Bird Species Classification on MLSP Dataset

CS 726: Advance Machine Learning Project

by

Deep Modh (140050002)
Himanshu Agarwal (163050001)
Prashanth Manjunath (163050043)
Vivek Kumar Arya (163050087)

under the guidance of

Prof. Sunita Sarawagi

Department of
Computer Science & Engineering
Indian Institute of Technology, Bombay

Contents

1 Introduction **1**

1.1 Problem statement 1

1.2 Motivation 1

1.3 Dataset 1

2 Related Literature **2**

2.1 Audio Processing 2

2.1.1 Mel-frequency cepstral coefficients 2

2.1.2 Chromagram 2

2.1.3 Melspectrogram 3

2.2 Winning model solution 3

2.3 Convolutional Neural Networks 3

2.3.1 VGG Neural Network 3

2.4 LSTM 4

2.5 Related Work 5

3 Approaches tried **7**

3.1 Feature Extraction 7

3.2 Image Networks 7

3.2.1 LSTM 8

3.2.2 Bidirectional LSTM 8

3.2.3 Variant of VGG network 8

3.2.4 Custom CNN architecture 8

3.2.5 Residual Neural Networks 8

3.3 Raw feature Networks 9

3.3.1 Feed forward neural network 9

3.3.2 CNN 9

4 Experiments and results **10**
4.1 Image Networks 10
4.1.1 LSTM 10
4.1.2 Bidirectional LSTM 10
4.1.3 Variant of VGG network[6] 11
4.1.4 Custom CNN architecture 12
4.1.5 Residual Neural Networks 13
4.2 Raw feature Networks 13
4.2.1 Feed forward neural network 13
4.2.2 CNN[5] 14
4.3 Libraries Used 14

5 Effort **15**
5.1 Time spent in different parts 15
5.2 Most challenging part 15

6 Future Work **16**

Bibliography **18**

Chapter 1

Introduction

1.1 Problem statement

Ours is a multi-label classification task based on a 2013 kaggle challenge. We are given audio samples which consists of bird sounds that are present in 10 second audio clips along with background noise. From this, we need to classify the set of bird species which are present in an audio recording. There are a total of 19 species.

1.2 Motivation

Bird behaviour and population trends have been of interest to wild life scientist, and can be helpful in predicting various environment changes. Earlier this was mostly done by human efforts, but implementing and improving machine learning techniques has automated this process.

1.3 Dataset

The dataset consists of 645 audio recordings of 10 seconds each wherein multiple and variable number of bird sounds are present along with background noise. Out of these, 322 audio recordings are for training and 323 audio recordings are test data. The training data is complemented with the bird species labels as a text file – ”reclabelstesthidden.txt”. In this text file, for each train sample multiple bird labels are given, and for each test sample no labels are given. Hence, this problem becomes a multi-label supervised classification task, wherein we need to predict the probability of each bird being present in the audio sample.

Chapter 2

Related Literature

In this chapter, we discuss various methodologies which have been found useful for audio classification tasks. CNNs and LSTMs have been shown to perform well for various image recognition tasks. Audio recordings can be mapped to spectrograms, and hence image classification literature is relevant in this task.

2.1 Audio Processing

2.1.1 Mel-frequency cepstral coefficients

The Mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. In MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear ”spectrum-of-a-spectrum”)[**7**]. We used **Librosa** library to extract MFCCs coefficients of each audio file of dataset.

2.1.2 Chromagram

Chroma-based features, which are also referred to pitch class profiles, are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the equal-tempered scale. Given a music representation (e.g. a musical score or an audio recording), the main idea of chroma features is to aggregate for a given local time window (e.g. specified in beats or in seconds) all information that relates to a given chroma into a single coefficient. Shifting the time window across the music representation results in a sequence of chroma features each expressing how the representation's pitch content within the time window is spread over the twelve chroma bands. The resulting time-chroma representation is also referred to as chromagram[**3**]. We used Librosa library to compute a chromagram from a power spectrogram of a audio file.

2.1.3 Melspectrogram

Melspectrogram is the spectrogram of a waveform computed on mel-scale. The mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The name mel comes from the word melody to indicate that the scale is based on pitch comparisons.

