Project Title

Optional Subtitle

Candidate Number: YPDR0[1]

MSc Data Science & Machine Learning

Supervisor: Dr. Ifat Yasin

Submission date: 11 September 2023

[1] **Disclaimer:** This report is submitted as part requirement for the MSc Data Science & Machine Learning at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged

Abstract

Summarise your report concisely.

Contents

1 Introduction 2

2 Background & related work 4

2.1 Evaluation metrics 4

2.2 Feature extraction 5

2.2.1 Features inspired by the human auditory system 7

2.2.2 Feature stacking 9

2.3 Classification 10

2.3.1 SVMs 11

2.3.2 Neural Networks 12

3 Methodology 20

3.1 Datasets 20

3.2 Pre-processing 21

3.3 Segmentation 22

3.3.1 Energy based syllable segmentation 23

4 Extensions of methodology 24

5 Conclusion 25

Chapter 1

Introduction

Birds play a crucial role in ecosystems around the world. They form an important link in the food chain, pollinate plants, and even plant trees [9]. The quantity and diversity of birds observed in an area can therefore be seen as a key indicator of the strength of its ecosystem.

Aside from being important ecological agents, birds also colour our lives with their sights, sounds and behaviours. The community of birdwatchers, known colloquially in the U.K. as 'birders' or 'twitchers', has grown considerably over the past few years. The Royal Society for the Protection of Birds (RSPB) reported a 70% increase in their website views over the first lockdown, with more than 50% of those views on pages looking at bird identification. The Bird Bird Garden Watch, an annual event which encourages people to note bird sightings in their own residences, brought in 1 million participants in January 2021, more than double the previous year's tally. As such there is a growing commercial demand for accurate and easily accessible bird identification tools.

Birds are typically shy and protective creatures and tend to reside out of harm's way in shrubs, trees and nests, and therefore are usually heard but not seen. The majority of their communication is done through distinctive vocalizations, known as birdsong, that are usually unique to an individual species. The consequence of this is that birds are typically identified through their birdsong rather than their visual sighting. Birdsong used for identification is usually recorded on microphones that may be running for a long time and/or located in a place that isn't necessarily optimum to capture the vocalization. The recordings may be corrupted from a wide range of sources, such as ambient background noise, large changes in birdsong amplitude, long periods of silence, and vocalizations from other birds. Recordings captured by long running microphones may be several hours in duration so it would be impractical to have a human expert identify the birds manually. There is therefore a need for robust birdsong identification tools that can handle these corruptions and that require minimal human intervention in order to classify unknown birdsong.

As with most machine learning classification problems, the work boils down to two key components. The first is feature extraction, that is, given an input signal usually in the form of an amplitude varying over time, how can it be transformed to a vector or matrix representation that captures the key information of the signal. The second is classification, i.e. given this representation in the feature space, how can it be used to train a model that can then perform classification on unseen samples of birdsong. There are further steps that can be taken to improve this process, such as pre-processing the signal in order to remove or reduce the influence of background noise [48].

There has been numerous research into audio classification problems both in birdsong and in other types of audio, such as speaker identification from human speech [31]. However, there remain novel methods that have been tested and have been shown to have good results in wider audio classification problems that have yet to be tried out in birdsong identification problems. These methods include both feature extraction and classification. The broad aim of this thesis is to attempt to evaluate these methods. In more detail, the aims of this thesis can be summarised as follows:

1. Explore the birdsong classification performance of feature representations shown to have promising results for non-birdsong related audio classification problems using simple classification models such as Support Vector Machines (SVM). The hypothesis is that feature representations shown to have good results in audio classification problems will have good results for birdsong identification.

2. Explore some of the hyperparameters available during the feature extraction process. The hypothesis is that using hyperparameter values more suited to birdsong classification problems will yield a higher classification accuracy.

3. Explore the performance of deep learning architectures such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) and compare the results with simpler statistical models. The hypothesis is that more complex and flexible architectures such as these will have superior performance when compared to simpler statistical models, such as SVMs.

Chapter 2

Background & related work

Describe here work that is connected to your thesis. This should include references to published work. There is no fixed rule, but I would expect a student to have read around 50 published research papers and reference them in a thesis.

2.1 Evaluation metrics

For multiclass classification problems, e.g. identifying a sample as a particular bird species from a last of $n > 2$ species, a simple classification accuracy is often used as an evaluation metric ([11], [50]). This is calculated as the percentage of correctly identified samples from a set of labelled samples previously unseen by the model.

Acevedo et al [1] used true positive (TP) and false positive (FP) rates to evaluate their models used to classify bird and amphibian calls. This evaluation method is sometimes preferred when analysing long recordings which may include several different species to be classified. TP gives an indication as to how well the model correctly identifies species present in the recording. FP indicates how often the model erroneously identifies species absent in the recording. A high performing model will have high values for TP and low values for FP.

Potamitis et al [48] used an alternative form of evaluation presented in terms of precision (P) and recall (R) which are defined as

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.1}$$

where FN is the number of false negatives. Informally, P and R can be thought of as

$$P = \frac{\text{relevant retrieved instances}}{\text{all } \textbf{retrieved} \text{ instances}}, \quad R = \frac{\text{relevant retrieved instances}}{\text{all } \textbf{relevant} \text{ instances}} \tag{2.2}$$

It's well known that there is usually a trade-off between P and R, so usually the F-score is reported along with the P and R metrics. The F-score is defined as

$$F = \frac{(1+\beta^2)PR}{\beta^2 P + R} \tag{2.3}$$

for some $\beta \in \mathbb{R}$.

For binary classification problems, i.e. classifying a sample as one of two classes (bird species), the Area Under Curve (AUC) metric is sometimes preferred [36]. The AUC ranges from 0 to 1, with a higher value indicating a better performing model. The AUC is sometimes desirable because it is scale-invariant (it measures how well predictions are ranked, rather than their absolute values) and classification-threshold-invariant (it measures the quality of the model's predictions regardless of what classification threshold is chosen.)

2.2 Feature extraction

As mentioned in Chapter 1, birds use vocalizations as a way to communicate with others. Birds have evolved over many thousands of years to make this form of communication very efficient in that it can travel long distances and cut through local ambient noise frequencies so that it can be received by other birds clearly. Some birds have even adapted their vocalizations so that they can be heard in local environments that have rapidly changed over the past decades, such as urban areas with increasing anthropogenic noise [38].

