INSTRUMENT ACTIVITY DETECTION IN POLYPHONIC MUSIC USING DEEP NEURAL NETWORKS

Siddharth Gururani[1] **Cameron Summers**[2] **Alexander Lerch**[1]

[1] Center for Music Technology, Georgia Institute of Technology , USA
[2] Gracenote, Emeryville, USA

{siddgururani, alexander.lerch}@gatech.edu, cameron.summers@nielsen.com

ABSTRACT

Although instrument recognition has been thoroughly research, recognition in polyphonic music still faces challenges. While most research in polyphonic instrument recognition focuses on predicting the predominant instruments in a given audio recording, instrument activity detection represents a generalized problem of detecting the presence or activity of instruments in a track on a fine-grained temporal scale. We present an approach for instrument activity detection in polyphonic music with temporal resolution ranging from one second to the track level. This system allows, for instance, to retrieve specific areas of interest such as guitar solos. Three classes of deep neural networks are trained to detect up to 18 instruments. The architectures investigated in this paper are: multi-layer perceptrons, convolutional neural networks, and convolutional-recurrent neural networks. An in-depth evaluation on publicly available multi-track datasets using methods such as AUC-ROC and Label Ranking Average Precision highlights different aspects of the model performance and indicates the importance of using multiple evaluation metrics. Furthermore, we propose a new visualization to discuss instrument confusion in a multi-label scenario.

1. INTRODUCTION

Music is an acoustic rendition of musical ideas. In most cases, one or more instruments are used for this acoustic rendition. As humans, we are easily able to identify the instruments being played in a song after exposure to their sound. However, the same cannot be said for computer algorithms. The task of recognizing musical instruments in an audio signal has been an active area of research in the field of Music Information Retrieval (MIR). While instrument recognition in monophonic audio (only one instrument is present in a signal) is reasonably successful [13], the task is much harder in a polyphonic setting. The challenges include, among others, the large variance in timbre and performance style within an instrument class combined with perceptual similarity of some instruments and the superposition of multiple instruments in time and frequency. Last but not least, the lack of data with relevant annotations for data-driven approaches is also a problem.

The identification of instruments and their activity in a song is important for music browsing and discovery, such as searching for songs with specific instruments or identifying the position of lead vocals or a saxophone solo. Instrument recognition can also inform other MIR tasks. For example, music recommendation systems can benefit from modeling a user's affinity towards certain instruments and music genre recognition systems could improve with genre-dependent instrument information. It can also be useful for tasks such as automatic music transcription, playing technique detection, and source separation in polyphonic music, where pre-conditioning a model on specific instruments present could possibly boost its performance.

An Instrument Activity Detection (IAD) system takes an audio track as input and outputs continuous instrument activity levels along the entire track. These activities may be binary (on/off) or on a continuous scale as likelihood. IAD systems may have varying time-resolutions for the instrument activity depending on the use case. For example, a solo detection use case would have a finer time-resolution that an instrument tagging system which would work on the track level. This paper proposes a deep neural network-based IAD system trained using multi-track datasets. We also address the problem of evaluation of an IAD system.

The following section reviews literature in instrument recognition and other related tasks. Section 3 describes the proposed IAD system starting with pre-processing the data, the model architectures and post-processing steps. Next, Section 4 describes the dataset used, the various experiments, the evaluation metrics and the proposed method to visualize confusion. We report the results for the experiments in terms of the evalution metrics and discuss these results in Section 5. Finally, in Section 6 we conclude the paper enumerating a few possible future directions for research on IAD.

2. RELATED WORK

The task of 'instrument recognition' can be divided into two distinct research problems based on the type of data being analyzed: (i) instrument recognition in monophonic audio and (ii) instrument recognition in polyphonic audio. This section presents an overview of past literature on instrument recognition as well as related topics such as automatic music tagging and sound event detection (SED).

