UNIVERSITY OF CALIFORNIA SAN DIEGO

Classification of Overlapping Bird Songs Using Spectogram-keypoint Based Analysis

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Data Science

by

Dylan Kailong Stockard

Committee in charge:

David Danks, Chair
Ryan Kastner
Sam Lau
Janine Tiefenbruck

2024

The Thesis of Dylan Kailong Stockard is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

EPIGRAPH

True ease in writing comes from art, not chance,
As those move easiest who have learn'd to dance.
'T is not enough to no harshness gives offence,—
The sound must seem an echo to the sense.

Alexander Pope

You write with ease to show your breeding,
But easy writing's curst hard reading.

Richard Brinsley Sheridan

Writing, at its best, is a lonely life. Organizations for writers palliate the writer's loneliness, but I doubt if they improve his writing. He grows in public stature as he sheds his loneliness and often his work deteriorates. For he does his work alone and if he is a good enough writer he must face eternity, or the lack of it, each day.

Ernest Hemingway

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge David Danks for his support as the chair of my committee. His assistance in the thesis process has been valuable and encouraging. This gratitude extends to Janine Tiefenbruck, Ryan Kastner, and Sam Lau for giving me their time by serving on my defense committee.

Many thanks to Sean Perry for his guidance through both technical and domain knowledge in this thesis, he allowed me to be creative in my approach while staying on a clear path.

I could not have completed this thesis without the help of Jessica Peurifoy, who made sure I was completing each step of my thesis preparation so I could have this final product here today.

ABSTRACT OF THE THESIS

Classification of Overlapping Bird Songs Using Spectogram-keypoint Based Analysis

by

Dylan Kailong Stockard

Master of Science in Data Science

University of California San Diego, 2024

David Danks, Chair

Wildlife biologists often rely on classification models to automate the labelling process of audio data that they collect in the wild. Bird populations are often a subject of interest in audio data, and tens to hundreds of species can be tracked in population-dense areas such as the Amazon rainforest. However, these population-dense environments also pose the issue of heavy noise, distortion, and overlapping bird calls to be classified. By transforming audio signals into their spectrogram form in the frequency domain, we can identify keypoints in the spectrogram signal by passing a sliding window over the signal and finding the local maxima. Describing an audio signal by its keypoint-spectrogram can result in higher classification accuracy for overlapping bird calls when passed as input to a CNN than if the entirety of the audio spectrogram were to be

used.

Introduction

Long-term remote data collection techniques such as soundscape monitoring and wildlife cameras are becoming more and more viable as our classification software improves to handle such data. Soundscape monitoring is done by deploying a microphone in the field and leaving it on to record for hours on end. This data is then retrieved days or weeks later, and analyzed to identify different animal calls. This can include frogs, wolfs, and deer, but often is used with birds due to their loud calls, as well as the difficulty of tracking them visually. The downside is that all other noises are also recorded, such as foliage rustling, wind, rain, or even cars if there are nearby roads. The audio itself is also often not perfect, as birds can be far away, or make calls simultaneously with other birds and noises. This is frequently the case when monitoring an area such as the Amazon Rainforest, where population density is so high that clear, isolated bird calls are infrequent. In the domain of smart-home devices, the same problem arises. Smart-home devices must detect audio commands even when there can be background noise such as talking, doors closing, outside street noise, or a dog barking. In this domain, keypoint-spectrogram classification has found success [6] as a faster and more accurate solution to the noisy signal problem. However, there are some differences between the domains. For one, the type of audio signal in bird classification is vastly different than that found in smart-home devices. Bird calls generally have greater range in frequency and amplitude, and the acoustics can be distorted to greater extremes due to scope of the environment they're detected from [1]. Smart-home devices also only need to classify a single signal when there is audio overlap since the device only performs one command at a time. Soundscape monitoring may yield multiple overlapping bird calls at one time that all need to be classified.

