# Classifying birds by their sounds using two different neural nets

**Efrem N. Eriksson F. Svedberg H.**

## Abstract

This paper aims to build a bird classifier from audio recordings of birds using a neural net architecture. Two models are considered in the paper, a Convolutional Neural Network (CNN) and Long Short-term memory (LSTM). The models classify birds into five genera according to their bird songs. The CNN achieves a test accuracy of roughly 70% while the LSTM achieves a test accuracy of roughly 52%. While the CNN outperformed a naive classifier assigning each observation to the most common class, the LSTM did not. Evaluating the confusion matrix for the CNN indicates that more data and improved pre-processing could yield better test accuracy.

## 1. Introduction

Recently there has been an increased interest in machine learning and its applications. One area of interest is audio classification in which machine learning has been used to identify human speech. The objective of this report is to predict the bird genera based on audio recordings of bird songs using a convolutions neural net (CNN) and long short-term memory net (LSTM). This is a feasible classification task for neural networks since it is known that different birds have different songs. CNN's and LSTM's can learn these differences in bird songs. The results would have potential applications in automatic bird identification using microphones. From a research perspective, this could be used to estimate the presence of different birds in remote places. The report will aim to maximize test accuracy for both models and then compare their performance. Further, in this paper we aim for the model to classify correctly at least 90% of the times and the model will not be considered reliably if it prediction rate is lower than classifying observations at random or classify all audio files to the most common group.

## 2. Data and Methods

### 2.1. Data

The dataset consists of 2161 audio files of bird sound retrieved from the website ttps://xeno-canto.org/. In total, there are 114 different bird species in the dataset and the bird species can be grouped into 24 genera of bird. Of the 114 bird species, 46 of the bird species have 30 audio files which is the largest number of audio files for any species. Aside from the species of bird, the dataset also includes the country in which the audio file was recorded and what type of sound the audio is. The bird species can be grouped into 24 genus where the least common genus has 3 audio files while the most common has 499.

Since all of the species have very few unique observations, there is unlikely to be enough data to build a good classifier on a species level. Instead, the number of categories to classify is reduced by grouping the species according to their genera and focusing only on the five most common genera in the dataset. The five genera included in the analysis are Cryturellus, Tinamus, Nortoprocta, Nothura, and Ortalis which all have at least 100 unique recordings. The first four genera mentioned come from the same family, called ”tinamus”, while the Ortalis belong to another family called ”Ortalis”.

Aside from the species and genera, the dataset also includes information about the type of sound and country in which the recording is from. The type of sound is used to further reduce the dataset to only consider bird songs while country is not included in the analysis. By only considering songs, the audio recordings from the same genus should be more similar and thus easier to classify.

The audio recordings differ in length from a few seconds up to 2 minutes. To avoid padding short recordings with minutes worth of zeros, the recordings are split into multiple audio clips of roughly 5 seconds in length. While this does mean that one recording will have multiple observations, it does ensure that as much of the data is used as possible.

### 2.2. Methods

### 2.3. CNN - Convolution neural network

The first model we will use is a convolution neural network (CNN). CNNs are a uni-directional regularized feed-forward neural network. Generally with CNN, the input data $X$ is being multiplied with Kernel matrix $W$, for which the output is called feature map. This is being done for each convolutional layer.

While CNNs are commonly used within computer vision the models can also applied to audio data by creating a visual representation of the audio (Das et al., 2020), (Thornton, 2019). A common approach is to create a spectrogram representation of each audio file which is then used in the CNN for classification. The modeling can be split into three parts; transforming, converting scale, and data representation.

As a first step, the audio signal is transformed to its frequencies by applying a Fourier transform:

$$X(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt$$

where $x(t)$ and $X(\omega)$ denote the original signal and is the frequency of the signal, respectively. Furthermore, $\omega$ is the angular frequency and $t$ stands for the time index. The frequency can then be shown in a 2D matrix which represents the change in frequencies over time visually.

