Implementation of Bird Species Detection Algorithm using Deep Learning

Niyati Jain[1,*] , *Medini* Kamble[1,**], *Amruta* Kanojiya[1,***], and *Chaitanya* Jage[1,****]

[1]Department of Electronics Engineering Ramrao Adik Institute of Technology Nerul, Navi Mumbai 400706

Abstract. Automatically identifying what types of the bird is present in the sound recording using the monitor reading. To distinguishing automatic birds based on their sound patterns.This is useful in the field of ornithology for studying bird species and their behavior based on their sound. Proposed method will be used to distinguish birds automatically using different sound processing methods and mechanical learning methods based on their chirping patterns. We propose a sequential model for audio features within a short interval of time. The model will be used Mel Frequency Cepstral Coefficients to extract features from the audio files and presented it in the model. The proposed work classifies the data set containing three species of bird, and outperform support vector machines.

1 Introduction

An increasing interest in exploring the different impacts of biodiversity is currently taking place around the world. With the rapid decline in wildlife populations around the world due to environmental pollution, there has been an ongoing effort over the years to monitor species that are seen as positive indicators of diversity. Tracking the number of birds in their habitat is one such endeavor, as birds are a good indicator of environmental change. For example, it allows researchers to capture important information such as climate change, migration patterns, pollution and disease outbreaks in the environment. Because birds play such an important role for the environment, a lot of effort has been focused on bird conservation[1]. In recent years, deep-reading strategies have changed the performance of machines in reading, viewing, and processing text. Significant improvements in many classification functions are reported using deep constructs, where deep accumulation neural networks are widely used in computer vision functions.

Human hearing frequency is usually between 20 to 20,000 Hz As we age, we all often lose the ability to hear high waves. Many bird songs have frequencies between 1000 Hz to 8000 Hz, making them readily audible to human hearing. At high altitudes, many warblers, sparrows, warblers, nightingales and many other birds produce sounds up to 8000 Hz or more[2]. When Convolutional Neural Networks(CNN) knows exhausted filters in terms of frequency and time, it faces the limit of deep neurons, lack of time and frequency. The use of deeper and more efficient CNN will also be common and will reflect modern performance in finding objects and image classification challenges. The use of CNN is also popular in voice recognition and voice recognition applications in which audio signals are often converted into spectrum and are taken in the form of inputs on CNN[3].

In audio classification systems, to perform or create models in machine learning, in the first place, we have to extract acoustic features from the audio files. By extracting features, generally the audio files are broken into single frames to make them easy to work. Mel Frequency Cepstral Coefficients(MFCC) is used to extract features from bird audio files. Our main contribution in this paper is create a model that detects a bird on providing an audio file of a bird[4, 5].

We have done an exploratory data analysis (EDA) on the audio files for examining and understanding the data and checking for unlabeled data and errors. To check missing values, we have used Pandas library. There are 2 libraries in Python to work on audio files and they are- Librosa and Scipy. In this paper, we have chosen Librosa because Librosa makes all the files into same sample rate (mono channel) and making the data easy to work on audio processing.

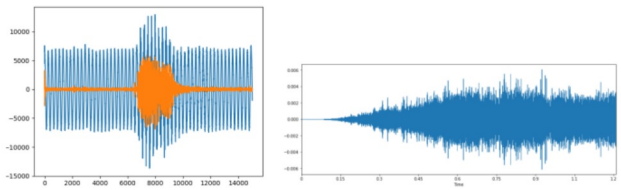

Figure 1: Data is shown by plotting it using Scipy(left) and Librosa(right)

Before creating independent and dependent features, we have to extract features of audio files by using MFCC. By extracting feature, we can uniquely identify the audio signal to predict that in which class it will belong. After features have been extracted, Data frame is created us-