2.1 Instrument Recognition in Monophonic Music

In monophonic music, instrument recognition may be performed on sounds at the note-level or on continuous audio signals of solo instrument performances. An extensive review of traditional feature extraction and classification approaches for note-level instrument recognition has been published by Herrera et al. [15]. For solo phrases, Essid et al. utilize MFCCs as features, Principal Component Analysis (PCA) for dimensionality reduction, and Gaussian mixture models (GMM) for classifying solo phrases of 5 instruments [8]. Krishna and Sreenivas propose the so-called Line Spectral Features (LSF). LSFs are used with a GMM and evaluated for instrument family classification and 14-class instrument classification [19].

In addition to extracting established pre-defined features, learned features have also been applied to this task. Yu et al. utilize sparse spectral codes and a support vector machine (SVM) for classifying single-source and multi-source (polyphonic) audio [31]. Han et al. propose to use sparse coding for learning features from mel-spectrograms extracted from a dataset of single-note audio clips for 24 instruments. A SVM is trained to classify the instruments using the learned features achieving a classification accuracy of around 95% for 24 instrument classes [13].

2.2 Instrument Recognition in Polyphonic Music

Recent work on instrument recognition has focused on polyphonic musical signals. Polyphonic audio synthesized from datasets of individual instrument sounds, such as the RWC dataset [10], as well as real-world audio recordings have been used for this task.

Kitahara et al. extract spectral and temporal features along with PCA and Latent Discriminant Analysis (LDA) for classification in duo and trio music [17]. Heittola et al. combine the results of Non-negative Matrix Factorization (NMF) with excitations of notes obtained from a multi-pitch tracking algorithm [18] to extract harmonic spectra from a mixture signal. The separated spectra are represented by MFCCs and classified with a GMM [14].

Fuhrmann et al. extract a large set of features representing an audio clip and perform predominant instrument detection in real-world audio signals using one SVM per instrument [9]. The 'predominant' instrument is defined as one with continuous presence in a snippet of audio and is easily audible for a human listener. Bosch et al. extend the work by utilizing source separation to segregate the polyphonic audio into streams: 'bass,' 'drums,' 'melody,' and 'other.' The segregated audio is subsequently used for classification using the aforementioned system [2].

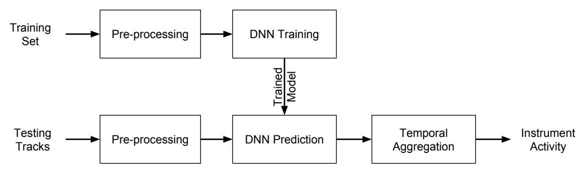

Figure 1. Block Diagram for DNN-based IAD System

Han et al. apply deep CNNs for the task of predominant instrument recognition and report a significant improvement of results over previous approaches [12]. The authors also provide an in-depth discussion of the model parameters and a qualitative analysis of the CNN models.

2.3 Related Tasks

In music tagging, a track is labeled with a variety of labels that describe it, such as genre, instruments, and mood. IAD may be considered a sub-task in music tagging since the tags often include instrumentation. Choi et al. use CNNs and CRNNs for the task of automatic tagging [5, 6]. Liu and Yang further proposed a method to localize the events in music tagging [20] which may be compared to IAD.

Sound Event Detection (SED) aims at detecting environmental sounds in a stream of audio. Some examples of sound events are gunshots, car horns, baby cries, dog barks, etc. Cakir et al. explore this task with deep neural networks on a dataset of environmental sounds [3, 4]. The main difference between SED and IAD is that in SED the sound events are uncorrelated and thus easier to discriminate while musical sources tend to have higher correlation in popular music. Music instrument sounds might also have a rich harmonic structure absent in most environmental sounds.

3. METHOD

A high-level block diagram for the presented IAD system is shown in Fig. 1. The individual processing steps are described in detail below.