Chapter 1

Data Processing

The goal of this paper is to evaluate the effectiveness of keypoint-spectrogram based classification techniques on bird audio data. Ideally, we could test directly on soundscape data that represents the data this technology would likely be applied on. However, soundscape data is often unlabelled, and so instead we relied on focal data provided by the online public database xeno-canto. Focal data refers to data that is recorded with the subject directly in front of the recorder. This means that there is a much higher signal-to-noise ratio in focal data, and less audial distortion from the environment. Therefore it is to note that the final model's accuracy is not fully representative of the accuracy in the real-world because the training data is not entirely representative. The xeno-canto database is a crowd-sourced bird audio dataset with recordings from all around the world. Although it is crowd-sourced so it is bound to have errors, it is generally recognized as one of a few primary bird audio sources. Being crowd-sourced data, many audio recordings have inconsistent lengths. Some samples will have a single bird call and end within a few seconds, while others will have long pauses as the call repeats for a total length of a minute. How these time lengths will be standardized will be discussed later. From the xeno-canto database, we downloaded 178 Straight-billed Woodcreeper recordings, 268 Screaming Piha recordings, 313 Squirrel Cuckoo recordings, and 171 White-necked Thrush recordings. These birds were selected based on their abundance of available data, as well as the fact that they all inhabit the Peruvian Amazon Rainforest so it is reasonable to have a situation

where these bird calls would overlap. All recordings were sampled at a rate of 44.1 kHz, solely because it was the sampling rate with the most available data. Data was also chosen only if it was classified as ”song” or ”call”. Bird songs and calls are longer, more distinct audio signals that are unique to each species, and so they are much better to classify on than other sounds like ”chirps” or ”mating calls”. Finally, some recordings were long, upwards of 60 seconds. To keep recordings short so that keypoints could meaningfully reflect the signal, samples were augmented into 10-second samples. This means samples longer than 10 seconds were truncated, and samples smaller than 10 seconds were padded with 0 values. This approach is assumed to be effective because most data collected that were longer than 10 seconds appeared to be multiple repetitions of a bird call, and so a 10 second clip would encapsulate at least one of those bird call repetitions. Audio is then normalized to avoid classifying by audio volume since birds can be close or far in the application of this algorithm. See Figure 1.1 for examples of our bird species calls after this normalization and time standardization. Two samples can then be chosen at random and overlapped by adding the waveforms together. Note that we only tested overalapping different species samples, so there were no samples of the same bird species twice. Those samples are then converted into spectrograms using Short-Time Fourier Transforms (1.2).

Then, some basic preprocessing is done to reduce background noise. First, a butterworth bandpass filter is applied to the signals to keep the signals between 1-8 kHz. This is done in accordance with domain knowledge that most birds only vocalize between these ranges [2]. Anything outside of these ranges can be treated as unwanted noise. Then, Per-Channel Energy Normalization [4] is done to filter constant frequency bands that get picked up, as these are generally background noise such as crickets. Figure 1.3 shows the resulting spectrogram after these filtering operations, which we will call a ”full-spectrogram” for future baseline testing. Note that although it may appear like the filtering actually makes the signal harder to discern from background noise by human eyes, the algorithm does perform better with this preprocessing.

