Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Demonstrations Track

A Human-in-the-Loop Tool for Annotating Passive Acoustic Monitoring Datasets

Hannes Kath[1,2] , **Thiago S. Gouvêa**[1] and **Daniel Sonntag**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI), Oldenburg, Germany
[2]Applied Artificial Intelligence, Oldenburg University, Oldenburg, Germany
{hannes_berthold.kath, thiago.gouvea, daniel.sonntag}@dfki.de

Abstract

Deep learning methods are well suited for data analysis in several domains, but application is often limited by technical entry barriers and the availability of large annotated datasets. We present an interactive machine learning tool for annotating passive acoustic monitoring datasets created for wildlife monitoring, which are time-consuming and costly to annotate manually. The tool, designed as a web application, consists of an interactive user interface implementing a human-in-the-loop workflow. Class label annotations provided manually as bounding boxes drawn over a spectrogram are consumed by a deep generative model (DGM) that learns a low-dimensional representation of the input data, as well as the available class labels. The learned low-dimensional representation is displayed as an interactive interface element, where new bounding boxes can be efficiently generated by the user with lasso-selection; alternatively, the DGM can propose new, automatically generated bounding boxes on demand. The user can accept, edit, or reject annotations suggested by the model, thus owning final judgement. Generated annotations can be used to fine-tune the underlying model, thus closing the loop. Investigations of the prediction accuracy and first empirical experiments show promising results on an artificial data set, laying the ground for application to a real life scenario.

1 Introduction

Machine learning (ML) with deep neural networks has achieved excellent performance in many tasks. Yet, the impact of ML on several domains is limited by technical entry barriers, as well as by lack of domain-specific annotated data for supervised learning. Motivated by the quest to improve efficiency of passive acoustic monitoring (PAM) of animal biodiversity, we are developing a graphical interactive ML tool for detection and annotation of events in PAM datasets.

PAM is an increasingly popular method for continuous, reproducible, scalable, and cost-effective monitoring of animal wildlife [Sugai *et al.*, 2018]. While available low-cost recording devices have allowed large-scale data collection [Hill *et al.*, 2019], processing this data is a bottleneck.

Due to the low quality of automatically generated annotations for PAM datasets, annotation is usually done manually: domain experts listen to each audio file, annotating events by manually selecting time segments on a graphical representation of the sound (e.g. amplitude envelope or spectrogram) [Audacity Team, 1999; Tkachenko *et al.*, 2020; Perry *et al.*, 2021]. This approach is laborious and incompatible with the large volume of data generated by PAM. Seadash proposes a graphical implementation of data programming—simultaneous whole-dataset annotation with a set of user-defined heuristics [Ratner *et al.*, 2016]—but hasn't been evaluated on real life datasets [Gouvêa *et al.*, 2022]. DetEdit is a ML-free tool that allows simultaneous detection of bouts of events through a configurable signal processing pipeline that includes a GUI for accepting/rejecting detections; it runs on a proprietary platform, and has only been evaluated on odontocete echolocation click datasets [Solsona-Berga *et al.*, 2020]. scikit-maad is a tool for large scale PAM data analysis by spectrogram segmentation and clustering [Ulloa *et al.*, 2021]; as a command line tool, it lacks interactivity.

We present an interactive ML-based tool for annotating PAM datasets[1]. Implemented features are derived from audio annotation tools and domain expert experience. Our approach addresses three shortcomings of existing tools by allowing multiple events to be annotated simultaneously, using additional annotations to continuously speed up the process rather than following a linear annotation speed, and using clickable labels rather than error-prone manual input. The underlying deep generative model (DGM) [Rezende *et al.*, 2014] improves the machine predictions using the human-in-the-loop concept [Monarch, 2021] and distorts the latent space to represent events as outliers. Interactive tools using the latent space of ML systems can facilitate data interaction [Prange and Sonntag, 2021]. While many datasets (e.g. Xeno-canto[2]) are annotated weakly (i.e. on file level) and current tools create strong (i.e. timestamp level) labels [Perry *et al.*, 2021; Grover *et al.*, 2020], we use time- and frequency-aligned labels visualised as bounding boxes, allowing noise reduction and time-overlapping annotation (see figure 1, spectrogram).