3.1 Pre-processing

All tracks are downsampled to 22.05 kHz, downmixed to mono and normalized by the root mean square energy. Each track is the chunked to 1 s long snippets. Each snippet is transformed into a mel-spectrogram, which is motivated by the non-linear frequency resolution of the human auditory system [22], and has been proven to be a useful input representation for multiple MIR tasks such as automatic tagging [5], onset detection [25], and feature learning [29].

The mel-spectrograms are calculated using Librosa [21] with 96 mel bands from 0–11.025 kHz. The block size and hop size are 46.4 ms and 11.6 ms, respectively. Decibel scaling is applied to the Mel-Spectrogram energies. The result is a matrix of dimension 96×86.

CNN	CRNN
Conv2D $k = 3 \times 3, d = 64$ MP $(p = 2, 2)$	
Conv2D $k = 3 \times 3, d = 128$ MP $(p = 2, 2)$	
Conv2D $k = 3 \times 3, d = 256$ MP $(p = 3, 3)$	Conv2D $k = 3 \times 3, d = 256$ MP $(p = 2, 2)$
Conv2D $(k = 3 \times 3, d = 640)$ MP $(p = 3, 3)$	Conv2D $(k = 3 \times 3, d = 256)$ MP $(p = 2, 2)$
FC $(h = 128)$	GRU $(h = 256)$
FC $(h = 18)$	

Table 1. Model Architecture. (Conv2D: 2D Convolutional Layer, MP: 2D Max-Pooling, k: kernel size, d: filter depth)

3.2 Model Architectures

Deep Neural Networks (DNNs) have consistently outperformed traditional MIR approaches in several tasks such as, music transcription [26, 30], onset detection [25], music tagging [5]. As this is also true for predominant instrument classification (compare Sect. 2), we choose to investigate DNNs for the task of IAD. Our architectural choices are influenced by the work of both Choi and Cakir [4–6].

The usability of DNNs stems from their ability to approximate complex non-linear functions mapping an input feature space to the outputs. This enables researchers to provide raw or minimally processed data to a DNN so that it may learn features relevant for the task at hand.

We compare the three broad classes of DNNs: multi-layer perceptrons, convolutional neural networks, and — since convolutional networks are useful for acoustic modeling [5]— a convolutional-recurrent network instead of a traditional RNN. The benefit of CRNN lies in the fact that it is able to learn both local and temporal features. Note that the model hyperparameters have been chosen so that the number of parameters for the three models is comparable.

3.2.1 Multi-Layer Perceptron

The input mel-spectrogram matrix is flattened into a vector for the MLP model. A fairly simple architecture is chosen: 4 hidden layers with 256 hidden units in each layer and an output layer of 18 hidden units. Dropout [28] is used with a keep probability of 0.5 at each layer.

3.2.2 Convolutional Neural Network

The CNN architecture is shown in Table 1 (left). Small square filters are chosen in order to facilitate hierarchical feature learning from local patches that grow larger in size with network depth. In order to preserve spatial dimensions, stride of 1 and *Same* zero-padding scheme is used for all the convolutional layers. Each Conv2D layer is followed by batch-normalization [16] and the Exponential Linear Unit (ELU) [7] activation function. The final convolution layer's output is flattened before feeding it to a fully connected layer. Finally, we connect to an output layer of 18 units with a sigmoid activation function.

		Train		Test	
Instrument	Abbr.	T	#	T	#
drum set	dru	300	720036	79	15957
electric bass	bgtr	253	620592	62	13344
male singer	ms	200	351384	62	10038
dist. elec. gtr	dgtr	171	396204	40	7522
clean elec. gtr	cgtr	119	225456	34	5875
synthesizer	syn	118	295524	33	5712
acoustic gtr	agtr	91	230556	25	5241
piano	pf	89	187536	24	4063
vocalists	vox	84	154596	12	1895
female singer	fs	79	149232	23	3733
string section	str	24	39444	10	1278
elec. piano	epf	24	52680	14	2075
elect. organ	eorg	22	39516	11	2117
double bass	db	21	40116	9	1786
cello	vc	13	22176	9	1623
violin	vn	10	28452	15	2385
tabla	tab	9	41640	3	806
flute	fl	7	9972	7	1171

