# MUSICAL INSTRUMENT RECOGNITION AS A TOOL FOR AUDIO PRODUCERS WITH VISION IMPAIRMENTS

*Hugo Flores García, Jack Wiig, Cooper Barth*

Northwestern University

## ABSTRACT

The advent of Digital Audio Workstations (DAWs) has popularized the craft of audio production by giving everyday users access to audio production tools and software at a cost that is orders of magnitude smaller than its analog counterparts. However, a lack of accessibility software for visually impaired users continues to be a prominent issue in digital audio production workflows. This paper proposes a fully accessible web-based tool for audio editing that incorporates a musical instrument recognition system to automatically annotate audio tracks with musical instrument labels. We discuss the design of our musical instrument recognition model, taking special care in improving the calibration and predictive uncertainty of our model. We hypothesize that the use of this system will remove a significant amount of navigational overhead required by visually impaired users to be able to complete common audio production tasks.

## 1. INTRODUCTION

Digital Audio Workstations (DAWs) have become a common household tool for audio production professionals and enthusiasts. Prior to the invention of DAWs, people interested in producing professional quality audio recordings were forced to resort to renting a professional recording studio at a costly rate, as well as paying for a studio engineer to operate a complex set of analog recording audio production devices. Now, a user needs only an audio interface and a DAW to produce professional quality audio recordings by themselves.

However, DAWs use primarily visual interfaces for representing audio content. This poses an inherent challenge to visually impaired users, who cannot benefit from the visual interfaces designed for a DAW. To worsen this matter, most DAWs do not have full screen reader support, as noted by [1]. But even if some DAWs have full screen reader support, the screen reader workflow is not optimized for common tasks that users may face. While a sighted user benefits from a meticulously designed visual interface meant to maximize efficiency, a non-sighted user must use an unoptimized interface designed with little to no concern about the user's ease of use and efficiency. As a result, blind users end up having to learn unintuitive sequences of keyboard shortcuts and screen reader sequences to complete trivial tasks.

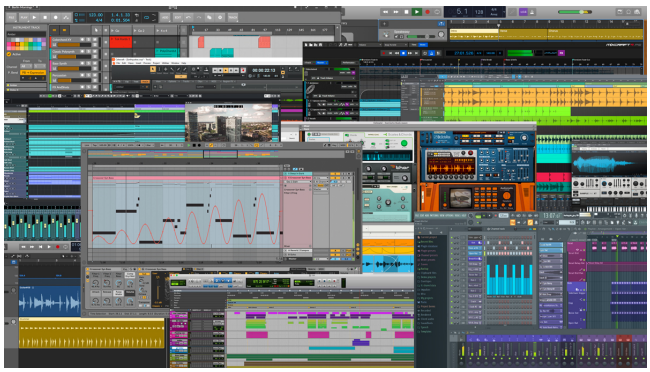

**Fig. 1**. DAWs have a dense interface that is highly visual in design.

In the field of audio production, there have been several efforts to design multimodal, tangible interfaces for accessible non-visual audio creation [2, 3, 4, 5]. In a similar content creation domain, Jayant et. al [6] use computer vision, audio feedback and face detection to help users with visual impairments take better photographs by providing them information about the location of faces within the camera frame.

To our knowledge, no work has yet explored the use of sound object detection within a DAW for improving accessibility to visually impaired users. This paper aims to contribute to this growing body of research in accessible content production interfaces by proposing a set of accessible DAW tools that use musical instrument recognition to improve user experience and optimize workflows for audio producers with visual impairments.

## 2. RELATED WORK

### 2.1. Accessibility in Audio Production for People with Visual Impairments

A recent review [7] of research in the design of accessible digital musical instruments finds that "little work has been done" for visually impaired users. To fill this gap in the literature, recent work [1, 8] has focused on understanding the current practices and struggles of visually-impaired audio production software users, in hopes of inspiring the design of novel, ac-

cessible interfaces for audio production. One of the key findings in [1] suggests that the involvement of visually impaired DAW users in the software development process is crucial to the success of the software at improving the user experience for visually impaired users.