Bird vocalizations can be broadly divided into two main categories, calls and songs. Calls are typically short vocalizations that carry some specific function, such as warning others to the presence of a predator or calling others to flight [39]. Songs are usually longer and more acoustically complex and occur more spontaneously. Songs are typically employed as breeding calls or territorial defence. While all birds produce calls, in many species of birds only the males utilise songs and often only during breeding season. The song and call of a Eurasian Wren can be seen and contrasted in figure 2.1. Check Crous [14] for references on call types etc.

Doupe et al [19] showed that there were striking similarities between birdsong and human language. Similar to human language, birdsong can be thought of as comprised of hierarchical levels of phrases, syllables, and elements [10]. ¡Figure here¿. Raw recordings of birdsong often contain periods of silence that occur between phrases which are unlikely to provide useful information in order to train a machine learning model. Fagerlund [22] showed that a good level of accuracy can be achieved by training models on features extracted from segmented syllables which were used as training samples. An algorithm for robust syllable segmentation was proposed in [21] and is used in this thesis. The algorithm is described in more detail in Section.

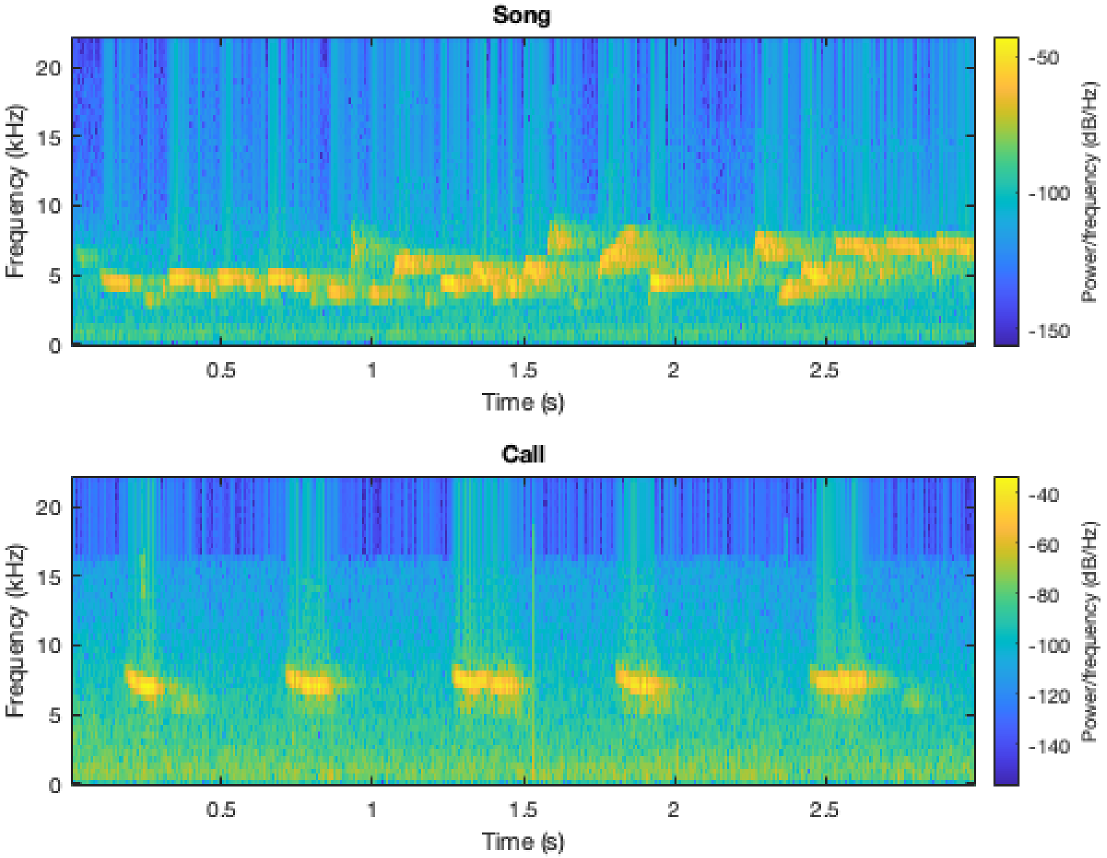

Figure 2.1: Spectrograms of the call and song of the Eurasian Wren (*Troglodytes troglodytes*). As can be seen, the song is much more complex in terms of pitch and acoustic structure compared to the call

Once the syllables have been segmented, there exists an abundance of options to turn the raw syllable input signal into features that might be used as training samples. In the following sections some of the more popular feature representations are described, along with some novel methods that have yet to be tried on birdsong. Note that the following list is certainly not exhaustive and emphasis has been placed on feature representations that are relevant to this thesis.

2.2.1 Features inspired by the human auditory system

Research has shown that birdsong and human language share many similarities in terms of the neural mechanisms employed to form the language/song, the impact of social contact in learning the language/song, and so on [19]. Give that the human auditory system has evolved other thousands of years to best process human speech, it seems reasonable to suggest that using features based on the human auditory system may be effective when it comes to birdsong.

At a high level, the human auditory system works by translating changes in air pressure originating from a source and reaching the outer ear of a listener into vibrations that travel along an internal organ known as a cochlear. The vibrations trigger electric signals that move along auditory nerves to the brain, where they are interpreted as sounds. The physiology of the cochlear means that certain parts of the organ are more sensitive to certain frequencies of vibrations, so different frequencies will lead to different electrical signals moving to the brain. This allows the brain to learn to differentiate between frequencies.

Mel frequency cepstrum coefficients

The makeup of the human auditory system means that humans have a greater ability to differentiate pitch at lower frequencies then they do at higher frequencies. In other words, humans perceive pitch non-linearly. This has led to the development of a logarithmic scale, known as the mel scale, such that equal distances on the scale have the same *perceptual* distance.

The mel frequency cepstrum coefficients (MFCC) are part of a family of cepstrum coefficients that capture information about the rate of change in different spectrum bands. MFCC differs from other cepstrum coefficients in that it uses the mel scale to transform the spectrum of an input signal, thus utilising the human auditory system in the calculation of its coefficients.

The i-th mel cepstral coefficient is computed as [16]

$$\mathrm{MFCC}_i = \sum_{k=1}^{K} X_k \cos\left(\frac{i(k-0.5)\pi}{K}\right) \tag{2.4}$$

where X_k is the logarithmic energy of the k-th mel-spectrum band, and K is the total number of the mel-spectrum bands. Usually 8–13 MFCC coefficients are used as the feature vector representing one time frame of the signal. The 0^{th} coefficient is often excluded as it represents the average log energy of the signal and is unlikely to carry any relevant information to help with classification.

MFCCs are often presented with their delta (Δ) and double-delta ($\Delta\Delta$) values that capture the local temporal dynamics and temporal changes of the delta values respectively.