Table 2. Dataset distribution: T denotes tracks and # denotes 1 s snippets

3.2.3 Convolutional Recurrent Neural Network

The CRNN architecture is shown in Table 1 (right). CRNNs have been applied to tasks such as music tagging [6] and sound event detection [4]. We hypothesize that it is a good choice for IAD since we want the model to learn from the evolution of spectra over time. The same configuration of padding and striding, batch-normalization, and non-linear activation is used for the convolutional modules. Only the depth and height of the final Conv layer output is flattened, thus preserving the temporal structure of the high-level ConvNet features. Finally, the last GRU output is connected to the output layer consisting of 18 units with a sigmoid activation function.

3.2.4 Training Procedure

Binary cross-entropy is used as the loss function for all models. Stochastic gradient descent with a learning rate of 0.0001 and momentum of 0.9 is used to optimize the loss function. The models are trained using batches of 32 instances for 20 epochs, which is sufficient for the training and validation loss to converge for each of the architectures.

3.3 Temporal Aggregation

Since the neural networks are trained using 1 s snippets of audio, a prediction is made for every 1 s in the test track. For experiments and evaluation with varying time-resolution, we max-pool the predictions and the ground truth over non-overlapping segments according to the desired time-resolution. For example, in order to have a 5 s resolution, the maximum across 5 continuous predictions for every instrument is chosen as the predicted score for the corresponding 5 s snippet in the track.

4. EVALUATION

4.1 Dataset

The dataset used in previous work on predominant instrument detection [2, 9, 12], IRMAS, consists of a training set with 3 s audio snippets manually annotated with one of 11 predominant (but non-percussive) instrument labels. These snippets may contain other instruments. The testing set contains audio snippets of variable length with 1 or more predominant instruments. We believe that training using polyphonic audio labeled with a single instrument may not be the ideal strategy for IAD. In this paper, we used multi-track audio to construct a dataset for IAD. The motivation behind using multi-track datasets is that the annotations for instrument activity can be generated automatically using stem energy as opposed to human annotations which may contain more errors. In addition, each snippet may contain multiple instrument labels, providing the models richer ground truth.

Two publicly available multi-track datasets are used for training and testing of the models. MedleyDB [1] and Mixing Secrets [11] were combined for this task in order to increase the number of tracks. In a pilot study involving only MedleyDB, we observed a significant improvement in model performance as the amount of training data increased. MedleyDB contains 330 multi-tracks and Mixing Secrets contains 258 multi-tracks. The two datasets combined contain tracks with approximately 100 different instruments. For this paper we consider 18 most frequently occurring instruments. The instruments considered are listed in Table 2. Note that the tracks may contain other instruments that the IAD system is not trained to detect.

Each multi-track in the dataset is associated with a mixed track. Instrument activation confidence is annotated automatically according to the process described in [1]. These annotations are computed with time-resolution of 0.0464 s. For our IAD system, however, we defined the minimum time-resolution to be 1 s. The annotations are aggregated by picking the maximum value across the time-axis to obtain one activation value per instrument per snippet. This allows for instruments to have a large activation value in the snippet even if they were active for a small period of time, as opposed to a value close to 0 if the mean was chosen for aggregation. Finally, the activations are binarized with a fixed threshold $\theta = 0.5$.

The datasets contain tracks where the stems have cross-talk or bleed. For these tracks, stem activations for a certain instrument may contain activity from another instrument. To prevent incorrect annotations, tracks with bleed are not considered for the IAD dataset, although we make exceptions for rare instruments such as tabla. Additionally, tracks without a single instrument of interest are not considered.