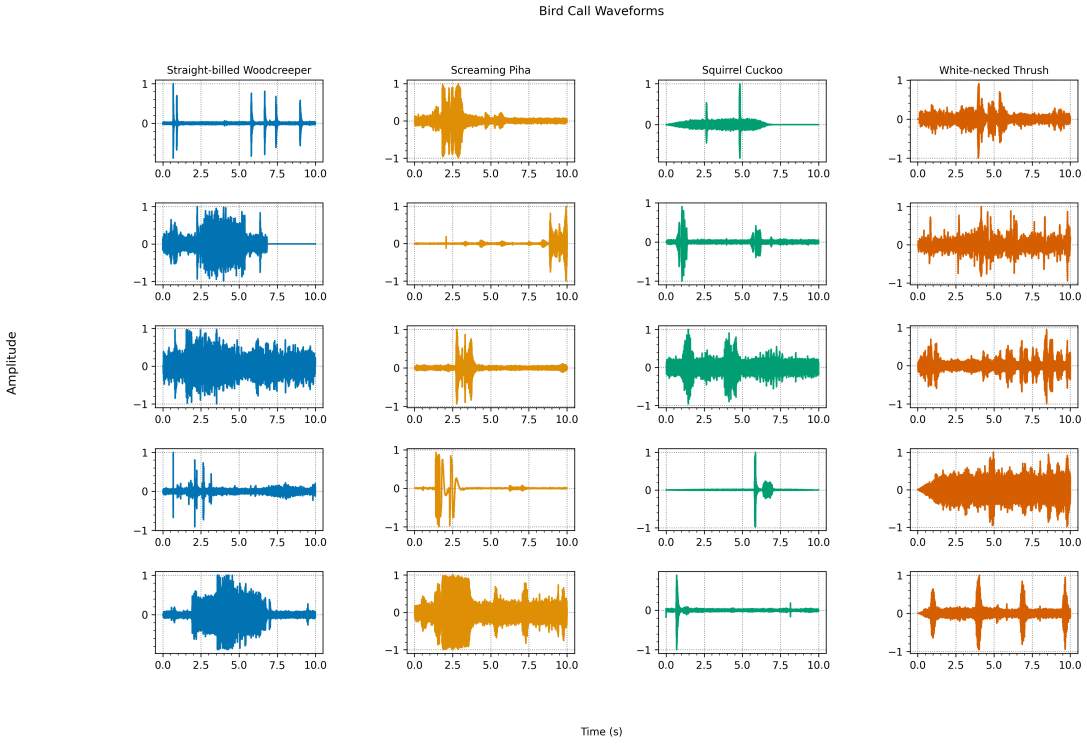

Figure 1.1. Wavelength representation of two different bird species calls after normalization and time standardization.

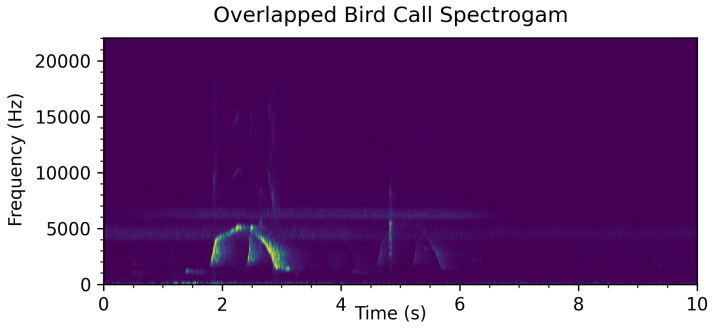

Figure 1.2. Spectrogram representation of two different overlapped bird species calls.

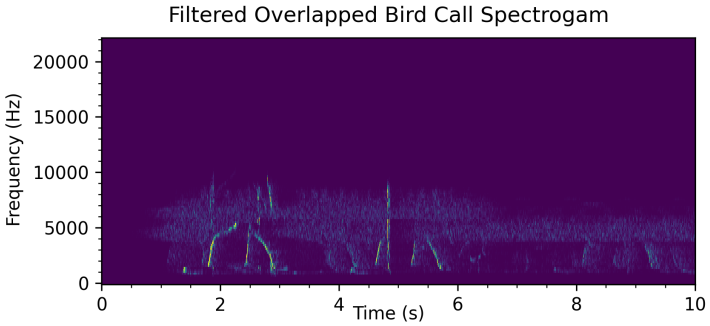

Figure 1.3. Spectrogram representation of two different overlapped bird species calls after butterworth and PCEN filtering.

Chapter 2

Methodology

Once the audio sample has been pre-processed with the techniques listed above, we begin the keypoint-spectrogram detection (2.1). As the Wang paper states,

> Keypoints are detected at locations that are local maximum across both frequency and time [6].