MFCCs are used as feature representations since they are simple to compute and have been shown to have good performance in a wide range of audio classification tasks, such as speaker identification [42] and emotion recognition [37]. MFCCs have also been shown to lead to good classification accuracy for birdsong identification problems ([22] and [50]).

¡include graph of filterbank¿

Gammatone cepstrum coefficients

Gammatone cepstrum coefficients (GTCCs) are similar to MFCCs except their calculation uses gammatone filterbanks instead of mel scale filterbanks. Gammatone filterbanks are designed to simulate the motion of the membrane inside the cochlear, known as the basilar membrane, when it is exposed to vibrations transmitted by the outer and middle ear [47].

A gammatone filter with a centre frequency f_c is defined as

$$g(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi f_c + \psi) \tag{2.5}$$

where t is the time in seconds, ψ is the phase in radians (usually set to 0), $a \in \mathbb{R}$ controls the gain, n is the filter's order and b is the filter's bandwidth in Hz. To simulate the human auditory system, the centre frequencies are uniformly spaced on the equivalent rectangular band-width (ERB) scale.

Similar to MFCCs, GTCCs are typically used with their Δ and $\Delta\Delta$ values and have also been shown to have a good performance in non-speech audio classification problems [60].

¡include graph of filterbank¿

Multi resolution cochleagram

Chen et al [12] proposed a novel feature known as a multi-resolution cochleagram (MRCG) that is formed by combining four cochleagrams at different resolutions, designed to capture both local and contextual information.

A cochleagram is formed by passing an input signal through a gammatone filterbank, similar to the first steps of the formation of the GTCCs. Each response signal from the gammatone filterbank is then divided into frames of a certain length with a certain overlap or frame shift. The cochleagram is then generated by calculating the power of each frame at each channel.

¡example cochleagram¿

A MRCG is formed by first taking one cochleagram with a frame length of 20ms and frame shift of 10ms, using a gammatone filterbank with 64 output channels. A log operation is applied to each time-frequency (T-F) unit of the output cochleagram, denoted CG1. CG2 is calculated in the same way, but with a frame length of 200ms and the same frame shift. CG3 is calculated by averaging CG1 across a square window of 11 frequency channels and 11 time frames centred at the given T-F unit. Zero padding is used here. CG4 is calculated in the same way as CG3, but with a 23×23 square window. CG1–4 are then stacked vertically to obtain the full MRCG feature, which will be a $256 \times p$ matrix, where p is the number of time frames.

¡example mrcg¿

The MRCG feature has been shown to outperform both MFCC and GTCC at separating human speech at various SNR levels with different types of background noise [12]. Abdullah et al [7] showed that the MRCG outperforms the Auditory Image Model (AIM) when classifying noise. Wang et al [63] showed that improved speech enhancement for hearing impaired listeners was achieved when the MRCG feature was added to the feature set.

One potential drawback when comparing the MRCG with MFCC and GTCC is the higher dimensionality of the MRCG feature. When including the Δ and $\Delta\Delta$ features, as is standard practice ([7], [63]), the MRCG feature vector for a single time frame has a dimensionality of 768. For MFCC or GTCC the equivalent dimensionality is 24–39, depending on the number of the coefficients used.

2.2.2 Feature stacking

It has been shown that appropriate combining features to be used as training inputs can lead to improved classification accuracy in audio related problems such speech separation [62]. Ramashini et al [50] showed that a combination of GTCC and MFCC yielded higher birdsong classification accuracy than MFCC alone. However, the combination did not improve on the accuracy of GTCC alone. Yan et al [64] demonstrated that a combination of MFCC with two other feature types (Log-mel spectrogram and Chroma) yielded higher birdsong classification accuracy than combinations of two of the features alone. Fagerlund [22] showed that combining MFCC with spectral and temporal features, such as frequency range and zero crossing rate, can lead to higher birdsong classification accuracies.

A consequence of feature stacking is that feature vectors will have increased size. This can lead to problems such as longer training times and overfitting, especially when used in conjunction with deep learning (DL) architectures. As a result, dimensionality reduction techniques such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) [51], are often used to lower the dimensionality of the feature vectors while still capturing enough information to be able to train a model effectively.

2.3 Classification

Once a suitable feature representation has been selected and performed, the remaining task is to pick an appropriate model and train the model using the set of training samples generated from the feature extraction.

There exist many different options in terms of models, and no one model is superior to the others. Each problem must be considered and a model which fits the problem at hand must be selected, or indeed a selection of suitable models tested and the evaluated for performance.

In the literature, birdsong classification has been tackled with a wide variety with models. Ramashini et al [51] used the simple Nearest Centroid (NC) classifier to assign birdsong samples to classes. It was compared to a K Nearest Neighbours (KNN) classifier and shown to have superior accuracy. Lasseck [33] used decision trees with bagging to identify birds. Trifa et al. [59] used Hidden Markov Models (HMM) to identify species of birds based on their vocalizations. They considered each sample as a discrete-time dynamical system, where an unobserved state generates observed features, such as pitch and MFCC. A different HMM can be learned for each class and a new sample can be assigned to a class by calculating which HMM gives the highest likelihood of the observed features. Kwan et al. [32] used a Gaussian Mixture Model (GMM) to classify birds. In their experiments, a Gaussian was learned for each class of bird. A new sample was then classified according to whichever class best describes the new sample.

The following sections describe some models that are relevant to this thesis in more detail

2.3.1 SVMs

SVMs are widely used in machine learning applications as a classification tool. They are well-established due to their high accuracy results [22] and relative simplicity to implement. Most implementations of SVMs also require little or no tuning of hyperparameters.

At its core, a SVM is a binary classifier that separates two classes by finding a hyperplane that maximizes the margin from the nearest vectors in the feature space from both classes. Classifications are then made by computing in which side of the hyperplane a test feature vector lies.

Binary classification

Let $\mathbf{x}_i \in \mathbb{R}^m$ be a feature vector of dimensionality m. Let $y_i \in \{+1, -1\}$ be its class label. For linearly separable data, the separating hyperplane satisfies

$$y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) \geq 1, \quad i = 1, \ldots, n \tag{2.6}$$

where $\mathbf{w} \in \mathbb{R}^m$ is a vector of weights and $b \in \mathbb{R}$ is a bias term. The margin between the hyperplane and the nearest feature vectors from each class is given by

$$\begin{aligned} \mathrm{d}(\mathbf{w}, b) &= \min_{\mathbf{x}_i, y_i = 1} \frac{|\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b|}{\|\mathbf{w}\|} + \min_{\mathbf{x}_j, y_i = -1} \frac{|\langle \mathbf{w} \cdot \mathbf{x}_j \rangle + b|}{\|\mathbf{w}\|} & (2.7) \\ &= \frac{2}{\|\mathbf{w}\|}. & (2.8) \end{aligned}$$

The optimal hyperplane can now be found by maximizing (2.8) subject to (2.6). This can be solved using Lagrange multipliers.