Subsequently, the dataset is split into a training and a testing set. We generate a random artist-conditional split to prevent the album or artist effect in the testing phase. The split is chosen such that there is a reasonable number of tracks per instrument. Table 2 lists the distribution of the data for the split. The training set consists of 361 tracks and the testing set consists of 100 tracks. [1] The training set is augmented using pitch-shifting: 6 semitones lower to 5 higher than the original with 1 semi-tone increments.

[1] The track IDs for the dataset splits used are available at https://github.com/SiddGururani/ISMIR2018

4.2 Experimental Setup

First, we preprocess both splits of data as described in Sect. 3.1 resulting in a time-frequency input representation and ground truth pair for each 1 second snippet. Table 2 lists the distribution of the different instrument classes in terms of 1 second snippets. Next, we train each of the DNN architecture as described in Sect. 3.2. Since the models were observed to converge to a solution in 20 epochs, we do not perform any form of early-stopping. In addition, we generate a validation set using a randomly sampled set of tracks from both the training and testing set due to lack of data. 50 tracks from the training and testing splits are picked, resulting in a validation set of 100 tracks. We use this scheme since we want to validate on unseen data while not using the entire test set. The validation set is used to evaluate the models at the end of each epoch. Finally, we test the best performing model for each class of DNNs. We test the models for various time-resolutions of activity detection: 1 s, 5 s, 10 s and track-level aggregation.

4.3 Evaluation Metrics

Evaluation of IAD systems, when looked at in detail, poses some challenges. Since each snippet has zero or more instruments, IAD is a multi-label classification problem. The sigmoid activation leads to an output between 0 and 1, denoting the predicted activity of that instrument. However, as pointed out by Han et al. [12], binarizing the outputs using a fixed threshold and evaluating the accuracy depends on the selected threshold. Additionally, the dataset is not balanced across the instrument classes, hence stressing the need for metrics robust against unbalanced class distribution.

Previous work on predominant instrument recognition uses metrics relevant for multi-class classification systems such as precision, recall and f-measure [2,12]. Since IAD is a multi-label classification problem, we use Label Ranking Average Precision (LRAP) and the Area Under Receiver Operating Characteristic curve (AUC-ROC).

4.3.1 Label Ranking Average Precision

LRAP was proposed in [24] to evaluate multi-label classification systems. Intuitively, the LRAP measures the ability of a model to assign better ranks to true labels for an instance. For example, if all the true labels for an instance are ranked higher than other labels in consideration, the ranking precision for this instance is 1. LRAP measures the average ranking precision across all the instances. In our experiments, we compute LRAP using 2 approaches: (i) Micro: LRAP computed using the concatenated outputs for all testing tracks. (ii) Macro: computed on the track level and averaged. This normalizes any effect of track length on the model performance, which could skew the results, for instance, if the model performs well for a particular long song but poorly for shorter songs with fewer snippets.

4.3.2 Area Under ROC Curve

The AUC-ROC or, in short, AUC is computed by first plotting the true positive rate and false positive rate on a plane for various classification thresholds, which results in a curve.

