Using Triplet Loss for bird species recognition on BirdCLEF 2020

Thailsson Clementino[1] and Juan G. Colonna[1]

Institute of Computing (Icomp), Federal University of Amazonas (UFAM),
Av. General Rodrigo Octávio 6200, Manaus, Amazonas 69077-000, Brazil
{thailsson.clementino,juancolonna}@icomp.ufam.edu.br

Abstract. This paper presents the approach used in the BirdCLEF 2020 Competition. The objective of the competition is to try to recognize bird species through its sings and calls among 960 species in soundscapes. We use a MultiScale CNN + Triplet Loss to learn the mel-spectrogram characteristics that differentiate the species from each other. The CNN create a 128-D embedding used to classify the species. The approach achieves third place on the competition with c-mAP of 0.009752 and r-mAP of 0.008. We also present some changes on train parameters done after the competition that achieve our best evaluation with c-mAP 0.062877 and r-mAP of 0.108.

1 Introduction

LifeCLEF 2020 Bird is a machine learning competition that was organized for the 2020 edition of CLEF – Conference and Labs of Evaluation Forum [6, 5]. The objective of the competition was to design, train and apply a classification algorithm that reliably recognize the sounds of birds in raw recordings contaminated with several environmental noises.

The audio dataset available for training on the competition is composed by 73.267 records of 960 different species from South and North America and Europe. All these records are taken from the website https://www.xeno-canto.org/. The audios present in the validation and test sets belongs to four different record sites: Peru (PER), Sapsucker Woods, Ithaca, USA (SSW), High Sierra Nevada, USA (HSN) and Germany (GER). The validation and test sets have 12 and 153 recordings, respectively, all of which are 10 minutes long and were recorded on the mentioned sites.

For our solution, we chose to use a Siamese Convolutional Neural Network (CNN) approach with Triplet Loss. This choice was supported by the promising results that state-of-the-art works achieved using Siamese Networks. This type of neural network is based on the concept of One-shot learning, which consists of comparing an embedding vector, generated at the output of the Neural Network, against embeddings of new samples using a similarity by distance calculation. The Triplet Loss helps to learn which features, represented by the embedding vectors, differentiate samples of different species. Related works that make use of Siamese Neural Networks have been applied in different contexts [8, 15, 13], and also in the context of audio recognition [4, 11, 14]. After reviewing the related works, we decided to adopt an approach in which we feed the Neural Network with Mel-spectrograms to generate a unique feature representation of each species call.

2 Preprocessing

As the competition task is to recognize species in every 5 seconds of audio in a recording, the first preprocessing was to take all the recordings contained in the training set and split them into five-second chunks. There were some recordings in the training set with less than five seconds long, in these cases, the records were padded with copies of themselves until they achieved five seconds. After this stage, we obtained a total of 623.981 segments.

Similarly to [11, 4, 9], we opted for a visual approach to model input, using Mel-scale spectrograms taken from each chunk of audio, as mentioned above. Furthermore, were also extracted the harmonic and percussive components from Mel-spectrograms [3, 14], once different species can has a call more expressive in one of its components. Figure 1 shows an input example of five seconds audio belonging to the species *Alder flycatcher*. Due to the high computational demands of preprocessing, we decided to perform a sub sampling in which 70 thousand spectrograms were separated for training and 20 thousand for validation.

All preprocessing steps were carried out with two Python Libraries for audio processing: PyDub [2] and LIBROSA [1]. Table 1 shows the parameters used to extract the Mel-spectrograms.

Table 1. Mel Spectrogram Parameters.

Sampling Rate	22050Hz
Number of Mel coefficients	128
Window overlap	512
Window length	1024

3 Approach

3.1 Model

Our approach was inspired on [14]. Our key idea is to use a Siamese Convolutional Neural Network to extract features from Mel-spectrograms that can be arranged