Often with real world data, classes are not linearly separable. This can remedied by projecting the data using a nonlinear mapping into a new feature space where the data are linearly separable. Equation (2.6) can then be re-written as

$$y_i \left(\langle \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i) \rangle + b \right) \geq 1, \quad i = 1, \ldots, n \tag{2.9}$$

where $\mathbf{\Phi}$ is the nonlinear mapping. Instead of explicitly finding $\mathbf{\Phi}$, (2.9) can be re-written in dual form

$$y_i \left(\sum_{j=1}^{l} \alpha_j y_j \langle \mathbf{\Phi}(\mathbf{x}_j) \cdot \mathbf{\Phi}(\mathbf{x}_i) \rangle \right) + b \geq 1, \quad i = 1, \ldots, n \tag{2.10}$$

and replacing the inner product with a kernel function $K(\mathbf{x}_j, \mathbf{x}_i) = \langle \mathbf{\Phi}(\mathbf{x}_j) \cdot \mathbf{\Phi}(\mathbf{x}_i) \rangle$.

In practice, a hyperplane that separates the classes perfectly may suffer from poor generalization ability. To improve this, nonnegative slack variables ξ_i are introduced to (2.9) to allow for some missclassification of training samples in order to improve generalization ability. The slack variables are introduced like so

$$y_i \left(\langle \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \ldots, n \tag{2.11}$$

The amount of regularization is controlled by the constant C such that the maximization problem (2.8) becomes

$$\frac{2}{\|\mathbf{w}\|} - C \sum_{i=1}^{n} \xi_i \tag{2.12}$$

Therefore for large values of C the classifier will behave like a hard-margin SVM and attempt to perfectly classify all training samples.

The last remaining question is around what determines a valid kernel function. A function in the input space is a kernel function if its kernel matrix $\mathbf{K} = [K(\mathbf{x}_j, \mathbf{x}_i)]_{i,j=1}^{n}$ is positive semidefinite. Typical kernel functions include

- linear: $K(\mathbf{x}_j, \mathbf{x}_i) = \langle \mathbf{x}_j, \mathbf{x}_i \rangle$
- polynomial: $(\langle \mathbf{x}_j, \mathbf{x}_i \rangle + c)^p$ for some $c \in \mathbb{R}$
- Gaussian or RBF: $\exp\left(-\gamma \|\mathbf{x}_j - \mathbf{x}_i\|^2\right)$ for some $\gamma \in \mathbb{R}, \gamma > 0$.

Multiclass classification

SVMs can easily be extended to work with multiclass classification using standard procedures. One procedure which is optimal with respect to the Hamming Loss, which relates to the number of false positives or false negatives, is the one-versus-all technique. Here, multiple SVMs are trained, each one learning the hyperplane for samples belonging to one class versus samples belonging to all the other classes. A new sample can then be tested against all the models and predictions are made using the model that is most confident.

2.3.2 Neural Networks

Although neural networks have existed in some form since the 1950s with inception of the perceptron algorithm [53], they have experienced a huge surge in popularity over the past decade or so. This has largely been due to remarkable progress being made thanks to deep neural networks in fields such as image classification [30] and language models [41].

Another reason for DL's increasing popularity in recent years is due to the availability of more performant hardware and software. The 'deep' aspect to deep learning refers to the fact that the software architecture depends on large amounts of parameters and needs massive amounts of training data in order to learn. In order to run training routines in a reasonable time and store large amounts of data in memory, access to powerful hardware and/or machine parallelism tools can be an essential part of the process.

The progress of audio problems related has also been accelerated due to deep neural networks. Hinton et al. [26] showed improved performance in speech recognition tasks using feed-forward neural networks when compared to more classical approaches like HMMs and GMMs. Speech enhancement [3] and speech separation [20] problems leveraging DL paradigms have also been shown to outperform more classical statistical models.

The world of birdsong classification has also benefited from DL, but perhaps to a lesser extent than other audio related problems so far. Approaches have mainly focused on Convolutional Neural Networks (CNNs), applying convolutional layers to an image representation of an input signal, such as a (Mel-)spectrogram ([6], [43]). Further birdsong classification solutions have approached the problem in a similar fashion, but utilised transfer learning to fine-tune an existing model ([18], [34]), such as ResNet [24] and ImageNet [17].

Disabato et al. [18] went a step further with their research in that they considered the computational and memory demands of their proposed bird detection DL network, ToucaNet. As mentioned earlier, DL applications can be extremely demanding in terms of computational resources which can make them unsuitable for running on devices with limited resources, such as smartphones and autonomous recording units (ARUs). Their research showed that a level of accuracy in line with the literature can be achieved but with lower computational complexity and memory demands.

Although the architecture of DL tools for birdsong classification may vary widely, the final layer is typically the same. This consists of a fully connected layer of n units, where n is the number of classes, with a softmax activation so that the output of the network can be considered as a probability distribution over the classes. The softmax function is defined as

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad i = 1, \ldots, K \tag{2.13}$$

where K is the number of classes.

A classification of an unseen sample can then be performed by assigning the sample to the class with the highest corresponding probability.

Feedforward Neural Network

A feedforward neural network (FNN) consists of a system of connected nodes, organised in layers, where information flows in one direction only — forward — from the input nodes, through the hidden nodes (if any), and to the output nodes. This differs fundamentally from e.g. a RNN, where information can flow forward and backwards. However, both FNNs and its derivatives and RNNs share many similarities, such as learning using backpropagation, and using nonlinear activation functions in order to model nonlinear data.

A FNN in perhaps its most basic form while still being able to solve non-trivial problems is known as a multilayer perceptron (MLP). An MLP consists of at least three layers of fully connected nodes with a nonlinear activation function. Since their inception in late 21st century MLPs have had extensive research conducted into their capabilities and efficiency ([28]).

¡description of MLP with diagram¿

MLPs have largely been superseded by more modern DL architectures such as CNN and RNN when it comes to audio problems such as birdsong classification, however there have been examples of more vanilla neural networks being used. McIlraith and Card [40] used a two layer perceptron network using backpropagation to minimize a mean squared error loss to classify samples from 7 birds. Murcia and Paniagua [44] used a simple FNN to classify birdsong from 35 different bird species, achieving 0.74 AUC and in doing so winning the International Conference on Machine Learning Bird Challenge 2013.