2.2 Winning model solution

MLSP 2013 kaggle competition has a best score of 0.95611, this models uses raw features from audio and feeds it to a random forest classifier[**9**]. Code can be found at https://github.com/gaborfodor/MLSP2013

2.3 Convolutional Neural Networks

CNNs are a subclass class of feed forward neural networks, which are known for effectively modelling time and space invariance in a input sequence. CNNs are famous in the literature for image recognition tasks and involve minimal preprocessing. CNNs usually share weights among the filters which makes them more effective in learning an image filter, also reducing the number of parameters. A cnn generally has following sequence of layers

1. Pair of convolution and pooling layers, several times

2. Fully connected layers

2.3.1 VGG Neural Network

Vgg Network is a standard cnn architecture from the visual geometry group, which has shown to perform well on images. It comes in two variants 16-layer and 19-layer.

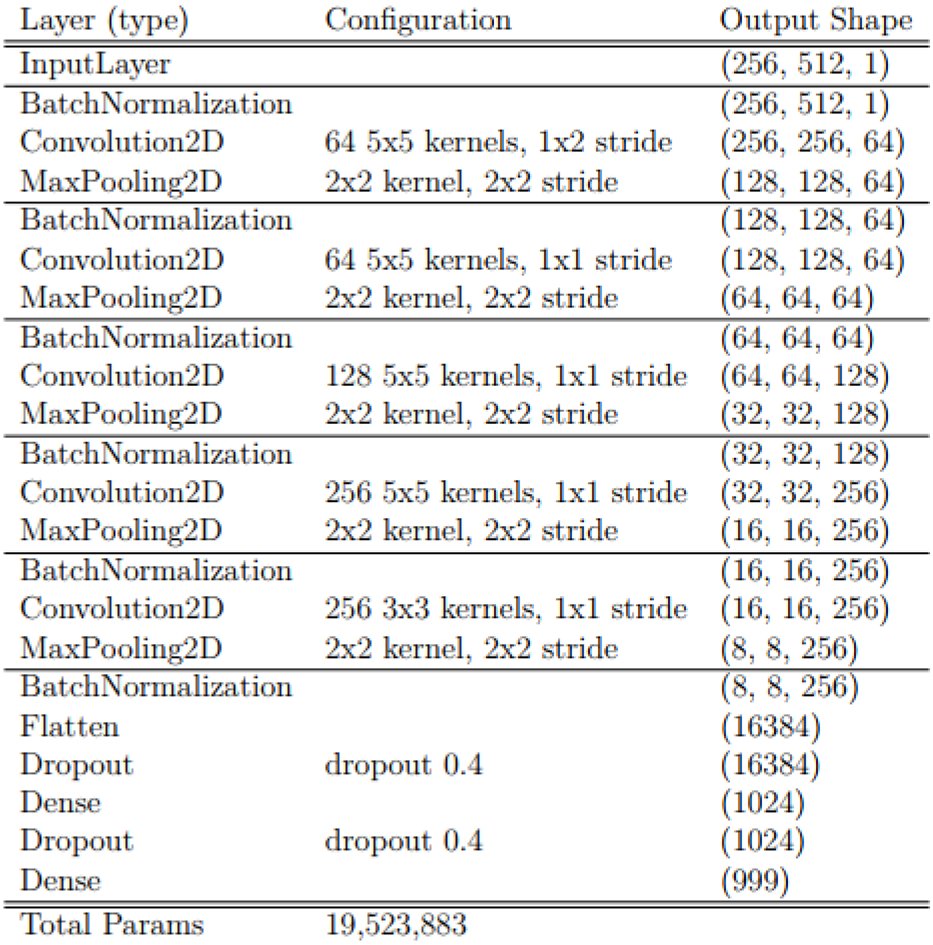
The main objective of this network was to explore the results from increasing the depth in cnn model and the results were quite successful.

2.4 LSTM

Long short term memory networks have been designed for sequence prediction in spatial inputs such as images, videos and have been found to produce good results. These are used in recurrent neural networks to effectively model the state of the model, which means to effectively remember the long and short term context in a model. LSTMs commonly have a cell and three gates − an input gate, output gate and a forget gate.

