CS230

Detection of Bird Species Through Sounds

Shirley Cheng
Department of Computer Science
Stanford University
scheng3@stanford.edu

Julie Wang
Department of Computer Science
Stanford University
jwang09@stanford.edu

Abstract

We address the problem of accurately identifying bird species through bird calls and songs. In this paper, we convert audio snippets into spectograms and use a convolutional neural network to classify these images. We first built our training model from a baseline CNN architecture of just two convolutional layers, then improved the model with hyperparameter tuning in learning rates, number of epochs and hidden layers, as well as dropout rates for regularization. The training dataset consists of 4,327 audio snippets containing 101 distinct bird species found across the United States. Our performance achieved an accuracy of about 98% for the training set and about 67% for the test set.

1 Introduction

Monitoring wild bird species helps scientists identify population diversity and is considered important for preservation of ecosystem health. With the growing impact of climate change, bird populations are expected to fluctuate in size and distribution. Therefore, it is essential to survey bird species. Traditionally, scientist have done this through manual data collection, often with help of volunteers. However, this time-consuming approach is inefficient as data is not readily scalable and many species dwell in areas that are physically challenging for volunteers to access.

Although data collection still proves to be challenging, deep learning architectures like CNNs have helped classify birds based on few examples of their calls and songs in recordings. Because of the importance of monitoring bird species, many competitions such as BirdCLEF has been hosted with the goal of detecting bird calls through soundscape audio recordings. Likewise, we use five-second audio recordings of bird calls in both our train and test datasets. From previous research and competition results, we have found that CNN-based models are the most common approach in bird call detection as features can be effectively extracted from spectograms and classified as images.

2 Related Work

Many past research have used CNN architectures that evolved from competitions such as BirdClef. One main reason is that extracting deep features based on image representation of sounds has been very effective, especially when applied to audio classification of bird calls [7]. For data preprocessing, the most common approach has been transforming audio into spectrograms, then exploring the best CNN architecture with respect to dataset diversity and the number of classes [7][8]. In Sprengle et al, to train the convnet, batches of size 16 or 8 and drop-outs are used on the input layer and the soft-max layer[3]. The common learning rates are 0.01 and 0.001, while most research trained the model for at least 100 epochs.

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Datasets and Preprocessing

Our preliminary dataset consisted of 4,327 five-second recordings of 101 popular bird species across the United States. To first gather the recordings, we used a python script that queried JSON data and downloaded respective sound files from the xeno-canto public dataset[1] [6].

Parameter	Value
Number of Audio Channels	1
Sampling rate	16000 Hz
Window length	400
Hop length	400
Number of Mel filters banks	128

Table 1: Parameters of spectrograms

Because the data downloaded from the website was originally stored as mp3 files, we used Pydub[2] to convert mp3 to wav. These 32-bit wav audio files are then normalized and loaded into a tensor, spliced into five-second audio snippets, and converted to spectrograms using the mel scale (see Table 1 above). Examples of the spectrograms are shown in Fig. 1. Because original audio files can be a few minutes long, splicing a file yielded hundreds of distinct audio recordings. As a result, each of the 101 species had around a couple hundred samples. Since the BirdCLEF dataset - one of the largest bird song datasets used in competitions - only used on average of 30 samples per species, we took up to 30 recordings per species for our final dataset. Lastly, the dataset is divided into $\frac{4}{5}$ train data and $\frac{1}{5}$ dev/test data.

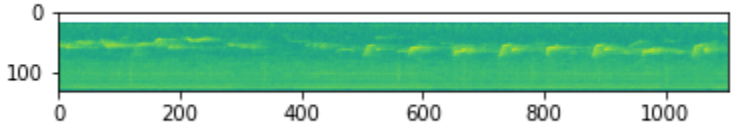


Figure 1: Mel spectogram of an Eurasian blackcap

4 Methodology

As we are representing our data as spectrograms, this essentially becomes an image classification problem. Due to the two-dimensional nature of this problem, we find convolution neural networks to be the appropriate type of neural network to use. We define a CNN architecture inspired by the LeNet architecture, with structure Conv -> pool -> Conv -> pool with ReLu activation function and then a fully connected layer at the end. The specific structure choices we explore are detailed later in this section. We use a cross entropy loss function as this is a multi class classification problem. We use pytorch's library for this:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

[1] https://www.xeno-canto.org/

[2] https://github.com/jiaaro/pydub

As our data set is small, with only 4327 images, we use a .9/.1 train/test split. We also do not use a validation set due to the dataset size constraints. We train our model with 3894 training images for 180 epochs. We have 433 images used for testing. We then look at the losses for hyperparameter tuning. For the specific design of the layers of the network, we explore three different models.