*e-mail: niy.jai.rt19@rait.ac.in
**e-mail: med.kam.rt19@rait.ac.in
***e-mail: amr.kan.rt19@rait.ac.in
****e-mail: chaitanya.jage@rait.ac.in

ing those features and the class name. Now that we have both the independent and dependent features, well split the dataset into training and testing dataset for our model.

```
Out[20]: array([[-5.0569110e+02, -4.5887506e+02, -4.3998956e+02, ...,
                 -4.1403049e+02, -4.1441818e+02, -4.1519479e+02],
                [ 4.9381628e+00,  2.7463079e+01,  4.1891090e+01, ...,
                  2.6102783e+01,  2.4007418e+01,  2.6515617e+01],
                [-2.7153938e+01, -5.2383221e+01, -5.2293556e+01, ...,
                  -7.1757324e+01, -7.5127136e+01, -7.2742760e+01],
                ...,
                [-6.6145706e-01,  6.2263441e+00,  5.5203032e+00, ...,
                  -3.4690235e+00, -5.3325167e+00, -5.6686220e+00],
                [ 8.7596697e-01,  3.1809101e+00,  2.9348588e-01, ...,
                  -5.0395322e+00, -4.3937540e+00, -4.4510584e+00],
                [ 7.5026476e-01, -5.1273794e+00, -1.0170565e+01, ...,
                  -1.2309712e+00,  1.1300253e+00,  1.5362844e+00]], dtype=float32)
```

Figure 2: MFCC values in an array

We provide our training x and y dataset to the model and train the model and check the accuracy. Finally, we check our model by providing a testing dataset sample and is being checked if the model is working properly.
This paper is organized as follows: Section 2 comprises of the basic concept of Mel Frequency Cepstral Coefficients has been presented briefly.Section 3 has a detailed explanation of Deep Learning and the model we have used in this work has been written. Section 4 represents the Implementation steps which were performed while designing the algorithm. Finally all the results are given and the conclusions drawn from this work are presented in section 5 and section 6.

2 Mel Frequency Cepstral Coefficients(MFCC)

The first thing for any automated speech recognition system is to extract or remove the features that is identifying the audio signal which are suitable to identify language content and to discard all other data or information which contains elements such as extra background sound.

Therefore the Mel Frequency cepstral coeffiients considers human perception of sensitivity to appropriate frequencies by converting normal frequency into a Mel scale,and thus better suited to speech recognition activities.

MFCCs were introduced by Davis and Mermelstein in 1980 and is the most widely used feature in automated speech recognition system. The feature count is small enough to force us to learn the information of the audio. 12 parameters are related to the amplitude of frequencies. It provides us enough frequency channels to analyze the audio[4, 6–8].

The flow of extracting the MFCC features is given below, As shown in Figure 3:Flow Diagram of MFCC Calculation, MFCC consists of seven computational steps[9**?**]. Each step has its function and mathematical approaches as discussed briefly in the following:

- Step 1:In this steps it detects signal passing through a filter which will emphasize high waves. This will increase the signal strength at higher frequencies.

$$Y\,[n] = X\,[n] - 0.95X\,[\,n\text{-}1] \tag{1}$$

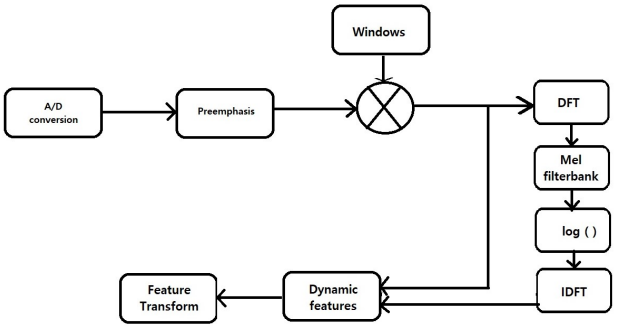

Figure 3: Flow Diagram of MFCC Calculation

Now lets consider a = 0.95 which will make 95% of any one sample is presumed to originate from previous sample.

- Step 2: Framing is the procedure of segmenting the speech samples obtained from analog to digital conversion (ADC) right into a small body with the period in the range of 20 to 40 msec. The voice sign is divided into frames of N samples. adjacent frames are being separated by using M (M < N).

