IMPROVED AUTOMATIC INSTRUMENTATION ROLE CLASSIFICATION AND LOOP ACTIVATION TRANSCRIPTION

Jake Drysdale * *and Jason Hockman*

Sound and Music Analysis Group, Digital Media Technology Lab
School of Computing and Digital Technology
Birmingham City University
Birmingham, UK
jake.drysdale@bcu.ac.uk

António Ramires * † *and Xavier Serra*

Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
antonio.ramires@upf.edu

ABSTRACT

Many electronic music (EM) genres are composed through the activation of short audio recordings of instruments designed for seamless repetition—or loops. In this work, loops of key structural groups such as bass, percussive or melodic elements are labelled by the role they occupy in a piece of music through the task of automatic instrumentation role classification (AIRC). Such labels assist EM producers in the identification of compatible loops in large unstructured audio databases. While human annotation is often laborious, automatic classification allows for fast and scalable generation of these labels. We experiment with several deep-learning architectures and propose a data augmentation method for improving multi-label representation to balance classes within the Freesound Loop Dataset. To improve the classification accuracy of the architectures, we also evaluate different pooling operations. Results indicate that in combination with the data augmentation and pooling strategies, the proposed system achieves state-of-the-art performance for AIRC. Additionally, we demonstrate how our proposed AIRC method is useful for analysing the structure of EM compositions through loop activation transcription.

1. INTRODUCTION

Affordable music production technologies (e.g., digital audio workstations) for incorporating and manipulating samples have democratised the EM creation process, allowing users with varying levels of musical knowledge to experiment in the creation of EM. A large majority of popular music is composed in this manner, inheriting some characteristics of EM, such as the use of samples, sequence-driven composition and a fixed tempo throughout the piece.

For sampled content, EM producers often rely on well-known sample libraries (e.g., Splice), which consist primarily of individual sounds and loops—short audio recordings (one, two or four bars in length) of instruments designed for seamless repetition [1]. Loops may be created by sequencing individual sounds or sampling a short musical phrase from a solo or polyphonic instrumental recording. These loops serve as the material from which music makers can generate EM compositions through various editing and combinatory processes (e.g., layering, splicing, rearranging). Figure 1 depicts a simplified representation of the EM creation process involving the layering and repeated activation of loops with different roles.

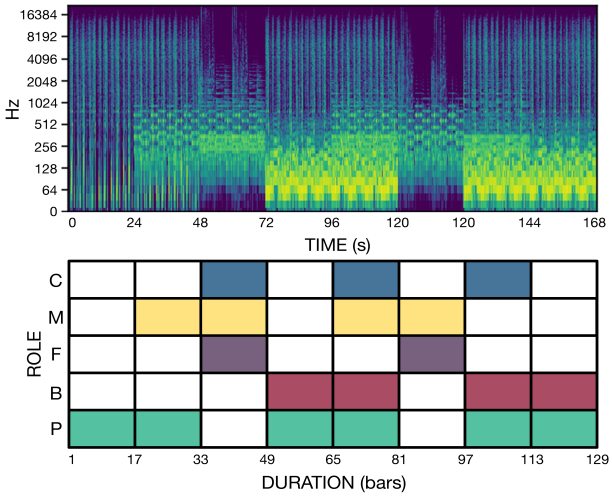

Figure 1: A simplified EM composition structure, built with five loop layers. Log-scaled STFT power spectrogram (*top*) and corresponding role activations: Chords (C), Melody (M), Sound Fx (F), Bass (B), and Percussion (P) at 4-bar intervals (*bottom*).

In this paper, we propose a system for automatic instrumentation role classification (AIRC) to label a loop by its specific function within a EM composition (e.g., drums, chords, melody, bass, sound Fx). System performance is measured through evaluation with state-of-the-art audio classification models and a data augmentation procedure that utilises common production techniques used in commercial music recordings. By estimating instrumentation roles for fixed-length segments of an EM composition, it is possible to retrieve an informative map of musical structure.