We first set a constant threshold to avoid picking up keypoints where no significant signal is found. For our experiments, we found that thresholding to only the top 99.5th percentile was most effective at removing keypoints in areas where only noise was present. We then pass a sliding window over the spectrogram, and any local maxima above the threshold are marked as keypoints. After testing a variety of window shapes and shifts, we settled on a window of 5ms x 240Hz with 50% overlap (shift of 120Hz). We found these dimensions to hit a balance of having quick performance when ran hundreds of times to generate samples, while keeping windows small enough to capture fine details. The overlap was kept to exaggerate strong keypoints that might be a maxima of 1.5x a window length. We store these keypoints by six summary statistics [6] to use as our features for our classification. The first three summary statistics are the energy, time, and frequency of the keypoint. These statistics describe the location and magnitude of the keypoint. The fourth summary statistic is the spectral-rolloff point of the window containing the keypoint. The fifth is the temporal-rolloff point of the window containing a keypoint. The last statistic is the short-time energy of the window. The spectral-rolloff point is defined as the frequency where N% of the energy in the window is below that frequency [5]. Temporal-rolloff

is the time duration where N% of the power in the window is within that duration, concentrated at the center of window time frame. These are essentially describing the spread of the energy in the window in the x and y directions. For this paper, we defined N to be 90. The short-time energy of the window is defined as the mean of the energy in that window [5].

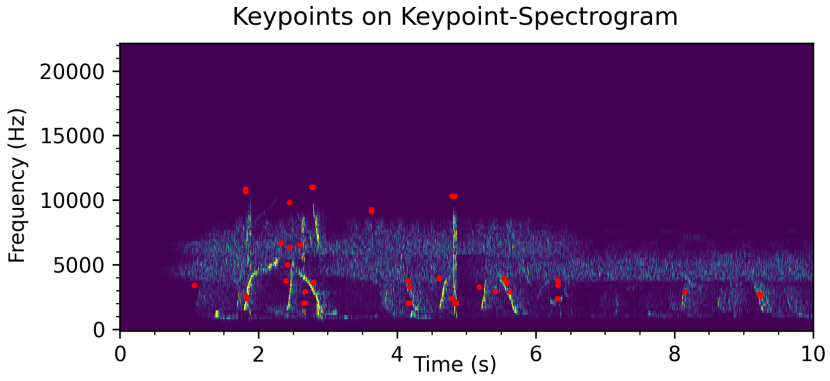

Figure 2.1. Keypoints marked as red dots on a filtered spectrogram of overlapped samples. Only some keypoints shown for clarity.

For each signal, we get a $k \times 4$ output array, where k is the number of keypoints found in the signal. We want to use these output arrays as our input to our classification models for our predictions [6]. That means our input has to be 2-dimensional and all samples must have the same number of features. Since our input will be a list of our $k \times 4$ output arrays, we first flatten that array to be 1-dimensional. Next, we need to ensure all signals have the same number of keypoints so that every input sample has the same dimensions. We do this by taking the m keypoints with the highest energy from the list of found keypoints in a spectrogram. We set m to be 200 for our models, as we tested 20 to 300 keypoints at an interval of 10 and found the best performance at 200-300 keypoints. Some shorter signals would have less than 500 keypoints, so we wanted to keep the number of keypoints as small as possible to not risk having a signal with less than m keypoints. Finally we can concatenate these arrays as our input data for our classification models to have a shape of $d \times 4000$ where d is the number of samples in our dataset. We chose to test an SVM classifier and a KNN classifier. An SVM classifier can generally handle high-dimensional data well and has predicted with decent accuracy in similar problems

[6]. The KNN classifier was chosen because we expect the distribution of keypoints relative to one another to be crucial for prediction. As mentioned, the input is a $d \times 4000$ where d is the number of samples in our dataset and 4000 is the first $m = 1000$ keypoints each with 4 statistical variables. We also model our data with a basic CNN. Our CNN keypoint-spectrogram model takes binary matrices as input that have the same shape as our original filtered spectrograms (full-spectrogram), with a 1 denoting a keypoint and 0 otherwise. Similar to our prior input, we only flag the m greatest magnitude keypoints with a 1, with m being 200 for our models. We compare our keypoint-spectrogram model with a full-spectrogram model that trains and tests on the full-spectrogram instead, but with the same architecture otherwise.