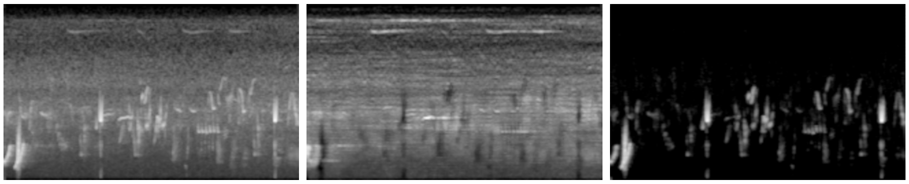

Fig. 1. Respectively Mel spectrogram, Harmonic Component and Percussive Component of Alder flycatcher

in a Euclidean space of n dimensions represented by an embedding vector. We tested two different CNN architectures based on [14]. In the first one, we reduce the number of Multiscale analysis modules from 4 to 2. In the second one, we keep the principal backbone architecture without the Dropouts layers between the two final dense layers. These adjustments were made empirically, based on test experiments. The table 2 shows the two implemented CNNs architectures. Both of them have inputs with shape 40x200x3, representing the three Mel-spectrogram with shapes 40x200 mentioned above. All convolutional layers use Relu as the activation function.

Table 2. The Two CNNs Architectures Used.

CNN1	**CNN2**
Input (40x200x3)	Input (40x200x3)
Conv(64,(3x3),Stride=(1x1))	Conv(64,(3x3),Stride=(1x1))
MAM1()	MAM1()
Conv(64,(3x3),Stride=(2x1))	Conv(64,(3x3),Stride=(2x1))
MAM2()	MAM2()
Conv(64,(3x3),Stride=(5x1))	Conv(64,(3x3),Stride=(2x1))
GlobalAveragePooling2D()	MAM3()
Dense(256)	Conv(64,(3x3),Stride=(2x1))
Dropout(0.5)	MAM4()
Dense(256)	Conv(64,(3x3),Stride=(5x1))
Dropout(0.5)	GlobalAveragePooling2D()
Dense(128,activation=linear)	Dense(256)
	Dense(256)
	Dense(128,activation=linear)

The Siamese network with triplet loss aims to generate close embedding vectors when the inputs belong to the same species and, at the same time, keep them away from vectors representing other species, considering a Euclidean vector space [12]. In this way, we hope that the new vector representation would make the classes more easily separable, therefore, easier to classify using a standard classifier such as kNN (k-Nearest Neighbors).

To achieve faster training convergence, we select the semi-hard triplets, which are the triplets composed by a negative sample more distant than the positive sample from the anchor sample. However, this distance lies inside a predefined margin α. Thus, with this loss function the lower the difference between positive and negative distances, the greater is the Loss. Equation 1 shows the loss function we want to minimize, where $f(x)$ is the embedding vector generated by the Siamese Network for the input sample x. For the i-th triplet x_i^a represent the anchor, x_i^p a positive sample, and x_i^n a negative sample.

$$L = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ \tag{1}$$

Once we convert all Mel-spectrograms into embeddings vectors, we can use them to classify each species. In our first submission, we chose a kNN classifier with Euclidean distance and, in our other submissions, a Multilayer Perceptron (MLP) neural network. The MLP architecture has an input layer, a hidden dense layer with 256 neurons and an output layer with softmax activation and 960 neurons, equivalent to the number of species we wish to recognize.

Table 3. MLP architecture.

MLP
Input (128)
Dense(256)
Dropout(0.5)
Dense(960,activation=softmax)

3.2 Train

We tried various combinations of hyperparameters until we reached our results. Table 4 shows the configuration used in each submission trial. The MLP was always trained with 200 epochs, batch size of 256 and learning rate equal to 0.001.

3.3 Submission

The test data has 153 soundscapes of 10 minutes each, for our submission, we divided them into segments of 5 seconds. For each segment, we build a Mel-spectrogram and its components. After that, we passed the spectrograms through

Table 4. Configuration of train per Submission

Submission	Model	Batch Size	Learning Rate	epochs	Margin (α)
1	CNN1+Knn	128	0.01	10	0.5
2	CNN1+MLP	128	0.001	20	0.5
3	CNN2+MLP	128	0.001	20	1.0
4	CNN2+MLP	32	0.001	30	1.0
5	CNN2+MLP	32	0.001	30	1.0

a CNN trained to obtain a 128-dimensional embedding vectors. Once we obtained the vectors, we use this to predict, using a classifier, a possible specie present on this part of that audio.

As our model is still not very accurate, for each 5 seconds piece of audio we get the 10 first possible species pointed out by the classifier and put them into submission file, given a higher weight to the first one and lower weight to the last one. We also take into account, and cut off, the species that does not belong to that region where the current soundscapes were recorded. The list with species per region and the information where each soundscape were recorded was provided along with the dataset.

