Convolutional Neural Network for Bird Species Classification

Adam Aboode
aboode@kth.se

Prashant Kumar
pkum@kth.se

Erifili Ichtiaroglou
erifili@kth.se

Abstract

This paper describes a convolutional neural network based deep learning approach for bird species classification, which was applied to the dataset for the "MLSP 2013 Bird Classification Challenge" hosted on Kaggle. The full dataset consists of 645 ten-second audio recordings in uncompressed WAV format. The recorded waveforms contain a varying amount of noise and zero or more bird species. Using spectrograms of the recordings we trained and evaluated a convolutional neural network. Unfortunately our network was not able to learn the features of the data well enough and the performance for classifying bird species turned out to be poor.

1 Introduction

Extensive scale, accurate bird species classification is necessary for avian biodiversity conservation. It helps us quantify the consequence of land use and land management on bird species and is requisite for bird watchers, conservation organizations, park rangers, ecology consultants, and ornithologists all over the world. Automatic bird species classification using audio constitutes a good, cheap and fast alternative to studies conducted by humans. It does also enable us to process large amounts of data from remote and hard to reach locations.

1.1 MLSP 2013 Bird Classification Challenge

The project is inspired by the bird classification challenge held by the Machine Learning for Signal Processing conference in 2013 and is available on Kaggle [1]. It is a open-end multi-label supervised classification problem for signal processing and machine learning to presumably catalogue bird sounds in real-world audio data accumulated in an acoustic monitoring scenario. Traditional methods for accumulating data about birds include expensive human effort. A likely alternative is acoustic monitoring. There are several benefits to recording audio of birds compared to human surveys, including increased temporal and spatial resolution and extent, applicability in remote sites, reduced observer bias, and potentially lower costs.

Alternative datasets from similar challenges are available online. One of them is the NIPS 2013 Multi-label Bird Species Classification challenge [2] which encompassed 87 classes (species). A more recent and more extensive dataset is BirdCLEF 2017 which consists of 1500 bird species over 36496 recordings[3]. However, we weren't able to test our approach on this dataset, due to time constraints, as the training data alone consisted of more than 100 GB of data.

1.2 Previous work

There have been numerous attempts to develop and evaluate methods of automatic bird species recognition based on auditory data [6][7].

Elias Sprengel et. al[6] approach consisted of a neural network with five convolutional and one dense layer. As inputs they used chunks of spectrograms of preprocessed sound files. They separated the noise part and the song part of the sound files computed the spectrograms and then split them into

equally sized chunks. Each chunk was used as a training sample. Their method was able to reach high scores in predicting the correct species and they managed to win the international BirdCLEF Recognition Challenge in 2016.

Bálint Pál Tóth and Bálint Czeba [7] used two different neural networks for the same competition (international BirdCLEF Recognition Challenge in 2016). The first network was a modified version of AlexNet with five convolutional layers and a fully connected layer, while the second consisted four convolutional layers and a fully connected layer with a smaller number of nodes. They produced spectrograms from the provided sound files and preprocessed them by deleting regions which didn't contain useful information. Finally, they split the remaining spectrograms into five second segments and used them as inputs to the network.

Two notable approaches taken in the Kaggle competition upon which this project is based are the ones taken by Gabor Fodor and Maxim Milakov. Fodor who won the competition[9] used spectrogram template matching [10]. Milakov on the other hand ended up in fourth place on the leaderboard[9] using an CNN approach. In recent years, deep learning techniques have garnered a lot of attention and popularity as they have been proven to be performing well for a variety of different tasks, including for instance computer vision and speech recognition. Thus, in this project the aim is to evaluate if Convolutional Neural Networks (CNN) are a valid alternative when it comes to bird species classification using auditory data.

1.3 Convolutional Neural Networks

A convolutional neural network is a type of deep neural network which is built upon the ideas described below.

1.3.1 Convolution

Supposing there is a grid formulation of the input data, the convolutional layers consist of several grids of neurons, i.e. several filters that, slide over the grid of the previous layer. By computing the dot product of the elements of both grids a new matrix is formed, feature map, that contains information about specific features of the initial grid. By using different kind of filters, emphasis is given to different kind of features.

1.3.2 Leaky Rectified Linear Unit

Leaky Rectified Linear Units (Leaky ReLU) are used after each convolutional layer, to introduce non-linearity to the method, since almost all real-life data we want to train the CNN on are non-linear. This operation multiplies the values of all negative neurons with a small constant, α.