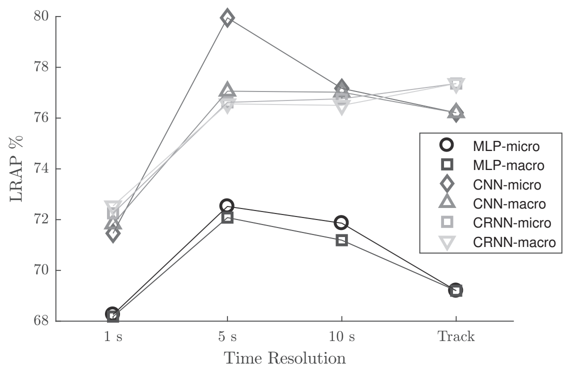

Figure 2. LRAP for various time-resolutions

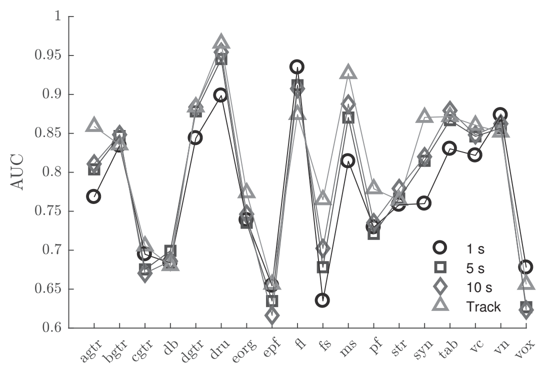

Figure 3. AUC per instrument for CRNN model

AUC is the area under this curve. It measures the probability that the model assigns a higher score to a randomly selected positive instance than a negative instance. The AUC gives a summary of the model performance without the need to adjust a threshold for binarization.

Since AUC is usually applied to binary classification, we compute it per instrument class. Only the micro AUC is computed as not all tracks contain all instruments. We report an average AUC by taking the mean of the micro AUC per class.

The reason for selecting AUC instead of precision, recall or f-measure is that most literature on instrument classification tends to use a common fixed threshold for all classes which bears the risk of being suboptimal. Han et al. suggest the use of a different threshold per instrument class [12]. Using the AUC to summarize model performance alleviates the problem of threshold selection while making it easier to directly compare model performance.

4.4 Confusion Visualization

In multi-class classification, every data sample has only one possible prediction and one ground truth label. A confusion matrix visualizes the frequency of confusion between every pair of predicted class label vs. ground truth class label. In a multi-label classification problem such at IAD, every instance has multiple possible predictions and zero or more ground truth labels. Hence, a traditional confusion matrix cannot be computed. However, as a confusion matrix is an intuitive way to gain insights into the model, we propose an alternative form of confusion visualization computed from the binarized predictions and the ground truths.

We hypothesize that an instrument is wrongly detected due to the activity of some instrument present in the audio. We are particularly interested in looking at which instruments were incorrectly missed (false negative) when an instrument was wrongly detected (false positive). For a particular false positive instrument, this is equivalent to looking at the probability of observing false negatives for the other instruments. This probability can be estimated using a histogram of false negatives. Vertically stacking these histograms for each instrument results in a matrix of dimension $C \times C$ (C =number of instrument classes). We convert the histograms to probabilities by normalizing each row of the matrix to a sum of 1.

Note that unlike a traditional confusion matrix, this is not a symmetric matrix. We only focus on one row at a time in order to compare probabilities of observing false negatives for a given false positive instrument.

5. RESULTS AND DISCUSSION

A comparison of model performance is summarized in Figure 2 and Table 3. It can be observed that CNN and CRNN outperform MLP in both metrics. This is expected since the convolutional layers allow the model to learn hierarchical acoustic features from the time-frequency representation more efficiently. However, the CRNN does not outperform the CNN, which may be attributed to the fact that only 1 s second snippets are used. The temporal dimension of the input is reduced to only 5 time steps after the 4 CNN layers. The benefits of using recurrent layers are more noticeable when longer sequences are involved as in work by Choi et al. where they use inputs of length 29 s [6]. In addition, the receptive field of the deeper layers of the CNN is large enough for learning temporal features. Another observation is that output aggregation tends to improve models' label ranking performance and mean AUC.

Figure 3 shows the AUC per instrument of the CRNN model for the chosen time-resolution aggregation. We observe that using output aggregation in time leads to better performance in almost all instrument classes. The model achieves high AUC not only for majority instruments in the dataset but also for minority instruments such as flute, violin and cello, suggesting that it not simply predicting the majority. We also observe that the model does not seem to perform well for vocals in general. While it does achieve high AUC for male singers, the AUC for female singers and vocalists is low. We investigate this further using the

	MLP	CNN	CRNN
1 s	71.28	**77.55**	77.5
5 s	70.85	78.35	**78.76**
10 s	70.82	78.59	**79.22**
Track	71.1	**80.92**	80.1