Convolutional Neural Network

CNNs have seen a huge surge in development and popularity over the last decade, largely thanks to huge strides being made in fields like machine vision and image classification using architectures such as AlexNet [30]. Since image and audio share many conceptual similarities, it is reasonable to suggest that CNNs may enjoy similar success in the world of audio classification problems. This hypothesis is strengthened when one considers that many feature representations of audio signals can be displayed as images, such as spectrograms (figure 2.1), MRCG outputs and MFCC outputs ¡add links to figures here¿. Indeed, there has been significant progress in recent years thanks to CNNs in fields such as audio event recognition [58] and automatic speech recognition [55].

A CNN is a feedforward neural network with at least one convolutional layer. A convolutional layer is so called because it performs a convolution on an input, usually an image, using a small matrix of weights, known as a filter or kernel. The output is considered as a matrix of activations. The activation is higher when the values in the corresponding area in the input are similar to those in the kernel, meaning that the activation output encodes the location of where the input matches certain features. Usually, several kernels are applied at each layer, and so the output takes the shape of a 3D matrix, or a volume. As with all neural nets, the output is passed through a nonlinear activation function in order to allow for the network to be able to learn from nonlinear data. The weights in the kernel are learned through backpropagation. Usually a max pooling layer is added in order to downsample the activation volume and reduce the number of weights, or parameters, needed by the network.

¡diagram of cnn¿

CNNs lend themselves well to image related problems since their structure allows them parse an image as a composition of meaningful elements rather than a collection of unrelated pixels [35]. This translates well to birdsong as e.g. a spectrogram of a sample of birdsong can conceptually be thought of as a composition of syllables, i.e. elements.

CNNs have been applied to birdsong classification challenges with great success. Kahl et al. [29] achieved a remarkable mean average precision of 0.605 for a dataset of 1500 different bird species using a CNN trained on spectrograms of 4 second samples. The spectrograms were pre-processed to reduce noise and augmented with a variety of augmentations, such as adding Gaussian noise and real noise samples from the original dataset. Ruff et al. [54] used CNNs to detect owl vocalizations in spectrograms from unprocessed field recordings, performing as well or better than human experts. Narasimhan et al. [45] used a CNN with encoder-decoder architecture based on Segnet [4] to simultaneously segment and classify birdsong spectrograms. This approach benefits from the fact that segmentation and classification are intuitively strongly interrelated, since when comparing two different segmentations, the one that leads to a higher classification accuracy will likely be a better segmentation.

Ruff [54] looks like a good paper for this. Kahl [29] also.

Recurrent Neural Network

Since FNNs have a fixed size of input and output, they become unsuitable when either the input size is variable and the output is fixed (e.g. with AI image generation tools like

Dall-E [52]), the input size is fixed and the output is variable (e.g. image captioning) or both the input size and output size are variable (e.g. language translation). RNNs solve this problem by allowing information in the network to flow in two different directions, both forward and backward, or more precisely, in directed cycles. This is done by feeding the output from a previous step as input to the current step, therefore the state of the hidden units depends on the previous state of the network. This gives the network the ability to learn long term dependencies thanks to an unbounded previous history. The architecture of a standard RNN is shown in figure

¡figure of rnn¿

Similar to FNNs, classification RNNs learn by minimizing a loss function, usually the cross-entropy loss [56], by backpropagation. However, a slight variation on the algorithm is used to incorporate the sequential nature of the input data. The variation is known as backpropagation through time (BPTT). Here, in order to update the weights matrices, all weights are first treated as independent. The standard backpropagation algorithm is then run and all the corresponding gradients are averaged together. BPTT however suffers from a problem whereby gradients larger than $\mathbf{1}$ are multiplied together, which causes the gradient updates further back the network to exponentially increase, which may cause memory problems. This is known as the gradient exploding problem. The equivalent problem whereby gradients less than $\mathbf{1}$ vanish towards 0 is known as the gradient vanishing problem. This results in the model having no ability to learn long term dependencies. Both of these problems are compounded by RNNs learning from long sequences of inputs, causing longer multiplicative chains of updates. Two common approaches to dealing with these problems is to replace vanilla RNNs with long short-term memory (LSTM) units or gated recurrent units (GRU).

LSTM

LSTM was first proposed by Hochreiter and Schmidhuber [27] and has seen excellent results in fields such as speech recognition [25] and acoustic modelling [49]. LSTM describes an alternative structure for RNN whereby the hidden units are replaced with a memory cell which can store information for extended periods of time. The flow of information in and out of the cell is controlled by three boolean gates: a forget gate (2.14) which controls how information is lost from memory, an input gate (2.15) which controls how new information is stored in memory, and an output gate (2.16) which controls which information in memory should be output to the next layer and for the next time step.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.14}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.15}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{2.17}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.18}$$

Here, $\odot$ represents the Hadamard product (element-wise multiplication) and $W_{\{f,i,o,c\}}$ and $b_{\{f,i,o,c\}}$ are trained weights matrices and biases respectively. The above equations describe how an input vector x_t at time step t, the previous output vector h_{t-1} and cell state c_{t-1} combine to compute the next output h_t and cell state c_t. This process can be visualized in figure

¡figure of lstm¿ Yu [65] has some nice diagrams

The key conceptual difference between LSTM and RNN is that LSTM separates memory from the hidden state. LSTM deals with the gradient vanishing problem since the memory cells are additive with respect to time [15]. The gradient exploding problem can be tackled using strategies such as gradient clipping, whereby the gradient is reset to a small number after it exceeds a threshold value using

$$g = \frac{\text{threshold}}{\|g\|} g \tag{2.19}$$

Yan et al. [64] used LSTM with a CNN to achieve a mean average precision (MAP) of 0.979 in their classification experiment with 4 different species of birds.

GRU

GRU was first proposed by Cho et al. [13]. GRU shares many similarities with LSTM in that it replaces hidden units in a RNN with a memory cell, however it improves on LSTM in that it has few parameters, and therefore has lower computational burden. The reduction in parameters is achieved by combining the input gate and the forget gate of an LSTM to form an update gate, meaning that the GRU has only two gates, an update gate (2.21) and a reset gate (2.20).

$$r_t = \sigma\left(W_{rh}h_{t-1} + W_{rx}x_t + b_r\right) \tag{2.20}$$

$$z_t = \sigma\left(W_{zh}h_{t-1} + W_{zx}x_t + b_z\right) \tag{2.21}$$

$$\overline{h}_t = \tanh\left(W_{hh}(r_t \odot h_{t-1}) + W_{hx}x_t + b_z\right) \tag{2.22}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \overline{h}_t \tag{2.23}$$

¡figure of GRU¿

The reduction in parameters and number of gates does mean that the GRU is less powerful than the LSTM and does not work for problems such as translation [8]. However, it has been applied birdsong related problems with promising results ([46], [2]).