Moreover, Saha and Piper [1] find, through a series of interviews, that for blind audio professionals and hobbyists, learning the craft of audio production (e.g. mixing, editing, mastering) is inseparable from learning a complex set of workarounds and non-optimized workflows, since modern DAWs lack support for efficient accessible workflows.

### 2.2. Musical Instrument Recognition

Musical Instrument Recognition can be performed in monophonic contexts [9, 10, 11], where only a single sound source may be active at any given time, as well as in polyphonic contexts [12, 13, 14, 15], where multiple sound sources may be active at the same time. We consider the monophonic case, as the vast majority of audio present in a DAW is monophonic.

Obtaining labeled datasets for musical instrument recognition (as well as more general sound event detection tasks) can be difficult and costly, as collecting human annotations for a large number of audio files is a tedious, time consuming task [16]. As a result, researchers are often forced to train and evaluate their models on small datasets that do not generalize well to out-of-distribution data.

To mitigate this issue, recent work has investigated methods for learning audio representations from a proxy task where self supervision is possible or obtaining labeled data is not a costly endeavor. These learned representations can then be used for downstream classification tasks where labeled data may be scarce. Arandjelovic et. al [17] and Cramer et. al [18] showed that using an audio embedding pre-trained on a self-supervised audio-visual correspondence (AVC) task can improve generalization performance on downstream audio classification tasks with smaller datasets. Because our dataset is considerably small, we make use of the pre-trained audio embedding model in [18].

### 2.3. Predictive Uncertainty in Classification Models

Since our dataset does not cover every musical instrument that may appear in an audio production scenario, we must deal with out-of-distribution data without making overconfident predictions and potentially worsening the user experience for visually-impaired users by introducing erroneous information that they may have to manually correct.

Deep learning models trained with only one-hot encoded labels are likely to make low-entropy predictions regardless of the classifier's true uncertainty, or whether the provided input lied within the training distribution or not. Thulasidasan et. al [19] find that models trained using mixup [20] exhibit significantly better calibration properties for both in and out of distribution data when compared to models trained using effective risk minimization, under the argument that mixup provides a form of entropic regularization on the training signals. We refer the reader to [20] for an overview on the mixup algorithm.

## 3. SYSTEM DESIGN

Building a set of musical instrument recognition tools for visually impaired users comprises of two major tasks: building a monophonic musical instrument recognition model, and developing a set of navigational tools that take advantage of the labels created by the musical instrument recognition model to optimize non-visual DAW workflows. In this section, we discuss the design choices for our musical instrument model and evaluate different model variants on standard classification metrics as well as predictive uncertainty. We make all of our experiment code available on GitHub [1].

### 3.1. Musical Instrument Recognition Model

In order to perform musical instrument recognition, we design a deep audio classification model capable of classifying 1 second audio frames into a set of musical instrument classes. We use the audio subnetwork from $L^3$-Net [17], using the pre-trained music model variant proposed by [18] as an audio embedding. The audio embedding model uses a 128 bin log-mel spectrogram of 48kHz audio as an input representation, and is composed of convolutional blocks with batch normalization and maxpool layers. We refer the reader to [17] for more information about the $L^3$-Net audio subnetwork architecture. The embedding model produces a 6144-dimensional embedding that is used as input for our classifier model. We refer the reader to our blog post[2] for more detail on our classifier architecture.

### 3.2. Data and Data Augmentation

We train the model using a 20 instrument subset the MedleyDB dataset [21]. The MedleyDB dataset is a multi-track dataset with 122 mixtures of real-life recordings of musical instruments and vocals and their corresponding stems. The dataset is split randomly into a train and validation set with an artist conditional split. For the full list of classes, we refer the reader to our blog post [2].

We preprocess the data by removing silent regions and splitting each stem into 1s segments with a hop size of 250ms. We augment samples in the training dataset by introducing random time stretching, pitch shifting, overdrive, flanger, phasor, compression, and EQ. Our augmented dataset consists of 288k training samples and 56k validation samples. The model is optimized to minimize the cross-entropy loss between model probabilities and ground truth labels.

### 3.3. Evaluation Metrics

We consider two primary evaluation metrics: micro-averaged F1-score and expected calibration error (ECE). We assume the reader is familiar with the definition of F1-score.