Table 3. Mean AUC for various time-resolutions

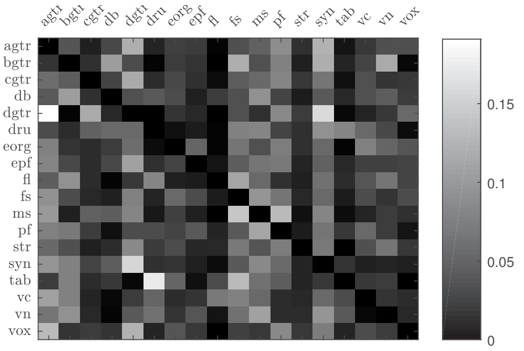

Figure 4. Distribution of false negative instruments conditioned on a true positive of a particular instrument

visualization method described in Sect. 4.4.

To construct the confusion visualization as described in Sect. 4.4, we pick the CRNN model and use the 1 s time-resolution outputs for the test set. Picking the threshold for binarizing the predictions is not straightforward. A fixed threshold of 0.5, for example, led to 0 detections for string section and electronic organ. Therefore, we adjusted the best thresholds for each class. These thresholds are determined by computing the class-wise f-measure at all score thresholds and selecting the threshold giving the best f-measure in the validation set. Figure 4 shows the constructed visualization. A high value in a row implies that the model may be confusing that particular pair of instruments more often than others. The following observations can be made from the figure:

- (*bgtr*, *db*): This confusion is possibly due to similar frequency range of the electric bass and double bass.

- (*dgtr*, *agtr*), (*dgtr*, *cgtr*), (*cgtr*, *dgtr*), and (*agtr*, *dgtr*): While confusion between acoustic and distorted guitar is unusual, the confusion between clean and distorted guitar is possibly explained by the variety in tone for both the clean and distorted guitars. A light crunch or low gain setting may possibly get misclassified. This could also explain the poor performance for clean electric guitar.

- (*dru*, *tab*) and (*tab*, *dru*): Both drum set and tabla are percussive instruments. In addition, one of the test tracks containing tabla has a ‘drum machine’ label which possibly causes drum false positives.

- (*dgtr*, *syn*) and (*syn*, *dgtr*): This is an interesting case since the variance in sound for both, the distorted guitar and synthesizers, is very large. Further investigation is needed to understand this case.

Next, we investigate the poor model performance on vocal classes. Figure 4 shows confusion between male and female singers implying that the model might be incorrectly classifying female singers as male. In order to investigate this phenomenon, the three vocal classes were combined for a follow-up experiment. We max-pool the predictions and the ground truth for these three classes, and recompute the AUC for this new ‘vocals’ class. Table 4 shows the average AUC of the three classes and the AUC of the combined ‘vocals’ class. The model performs significantly better for the combined class confirming our hypothesis that it confuses the vocal classes.

	1 s	5 s	10 s	Track
Average AUC (ms, fs, vox)	0.709	0.725	0.737	0.782
AUC vocals	0.822	0.96	0.975	0.998

Table 4. AUC for different time resolutions comparing pooled vocals against averaged AUC for vocal classes

Another interesting finding is that the best threshold chosen per instrument for binarization ranges from 0.02 to 0.55 with lower thresholds for minority instruments in general. We observe a correlation coefficient of 0.9 between the thresholds and the training data distribution suggesting that the model has learned biases in the dataset. The impact of this finding requires further experiments.

6. CONCLUSION

We presented a DNN-based IAD system trained using multi-track datasets to detect 18 instruments. The CRNN and CNN outperform MLP architectures for the task and perform well for detecting instruments common in popular music, such as drums, electric bass, acoustic guitars, distorted guitars and vocals. It also performs well for instruments in classical music such as flute, cello, violin even though they were under-represented in the dataset. We also stress the need for multiple metrics and visualizations for evaluation of systems such as IAD which is non-trivial to evaluate.