While the usage of RNNs, LSTMs or GRUs alone for birdsong classification has been rare in the literature, there are several examples of experiments where convolutional layers and recurrent layers have been combined to form a Convolutional Recurrent Neural Network (CRNN) ([64], [43]) in order to leverage the benefits of both architectures. A typical approach using a CRNN with regards to birdsong classification might look something like the following [14]

1. Convert an audio segment to a frequency domain representation to be interpreted as an image, e.g. a spectrogram.
2. Apply a convolutional layer (or layers) to each time step of the representation in order to extract local features. A time step is defined as a single unit along the time axis of the spectrogram.
3. Apply a recurrent layer to spread the local features of a single time step to its temporal surroundings. The spread features can then be processed by a fully connected layer with softmax activation as described in Section 2.3.2 to make a prediction in the form of a distribution over output classes for each time step.

The recurrent layer provides two key advantages compared to a plain CNN. Firstly, the network will be able to learn from previous and future features in order to strengthen predictions. For example, if the current time step equally supports a blackbird or a nightingale prediction by itself, it will be skewed towards a blackbird if the previous time step and future time step both predicted a blackbird. Secondly, the network can learn to include moments of silence in a prediction. Note that this benefit is lost if just considering audio segments of isolated syllables as these segments should contain little or no silence.

Check refs in the Narasimhan paper [45].

Chapter 3

Methodology

Describe your method in detail and with great clarity, distinguishing it from other works (if it is indeed a novel idea). It is very important to clearly motivate your method.

Describe the results of your method here in this chapter.

All computational work for this thesis was performed in *MATLAB vR2023a* unless stated otherwise.

3.1 Datasets

Crous [14] has some data on Xeno-canto.

Since all the models described in Section 2.3 are trained in a supervised learning fashion, this requires a dataset of labelled training examples. The website xeno-canto.org (XC) houses the largest and most comprehensive publicly available collection of birdsong samples in the world. It has made an enormous impact in the field birdsong recognition since its inception in 2005 and has been source of the datasets for the annual BirdCLEF challenge since 2014 [61]. All samples available on XC are labelled with the bird species and contain rich metadata, such as the time and location of the recording. Importantly, all samples have a crowd-sourced rating from A — E which signifies how clear the recording sample is, where A denotes samples with the highest quality and clarity. This is especially important should we wish to experiment with different signal-to-noise ratios (SNR) as the noise can be manually added to a clean recording at precise SNR levels.

Due to its scale and reputation in the birdsong classification community, all samples used in this thesis have been downloaded from the XC repository. All samples downloaded have an A rating and have been labelled as a ‘song’ rather than a ‘call’. The samples are stored in a directory labelled according to the first three letter of species' binomial name. For example, samples from the common blackbird (*Turdus merula*) are stored in a directory called 'TURMER'. This directory name acts as a label for training and test samples.

In this work we are mostly concerned with testing relative improvements in model performance after tuning various components, such as novel feature representations. To this end, we select the most simple form of classification experiment: binary classification. The two bird classes selected for all experiments herein are the common blackbird and the common nightingale (*Luscinia megarhynchos*) due to the two birds' vocalizations being acoustically similar and hence more difficult to distinguish resulting in more variable accuracy results, and the author's personal preference.

3.2 Pre-processing

Some recordings available on XC are recorded in stereo sound, so in order to reduce dimensionality without losing too much information, all recordings are first converted to mono by taking a mean average of both channels. Leading and trailing sections of background noise are then stripped from the recording using the *detectSpeech* function provided by *MATLAB*. This function uses a thresholding algorithm to detect onset and offset indices of speech [23]. The function works for birdsong since birdsong frequencies reside in a similar range to that of human speech. The function accepts various arguments to determine properties such as window length and threshold value, but from initial experiments, the function was shown to work well with birdsong with the defaults, see figure 3.1. The detected onset and offset of birdsong allows for the leading and trailing bits of background noise to be removed from the recording, thus reducing unnecessary computational time in segmenting the audio.

A highpass filter is then applied to the recording with a passband frequency of 450Hz. This value was chosen as most birdsong sits in the range of 1KHz — 10Khz, and so lower frequencies can be attributed to background noise and therefore their removal should reduce noise in the system without losing important information. The reason why frequencies above 10Khz are note removed is that, as can be seen from figure 2.1, birds typically produce higher frequency harmonics when vocalizing, shown as vertical bands moving upwards from where syllables are located on the spectrogram. These harmonics may capture useful information for the learning algorithm to utilise.

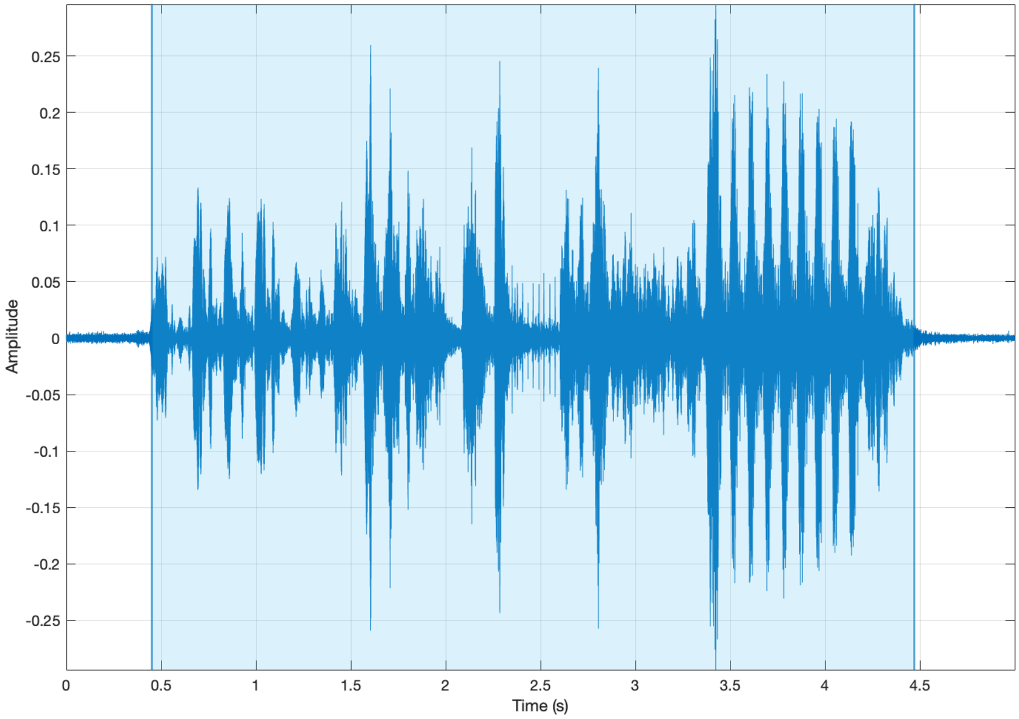

Figure 3.1: Output of the *detectSpeech* function for a recording of birdsong from a Eurasian wren. The highlighted section shows where the algorithm has detected ‘speech’, or in this case, birdsong.