[1] github.com/hugofloresgarcia/instrument-recognition

[2] cod-audio.github.io/cod-site

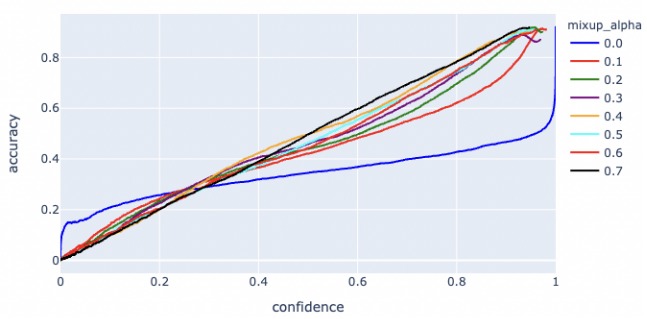


**Fig. 2**. Reliability diagrams for varying levels of $\alpha$. The reliability diagram plots accuracy vs average confidence a set of binned predictions.

We use the expected calibration error (ECE) as a metric indicative of our model's predictive uncertainty. The ECE of a model is defined as the expected difference between model confidence and prediction accuracy. The ECE is computed by partitioning a set of test set predictions into $M$ bins and calculating a weighted average of the difference between the accuracy and average confidence for each bin:

$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$

where $B_m$ is a bin with predictions, $n$ is the number of samples, and $acc(B_m)$ and $conf(B_m)$ are the prediction accuracies and model softmax probabilities, respectively.

### 3.4. Experimental Setup

We evaluate the effect of mixup training on the predictive uncertainty and classification performance of our models. We train model variants with varying degrees of $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, where $\alpha = 0$ is the baseline case, with no degree of interpolation between training samples. Mixup is performed at the embedding level.

For all trials, each model was trained for 100 epochs, with 288k training examples for each epoch. All models were trained using a batch size of 512 and the Adam optimizer with an initial learning rate of $3 \times 10^{-3}$, decaying to $3 \times 10^{-4}$ and $3 \times 10^{-5}$ after 50 and 75 epochs, respectively. All models are trained to minimize the cross entropy loss between model predictions at the softmax layer and ground truth labels. All model weights are initialized to the same random state. We evaluate each classification model on the validation dataset using a micro-averaged F1-score and ECE.

## 4. RESULTS

Experimental results are shown in Table 1. We find that setting $\alpha$ to 0.4 achieves both the lowest ECE and highest F1. Additionally, we observe that classification performance starts to degrade with higher values of $\alpha$, likely due to underfitting.

Reliability diagrams for each model variant are shown in Figure 2. To create a reliability diagram, we split our test set predictions into M bins and calculate the accuracy and average confidence for each bin. Then, we sort the bins by confidence (X axis) and plot the accuracy with respect to confidence for each bin.

| mixup-$\alpha$ | ECE | F1-score |
|---|---|---|
| 0.0 | 24.4% | 70.3% |
| 0.1 | 11.8% | 70.0% |
| 0.2 | 06.1% | 70.3% |
| 0.3 | 04.7% | 70.2% |
| 0.4 | **02.5%** | **71.4%** |
| 0.5 | 03.6% | 69.7% |
| 0.6 | 04.9% | 68.8% |
| 0.7 | 03.3% | 69.4% |

**Table 1**. Metrics for experiments on mixup: Expected Calibration Error (ECE, lower is better) and F1-score (higher is better). Best results shown in bold.

We observe that the model trained with no mixup ($\alpha$ = 0) tends to make substantially overconfident predictions when its accuracy is higher, and underconfident predictions when its accuracy is lower. On the other hand, models trained with mixup exhibit better calibration properties; that is, their predictive confidences are more aligned with their accuracies.

## 5. WEB BASED AUDIO EDITOR

We developed Cod[3], a simple web-based audio editor, in order to demonstrate the use of the model for automatic label generation. Users can either add labels to loaded audio manually or choose to generate them, which will call the model with the audio buffer as input and return a set of labels to display.

The interface is fully accessible to screen readers, with keybinds to create, edit, and delete labels, move the playhead to selected labels, and jump between labels of the same type.