As future work, a few extensions and research directions are: (i) pre-training the network using monophonic stems from the multi-track datasets and subsequently training and testing for IAD, (ii) designing the convolutional network for the CRNN as proposed by Jordi et al. [23] instead of the currently used 3×3 filters as is common in computer vision, (iii) converting the proposed monolithic model architecture for IAD to a hierarchical architecture for instrument family classification first and subsequently instrument classification. While this paper treats the model as a black box and focuses on evaluation and analysis of model outputs, it is worth studying the model to understand the internal representations by means of visualization tools such as t-SNE and saliency maps [27] as performed by Han et al. [12].

By drawing attention to challenges in IAD with this paper, we hope to encourage the MIR community to explore this task. IAD is a rewarding avenue for research due to its real-world use cases as well as the potential to augment and improve performance in other tasks in MIR.

7. ACKNOWLEDGMENTS

This work was funded by Gracenote. We thank them for their generous support. We would also like to thank Nvidia for supporting us with a Titan Xp awarded as part of the GPU grant program.

8. REFERENCES

[1] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 155–160, 2014.

[2] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 559–564, 2012.

[3] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2015.

[4] Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, Tuomas Virtanen, Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(6):1291–1303, 2017.

[5] Keunwoo Choi, György Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. In *Proc. of the International Soceity of Music Information Retrieval Conference (ISMIR)*, pages 805–811, 2016.

[6] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 2392–2396, 2017.

[7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations (ICLR)*, 2015.

[8] Slim Essid, Gaël Richard, and Bertrand David. Musical instrument recognition on solo performances. In *Proc. of the 12th European Signal Processing Conference*, pages 1289–1292, 2004.

[9] Ferdinand Fuhrmann, Martín Haro, and Perfecto Herrera. Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 321–326, 2009.

[10] Masataka Goto. Rwc music database: Music genre database and musical instrument sound database. In *Proc. International Conference on Music Information Retrieval, 2003*, pages 229–230, 2003.

[11] Siddharth Gururani and Alexander Lerch. Mixing secrets: A multitrack dataset for instrument detection in polyphonic music. In *Late Breaking Demo (Extended Abstract), Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, 2017.

[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 Transactions on Audio, Speech and Language Processing (TASLP)*, 25(1):208–221, 2017.

[13] Yoonchang Han, Subin Lee, Juhan Nam, and Kyogu Lee. Sparse feature learning for instrument identification: Effects of sampling and pooling methods. *The Journal of the Acoustical Society of America*, 139(5):2290–2298, 2016.

[14] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 327–332, 2009.

[15] Perfecto Herrera-Boyer, Geoffroy Peeters, and Shlomo Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 448–456. PMLR, 2015.

[17] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno. Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. *EURASIP Journal on Applied Signal Processing*, 2007(1):155–155, 2007.

[18] Anssi Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 216–221, 2006.

[19] AG Krishna and Thippur V Sreenivas. Music instrument recognition: from isolated notes to solo phrases. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, volume 4, pages iv–iv, 2004.

[20] Jen-Yu Liu and Yi-Hsuan Yang. Event localization in music auto-tagging. In *Proc. of the 2016 ACM on Multimedia Conference*, pages 1048–1057. ACM, 2016.

[21] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *14th python in science conference*, pages 18–25, 2015.

[22] Brian CJ Moore. *An Introduction to the Psychology of Hearing*. Brill, 2012.

[23] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *Proc. of the 25th European Signal Processing Conference*, Kos island, Greece, 28/08/2017 2017. IEEE.

[24] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[25] Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 6979–6983, 2014.

[26] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(5):927–939, 2016.

[27] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations (ICLR)*, 2013.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[29] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2643–2651, 2013.

[30] Richard Vogl, Matthias Dorfer, and Peter Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 201–205, 2017.

[31] Li-Fan Yu, Li Su, and Yi-Hsuan Yang. Sparse cepstral codes and power scale for instrument identification. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 7460–7464, 2014.