  Normal values used are M=100 and N=256

- Step 3: Hamming window is used as window form through thinking about the subsequent block in function extraction processing chain and integrates all the closest frequency lines. The Hamming window equation is given as: If the window is described as W (n), $0 \leq n \leq N-1$

  Wherein,

  N = numeric values of samples in each frame

  Y[n] = Output signal

  X (n) = input signal

  W (n) = Hamming window,

  then the result of windowing signal is shown below:

$$Y(n) = X(n) * W(n) \tag{2}$$

$$W\,(n) = 0.54 - 0.46cos[2\pi n/N - 1] \tag{3}$$

$$\text{for } 0 \;\leq\; n \leq N\text{ - }1$$

- Step 4: Fast Fourier Transform is used to convert N samples from time domain to frequency domain.The Fourier Transform is to convert the convolution of the glottal pulse U[n] and the vocal tract impulse response H[n] in the time domain. This statement supports the equation below:

$$(Y(w) = FFT[h(t) * X(t)] = \; X(w) * X\,(w) \tag{4}$$

If X (w), H (w) and Y (w) are the Fourier Transform of X (t), H (t) and Y (t) respectively.

- Step 5: Mel filter bank processing - The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. The bank of filters according to Mel scale as shown in figure 4:Triangular filters of MFCC is then performed.

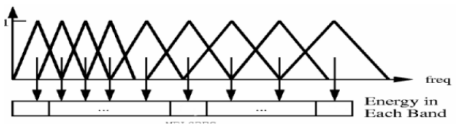

Figure 4: Triangular filters of MFCC

This figure shows a hard and fast of triangular filters that are used to compute a weighted sum of filter spectral components so that the output of system approximates to a Mel scale.Every filters magnitude frequency response is triangular in form and same to unity at the centre frequency and reduce linearly to 0 at centre frequency of two adjoining filters [7, 8]. Then, every filter output is the sum of its filtered spectral components. After that the subsequent equation is used to compute the Mel for given frequency f in HZ:

$$(\text{F (Mel)} = [2595 * log[1 + f]700] \tag{5}$$

- Step 6: Discrete Cosine rework this is the process to convert the log Mel spectrum into time domain the usage of Discrete Cosine rework (DCT). The end result of the conversion is known as Mel Frequency Cepstrum Coefficient.
- Step 7: Delta strength and Delta Spectrum The voice signal and the frames changes, along with the slope of a formant at its transitions. Thirteen delta or pace functions (12 cepstral functions plus power), and 39 capabilities a double delta or acceleration function are introduced. The energy in a body for a sign x in a window from time pattern t1 to time pattern t2, is represented at the equation underneath:

$$\text{Energy} = \Sigma X^2[t] \tag{6}$$

Every thirteen delta functions represents the exchange among frames inside the equation 8 corresponding cepstral or strength characteristic, even as every of the 39 double delta capabilities represents the trade among frames inside the corresponding delta capabilities.

$$D(t) = (c(t + 1) - c(t - 1))/2 \tag{7}$$

3 Deep Learning

Deep learning allows computer models composed of multiple processing layers to read data presentations with multiple output levels[10].These methods have greatly improved the state of the art in speech recognition, material recognition, object discovery and many other fields such as drug discovery and genomics. In-depth learning finds complex structures in large data sets using a back-to-back algorithm to show how a machine should modify its internal parameters used to calculate representation in each layer from representation in the previous layer. Deep convolutional networks have brought success to image, video, speech and audio processing, while repetitive networks illuminate successive data such as text and speech[11].

ConvNets or CNNs can process data that comes in many categories, such as speech, text, image, and video. ConvNets is made up of different categories. The first stage is the convolutional layer, where a weight set called the filter bank is combined with the insertion vector. This local measured amount is then transferred to an indirect object such as ReLU called activation function. Two or three stages of conversion, opening operations, and integration layers are packaged, followed by fully integrated layers. Back gradients spread by ConvNet train all the weight of the filter banks. This sequence structure will allow high-level features (details) to be achieved by designing a low level. Convolutional and integration layers in ConvNets are promoted by cells in visual neuroscience

In this work model we use the Keras model. Keras is a neural network application programming interface (API) for python integrated with tensorflow[12].TensorFlow is used to build machine learning models. It gives us a simple and easy-to-use way to define a neural network, which can be built by TensorFlow. Keras facilitates the implementation of complex neural networks with its easy-to-use framework[13, 14].