## 6. CONCLUSION

It is of utmost importance that ongoing research addresses the problem of lack of accessibility in content creation interfaces, as these experiences should be accessible to everybody. We proposed a musical instrument labeling system that aims to reduce the navigational overhead for visually impaired users navigating a DAW. Moreover, we observed the effect of mixup training the calibration of our model and found that it significantly improves our model's predictive uncertainty.

Since we have achieved good predictive uncertainty estimates, future work will explore using an open vocabulary classification setting where the user can add manual annotations to low confidence predictions, which can in turn be used in a few-shot learning scenario to expand the vocabulary of our model. We hope our work helps lower the technical barrier that many visually impaired users have to overcome when interacting with content creation interfaces.

[3]https://cod-audio.github.io/cod/

## 7. REFERENCES


[1] Abir Saha and Anne Marie Piper, "Understanding audio production practices of people with vision impairments," in *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20), October 26–28, 2020, Virtual Event, Greece.* IEEE, 2020.

[2] Thomas Haenselmann, Hendrik Lemelson, Kerstin Adam, and Wolfgang Effelsberg, "A tangible midi sequencer for visually impaired people," in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, MM '09, p. 993–994, Association for Computing Machinery.

[3] Shotaro Omori and Ikuko Eguchi Yairi, "Collaborative music application for visually impaired people with tangible objects on table," in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, NY, USA, 2013, ASSETS '13, Association for Computing Machinery.

[4] Oussama Metatla, Fiore Martin, Adam Parkinson, Nick Bryan-Kinns, Tony Stockman, and Atau Tanaka, "Audio-haptic interfaces for digital audio workstations: A participatory design approach," *Journal on Multimodal User Interfaces*, vol. 10, 05 2016.

[5] Atau Tanaka and Adam Parkinson, "Haptic wave: A cross-modal interface for visually impaired audio producers," 05 2016, pp. 2150–2161.

[6] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham, "Supporting blind photography," in *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, NY, USA, 2011, ASSETS '11, p. 203–210, Association for Computing Machinery.

[7] Emma Frid, "Accessible digital musical instruments—a review of musical interfaces in inclusive music practice," *Multimodal Technologies and Interaction*, vol. 3, no. 3, pp. 57, Jul 2019.

[8] Fabiha Ahmed Lisa Ye William C. Payne, Alex Yixuan Xu and Amy Hurst., "How blind and visually impaired composers, producers, and songwriters leverage adapt music technology," in *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20), October 26–28, 2020, Virtual Event, Greece.* IEEE, 2020.

[9] E. Benetos, M. Kotti, and C. Kotropoulos, "Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006, vol. 5, pp. V–V.

[10] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, 2000, vol. 2, pp. II753–II756 vol.2.

[11] Vincent Lostanlen and Carmine-Emanuele Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," *CoRR*, vol. abs/1605.06644, 2016.

[12] Yoonchang Han, Jaehun Kim, Kyogu Lee, Yoonchang Han, Jaehun Kim, and Kyogu Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 25, no. 1, pp. 208–221, Jan. 2017.

[13] Y. Hung, Y. Chen, and Y. Yang, "Multitask learning for frame-level instrument recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 381–385.

[14] Yun-Ning Hung and Y. Yang, "Frame-level instrument recognition by timbre and pitch," *ArXiv*, vol. abs/1806.09587, 2018.

[15] Siddharth Gururani, Mohit Sharma, and Alexander Lerch, "An attention mechanism for musical instrument recognition," *CoRR*, vol. abs/1907.04294, 2019.

[16] Bongjun Kim and Bryan Pardo, "I-sed: An interactive sound event detector," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, New York, NY, USA, 2017, IUI '17, p. 553–557, Association for Computing Machinery.

[17] Relja Arandjelovic and Andrew Zisserman, "Look, listen and learn," in *IEEE International Conference on Computer Vision (ICCV))*. IEEE, Oct 2017.

[18] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello, "Look, listen and learn more: Design choices for deep audio embeddings," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019, pp. 3852–3856.

[19] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," 2020.

[20] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[21] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.