After these pre-processing steps have been taken, the audio is ready for syllable segmentation.

3.3 Segmentation

Recordings downloaded from XC were also chosen based on their length. Recordings of duration 40 seconds to 120 seconds were preferred since they were likely to be long enough to include enough variety of vocalizations from one individual bird, but not too long so as to take up too much space on disk. In this thesis it was preferred to have fewer samples from more individual as opposed to more samples from fewer individuals for each species. The intention of this was to introduce more variation in the training samples for each class, especially when considering that many birds of the same species but residing in different locations have demonstrated subtle variations in vocalizations, known as dialects [5]. As a general rule of thumb, we tried to have roughly 50 training samples per individual.

This leads to the question of how to generate samples. In the literature there seem to be two main ways of segmenting recordings into samples to be used for training. The first is segmenting a recording into overlapping segments of a certain length, typically in the range of 4 — 11 seconds ([64], [14]). This has the advantage that all samples will be of fixed length and that the segmentation algorithm is very simple. However, it will likely mean that some samples will contain only background noise which, if used for training, will introduce noise into the system. These noise samples therefore may need to be removed, either manually [64] or using an algorithm [45]. The other method involves segmenting the recording into syllables ([22], [50]). This has the advantage that all training samples are likely to contain little or no noise. However the samples will be of variable length, so steps will be needed to be taken in order to compare the samples, such as padding or more advanced techniques like dynamic time warping [57]. Syllable segmentation also has the enticing prospect of being able to identify a bird species from a very short sample. This could be useful in situations where a recording is mostly corrupted by background noise but has a few small segments of clear birdsong.

In this work we attempt to combine the benefits of both approaches by first segmenting a recording to retrieve the syllables using a process described in Section 3.3.1. Then, for each syllable, the following syllables are appended until a fixed sample length is reached. If a syllable is added which pushes the sample length over the limit, then the sample is trimmed at the fixed length. This approach is motivated by Somervuo et al.'s work [57] in showing that training and classifying using single syllables returns suboptimal results, whereas using sequences of syllables gives much improved accuracy. Of course there is a tradeoff here between training with longer sequences, and hence more information, versus increased computational demand to perform training. With this tradeoff in mind, a fixed length of 13230 is chosen. Add figure of histogram here showing lengths of syllables. Typical syllable length is X, and so the fixed length should contain roughly Y syllables. This approach ensures the influence of background noise is kept to a minimum, while still maintaining information related to the temporal evolution of a particular birdsong.

3.3.1 Energy based syllable segmentation

This algorithm was first proposed by Fagerlund [21] and has been used in various research papers since ([57], [50]).

Chapter 4

Extensions of methodology

It is unlikely that everything you tried worked well, so in this chapter you may wish to describe a modified version of your method and the associated results. Explain why you were motivated to try this extension and how you think it might help to address some of the shortcomings of the method is Chapter 3.

Chapter 5

Conclusion

Summarise what you have achieved and evaluate honestly if you feel the approach has been largely successful. Explain what could be improved still and perhaps why the method is not working well (if that is the case).

Bibliography

[1] Miguel A Acevedo, Carlos J Corrada-Bravo, Héctor Corrada-Bravo, Luis J Villanueva-Rivera, and T Mitchell Aide. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. *Ecological Informatics*, 4(4):206–214, 2009.

[2] Sharath Adavanne, Konstantinos Drossos, Emre Çakir, and Tuomas Virtanen. Stacked convolutional and recurrent neural networks for bird audio detection. In *2017 25th European signal processing conference (EUSIPCO)*, pages 1729–1733. IEEE, 2017.

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121*, 2018.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] Myron Charles Baker and Michael A Cunningham. The biology of bird-song dialects. *Behavioral and Brain Sciences*, 8(1):85–100, 1985.

[6] Franz Berger, William Freillinger, Paul Primus, and Wolfgang Reisinger. Bird audio detection-dcase 2018. *DCASE2018 Challenge, Tech. Rep.*, 2018.

[7] S Binti Abdullah, Andreas Demosthenous, and Ifat Yasin. Comparison of auditory-inspired models using machine-learning for noise classification. In *International Journal of Simulation Systems, Science & Technology Special Issue: Conference Procedings UKSim2020, 25 to 27 March 2020.* United Kingdom Simulation Society, 2020.

[8] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[9] Richard K Broughton, James M Bullock, Charles George, Ross A Hill, Shelley A Hinsley, Marta Maziarz, Markus Melin, J Owen Mountford, Tim H Sparks, and Richard F Pywell. Long-term woodland restoration on lowland farmland through passive rewilding. *PloS one*, 16(6):e0252466, 2021.

[10] Clive K Catchpole and Peter JB Slater. *Bird song: biological themes and variations.* Cambridge university press, 2003.

[11] Deep Chakraborty, Paawan Mukker, Padmanabhan Rajan, and Aroor Dinesh Dileep. Bird call identification using dynamic kernel based support vector machines and deep neural networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 280–285. IEEE, 2016.

[12] Jitong Chen, Yuxuan Wang, and DeLiang Wang. A feature study for classification-based speech separation at low signal-to-noise ratios. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1993–2002, 2014.

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[14] Maximilian Crous. Polyphonic bird sound event detection with convolutional recurrent neural networks. *10.13140/RG. 2.2*, 11943, 2019.

[15] Michał Daniluk. Back-to-the-future networks: Referring to the past to predict the future. Master's thesis, University College, London, 2016.

[16] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[18] Simone Disabato, Giuseppe Canonaco, Paul G Flikkema, Manuel Roveri, and Cesare Alippi. Birdsong detection at the edge with deep learning. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 9–16. IEEE, 2021.

[19] Allison J. Doupe and Patricia K. Kuhl. Birdsong and human speech: Common themes and mechanisms. *Annual Review of Neuroscience*, 22(1):567–631, 1999. PMID: 10202549.