3.1 Keras Model Overview

Models are a core business that we will be working with when using Keras. Models used to describe TensorFlow neural networks by specifying the features, functions, and layers we want. Keras provides a number of APIs we can use to define your neural network, including:

1. Sequential API, which allows you to create a model layer according to the multi-problem layer. Straightforward (just a simple layer of layers), but limited to single input, single layer output stacks[15].

2. Functional API, which is a full-featured API that supports structural models. It is much more flexible and sophisticated than a sequential API.

3. Model Subclassing, which lets you use everything from scratch. It is suitable for research and very complex application situations, but is rarely used in practice[16].

TensorFlow provides a complete machine learning platform that provides advanced and low-level skills to build and supply machine learning models. The sequential model is suitable for a blank layer stack where each layer has one input tensor and one outgoing tensor. In this model we have used a Dense layer, a function of dropout, a function of Activation and flatten.

1. Dense layer: These layers are used when the connection can be between any element to any other

object in the data area. As between two layers of size n1 and n2, there may be a connection of $n1 * n2$ and this is called Dense.

2. Dropout function: Stopping is a way to cut as many connections between the elements by reducing weight (edges) as much as possible. Reducing associations can be used between any layers that stop weight development at the edges. Another important difference here is that it does not have the corresponding weights. There is just throwing things away.

3. Activation function: Opening functions give the model the ability to add compliance to the model. Here a relaxation exercise is performed, which removes a lot of weight.

4. Flatten function: This layer is used when you find a multi-dimensional output and want to edit it to transfer it to a dense layer. Outputs from flat flats are transferred to the phase model or retrieval function you want to achieve.

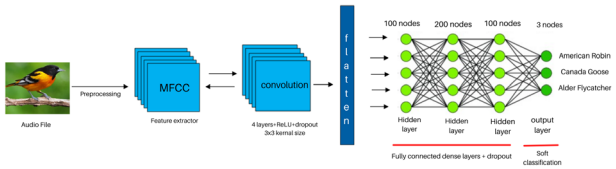

Figure 5: Bird audio detection using Convolutional Neural Network

The first layer by default is the input layer.Afterwards:

Table 1: Layer details of our Model

Layer Index	Layer type	Note
1	Dense	40 Inputs,ReLU activation with 100 dense nodes
2	Dense	ReLU activation with 200 dense nodes
3	Dense	ReLU activation with 100 dense nodes
4	Dense	3 output nodes,Softmax activation

4 Design Algorithm and Result

4.1 Design of Implementation

Some of the functions and libraries used while detecting bird audio files on Jupyter using Python are:

Figure 6: Model workflow on how the project is created

- Librosa: This Library is a python package for music and audio analysis. Libroso is basically used when dealing with audio data such as making music (using LSTM, s), Automatic Speech Recognition. Librosa helps to visualize sound signals and performs the extracts from it using different signal processing techniques [17].
- MFCC: This function comes under Librosa. In short, The mel frequency cepstral coefficients (MFCCs) of the signal are a small set of factors (usually about 10-20) that briefly describe the general shape of the spectral envelope. This feature is one of the most important ways to remove the spectral envelope feature. audio signal and is widely used whenever it works on audio signals[3, 8].
- Tensorflow: This library is uses for data flow graphs to represent calculations, shared status, and functions that change that status. Map graphs of data flow across multiple machines in the collection, and within the machine on all multiple computer devices, including multicore CPUs, standard GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs). This structure provides flexibility for the application developer[14].
- Pandas: Python library for the rich data structures and working tools and standard data sets mathematics, finance, social science, and many other fields. The library provides integrated, accurate methods for committing common data fraud as well analysis on such data sets. It aims to be the basic layer of the future mathematical computer in Python[12].
- Sklearn: scikit-learn, which is a Python module that integrates a wide range of modern machine learning algorithms for moderate and unregulated intermediate scale

problems, is one of the most popular machine learning tools [18].

- Label Encoder: This function is used to make the text of the label simply provide the total value for the total possible variance of the category [19].
- Sequential: It is based on a rotating process between the suggestion of a new hyperparameter configuration for testing and a review of the dynamic model of the relationship between hyperparameter configuration and holdout set performances. So, as the model learns about this relationship, increases its ability to lift better hyperparameter setting and gradually converts to the best solution[15].