1.3.3 Pooling

The next layer downsamples the large matrices obtained from the convolutional layer. The pooling operation degrades the matrix by replacing non-overlapping sections of the matrix with the average (average pooling) or max (max pooling) value found in the specific sections. In this way we achieve reduction of the dimensionality of the network, while at the same time we maintain the most informative neurons.

1.3.4 Classification

The operation that takes place in the last layers of a CNN is the classification. These layers are called fully connected layers, since all the neurons in the previous layers are connected with all the neurons in the next and it is used to actually train the network.

The above process of convolving then downsampling is replicated until key features are extracted, and is then fed to the fully connected layers. A useful property of CNN is that the key features of the input do not have to be specified, but will be extracted by the CNN itself. Thus, not much pre-processing has to be done to the input and the method is therefore suitable for automatic feature extraction.

2 Method

2.1 Data representation

The data provided by MLSP contained both the raw audio files and processed data such as the raw spectrograms, filtered spectrograms and spectrograms which had undergone automatic segmentation. We decided on using downsampled spectrograms as our input data. The raw spectrograms were produced by dividing the audio signal into frames and applying FFT to each frame [4]. The spectrograms were then downsampled by applying a mean function on 2x2 blocks of the image. Since the dataset contained 19 different classes (species) and each spectrogram could contain more than one species at a time we represented the labels as vectors of length 19 with absent classes represented with a 0 and present classes represented with a 1.

Figure 1: Sample spectrogram.

2.2 Architecture

Our network is an amalgamation of networks previously used for this task, namely the network created by Maxim Milakov [5] and Sprengel et al. [6]. The network of Milakov resulted in a 4th place on the leaderboard in the Kaggle competition whereas Sprengel et al. won the BirdCLEF 2016 Recognition Challenge.

Our network, Net-1 (see Table 1), consists of four convolutional layers with an increasing number of features. After each convolution we apply the Leaky ReLU activation function, max pooling and batch normalization. At the end of our network we put a fully connected layer with Leaky ReLU activation and finally a fully connected output layer with an sigmoid activation function. The network was implemented in the deep learning framework Keras [8].

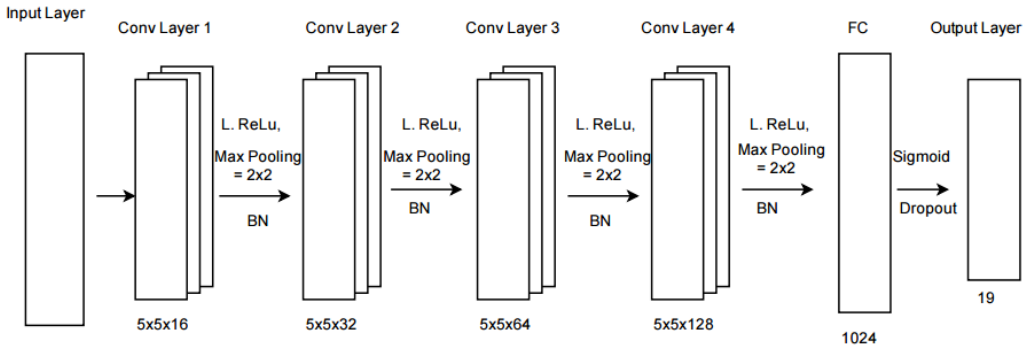

Figure 2: Architecture of the Network

Table 1: Architecture of our network

Net-1
CONV-16, 5x5
LeakyReLU, 0.3
M-P, 2x2
BN
CONV-32, 5x5
LeakyReLU, 0.3
M-P, 2x2
BN
CONV-64, 5x5
LeakyReLU, 0.3
M-P, 2x2
BN
CONV-128, 5x5
LeakyReLU, 0.3
M-P, 2x2
BN
FC, 1024
LeakyReLU, 0.3
DROPOUT, 0.5
SIGMOID, 19

CONV-N - Convolutional layer where N is number of filters with the corresponding size. LeakyReLU - Leaky ReLU with constant α. M-P - Max-Pooling with pooling size. BN- Batch normalization. FC - Fully connected layer with number of nodes. DROPOUT - dropout with rate. SIGMOID - sigmoid output layer with number of nodes.