2.5 Related Work

One of the related work we looked at is by Sprengel et. al [4]. This approach won international BirdCLEF 2016 Recognition Challenge. They have used convolutional neural network architecture which is summarized below. The first column contains the type of the layer, the second column contains the configuration of the layer, and the third column contains the output shape of the layer (rows, columns, channels).



Layer (type)	Configuration	Output Shape
InputLayer		(256, 512, 1)
BatchNormalization		(256, 512, 1)
Convolution2D	64 5x5 kernels, 1x2 stride	(256, 256, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(128, 128, 64)
BatchNormalization		(128, 128, 64)
Convolution2D	64 5x5 kernels, 1x1 stride	(128, 128, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(64, 64, 64)
BatchNormalization		(64, 64, 64)
Convolution2D	128 5x5 kernels, 1x1 stride	(64, 64, 128)
MaxPooling2D	2x2 kernel, 2x2 stride	(32, 32, 128)
BatchNormalization		(32, 32, 128)
Convolution2D	256 5x5 kernels, 1x1 stride	(32, 32, 256)
MaxPooling2D	2x2 kernel, 2x2 stride	(16, 16, 256)
BatchNormalization		(16, 16, 256)
Convolution2D	256 3x3 kernels, 1x1 stride	(16, 16, 256)
MaxPooling2D	2x2 kernel, 2x2 stride	(8, 8, 256)
BatchNormalization		(8, 8, 256)
Flatten		(16384)
Dropout	dropout 0.4	(16384)
Dense		(1024)
Dropout	dropout 0.4	(1024)
Dense		(999)
Total Params	19,523,883	

Another work in Masters thesis by John Martinsson titled *'Bird Species Identification using Convolutional Neural Networks'* [1] provides significant insight into the problem we are trying to solve. The data set being used here is from LifeCLEF 2016 bird identification task (BirdCLEF 2016). The architecture being is used is convolutional neural network with batch normalization.

The configuration of a basic block, e.g., "64 3x3 kernels, 2x2 stride" refers to the number of filters of the convolutional layers in the basic block is 64, their kernel sizes are 3x3, the stride size of the first convolutional layer is 2x2, whereas the stride size of the second convolutional layer is 1x1.

Layer (type)	Configuration	Output Shape
InputLayer		(256, 512, 1)
Convolution2D	64 7x7 kernels, 2x2 stride	(128, 256, 64)
MaxPooling2D	3x3 kernel, 2x2 stride	(64, 128, 64)
BasicBlock	64 3x3 kernel, 1x1 stride	(64, 128, 64)
BasicBlock	64 3x3 kernel, 1x1 stride	(64, 128, 64)
BasicBlock	128 3x3 kernel, 2x2 stride	(32, 64, 128)
BasicBlock	128 3x3 kernel, 1x1 stride	(32, 64, 128)
BasicBlock	256 3x3 kernel, 2x2 stride	(16, 32, 256)
BasicBlock	256 3x3 kernel, 1x1 stride	(16, 32, 256)
BasicBlock	512 3x3 kernel, 2x2 stride	(8, 16, 512)
BasicBlock	512 3x3 kernel, 1x1 stride	(8, 16, 512)
AveragePooling2D	8x16 pool size, 1x1 stride	(1, 1, 512)
Flatten		(512)
Dense	He normal, softmax	(999)
Total Params	11,691,751	

The architecture of a basic block is as following : The input layer is simply the output of the layer to which the basic block has been connected. Meaning that if the previous layer has an output of shape (n, m, d) then the input layer gets that output shape. Both convolutional layers use the number of filters which is specified when constructed, but only the first convolutional layer uses the specified stride size, the second always has a stride size of 1x1.

Layer (type)	Configuration	Output Shape
InputLayer		(n, m, d)
BatchNormalization		(n, m, d)
Activation	ReLU	(n, m, d)
Convolution2D	c kernels, sxs stride	(n/s, m/s, c)
BatchNormalization		(n/s, m/s, c)
Activation	ReLU	(n/s, m/s, c)
Convolution2D	c kernels, 1x1 stride	(n/s, m/s, c)
Merge	[InputLayer, Convolution2D]	(n/s, m/s, c)

Chapter 3

Approaches tried

In this chapter we discuss the approaches we have tried to feed the audio inputs to neural networks and discuss the network architectures we have tried on these audio recordings.