[20] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[21] Seppo Fagerlund. Automatic recognition of bird species by their sounds. *Finlandia: Helsinki University Of Technology*, 2004.

[22] Seppo Fagerlund. Bird species recognition using support vector machines. *EURASIP Journal on Advances in Signal Processing*, 2007:1–8, 2007.

[23] Theodoros Giannakopoulos. A method for silence removal and segmentation of speech signals, implemented in matlab. *University of Athens, Athens*, 2, 2009.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Tianxing He and Jasha Droppo. Exploiting lstm structure in deep neural networks for speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5445–5449. IEEE, 2016.

[26] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[29] Stefan Kahl, Thomas Wilhelm-Stein, Hussein Hussein, Holger Klinck, Danny Kowerko, Marc Ritter, and Maximilian Eibl. Large-scale bird sound classification using convolutional neural networks. *CLEF (working notes)*, 1866, 2017.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[31] Mineichi Kudo, Jun Toyama, and Masaru Shimbo. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20(11):1103–1111, 1999.

[32] Chiman Kwan, KC Ho, Gang Mei, Yunhong Li, Zhubing Ren, Roger Xu, Y Zhang, Debang Lao, M Stevenson, Vincent Stanford, et al. An automated acoustic system to monitor and classify birds. *EURASIP Journal on Advances in Signal Processing*, 2006:1–19, 2006.

[33] Mario Lasseck. Improved automatic bird identification through decision tree based feature selection and bagging. *CLEF (working notes)*, 1391, 2015.

[34] Mario Lasseck. Acoustic bird detection with deep convolutional neural networks. In *DCASE*, pages 143–147, 2018.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[36] Yi Ren Leng and Huy Dat Tran. Multi-label bird classification using an ensemble classifier with simple features. In *Signal and information processing association annual summit and conference (APSIPA), 2014 Asia-Pacific*, pages 1–5. IEEE, 2014.

[37] MS Likitha, Sri Raksha R Gupta, K Hasitha, and A Upendra Raju. Speech based human emotion recognition using mfcc. In *2017 international conference on wireless communications, signal processing and networking (WiSPNET)*, pages 2257–2260. IEEE, 2017.

[38] David Luther and Luis Baptista. Urban noise and the cultural evolution of bird songs. *Proceedings of the Royal Society B: Biological Sciences*, 277(1680):469–473, 2010.

[39] PETER MARLER. Chapter 5 - bird calls: a cornucopia for communication. In Peter Marler and Hans Slabbekoorn, editors, *Nature's Music*, pages 132–177. Academic Press, San Diego, 2004.

[40] Alex L McIlraith and Howard C Card. Birdsong recognition using backpropagation and multivariate statistics. *IEEE Transactions on Signal Processing*, 45(11):2740–2748, 1997.

[41] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

[42] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[43] Rajdeep Mukherjee, Dipyaman Banerjee, Kuntal Dey, and Niloy Ganguly. Convolutional recurrent neural network based bird audio detection. *DCASE challenge*, 2018.

[44] Rafael Hernández Murcia and Vıctor Suárez Paniagua. Bird identification from continuous audio recordings the icml 2013 bird challenge. *Email list of participants*, page 96, 2013.

[45] Revathy Narasimhan, Xiaoli Z Fern, and Raviv Raich. Simultaneous segmentation and classification of bird song using cnn. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 146–150. IEEE, 2017.

[46] Alberto García Arroba Parrilla and Dan Stowell. Polyphonic sound event detection for highly dense birdsong scenes. *arXiv preprint arXiv:2207.06349*, 2022.

[47] Roy D Patterson, KEN Robinson, John Holdsworth, Denis McKeown, C Zhang, and Michael Allerhand. Complex sounds and auditory images. In *Auditory physiology and perception*, pages 429–446. Elsevier, 1992.

[48] Ilyas Potamitis, Stavros Ntalampiras, Olaf Jahn, and Klaus Riede. Automatic bird sound detection in long real-field recordings: Applications and tools. *Applied Acoustics*, 80:1–9, 2014.

[49] Zhongdi Qu, Parisa Haghani, Eugene Weinstein, and Pedro Moreno. Syllable-based acoustic modeling with ctc-smbr-lstm. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 173–177. IEEE, 2017.

[50] Murugaiya Ramashini, Pg Emeroylariffion Abas, Kusuma Mohanchandra, and Liyanage C De Silva. Robust cepstral feature for bird sound classification. *International Journal of Electrical and Computer Engineering*, 12(2):1477, 2022.

[51] Murugiaya Ramashini, Pg Emeroylariffion Abas, Ulmar Grafe, and Liyanage C De Silva. Bird sounds classification using linear discriminant analysis. In *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–6. IEEE, 2019.

[52] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[53] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[54] Zachary J Ruff, Damon B Lesmeister, Leila S Duchac, Bharath K Padmaraju, and Christopher M Sullivan. Automated identification of avian vocalizations with deep convolutional neural networks. *Remote Sensing in Ecology and Conservation*, 6(1):79–92, 2020.

[55] Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very deep multilingual convolutional neural networks for lvcsr. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4955–4959. IEEE, 2016.

[56] John Shore and Rodney Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on information theory*, 26(1):26–37, 1980.

[57] Panu Somervuo, Aki Harma, and Seppo Fagerlund. Parametric representations of bird sounds for automatic species recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2252–2263, 2006.

[58] Naoya Takahashi, Michael Gygli, and Luc Van Gool. Aenet: Learning deep audio features for video analysis. *IEEE Transactions on Multimedia*, 20(3):513–524, 2017.

[59] Vlad M Trifa, Alexander NG Kirschel, Charles E Taylor, and Edgar E Vallejo. Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *The Journal of the Acoustical Society of America*, 123(4):2424–2431, 2008.

[60] Xavier Valero and Francesc Alias. Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE transactions on multimedia*, 14(6):1684–1689, 2012.

[61] Willem-Pier Vellinga and Robert Planqué. The xeno-canto collection and its relation to sound recognition and classification. In *CLEF (Working Notes)*, 2015.

[62] Yuxuan Wang, Kun Han, and DeLiang Wang. Exploring monaural features for classification-based speech segregation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(2):270–279, 2012.

[63] Zhong-Qiu Wang and DeLiang Wang. A joint training framework for robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):796–806, 2016.

[64] Na Yan, Aibin Chen, Guoxiong Zhou, Zhiqiang Zhang, Xiangyong Liu, Jianwu Wang, Zhihua Liu, and Wenjie Chen. Birdsong classification based on multi-feature fusion. *Multimedia Tools and Applications*, 80:36529–36547, 2021.

[65] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.