The spectrogram is further processed by taking the natural logarithm of its entries $M = ln(X(\omega))$. This scale transformation is performed since changes in frequency are expected to be small for voice recordings (Hansen, 2020). It is reasonable to believe that this is the case for bird songs as well.

This yields a spectrogram where width represents time and height represents the frequency on a logarithmic scale. Furthermore, color is used to indicate the intensity of the frequency.

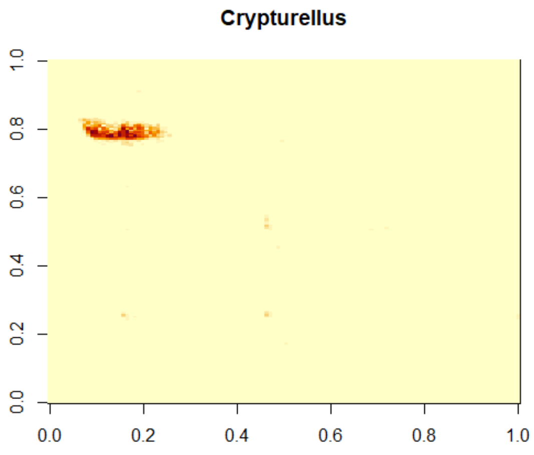

*Figure 1.* Example spectrogram from the crypturellus genus

Figure 1 visualizes the spectrogram for one of the observations after transforming the audio to a spectrogram on a log scale. This is the type of image that is used as input into the CNN.

### 2.4. Long Short-Term Memory

For the second model in the paper, a Long Short-Term Memory (LSTM) network is fit which is a type of a Recurrent neural network (RNN). An RNN is a natural choice for modeling time series since the neural net can learn time dependencies in the dataset. Each hidden layer takes the previous hidden layer and the current data point as input. This is a reasonable choice for audio data since the frequency at time point $t$ can be expected to depend on some previous frequency.

LSTM, compared to RNN, is a network that addresses the problem of gradient that tends to explode, or vanish, while allowing for the ability to learn long-term dependencies. The LSTM adds a forget gate which allows the model to forget previous states. This is useful for classifying birds since the audio recordings also include periods when the birds are not singing. If the model is able to forget these states which do not aid classification, the classifier should be able to perform better.

### 2.5. Practical Methodology and Computation

To build the models, we aim to maximize capacity while avoiding overfitting and minimizing computational intensity. This is achieved by first building an overfitted baseline model which can achieve a training accuracy of at least 95% as our target accuracy. Capacity is increased manually in the CNN by increasing filter size, kernel size, unit size, and the number of layers. In the LSTM capacity is increased by iteratively adding layers and units.

When the model achieves a training accuracy of at least 95%, a random search is used to optimize the hyper parameters learning rate and weight decay. The learning rate is tuned to further increase capacity while weight decay is added to regularize the overfitted model. The learning rate and weight decay are randomly sampled from the distribution

$$10^{\mathrm{U}(-5,-1)}$$

as proposed by Goodfellow et al. (2016, pg.434). The optimal hyperparameters are selected by using the hyperparameters which result in the highest validation accuracy.

Finally, the models are manually regularized by adding dropout layers if there is a large gap in training- and validation accuracy.

### 2.6. Evaluation

The dataset is split into a training set and test set consisting of 80% and 20% of the total data set respectively. When training the models, we use the training set and for each epoch, 20 % percent of the training set is used as validation data as an estimate for the error for unseen data.

The test dataset is used to evaluate the final models. This test set has not been used to train the models in any way. This will ensure that our prediction accuracy is not due to training an overfitted model.