Chapter 3

Results

The results from our SVM classification show an accuracy on single samples of around 25% (3.1). Since we are testing on 4 classes of bird, this is as effective as random guesses. Our KNN classifier has similar results, with accuracy also around 25% (3.2). Despite hyperparameter tuning and trying different combinations of inputs, we were unable to get the results to be much better than random guesses. Because of the poor results in predicting even just single samples of bird calls, we decided to move on from the SVM and KNN classification models in favor of the CNN model for futher testing on overlapping audio signals.

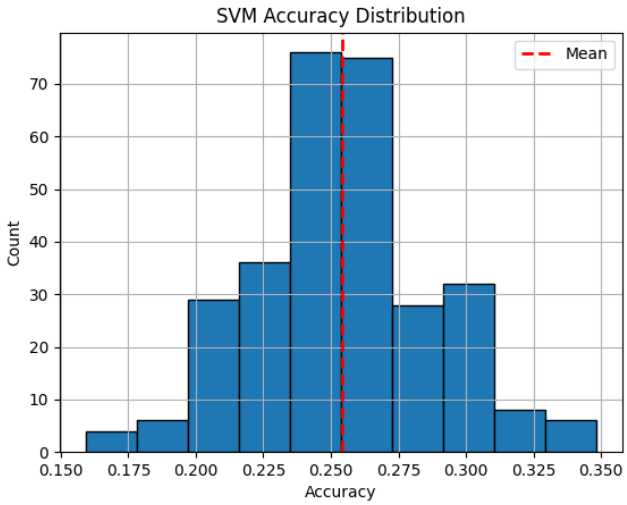

Figure 3.1. Accuracy of SVM model after 300 iterations.

The CNN had more promising results. We tested the CNN model on single keypoint-

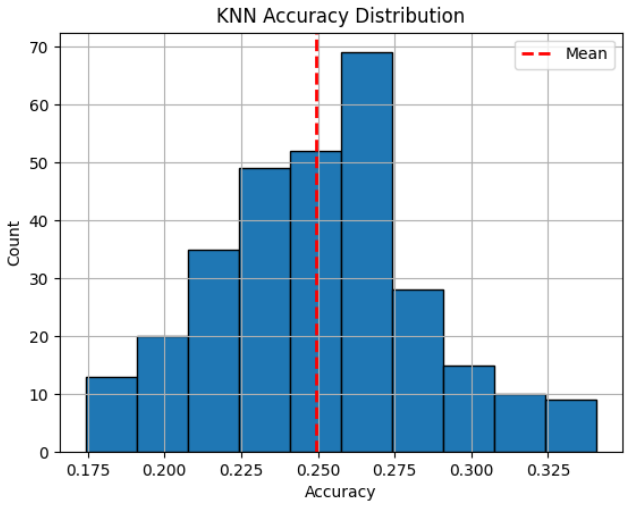

Figure 3.2. Accuracy of KNN model after 300 iterations.

spectrogram input, single full-spectrogram input, overlapped keypoint-spectrogram input, and overlapped full-spectrogram input (3.4). The single sample inputs were trained on 412 samples. It was validated on 136 samples, and tested on another 136 samples. The full dataset was uniformly distributed between all the classes, and this comes out to a 60/20/20 train/test/validation split. The model had two convolution layers, two max pooling layers, and three dropout layers to avoid over-fitting 3.3. The single sample model used a softmax activation function and categorical cross-entropy loss for multi-class predictions. While the full-spectrogram was still only barely better than random guessing when predicting a single sample, the keypoint-spectrogram was classified correctly 20% more often with the same model. These were promising results that the keypoints could summarize class signals better for the CNN to predict, so we tried the overlapping audio data. The overlapped sample tests were measured based off of a model ”score”. Since our models are predicting 2 classes, score = 0 when both predicted classes are wrong, score = 1 when one class is predicted correctly, and score = 2 if both classes are predicted correctly. We can calculate a score for each prediction by the model, and then calculate the mean as the overall model score. We calculate the expected score from random guessing in Equation 3.1 as a baseline to compare our model scores. The model architecture remained the same, with the

