Real Time Bird Call Classifier

Iordan Tchaparov
&
Kavilan Nair

University of the Witwatersrand

School of Electrical & Information Engineering

ELEN4012 - Lab Project

September 13, 2018

Overview

1. Introduction
2. System Components
3. Data Acquisition
4. Feature Extraction
5. CNN Architecture
6. Results
7. iOS Application
8. API Server
9. Future Recommendations
10. Conclusion

Introduction

- Project aims to develop a classifier to identify bird species through their calls
- The classifier must have accuracy of 80% or higher
- Needs to deliver classification results in real time
- Classification results, audio recordings and geographical data must be saved to a database

System Components

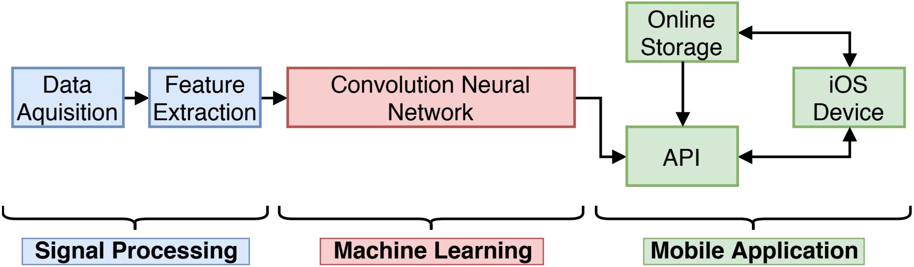


Figure: System components of the real time bird call classifier

Data Acquisition

- Data is needed to train any Artificial Neural Network (ANN)
- Xeno-Canto is an open platform database containing bird calls from around the world
- A total of 590 South African bird species recordings were obtained
- Due to processing power limitations, 10 bird species are used
- Recordings were cut and processed through mean analysis
- Resulted in a total of 3560 sound files to be used to train an ANN

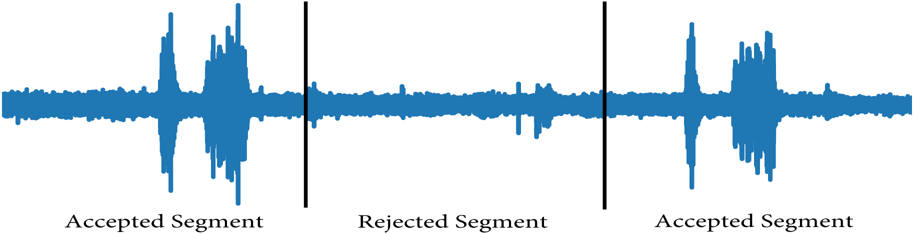


Figure: Mean analysis of an audio recording

Feature Extraction

- The bird call recordings were filtered from 20 Hz to 12 000 Hz to eliminate noise
- Mel-Frequency Cepstral Coefficients (MFCCs) were used as the features that the Neural Network must learn
- Only the first 13 MFC coefficients are computed and used
- Images of the MFCC plots were generated and saved
- The MFCC extraction is handled through the use of the Python library, Librosa

MFCC Key Equations

Discrete Fourier Transform

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j2\pi kn}{N}} \tag{1}$$

Mel-Scale Transformation

$$M(f) = 1127 \cdot ln(1 + \frac{f}{700}) \tag{2}$$

Discrete Cosine Transform with logarithm

$$MFCC = \sum_{n=0}^{N-1} log(X[k]) \cdot cos\left[\frac{\pi}{N}(n + \frac{1}{2})k\right] \tag{3}$$