Following are the steps to be followed while providing a audio file to the model in Jupyter software:

1. Start with choosing a file from the device. Again, we need to preprocess data where we need to extract the features of that specific data itself with the help of MFCC.
2. Now that we have features of the particular file, we predict the class labels with the help of model that is created. From this, model provides the label.
3. Finally, we have to inverse transform the label to get the class name that will provide us which bird the audio is relating to.

4.2 Result

In this work, we have provided 6342 audio files of 3 birds as the dataset. A random file name is selected from the dataset and then passed to the features-extractor class which returns the class label from the model(numeric value) which is later given out as bird type using inverse transform.

Prediction-feature saves the return extracted features. We have 2 variable x and y , where x represents extracted features and y represents name of the birds. By using the extracted feature, we can predict which type of bird it is by using CNN module. Now we have features by random file module is predicted from this. It is then compared with the given dataset and results the results in array. For a record, if the predicted value is equal to the actual value, it is considered accurate. We then calculate Accuracy by dividing the number of accurately predicted records by the total number of records.Here we get an accuracy of 70 percentage.

5 Conclusion

In this work, We have implemented an algorithm using Sequential model providing the best accuracy at 70 percentage when using the mean values of all features earlier mentioned. By implementing this model, we have classified three birds- Alder flycatcher, Canada Goose and American Robin using only audio data from xeno-canto dataset.This work will really help the researchers to implement the Deep Learning Sequential model to detect bird type using their audio files.

Figure 7: Alder Flycatcher(left) and American Robin(right)

```
test_accuracy=model.evaluate(X_test,y_test,verbose=0)
print(test_accuracy[1])
```

```
0.699999988079071
```

Figure 8: Accuracy of 70%

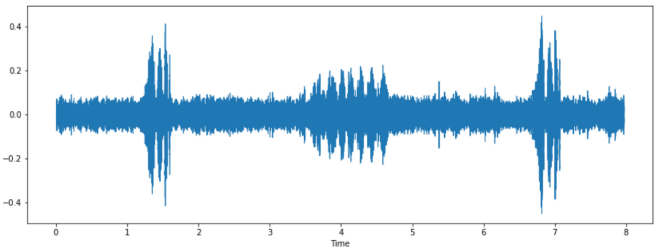

Figure 9: Audio waveform of Alder Flycatcher

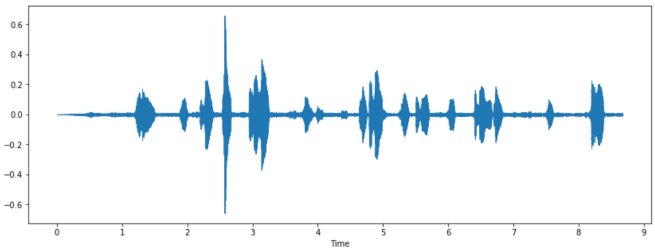

Figure 10: Audio waveform of Canada Goose

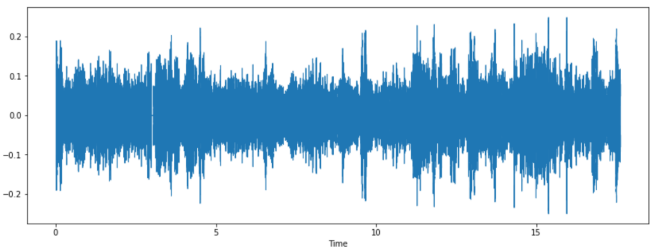

Figure 11: Audio waveform of American Robin

References

[1] J.A. Kogan, D. Margoliash, The Journal of the Acoustical Society of America **103**, 2185 (1998)

[2] D.E. Balmer, S. Gillings, B. Caffrey, R. Swann, I. Downie, R. Fuller, *Bird Atlas 2007-11: the breeding and wintering birds of Britain and Ireland* (BTO Thetford, 2013)

[3] C.H. Lee, Y.K. Lee, R.Z. Huang, Journal of Information Technology and Applications () **1**, 17 (2006)

[4] M. Likitha, S.R.R. Gupta, K. Hasitha, A.U. Raju, *Speech based human emotion recognition using MFCC*, in *2017 international conference on wireless communications, signal processing and networking (WiSPNET)* (IEEE, 2017), pp. 2257–2260