3 Experiments

3.1 Experimental setup

The training data and the test data were predefined by the competition and loaded into two separate sets. Next, the spectrograms were downsampled and the training set was shuffled and split into an validation set and a training set by assigning 90% of the data to the training set and 10% to the validation set. This ratio was chosen because of the low amount of total training data. The network was trained using mini-batch learning with an batch size of 8 samples per gradient update and for a total of 10 epochs. The batch size was chosen to be 8 as our training data is fairly limited and to large batches would worsen the performance. The number of epochs was chosen experimentally, as too many epochs lead the network to overfit to the data. During training we tried to minimize the cross-entropy loss, we used RMSProp as our optimizer and mini-batch learning as suggested by Tóth and Czeba [7]. The deep learning framework Keras [8] with the TensorFlow backend was used for data preparation, training and evaluation. Downsampling of the spectrograms was performed using the function `block_reduce` from the Python library scikit-image[11].

3.2 Evaluation of networks

After training, the network was evaluated on the test set by letting the network predict the labels of the spectrograms in the test set and comparing the outputs to the ground truth labels available in the dataset. We did only count the prediction as correct if all the labels for a specific spectrogram were correctly predicted.

4 Results

The accuracy and loss graph for the training and validation data over the 10 epochs can be seen in fig. 3 and fig. 4 respectively.

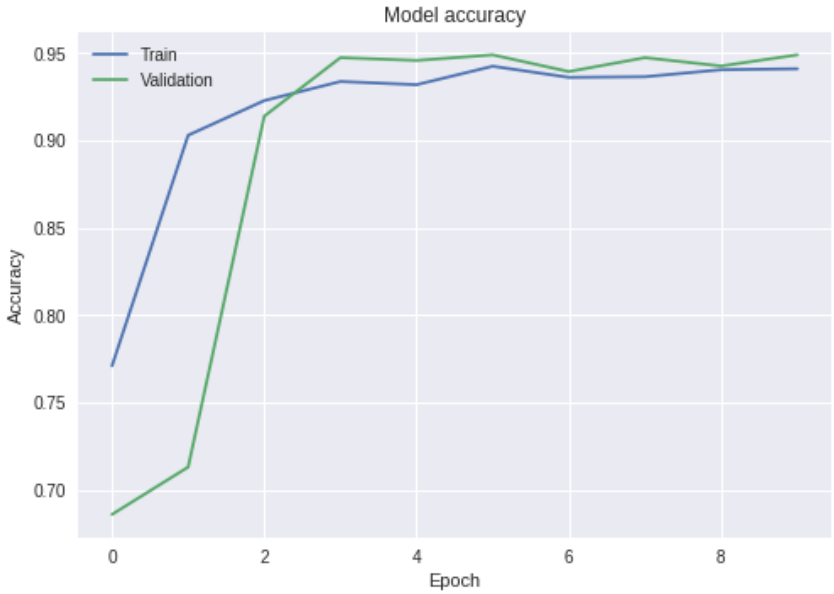

Figure 3: Accuracy on training and validation set during training.

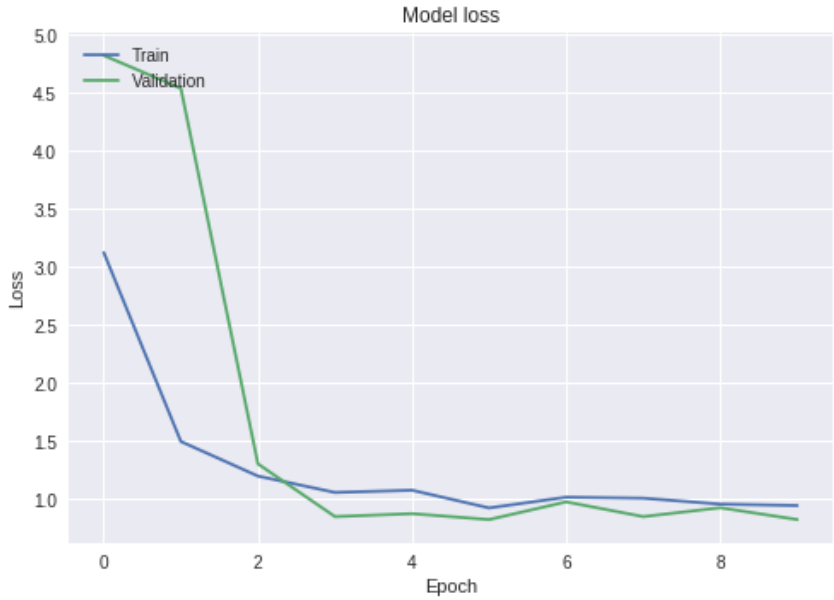

Figure 4: Loss on training and validation set during training.