4 Results

To evaluate the system, the competition provides a well-known ranking metric, the Mean Average Precision (MAP). But in this case, divided into two parts. The sample-wise mean average precision (r-mAP), the classic one, and the class-wise mean average precision, which takes the average precision (c-mAP) among the classes. Bellow, we have the equation for MAP, where Q is the number of test audio files and AveP(q) is an Average Precision for q file computed on equation 3:

$$MAP = \frac{\sum_{q=0}^{Q} AveP(q)}{Q}, \tag{2}$$

and

$$AveP = \frac{\sum_{k=0}^{N} (P(k) \times rel(k))}{num\_relevant\_documents}, \tag{3}$$

where k is the rank in the sequence of returned species, n is the total number of returned species, $P(k)$ is the precision at cut-off k in the list and $rel(k)$ is an indicator function equaling 1 if the item at rank k is a relevant species.

Table 5 shows the results for the submissions made, 1 and 2 made during the competition and the others made after the competition ended. The submission 4 and 5 use the same train configuration, the difference between them is that on submission time instead of given higher weight to the first one and lower weight, we assign a weight for all species equal to 1.

Table 5. Submission Evaluation.

Submission	(c-mAP)	(r-mAP)
1	0.006404	0.008
2	0.009752	0.008
3	0.020012	0.031
4	0.024094	0.041
5	**0.062877**	**0.108**

5 Conclusion

We applied a Siamese Network with Triplet Loss to the task of Bird Sounds Recognition. Our results were not the best, but we believe they can be improved. This is an initial study on this theme, from here a better study can be carried out. In this work we don't do any kind of data augmentation, a good work with augmentation techniques can be made with noise injection, as in [9, 7]. Therefore algorithms as Noise Reduction and per-channel energy normalization [10] still can be used on the preprocessing stage to make learning a simpler task.

On train stage, a better hyperparameter optimization should be done to achieve the optimal result, not only empirically choices. Furthermore, a different approach to the selection of the triplets can be tested, such as the dynamic triplet loss in which the margin varies according to the progress of the training [14].

The implementation made to run all the submissions mentioned here are available on `https://github.com/clementino1971/BirdCLEF_2020`. We would like to thank CLEF for organizing competitions of this type and encouraging research in this context of bioacoustic.

References

1. Librosa: audio and music processing in python, `https://librosa.org/`
2. Pydub: Manipulate audio with a simple and easy high level interface, `https://pydub.com/`
3. Driedger, J., Müller, M., Disch, S.: Extending harmonic-percussive separation of audio signals. In: ISMIR. pp. 611–616 (2014)
4. Honka, T.: One-shot learning with siamese networks for environmental audio (2019)
5. Joly, A., Goëau, H., Kahl, S., Deneu, B., Servajean, M., Cole, E., Picek, L., Ruiz De Castañeda, R., é, Lorieul, T., Botella, C., Glotin, H., Champ, J., Vellinga, W.P., Stöter, F.R., Dorso, A., Bonnet, P., Eggel, I., Müller, H.: Overview of lifeclef 2020: a system-oriented evaluation of automated species identification and species distribution prediction. In: Proceedings of CLEF 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
6. Kahl, S., Clapp, M., Hopping, A., Goëau, H., Glotin, H., Planqué, R., Vellinga, W.P., Joly, A.: Overview of birdclef 2020: Bird sound recognition in complex acoustic environments. In: CLEF task overview 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)

7. Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M., Eibl, M.: Large-scale bird sound classification using convolutional neural networks. In: CLEF (Working Notes) (2017)
8. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2. Lille (2015)
9. Lasseck, M.: Bird species identification in soundscapes. CLEF working notes (2019)
10. Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., Bello, J.P.: Robust sound event detection in bioacoustic sensor networks. PloS one **14**(10), e0214168 (2019)
11. Manocha, P., Badlani, R., Kumar, A., Shah, A., Elizalde, B., Raj, B.: Content-based representations of audio using siamese neural networks. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3136–3140. IEEE (2018)
12. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
13. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1420–1429 (2016)
14. Thakur, A., Thapar, D., Rajan, P., Nigam, A.: Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss. The Journal of the Acoustical Society of America **146**(1), 534–547 (2019)
15. Wang, J., Fang, Z., Lang, N., Yuan, H., Su, M.Y., Baldi, P.: A multi-resolution approach for spinal metastasis detection using deep siamese neural networks. Computers in biology and medicine **84**, 137–146 (2017)