## 3. Results

### 3.1. Convolutional neural network

As a first step, a baseline convolutional neural network model was fitted with four convolutional layers and four layers in the feed-forward network. Each convolutional layer had 12 filters and a 3 × 3 kernel size with a max pooling layer in between. Each layer of the feed-forward neural net contained 8 units. The output layer contained 5 units and a softmax activation function to predict the genus of the bird. The capacity of the model was then sequentially increased by increasing both the filter and units up to 32. This yielded a model with around 95 % training accuracy and 75 % validation accuracy for 20 epochs. Since the training accuracy was larger than the validation accuracy, this indicated overfitting. Using this as a baseline model for the random search method to tune the parameter values for a model with the same architecture, but with added l2 penalization, it was found that the best value for the learning rate was 0.001 and for the weight decay of the penalization, it was 0.002. The final CNN model was then based on these parameter values but also included a dropout layer of 0.2 after each layer. The results from the final model are presented in figure 2

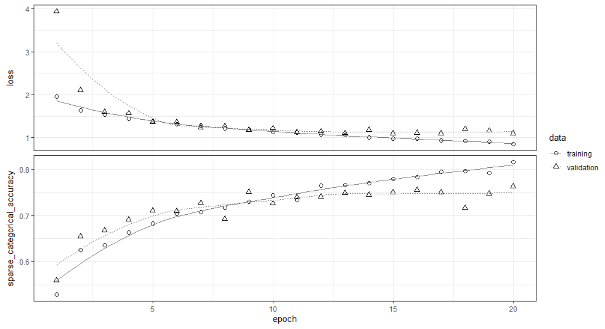

*Figure 2.* best CNN model training history

For this model, the training accuracy was around 0.82 and the validation accuracy was around 0.77. Since this gap was smaller compared to the baseline model without any penalization, it indicates that the overfitting has been reduced, although the validation accuracy did not improve by much.

Using this model to predict the holdout test data, the accuracy on this test data was 0.76 and hence close to the validation accuracy.

Looking at the results from table 1 then it seems that whereas it was able to correctly classify birds for all genera, it was more accurate for birds with many observations such as crypturellus or ortalis. It can also be seen that it often tended to predict observations as crypturellus which was the largest group.

*Table 1.* Confusion matrix for the best CNN model where columns represent the reference group and rows the predicted group

| | Crypturellus | Nothoprocta | Nothura | Ortalis | Tinamus |
|---|---|---|---|---|---|
| Crypturellus | 485 | 39 | 27 | 35 | 37 |
| Nothoprocta | 0 | 9 | 0 | 0 | 0 |
| Nothura | 12 | 10 | 21 | 4 | 2 |
| Ortalis | 22 | 5 | 0 | 184 | 5 |
| Tinamus | 8 | 0 | 0 | 0 | 26 |

### 3.2. LSTM

First, a baseline LSTM model was fitted with one LSTM layer and 2 dense layers, each with 8 units. This was then increased up to 32 units to overfit the data with a training accuracy of approximately 0.98%. In a similar way to the CNN model, a random search was used to find the optimal values of the learning rate and weight decay for the l2 regularizer for each layer. The optimal learning rate was found to be 0.0001 and the optimal weight decay was 0.001. The results from this model are presented in figure 3 and table 2

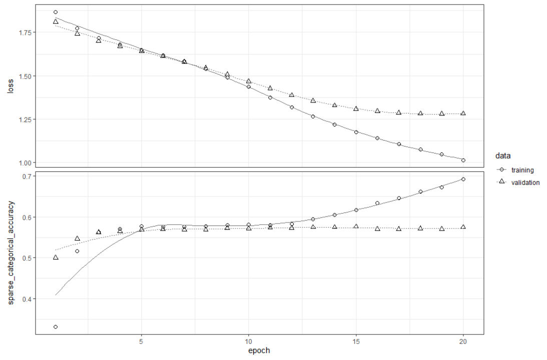

*Figure 3.* Best LSTM model training history

Looking at figure 3, the training accuracy for the last epochs is 0.69, whereas the validation accuracy does not improve by much over the epochs and fluctuates around 0.57. This suggests that the model is unable to learn any interesting features in the training set which yields good predictions on out-of-sample data.

Using this model on the test data gave an accuracy of 0.52. Hence this model did not perform as well as the CNN model.