The accuracy on the test set was 46.75%.

5 Discussion and Conclusions

In this report we wanted to investigate whether CNNs can be used for bird species classification. Unfortunately we were not able to create a system which could perform this task but judging from previous work promising results can be achieved using CNNs[5][6][7]. The performance of our network was very poor with an accuracy of 46.75% on the test set, which might appear to be fairly decent. However, if we examine the predictions we see that the network tends to almost always

guess that there is no bird in the recording and since almost half of the dataset consists of empty recordings we get an accuracy close to 50%. The actual accuracy for predicting bird species from non-empty recordings is close to 0%. We don't know how well our system performed compared to the contestants on the leaderboard as Kaggle does only publish a score which is calculated by submitting your results and do not publish the accuracy on the test set.

There were several problems with the dataset we used for our approach, the dataset was very small (only 322 spectrograms in the test set), almost 50% of the dataset consisted of recordings without any bird sounds and the fact that each spectrogram could consist of zero or more birds. Since, there were 19 species in the dataset and more than one bird could be present in each spectrogram, the number of potential combinations were very high. This high number of combinations combined with the fact that our dataset was very small made it unlikely and even impossible for all different combinations to be present in the data, which might have made it problematic for the network to generalize. There are also several limitations to this approach in general as two spectrograms of the same bird might look fairly different to the network based on if the bird chirps are occurring earlier or later in the recording. Besides that, the effectiveness of our validation set is also very questionable as it did only consist of 10% of the training set (32 samples). As the training data as a whole has a problem of representing all the combinations, this problem becomes even more apparent for the validation set which could potentially end up consisting of only empty recordings and therefore be very non-representative for the problem at hand.

One way to address the problem of limited training data and help the network to generalize better is by using some data augmentation techniques. As described in [6] one technique is to shift the training example in time by a random amount. That means that the spectogram is split in two parts and the second one is placed in front of the first. In this way the network can learn that the bird song can appear any time independent of the bird species. Another data augmentation technique is to combine audio files containing the same class and create one spectogram that is used as input to the model. In this way a lot of songs of same species birds are processed simultaneously and the neural network sees more important patterns at once, which lead to better accuracy. Yet another technique includes adding background noise on top of the original training sample in order to make the network less sensitive to noise and generalize better.

To summarize, our approach is not suitable for a dataset this small, we should either have used a larger dataset, done more pre-processing to the data or used a different approach altogether. By studying previous work we see that better accuracy can be achieved by performing clever preprocessing and feature engineering. Convolutional neural networks are most effectively trained on large amounts of data such as the BirdCLEF 2017 dataset [3], this dataset was unfortunately not used as we did simply not have enough time and resources to process it.

References

[1] MLSP 2013 Bird Classification Challenge. On Kaggle `https://www.kaggle.com/c/mlsp-2013-birds`

[2] NIPS 2013 Multi-label Bird Species Classification. On Kaggle `https://www.kaggle.com/c/multilabel-bird-species-classification-nips2013`

[3] BirdCLEF2017 `http://www.imageclef.org/lifeclef/2017/bird`

[4] Data Description for MLSP 2013 Bird Classification Challenge. On Kaggle `https://www.kaggle.com/c/mlsp-2013-birds/data`

[5] Maxim Milakov (2013) Description of solution. Kaggle `https://www.kaggle.com/c/mlsp-2013-birds/discussion/5457#29092`

[6] Sprengel et al. (2016) Audio Based Bird Species Identification using Deep Learning Techniques. `http://ceur-ws.org/Vol-1609/16090547.pdf`

[7] Tóth and Czeba (2016) Convolutional Neural Networks for Large-Scale Bird Song Classification in Noisy Environment. `http://ceur-ws.org/Vol-1609/16090560.pdf`

[8] Chollet, François and others (2015) Keras. On GitHub `https://github.com/fchollet/keras`

[9] MLSP2013 Bird Challenge Leaderboard `https://www.kaggle.com/c/mlsp-2013-birds/leaderboard`

[10] Solution Approach of the Kaggle competition winner https://www.kaggle.com/c/mlsp-2013-birds/discussion/5457

[11] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) http://dx.doi.org/10.7717/peerj.453