[1]https://www.youtube.com/watch?v=VOfohkiWevU
[2]https://xeno-canto.org/

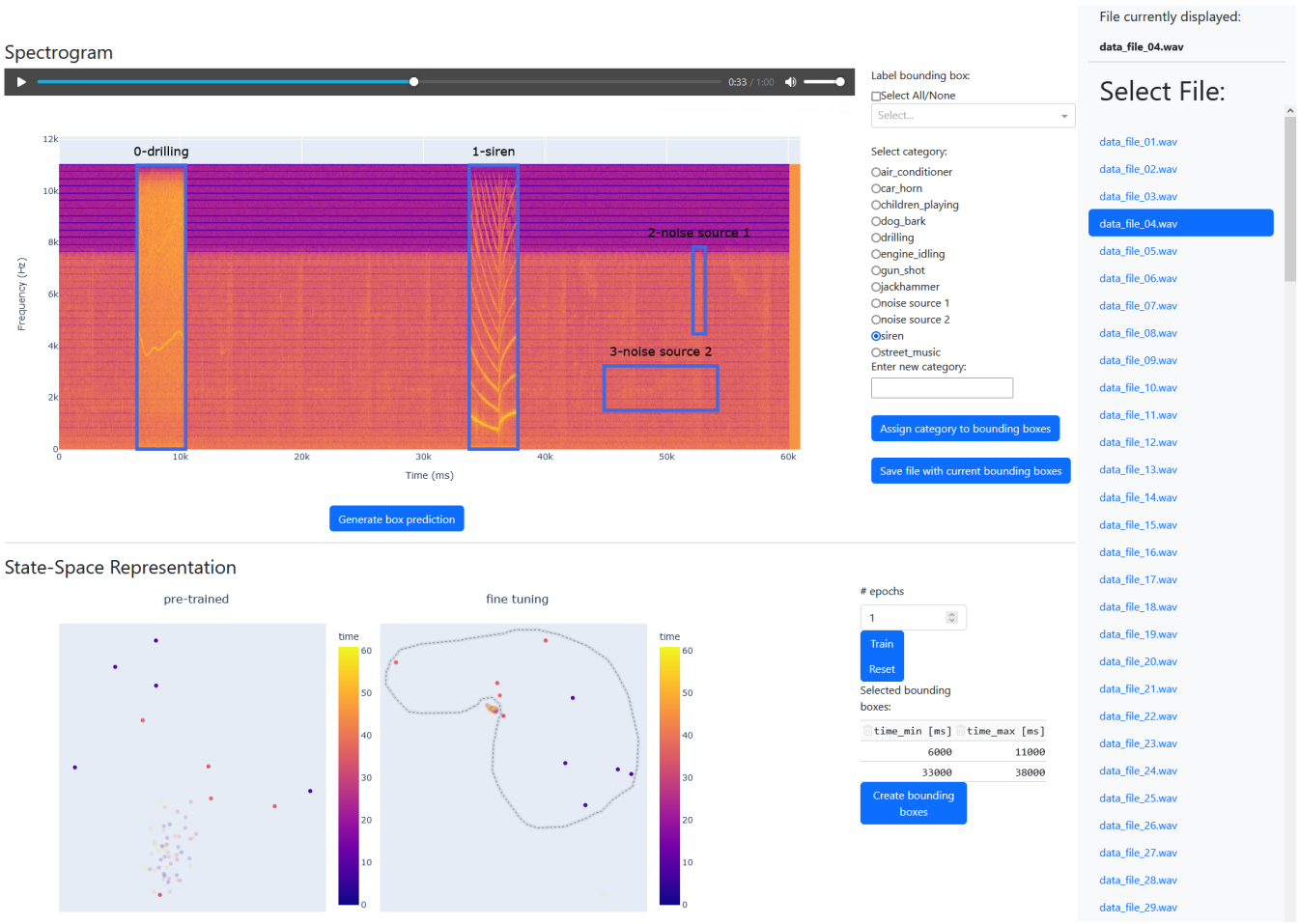

Figure 1: Layout of the user interface: file selection bar (right), spectrogram interaction row (top left), state-space interaction row (bottom left)