1.1. Related Work

In recent years, there has been an increased focus on research related to audio loops within the field of music information retrieval. There are several methods that exist for automated loop retrieval [2, 3, 4, 5, 6], and loop creation [7, 8, 9].

In addition, there are two recent methods proposed for loop activation transcription, a task that involves estimating the locations

* Both authors have contributed equally

† This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Nº 765068, MIP-Frontiers.

in which loops occur throughout a piece of music. López-Serrano et al. [10] proposed a method for decomposing loop-based EM using non-negative matrix factorization deconvolution (NMFD) [11] to estimate spectral templates and rhythmic activations from magnitude spectrograms. Following this work, Smith et al. [12] propose an alternative method to discovering loop activations of EM using non-negative tensor factorization (NTF) [13]. While non-negative matrix factorization approaches allow for separation of mixed audio into the constituent loops of a music composition, they rely on non-varying repetitions of loops and do not optimise independence between learned loop representations.

As an alternative to the aforementioned approaches, which seek to identify the instrument within a loop and its associated activation, music auto-tagging is a multi-label classification task that may be used to denote presence of an attribute or instrument. Previous approaches have applied standard Convolutional Neural Networks (CNNs) to this task [14, 15], while recent work has focused on the application of musical knowledge for pitch and loudness invariance through musically-motivated filter shapes [16]. Won et al. [17] achieved state-of-the-art accuracy for auto-tagging by using data-driven Harmonic filters, a harmonically-stacked trainable representation to preserve time-frequency locality in convolution layers.

AIRC is a music auto-tagging task that estimates the presence of active instrumentation role groups (e.g., percussion, bass, melody, chords, sound Fx) within audio recordings. Research in AIRC has been facilitated by the development of the Freesound Loop Dataset (FSLD), a large public collection of loops and corresponding instrumentation role annotations from Freesound.[1] Ching et al. [18] benchmark AIRC performance of neural network and non-neural network models on the non-sequenced loops of FSLD, and achieve the current state-of-the-art performance using a Harmonic CNN [17].

1.2. Motivation

As EM production (i.e., creation, selection, manipulation) is guided heavily by aesthetic preferences, producers often select sounds based on their function within loops. With a wide range of traditional and synthesized timbres from which to select, instruments are often utilised outside traditional roles. We thus follow the AIRC problem formalisation as represented in Ching et al. [18], which associates instrumental roles with short loops. We expand on this approach by applying AIRC to full EM compositions, in which multiple instrumentation roles (e.g., percussion, melody, bass playing together) are often active. [18] observed that accuracy and bias were reduced by the overuse of single labels, due to limited coverage of multi-label annotations in the FSLD. To mitigate this imbalance, we introduce a novel data augmentation technique to balance classes and experiment with several deep-learning architectures and pooling operations, resulting in a state-of-the-art AIRC system.

We then demonstrate the usefulness of AIRC in EM structural analysis by comparing our system with previous approaches for loop activation estimation. Additionally, the proposed AIRC system is shown to derive key structural information from full-length EM compositions in the form of instrumentation role activation maps, which would be of use in tasks such automatic DJing [19], mashups [20], and loop creation [4]. Finally, we show that the system is capable of identifying percussion only passages and then compare it against the previous state-of-the-art for breakbeat identification. For reproducibility, we provide open-source code for the proposed data augmentation method and AIRC system.

The remainder of this paper is structured as follows: Section 2, presents the proposed method for AIRC and loop activation transcription. Evaluation methodology and the datasets used in this study are detailed in Section 3 and the results and discussion are provided Section 4. Conclusions and suggestions for future work are presented in Section 5.

2. METHODOLOGY

In this study, several CNN architectures are evaluated to identify the best system for AIRC. Each architecture utilises different configurations of front-end filter shapes to learn a representation from spectrograms and pooling operations that derive the final predictions by summarising the information learned by the network.

As the data employed in AIRC contains different types of musical audio, from tonal melodies to noise-like sound Fx, this motivates the experimentation of architectures aimed at different sound classification tasks. Three front-end filter shapes are used: general domain square filters, vertical filters [16] tailored towards classifying the timbre of melodic instruments, and previous state-of-the-art for AIRC—harmonic band-passfilters which capture harmonic characteristics.