- Model 1: Simple model with 2 convolution layers and a limited amount of features. We use kernel size 3 with and 4 output features. One fully connected layer at the bottom.

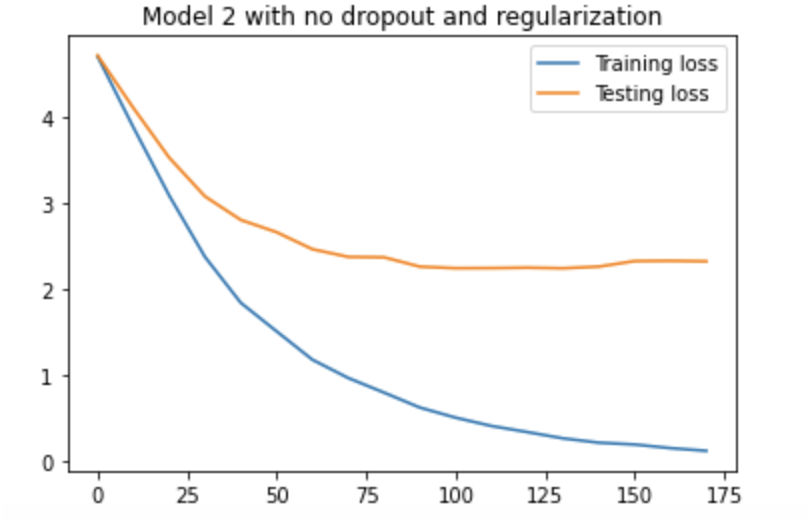
- Model 2: more complex model with 4 convolution layers, with kernel size 3 and 64 output features. One fully connected layer at the bottom.

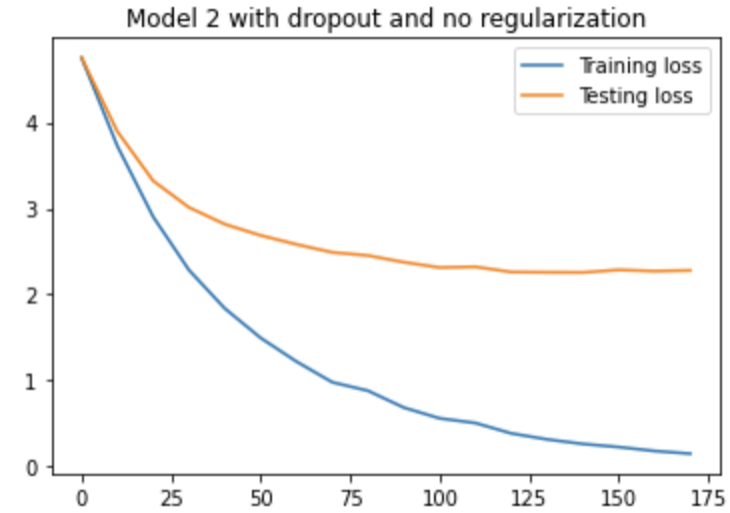
- Model 3: 4 convolution layers, with kernel size 3 and 128 output features. Two fully connected layers at the bottom.

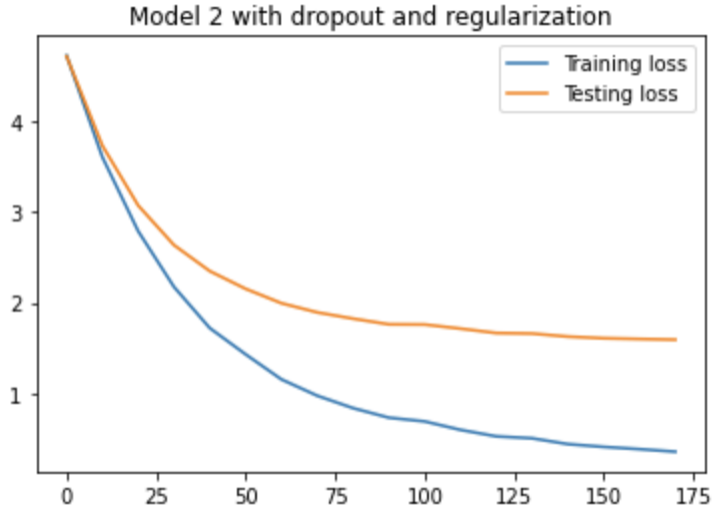
For all 3 models, we use batch normalization and max pooling in each convolution layer as well. We use accuracy score as our performance metric. From the 3 models, we choose a model with highest training accuracy while also generalizing reasonably well. Then we implement techniques of dropout and l2 regularization in order to improve test accuracy.