2 System Description

User Interface. We designed a Dash[3] interface composed of three parts, namely the file selection bar, the spectrogram interaction row and the state-space interaction row (see figure 1). The file selection bar allows the user to select an audio file for annotation from a list. The spectrogram interaction row displays annotation tools over and alongside the spectrogram of the selected file. Spectrograms are main elements of current audio annotation tools and lend themselves to an intuitive presentation of raw data. Hovering over the spectrogram reveals a toolbox that allows the user to zoom in and out, as well as create and edit time and frequency aligned bounding boxes to annotate regions of interest (ROIs). Selected ROIs can be played as audio. A button below the spectrogram allows quick annotation by suggesting bounding boxes and associated labels. Assigning, changing, and saving labels is possible through selection elements displayed alongside the spectrogram. More processed representations of the selected audio file are shown in the state-space interaction row, with each dot representing a colour-coded second in both figures. The left figure shows representations of the pre-trained unsupervised learning model, the right figure the fine-tuned model that also processes (generated) annotations. Hovering over the figures reveals a toolbox that allows users to zoom in and select data points, which are highlighted in the two figures and used to create a table of associated times on the right. A button below uses these times to create bounding boxes in the spectrogram. Continuous fine-tuning of the model using all (generated) annotations is possible by selecting a number of epochs and the corresponding button. Another button resets the model to the pre-trained state.

Workflow. Figure 2 shows the workflow of our system. Starting from an incompletely annotated dataset, the user loads the data used to train both models. The user selects a file for annotation in the file selection bar, which is displayed as a spectrogram and state-space representations of two different models. The spectrogram contains existing annotations. Adding annotations is done by one of three ways: The most intuitive way for experts is to draw bounding boxes directly on the spectrogram. Secondly, the user can retrieve a bounding box predicted by the fine-tuned system. And thirdly, the user can select one or more points in the state-space representation and create bounding boxes around the selected times. Our fine-tuned model is designed to represent events as out-

[3]https://dash.plotly.com/

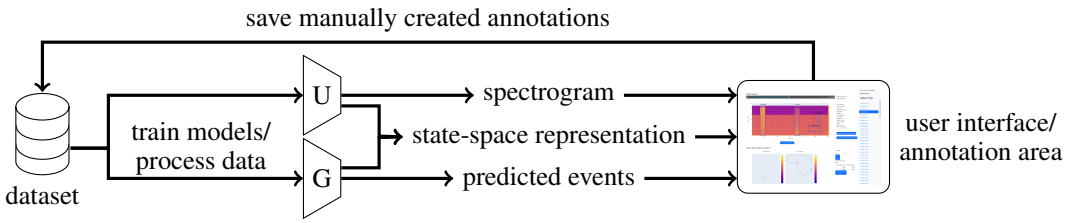

Figure 2: Workflow of the system: The unsupervised model (U) processes the entire dataset through unsupervised learning, the deep generative model (G) makes additional use of the existing annotations. After training, the dataset is processed by the models and used to create spectrograms, state-space representations and predict events, all presented to the user via an interface. Using these information, the user can annotate data, save the annotations and trigger re-training of the deep generative model.

liers, making it easy for the user to create accurate bounding boxes around any time-aligned regions of interest. Regardless of how the bounding boxes are created, the user can assign labels to them and move, scale and delete them in the spectrogram. Saving intermediate or final results is possible via a button and leads to an improvement of the dataset. This improvement can be used by selecting a number of epochs and re-training the model, resulting in more accurate bounding box suggestions as well as better separation in the state-space representation. The improvement of our model enables faster annotation of the next audio file.