[5] F. Briggs, R. Raich, X.Z. Fern, *Audio classification of bird species: A statistical manifold approach*, in *2009 Ninth IEEE international conference on data mining* (IEEE, 2009), pp. 51–60

[6] A. Coates, A.Y. Ng, in *Neural networks: Tricks of the trade* (Springer, 2012), pp. 561–580

[7] P. Somervuo, A. Härmä, *Analyzing bird song syllables on the self-organizing map*, in *Workshop on Self-Organizing Maps (WSOM03)* (2003)

[8] V. Tiwari, International journal on emerging technologies **1**, 19 (2010)

[9] A. Selin, J. Turunen, J.T. Tanttu, EURASIP Journal on Advances in Signal Processing pp. 1–9 (2006)

[10] Y. LeCun, Y. Bengio, G. Hinton, nature **521**, 436 (2015)

[11] S. Wei, S. Zou, F. Liao et al., *A comparison on data augmentation methods based on deep learning for audio classification*, in *Journal of Physics: Conference Series* (IOP Publishing, 2020), Vol. 1453, p. 012085

[12] W. McKinney et al., Python for high performance and scientific computing **14**, 1 (2011)

[13] L. Breiman, Machine learning **45**, 5 (2001)

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., {*TensorFlow*}*: A System for* {*Large-Scale*} *Machine Learning*, in *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (2016), pp. 265–283

[15] Y. Even-Zohar, D. Roth, arXiv preprint cs/0106044 (2001)

[16] C. Kwan, G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, K. Ho, *Bird classification algorithms: Theory and experimental results*, in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, 2004), Vol. 5, pp. V–289

[17] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, O. Nieto, *librosa: Audio and music signal analysis in python*, in *Proceedings of the 14th python in science conference* (Citeseer, 2015), Vol. 8, pp. 18–25

[18] F. Yang, X. Wang, H. Ma, J. Li, BMC Medical Informatics and Decision Making **21**, 1 (2021)

[19] J.T. Hancock, T.M. Khoshgoftaar, Journal of big data **7**, 1 (2020)

[20] F. Briggs, B. Lakshminarayanan, L. Neal, X.Z. Fern, R. Raich, S.J. Hadley, A.S. Hadley, M.G. Betts, The Journal of the Acoustical Society of America **131**, 4640 (2012)

[21] P. Somervuo, A. Harma, S. Fagerlund, IEEE Transactions on Audio, Speech, and Language Processing **14**, 2252 (2006)

[22] M.A. Acevedo, C.J. Corrada-Bravo, H. Corrada-Bravo, L.J. Villanueva-Rivera, T.M. Aide, Ecological Informatics **4**, 206 (2009)

[23] J. Pons, X. Serra, *Randomly weighted cnns for (music) audio classification*, in *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (IEEE, 2019), pp. 336–340

[24] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, arXiv preprint arXiv:1810.04805 (2018)

[25] M. Stracy, O. Snitser, I. Yelin, Y. Amer, M. Parizade, R. Katz, G. Rimler, T. Wolf, E. Herzel, G. Koren et al., Science **375**, 889 (2022)

[26] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, Z. Zhu, IEEE Access **7**, 88058 (2019)

[27] M. Aaron, M. Elad, Signal Processing, IEEE Transactions on **54**, 4311 (2006)

[28] T.M. Aide, C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega, R. Alvarez, PeerJ **1**, e103 (2013)

[29] R. Caruana, A. Niculescu-Mizil, *An empirical comparison of supervised learning algorithms*, in *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 161–168

[30] S.S. Stevens, J. Volkmann, E.B. Newman, The journal of the acoustical society of america **8**, 185 (1937)

[31] K. Seyerlehner, G. Widmer, P. Knees, *Frame level audio similarity-a codebook approach*, in *Proc. of the 11th Int. Conf. on Digital Audio Effects (DAFx-08)* (2008), p. 31

[32] P. Minka, Tech. rep., Tech. Rep., Microsoft Research (2003)

[33] R.E. Kass, P.W. Vos, *Geometrical foundations of asymptotic inference* (John Wiley & Sons, 2011)