5 Results

We found the Model 2 to be the best. Model 1 had high training and validation error while model 3 had over fitting problems with low training error but high testing error. We found that a learning rate of .001 was best. While learning was slow(we had to run around 180 epochs), we found that higher rates of learning led to convergence at suboptimal values. With 180 epochs and a learning rate of .001, we found model 2 to have an accuracy score of 99% on the training set and an accuracy score of 65.5% on the test set.







We found that applying optimization techniques to reduce overfitting did reduce testing error, but not substantially. The accuracy score for the test set after applying dropout was 66.5% and after applying l2 regularization, we can reach 67% accuracy. We tried several different dropout rates from .2 to .8 in intervals of .1 and found .5 to be the best dropout rate. For l2 regularization, we tried several different values of weight decay as well, including .1, .01, and .001. We found .001 to have the best results. While 67% accuracy does not sound impressive, considering the fact that we have 100 total species in the dataset, and only 30 images per species, this performance of this model is better than what one might expect.

6 Conclusion/Future Work

Detecting bird species from sound can present some challenges. First, within species, there can be variation. Another challenge is when multiple species are singing at the same time, it can be difficult to separate out all of them to give distinct identification to each. Finally, underlying datasets can present a challenge as well as it might be unbalanced towards popular birds[9]. Thus, future work will include improving the data preprocessing to handle more robust recordings, such as recordings that include various bird calls as well as noise distractions. Additionally, as our audios are simplified to one channel, it would be beneficial to keep audios separated into left and right inputs.

As for the model, because of RAM issues in Google Colab, the environment we used for this project, we were not able to use the bigger dataset of 36,018 spectrograms of 74 bird species which we additionally prepared. The dataset we used to train had an average of 30 images per species. Due to this small data size, we noticed a high variance in results as our model does not generalize well. If we had more time, we would train on more data per species.

7 Contributions

Shirley worked on building the training model, hyperparameter tuning to improve baseline models, and accessing the accuracy of the model through performance metrics. Julie collected bird audio recordings from the xeno-canto repository and preprocessed datasets as well as separating them into train and test sets. Both members reviewed previous research, searched for Github repos for baselines, writing the report, and making the final video.

References

[1] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[2] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 0.1038/s41586-020-2649-2.

[3] Milakov, M. "MLSP 2013 Bird Classification Challenge" Kaggle. 2013.

[4] Pulkis, S. "Build an Image Classification Model using Convolutional Neural Networks in PyTorch." Analytics Vidya. Oct 1, 2019.

[5] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller A., Grisel O., Niculae V., Prettenhofer, P., Gramfort A., Grobler J. et al. "Design for machine learning software: experiences from the scikit-learn project." ECML PKDD Workshop: Languages for Data Mining and Machine Learning. 2013.

[6] Mikołajczyk, A., Kortas, M., and Knorps, M. Bird Song Recognition. (2019) github.com/wimlds-trojmiasto/birds.

[7] Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, Danny., Ritter, M., and Eibl, M. (2017) Large-Scale Bird Sound Classification using Convolutional Neural Networks. In: CLEF (Working Notes)

[8] Chih-Yuan K., Chang, J., Tai, C., Huang, D., Hsieh, H., and Liu, Y. Bird Sound Classification using Convolutional Neural Networks.

[9] Sprengel, E., Jaggi, M., Kilcher, Y., and Hofmann, T. (2016) Audio Based Bird Species Identification using Deep Learning Techniques. LifeCLEF 2016 pp.547-559.

[10] Incze, A., Jancso, H., Szil, Z., Farkas, A., and Sulyok, C. (2018) Bird Sound Recognition Using a Convolutional Neural Network. IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY). DOI: 10.1109/SISY.2018.8524677.