*Table 2.* Confusion matrix for the best LSTM model where columns represent the reference group and rows the predicted group

| | Crypturellus | Nothoprocta | Nothura | Ortalis | Tinamus |
|---|---|---|---|---|---|
| Crypturellus | 497 | 64 | 53 | 217 | 67 |
| Nothoprocta | 2 | 1 | 1 | 0 | 0 |
| Nothura | 0 | 0 | 1 | 0 | 0 |
| Ortalis | 22 | 5 | 2 | 15 | 2 |
| Tinamus | 0 | 0 | 0 | 0 | 1 |

The confusion matrix in table 2 shows that this model tended to predict most of the observations as Crypturellus, which also constitute roughly half of the total observations. In practice, the model performs as a naive classifier which assigns each observations to the most common class. This indicates that the model has not learned any useful features from the data.

## 4. Conclusion

Comparing the CNN with the LSTM model in terms of their performance, it was found that the CNN model had higher accuracy both while training, and when evaluated on the test data set. It was also found that for the final CNN model, the training accuracy was relatively close to the validation accuracy, in contrast to the final LSTM model, indicating that the LSTM model is still somewhat overfitting the training data. This might be due to the processing of the audio files before using them as input in the models. For CNN, first, the audio files were transformed to frequencies and then converted to a log-scale spectrogram. For LSTM, the raw audio data was used. Since the audio recordings have a lot of background noise, some form of pre-processing may be necessary before model-fitting.

Since the models classify into five categories, a naive random classifier would be expected to achieve a prediction accuracy of approximately 20%. Both models surpassed this naive classifier. A naive classifier which assigns each observation to the most common class would be expected to achieve a prediction accuracy of approximately 55%. This naive classifier was surpassed by the CNN but not by the LSTM indicating that the LSTM model would not be preferable to a naive classifier.

The poor performance of the LSTM could be due to the data not being pre-processed before fitting the model. Background noise outside the normal frequency range of bird sounds was not removed while the CNN model used data transformed to a log-scale. Another potential drawback is that four of the genera belong to the same family and thus may already have quite similar songs. This would make it harder for the CNN and LSTM to distinguish between these genera.

Another potential issue with the methodology is that large audio files were split into multiple observations. If the majority of the observations are from the same audio file, the neural net may learn features of background noise which is similar across observations. The reason this was done was to use as much of the data as possible and avoid having to pad short audio clips with several minutes of 0s. However, this does still result in there existing dependencies between observations.

Aside from the methodological- and data processing issues, further improvements to the models was hampered by computational limits. The confusion matrix of the CNN indicates that it was easier to predict genera with more observations. With more data for the less well-represented genera, the model could become better at correctly classifying them. With the available computational resources, working with a larger dataset would have been infeasible.

A growing concern within machine learning research and application is the ethical ramifications of data-driven decision-making. Black box models trained on large amount of data can be used to make decisions which impact individual's lives with poor insight into how those decisions are taken. While this is a broad concern with deploying machine learning models, it is not a major concern with this report. The data used does not pertain to protected classes such as gender, sexuality or race and cannot be directly used in potentially harmful ways.

Future studies could look at using other transformations for the spectrogram data. Another available transformation in torchaudio is Mel-frequency cepstral coefficients (MFCC) which could result in an improved model if tested. Different pre-processing methods could also be tested on the LSTM data to remove frequencies which are known not to be relevant for the classification task. Finally, it would be interesting to build a classifier for all the genera or species. However, this would most likely require more data to build a good classifier.

## References

Das, J. K., Ghosh, A., Pal, A. K., Dutta, S. and Chakrabarty, A. (2020), Urban sound classification using convolutional neural network and long short term memory based on multiple features, *in* '2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)', IEEE, pp. 1–9.

Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press. `http://www.deeplearningbook.org`.

Hansen, M. (2020), 'Domain adapting english speech recognition to swedish', *LU-CS-EX* .

Thornton, B. (2019), 'Audio recognition using mel spectrograms and convolution neural networks'.