exception that we now used a sigmoid activation function and binary cross-entropy loss because we are doing multi-class multi-label classification. We see that once again the full-spectrogram model performs similar to random guessing. On the other hand, the keypoint-spectrogram again outperforms the both other methods by over 0.4 score. Furthermore, we can investigate the distribution of these scores, and we see that the keypoint-spectrogram model has both the most completely correct predictions, and the least completely incorrect predictions (3.5). In viewing the output, all classes had roughly the same amount of accuracy, with the best being the Screaming Piha. This is likely because the Screaming Piha has a very distinct call, and it generally has the highest signal-to-noise ratio samples due to the bird itself being especially loud. The CNN keypoint-spectrogram model outperforms full-spectrogram and random guessing models both by mean and proportion of true positive predictions. Although there is plenty of room for the model to improve on its predictions, it shows promise in the keypoint method when dealing with bird call classification.

Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 120, 3442, 32)	1,952
max_pooling2d_18 (MaxPooling2D)	(None, 40, 688, 32)	0
dropout_18 (Dropout)	(None, 40, 688, 32)	0
conv2d_19 (Conv2D)	(None, 31, 683, 64)	122,944
max_pooling2d_19 (MaxPooling2D)	(None, 10, 170, 64)	0
dropout_19 (Dropout)	(None, 10, 170, 64)	0
flatten_9 (Flatten)	(None, 108800)	0
dense_18 (Dense)	(None, 128)	13,926,528
dropout_20 (Dropout)	(None, 128)	0
dense_19 (Dense)	(None, 4)	516

Figure 3.3. Architecture of the CNN model.

$$P(First = 1) = \frac{2}{4}$$
$$P(Second = 1|First = 0) = \frac{2}{3}$$
$$P(Second = 1|First = 1) = \frac{1}{3}$$

$$P(Score = 0) = P(Second = 0) \cap P(First = 0)$$
$$P(Score = 0) = P(First = 0) * P(Second = 0|First = 0)$$
$$P(Score = 0) = (1 - \frac{2}{4}) * (1 - \frac{2}{3})$$
$$P(Score = 0) = \frac{1}{6}$$

$$P(Score = 1) = (P(Second = 1) \cap P(First = 0)) \cup (P(Second = 0) \cap P(First = 1))$$
$$P(Score = 1) = (P(First = 0) * P(Second = 1|First = 0)) + (P(First = 1) * P(Second = 0|First = 1))$$
$$P(Score = 1) = ((1 - \frac{2}{4}) * \frac{2}{3}) + (\frac{2}{4} * \frac{2}{3})$$
$$P(Score = 1) = \frac{2}{3}$$

$$P(Score = 2) = P(Second = 1) \cap P(First = 1)$$
$$P(Score = 2) = P(First = 1) * P(Second = 1|First = 1)$$
$$P(Score = 2) = (\frac{2}{4}) * (\frac{1}{3})$$
$$P(Score = 2) = \frac{1}{6}$$

$$E(X) = \sum_{i=0}^{2} xP(Score = x)$$
$$E(X) = (0 * \frac{1}{6}) + (1 * \frac{2}{3}) + (2 * \frac{1}{6})$$
$$E(X) = 1 \tag{3.1}$$

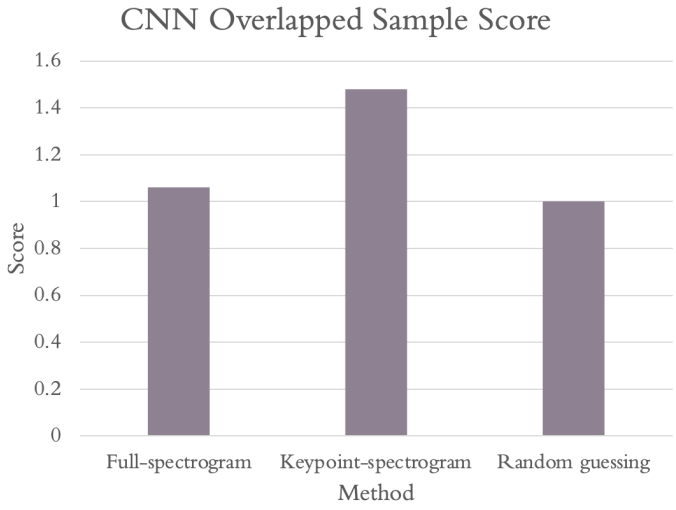

Figure 3.4. Scores of the full-spectrogram and keypoint-spectrograms, and expected random guessing score.