CNN Architecture - Layers

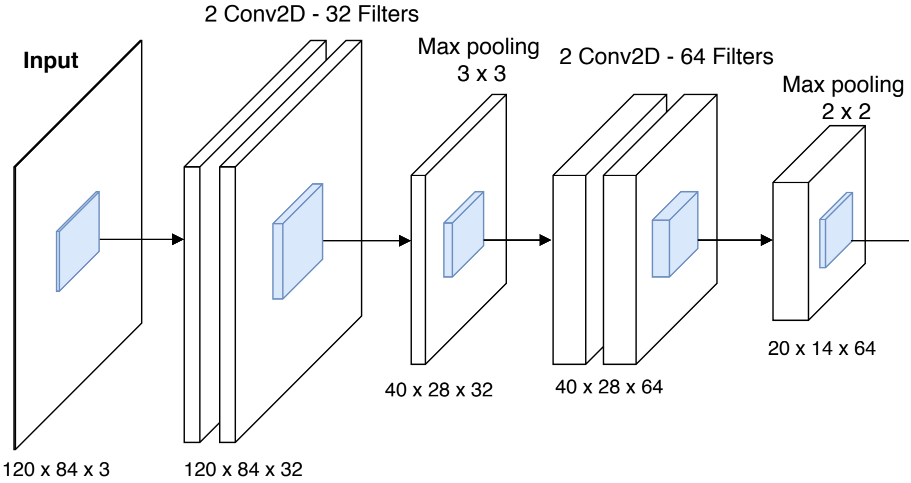


Figure: Layers 1 to 7 of the CNN

CNN Architecture - Layers

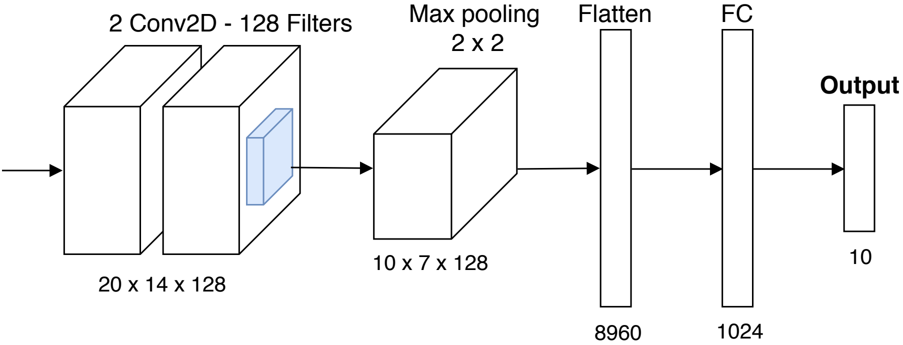


Figure: Layers 8 to 13 of the CNN

CNN Architecture - Hyperparameters

- Filters had size of 3 x 3
- Dropout of 50% utilized at each layer except for Dense 1024 layer, which uses 80%
- Rectified Linear Unit function (Relu) is used as the activation functions of each layer
- Adam Optimizer chosen as the optimizer
- Categorical Cross Entropy chosen as the loss function
- Softmax Activation function used for output layer
- Batch size of 5
- Learning Rate of 0.00001
- Trained over 3000 epochs

Results - Accuracy

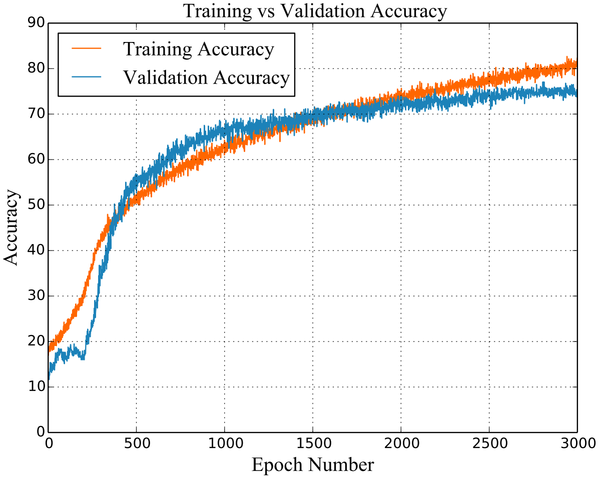


Figure: Training and validation accuracy over 3000 epochs

Results - Loss

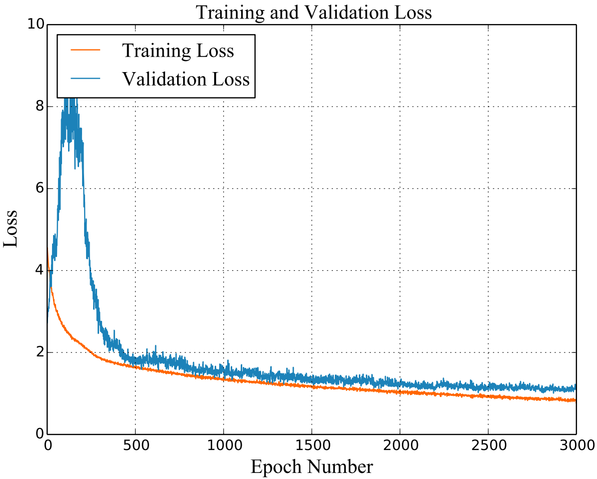
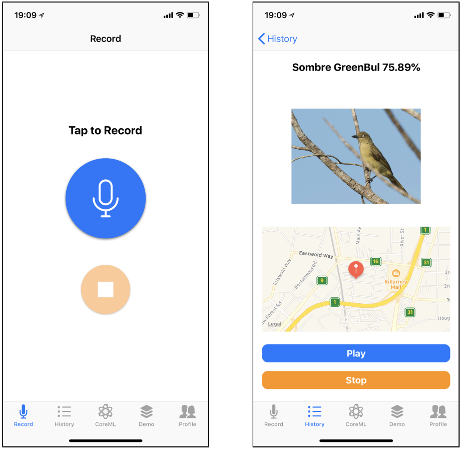


Figure: Training and validation loss over 3000 epochs

iOS Application

Functionality

- Utilizes devices built in microphone
- GPS for location data
- Firebase Realtime Database
- Firebase Storage Bucket



Sombre GreenBul 75.89%

(a) Record

(b) Classification

Figure: iOS application screenshots

API Server

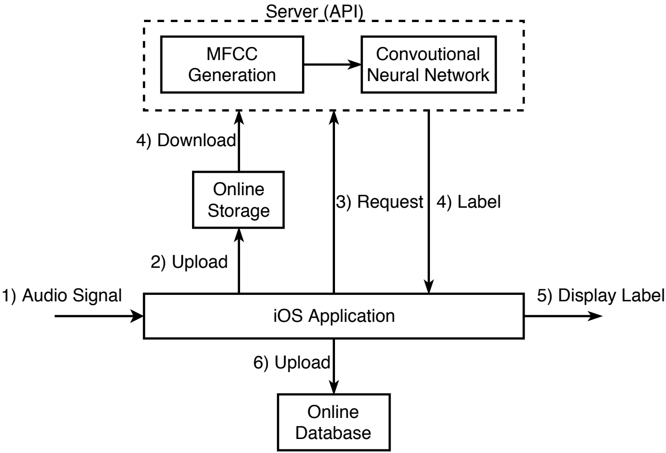


Figure: Mobile application data flow

Future Recommendations

- Increase the number of bird species to classify
- Perform data processing on the mobile device
- Ability to identify if a bird call is present
- Classify multiple birds in the same audio clip
- Validate and verify the quality of the data
- Develop a training pipeline that make use of user classified sounds to train and enhance the model

Conclusion

- CNN classified 10 South African birds with an accuracy of 77%
- Mobile application successfully communicated with classifier on a server and performs classification in real time
- Classification and location information is saved to database

The End

Questions?