3.1 Feature Extraction

Audio can be used as input to a network architecture in various ways, we have tried two methodologies for feeding the audio wav files to the neural networks namely

1. Extracting the spectrograms (Images) from the audio files.

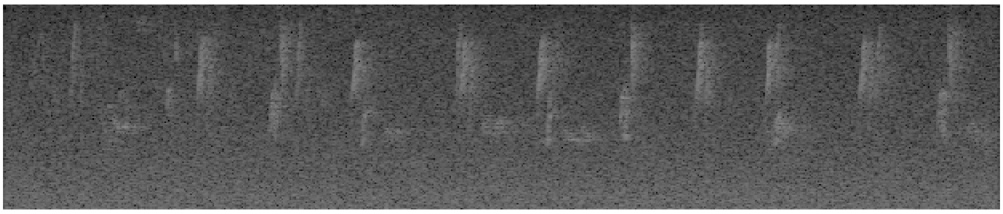


Figure 3.1: Spectrogram of an audio file.

2. Extracting raw features from the audio file : MFCCs, Chroma features, Mel-spectrogram and Spectral Contrast.

3.2 Image Networks

In this section we discuss about the networks we tried for image classification task i.e. classifying the audio spectrograms corresponding to audio recordings. We tried out the following networks

3.2.1 LSTM

As discussed in section 2.4, we ran few instances of lstm model, which takes as input a re-sized spectrogram image of size (623,128). This input is processed through the LSTM model, and finally passes through a 19 (number of bird species) dimensional dense sigmoid/softmax output layer

3.2.2 Bidirectional LSTM

We modified the above LSTM model into a bidirectional lstm model with similar settings and the results observed were quite similar.

3.2.3 Variant of VGG network

We ran a vgg like cnn network, which has 4 convolution layers, 2 max-pooling layers, 1 fully connected dense layer, along with dropout [6].

3.2.4 Custom CNN architecture

We built CNN model by considering different types of layers, changing their number of filters, kernel size and stride size. We also did max pooling wherever necessary and dropout at the last few layers. The input to the model is spectrogram image of the audio file.

3.2.5 Residual Neural Networks

Deep networks are usually harder to train because of vanishing or exploding gradients. This intuitively can be understood as data disappearing through too many layers of the network, meaning output from a shallow layer was diminished through the greater number of layers in the deeper network, yielding a worse result. To resolve the vanishing gradient problem Residual Neural Networks, ResNets, are used [8]. We ran Residual network with 18 layers and take spectrogram as input.

3.3 Raw feature Networks

3.3.1 Feed forward neural network

Winning solution in the kaggle competition used raw audio features of dimension 6079 and was fed to a random forest model. We modified this by feeding these features to a feed forward neural network. This gives an accuracy of 69.716, which is 10% lesser than our best model discussed in the next section. We believe that this is because of the 6079 input feature, as our training data is less model could not converge to optimum values.

3.3.2 CNN

In this model, we feed MFCCs, melspectrogram, chroma and spectral contrast as features of an audio file as input. We considered 40 mfccs coefficients and computed chroma, spectral contrast using short term Fourier transform of the audio file. These features (total 193) are fed as input to multiple 1D convolutional layers, 1D max-pooling layers interleaved as discussed in the **experiments section**. A fully connected dense output layer is added to the network along with a dropout rate. We trained this network to convergence, and it performs reasonably well as discussed in the **experiments section**.

Chapter 4

Experiments and results

In this chapter, we talk about the experiments we have performed and the results on the various network architectures discussed in chapter 3. Score of each model is calculated directly by submitting to kaggle. Kaggle score metric is area under the roc curve. All experiments were run in python using the Keras library. The machine used had a GPU 12GB TitanX.(Blossom Server). All of the below experiments takes nearly 20 mins for convergence for each hyperparameter combinations. Hence we had a lot of work regarding hyperparameter tuning and layer modifications for each of the below architectures.