Architecture of DGM. The requirements for the deep generative model include learning relevant data structures without annotations, mapping these structures into a 2D latent space, and a way to helpfully customise the latent space for the user based on the added annotations. Our derived model architecture is inspired by [Paige *et al.*, 2017] and extends a variational autoencoder (VAE) [Kingma and Welling, 2014] with a classification head. The input data X is processed by the encoder and mapped to the 2D-latent variable Z, which is presented to the user as a state-space and represents the input to the decoder and classifier. The decoder computes the reconstruction $\tilde{X}$. The classifier is implemented as a multilayer perceptron (MLP) and processes only annotated data by computing predicted labels $\tilde{Y}$ from Z for all files with a label Y, grouping in Z data points of the same category. The VAE and MLP are jointly optimized by minimizing the loss function

$$\mathcal{L} = \mathcal{L}_{\text{reconst}}(X, \tilde{X}) + D_{KL}\left(q_\phi(Z \mid X) \,||\, p(Z)\right) + \mathcal{H}(Y, \tilde{Y}),$$

where the first two terms are as in [Kingma and Welling, 2014], and $\mathcal{H}$ is the cross entropy between Y and $\tilde{Y}$. Efficient storage of bounding boxes is implemented using tidy data tables [Wickham, 2014]. Data pre-processing includes the calculation of the spectrogram and the subdivision of the audio files into second-long units. The DGM (referred to as fine-tuned model) implemented in Tensorflow[4], in the absence of labels, is identical to a VAE (referred to as pre-trained model) displayed for comparison purposes. While the autoencoder is trained on the entire dataset, the classification head only processes regions of bounding boxes, which is why conspicuous but uninteresting events (e.g. artefacts, geophony) can easily be ignored by the DGM.

[4]https://www.tensorflow.org/

3 Preliminary Evaluation

We ran preliminary evaluations of bounding box prediction and the usability of state-space representations. We created a dataset of 50 one-minute audio files composed of a PAM background (recorded in the Central Catchment Nature Reserve, Singapore) and foreground events from the UrbanSound8k dataset [Salamon *et al.*, 2014] inserted at random times (Poisson distributed, average of 4 per file). The VAE was trained on the entire dataset (pre-training); 30 files were used for fine-tuning (100 epochs), and the remaining 20 files for evaluation.

Bounding box prediction was evaluated by predicting three bounding boxes per file. Each bounding box capturing an event was considered correct. The prediction accuracy of the fine-tuned model is 79.9 %.

To obtain initial empirical results, we had a user create bounding boxes by selecting all elements in the state-space representations that were considered outliers of the pre-trained and fine-tuned model. For evaluation, we categorised selected items that contain events (true positive), selected items that do not contain events (false positive) and unselected items that contain events (false negative). To treat all events and predictions equally, we used the respective sums of all 20 evaluated files for the F-score calculation. The F-score of the pre-trained model is 77.0 %, the F-score of the fine-tuned model is 94.2 %.

4 Conclusion and Future Work

We propose an interactive, human-in-the-loop tool for ML-assisted annotation of PAM datasets. By modifying the latent space of a VAE through an added classifier head, we generate an actionable, low-dimensional representation of the input data that can improve efficiency of event detection and classification by the user. Future work includes making our tool applicable to real-world problems [Gouvêa *et al.*, 2023]. The webapp will be integrated with the users's PAM database and served remotely. To improve our tool in terms of implemented features (e.g. providing frequency units in the state-space representation, enabling playback of selected spectrogram regions, implementing existing libraries such as [Ulloa *et al.*, 2021], real-time update of the system) and interface design, we plan to follow a human-centred AI approach. Using design science research methods [Peffers *et al.*, 2008] we plan to conduct a user study with domain experts.

Ethical Statement

There are no ethical issues.

Acknowledgments

We thank Simone Dena and the Fonoteca Neotropical Jacques Vielliard (FNJV) for providing passive acoustic monitoring data. We thank Patrícia P. Serafini and Ivan Campos for sharing their experience in annotating passive acoustic monitoring datasets.

References

[Audacity Team, 1999] Audacity Team. Audacity. Available online at https://www.audacityteam.org/, 1999. Accessed: 2023-05-10.

