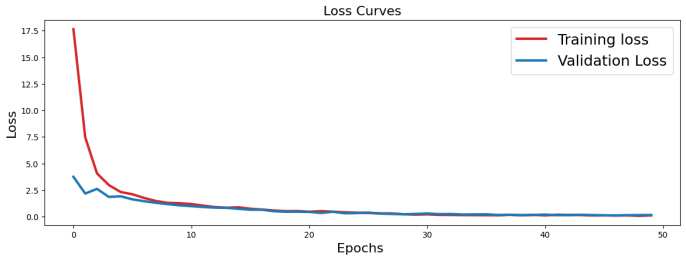

Fig. 3. Loss Curve

The loss curve demonstrates a consistent decrease over the epochs, with the training and validation loss closely aligned, indicating an absence of overfitting. The curve flattens with a near-zero slope after 50 epochs, signifying that the model has reached convergence.

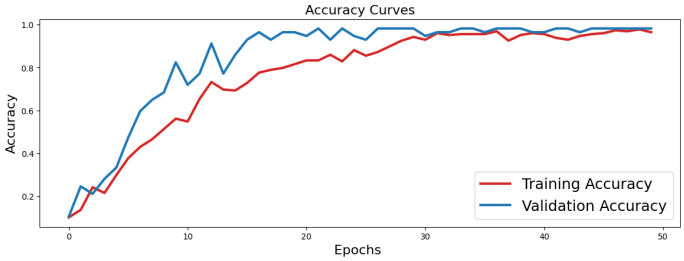

Fig. 4. Accuracy Curve

The accuracy curve shows a continuous increase across epochs for both the training and validation sets. An interesting observation is that the validation accuracy is slightly higher than the training accuracy. This could be due to the dropout layers, as only a subset of neurons is used during training, while all neurons are active during validation, potentially leading to higher accuracy on the validation set. Additionally, this may result from a slight imbalance in the distribution of classes between the training and testing sets, as they were divided randomly. Such imbalances can have a greater impact on a small dataset, like the one used in this project, where the species ratio may vary between sets. However, with both sets reaching an accuracy of 98%, the model appears to have converged effectively.

The confusion matrix for the test set is shown below:

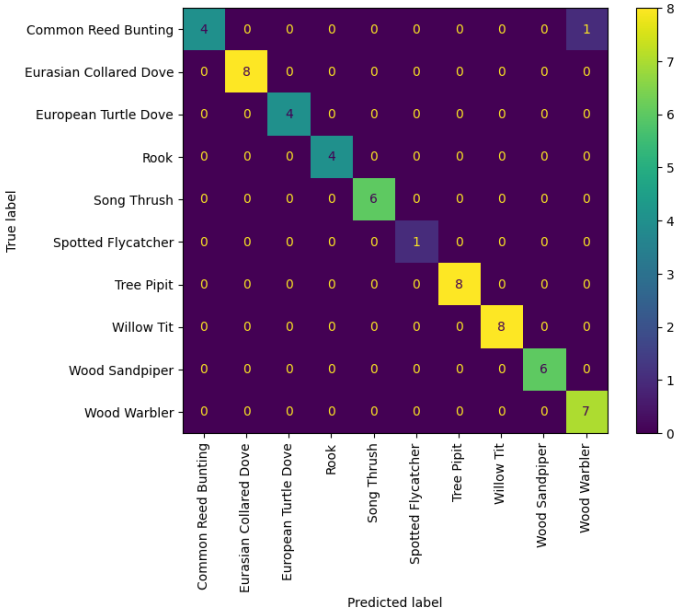

Fig. 5. Confusion Matrix

The classification performance parameters are presented in the table below:

TABLE II
CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Common Reed Bunting	0.80	1.00	0.89	4
Eurasian Collared Dove	1.00	1.00	1.00	8
European Turtle Dove	1.00	1.00	1.00	4
Rook	1.00	1.00	1.00	4
Song Thrush	1.00	1.00	1.00	6
Spotted Flycatcher	1.00	1.00	1.00	1
Tree Pipit	1.00	1.00	1.00	8
Willow Tit	1.00	1.00	1.00	8
Wood Sandpiper	1.00	1.00	1.00	6
Wood Warbler	1.00	0.88	0.93	8
Accuracy			**0.98**	57
Macro Avg	0.98	0.99	0.98	57
Weighted Avg	0.99	0.98	0.98	57

The confusion matrix and classification performance parameters indicate that this small model achieves excellent performance on a limited dataset with minimal training data. The results demonstrate that the combination of MFCC and CNN is effective for bird call classification, balancing accuracy and computational efficiency. This work establishes a solid foundation for future developments building on these achievements.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.