Link for the written code: https://github.com/himanshu123j/AML-Project

4.1 Image Networks

4.1.1 LSTM

Image Size	lstm layer1	lstm layer2	lstm layer3	Kaggle Score
(310, 64)	32	32	32	0.69907
(623, 128)	32	32	32	0.70101

Table 4.1: Accuracy results in lstm model

4.1.2 Bidirectional LSTM

This model takes as input a re-sized gray spectrogram image of dimension (623,128), which is fed to two bidirectional lstm layers. We fine tune the number of hidden node in the lstm layers and various settings and results are reported below

Image Size	Bidirectional layer1	Bidirectional layer2	Kaggle Score
(623,128)	32	32	0.70072
(623,128)	64	64	0.68964
(310,64)	32	32	0.69848

Table 4.2: Accuracy results in a bi-directional lstm model

4.1.3 Variant of VGG network[6]

This model gives similar kaggle scores as that of the LSTM and Bidirectional LSTMs.

1. Layer1: Conv2D, 64 units, stride (4x4), activation relu, Input: Spectrogram image of (623x128)
2. Layer2: Conv2D, 64 units, stride (4x4), activation relu
3. Layer3: MaxPooling, stride (2x2)
4. Layer4: Dropout-0.25
5. Layer5: Conv2D, 64 units, stride (3x3), activation relu
6. Layer6: Conv2D, 64 units, stride (3x3), activation relu
7. Layer7: MaxPooling, stride (2x2)
8. Layer8: Droput-0.25
9. Layer9: Flatten
10. Layer10: Dense, 128 units, activation relu
11. Layer11: Dropout-0.5
12. Layer12: Dense, 19 units, activation sigmoid

This is final model after hyperparameter tuning and the kaggle score is 0.69036

4.1.4 Custom CNN architecture

We used the model described by [1] and [2]

1. Layer1: BatchNormalization
2. Layer2: Convolution2D, 16 units, stride (5, 5), activation 'relu'
3. Layer3: MaxPooling2D (2, 2), stride (2, 2)
4. Layer BatchNormalization
5. Layer4: Convolution2D, 32 units, stride (5, 5), activation 'relu'
6. Layer5: MaxPooling2D (2, 2), stride (2, 2)
7. Layer BatchNormalization
8. Layer6: Convolution2D, 64 units, stride (5, 5), activation 'relu'
9. Layer7: MaxPooling2D (2, 2), stride (2, 2)
10. Layer8: BatchNormalization
11. Layer9: Convolution2D, 12 units, stride (5, 5), activation 'relu'
12. Layer10: MaxPooling2D (2, 2), stride (2, 2)
13. Layer11: BatchNormalization
14. Layer12: Convolution2D, 19 units, stride (4, 4), activation 'relu'
15. Layer13: BatchNormalization
16. Layer14: Flatten()
17. Layer15: Dropout(0.1)
18. Layer16: Dense 512 units, activation 'relu'
19. Layer17: Dropout(0.1)
20. Layer18: Dense 19 units, activation 'softmax'

This is the final model after hyperparameter tuning and kaggle score is 0.77387

4.1.5 Residual Neural Networks

We ran Residual network with 18 layers and take spectrogram as input. Its kaggle score is 0.62814, which is not so good. According to thesis [1] by John Martinsson, we find that Deep residual neural networks can be used to classify bird species based on acoustical data recordings. We used the resnet but end up with very less accuracy.

4.2 Raw feature Networks

4.2.1 Feed forward neural network

Here we describe the architecture of the model discussed in section 3.3.1. After fine tuning this gives kaggle score 0.69716

1. Dense 32, activation 'relu'
2. Dense 32, activation 'relu'
3. Dense 32, activation 'relu'
4. Dense 32, activation 'relu'
5. Dropout(0.5)
6. Dense 19, activation 'sigmoid'

4.2.2 CNN[5]

Here we used the library librosa to get a dense vector of dimension 193, which describes the raw features of the audio file2.1 (mfcc-40, chromagram, melspectrogram etc). Since the feature is a 1 dimensional vector, we feed these as input to 1D convolutional layers and the model architecture is described below

- Conv1D 32 units, 4, activation='relu'
- Conv1D 32 units, 4, activation='relu'
- MaxPooling1D (4)
- Conv1D 64 units, 4, activation='relu'
- Conv1D 64 units, 4, activation='relu'
- GlobalAveragePooling1D()
- Dropout(0.5)
- Dense 19, activation='sigmoid'

After hyperparameter tuning, this was the best deep learning model with a kaggle score of 0.83212.