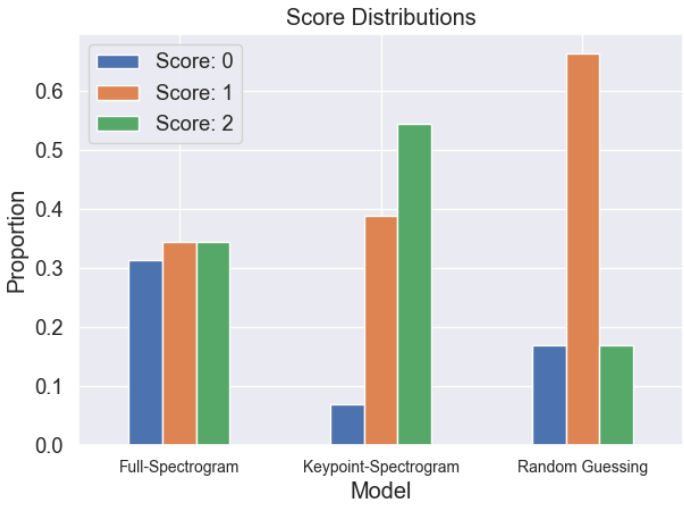

Figure 3.5. Score distribution of the three methods.

Chapter 4

Conclusion

The keypoint-spectrogram technique shows promise when combine with CNNs to classify bird calls, but not with more simple machine learning techniques like SVM and KNN classifiers. However, it is important to note that in our experiments we omitted much of the preprocessing that would be expected for a SOTA approach to bird call classification, and perhaps with this extra preprocessing we achieve better results with our SVM and KNN classifiers. Our theory is that the summary statistics cannot define the shape of the signal well enough to be classified, and for signals as intricate as bird calls the shape is very important as the main identifier of the class. Going forward, testing the keypoint-spectrogram CNN model with more domain-specific preprocessing such as the SOTA preprocessing seen in BirdNET [3] may lead to results that could be put into practice as rivals to SOTA techniques. This still remains to be tested however, and must also be trained with labelled soundscape data in order to determine real-world viability.

Bibliography

[1] Darras, K., Furnas, B., Fitriawan, I., Mulyani, Y., Tscharntke, T. (2018) 'Estimating bird detection distances in sound recordings for standardizing detection ranges and distance sampling', Methods in Ecology and Evolution, 9(9), pp. 1928–1938. doi:10.1111/2041-210x.13031.

[2] Do bird songs have frequencies higher than humans can hear? (2023) All About Birds. Available at: https://www.allaboutbirds.org/news/do-bird-songs-have-frequencies-higher-than-humans-can-hear/ (Accessed: 22 February 2024).

[3] Kahl, S., Wood, C. M., Eibl, M., Klinck, H. (2021a). BirdNET: A deep learning solution for avian diversity monitoring. Ecological Informatics, 61, 101236. doi:10.1016/j.ecoinf.2021.101236

[4] Lostanlen, V., Salamon, J., Cartwright, M., McFee, B., Farnsworth, A., Kelling, S., Bello, J. (2019) 'Per-channel energy normalization: Why and how', IEEE Signal Processing Letters, 26(1), pp. 39–43. doi:10.1109/lsp.2018.2878620.

[5] Mitrović, D., Zeppelzauer, M. Breiteneder, C. (2010) 'Features for content-based audio retrieval', Advances in Computers, pp. 71–150. doi:10.1016/s0065-2458(10)78003-7.

[6] Wang, W., Seraj, F., Meratnia, N., Havinga, P. (2019) 'Localization and classification of overlapping sound events based on spectrogram-keypoint using acoustic-sensor-network data', 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS) [Preprint]. doi:10.1109/iotais47347.2019.8980421.