[Gouvêa *et al.*, 2022] Thiago S. Gouvêa, Ilira Troshani, Marc Herrlich, and Daniel Sonntag. Annotating sound events through interactive design of interpretable features. In Stefan Schlobach, María Pérez-Ortiz, and Myrthe Tielman, editors, *HHAI 2022: Augmenting Human Intellect - Proceedings of the First International Conference on Hybrid Human-Artificial Intelligence, Amsterdam, The Netherlands, 13-17 June 2022*, volume 354 of *Frontiers in Artificial Intelligence and Applications*, pages 305–306. IOS Press, 2022.

[Gouvêa *et al.*, 2023] Thiago S. Gouvêa, Hannes Kath, Ilira Troshani, Bengt Lüers, Patrícia P. Serafini, Ivan Campos, André Afonso, Sérgio M. F. M. Leandro, Lourens Swanepoel, Nicholas Theron, Anthony M. Swemmer, and Daniel Sonntag. Interactive machine learning solutions for acoustic monitoring of animal wildlife in biosphere reserves. *IJCAI 2023*, 2023.

[Grover *et al.*, 2020] Manraj Singh Grover, Pakhi Bamdev, Yaman Kumar, Mika Hama, and Rajiv Ratn Shah. audino: A modern annotation tool for audio and speech. *CoRR*, abs/2006.05236, 2020.

[Hill *et al.*, 2019] Andrew P Hill, Peter Prince, Jake L Snaddon, C Patrick Doncaster, and Alex Rogers. Audiomoth: A low-cost acoustic device for monitoring biodiversity and the environment. *HardwareX*, 6:e00073, 2019.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[Monarch, 2021] Robert Munro Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.

[Paige *et al.*, 2017] Brooks Paige, Narayanaswamy Siddharth, Jan-Willem van de Meent, Alban Desmaison, Noah D. Goodman, Pushmeet Kohli, Frank D. Wood, and Philip H. S. Torr. Learning disentangled representations with semi-supervised deep generative models. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5925–5935, 2017.

[Peffers *et al.*, 2008] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *J. Manag. Inf. Syst.*, 24(3):45–77, 2008.

[Perry *et al.*, 2021] Sean Perry, Vaibhav Tiwari, Nishant Balaji, Erika Joun, Jacob Ayers, Mathias Tobler, Ian Ingram, Ryan Kastner, and Curt Schurgers. Pyrenote: a web-based, manual annotation tool for passive acoustic monitoring. In *IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2021, Denver, CO, USA, October 4-7, 2021*, pages 633–638. IEEE, 2021.

[Prange and Sonntag, 2021] Alexander Prange and Daniel Sonntag. A demonstrator for interactive image clustering and fine-tuning neural networks in virtual reality. In *KI 2021: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, September 27–October 1, 2021, Proceedings 44*, pages 194–203. Springer, 2021.

[Ratner *et al.*, 2016] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data Programming: Creating Large Training Sets, Quickly. In *Advances in Neural Information Processing Systems*, volume 29. NeurIPS, 2016.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.

[Salamon *et al.*, 2014] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22nd ACM International Conference on Multimedia (ACM-MM'14)*, pages 1041–1044, Orlando, FL, USA, Nov. 2014.

[Solsona-Berga *et al.*, 2020] Alba Solsona-Berga, Kaitlin E. Frasier, Simone Baumann-Pickering, Sean M. Wiggins, and John A. Hildebrand. Detedit: A graphical user interface for annotating and editing events detected in long-term acoustic monitoring data. *PLoS Comput. Biol.*, 16(1), 2020.

[Sugai *et al.*, 2018] Larissa Sayuri Moreira Sugai, Thiago Sanna Freire Silva, Jr Ribeiro, José Wagner, and Diego Llusia. Terrestrial Passive Acoustic Monitoring: Review and Perspectives. *BioScience*, 69(1):15–25, 11 2018.

[Tkachenko *et al.*, 2020] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020.

[Ulloa *et al.*, 2021] Juan Sebastián Ulloa, Sylvain Haupert, Juan Felipe Latorre, Thierry Aubin, and Jérôme Sueur. scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python. *Methods in*

Ecology and Evolution, pages 2041–210X.13711, September 2021.

[Wickham, 2014] Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014.