4.3 Libraries Used

Sox To get spectrograms from audio file

librosa To get raw audio features

keras Deep learning library with tensorflow as backend in python

numpy,pandas

Chapter 5

Effort

5.1 Time spent in different parts

- 10% of the time was spent on literature survey
- 40% of the time was spent on writing code
- 50% of the time was spent in hyperparameter tuning

5.2 Most challenging part

- Finding new ways of audio processing and feature extraction
- Deciding on relevant architectures for different features Hyperparameter tuning

Chapter 6

Future Work

We can change feature input of Spectrogram by segmenting out bird speech part from the spectrogram. This may enhance accuracy of model.

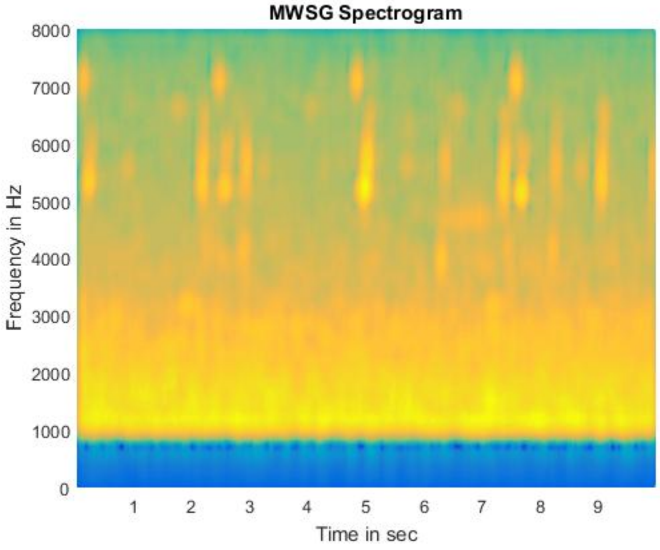


Figure 6.1: Spectrogram of an audio file.

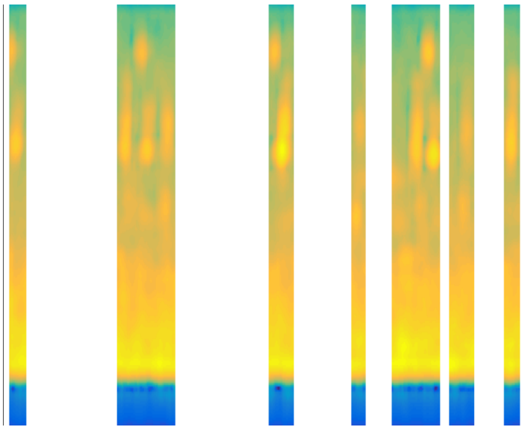


Figure 6.2: Segmented spectrogram of above spectrogram indicating bird speech.

Bibliography

[1] Bird species identification using convolutional neural networks. http://publications.lib.chalmers.se/records/fulltext/249467/249467.pdf.

[2] Birdclef 2016 challenge. https://github.com/johnmartinsson/bird-species-classification.

[3] Chroma feature. https://en.wikipedia.org/wiki/Chroma_feature.

[4] Elias sprengel, martin jaggi, yannic kilcher, and thomas hofmann. audio based bird species identification using deep learning techniques. 2016.

[5] Environmental sound classification. https://github.com/mtobeiyf/audio-classification.

[6] Kerasdocs. https://keras.io/getting-started/sequential-model-guide/.

[7] Mel-frequency cepstral coefficients. https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

[8] Residual networks. https://en.wikipedia.org/wiki/Vanishing_gradient_problem#Residual_networks.

[9] winningmodel. https://github.com/gaborfodor/MLSP_2013.