To improve the AIRC predictions, two methods for summarizing the information learned in the final convolutional layers of a CNN are investigated. The standard approach to is to use global max-pooling (GMP); however, this infers strict assumptions about the label characteristics of the data. In the closely related field of sound event detection, auto-pooling has been proposed to automatically learn the best suited operation by interpolating between max-, mean-, or min-pooling during training. We implement both GMP and auto-pooling and compare their performance for the task of AIRC.

2.1. Implementation

Audio is input into the networks as a spectrogram representation, from which features are extracted through convolutional layers. Output predictions return values between $[0., 1.]$ depicting the presence of active instrumentation roles.

For each network, the input layer is a four-dimensional tensor $t \in \mathbb{R}^{b \times w \times h \times c}$, with batch size b, number of frames w, number of frequency bins h, and channels c. Following [16], each model uses L2-norm regularization of filter weights to encourage loudness invariance with the exception to the harmonic CNN-based models, which use a weight decay of $1e$–4 [17].

2.1.1. Vertical filter network

The vertical filter network (`VF-CNN`) is based on the *multi-layer* architecture in [16] for musical instrument recognition. Figure 2 provides an overview of the `VF-CNN` configuration. The input spectrogram is set to be of size 500×128 to accommodate for longer observations of audio loops (see Section 2.2). The front-end utilizes several vertical convolution filter sizes (black rectangles in Figure 2) to efficiently model timbral characteristics present in the spectrogram. Custom filter sizes are used to capture both wide (e.g., bass, chords) and shallow spectral shapes (e.g, percussion).

[1] `https://freesound.org/`

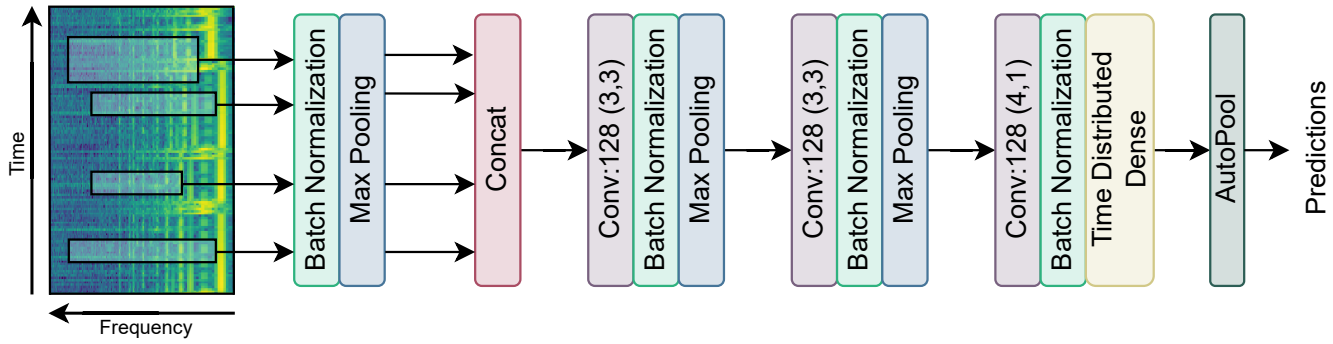

Figure 2: Block diagram showing the configuration of the vertical filter network with auto-pooling.

The numbers and sizes of filters used in the front-end are as follows: 128 filters of sizes 5×1 and 80×1; 64 filters of sizes 5×3 and 80×3; and 32 filters of sizes 5×5 and 80×5.

All convolutions in the front-end use *same* padding, and max-pooling is applied to obtain a 16×16 summary of each feature map. This is followed by two 2-D convolutional layers with batch normalisation [21] and exponential linear unit (ELU) [22] activation functions. The first 2-D convolutional layer is followed by strided $(2, 2)$ max-pooling. After the final 2-D convolutional layer, we experiment with two pooling operations to summarise the information learned by previous layers prior to predictions (see Subsection 2.1.4).

2.1.2. *Square filter network*

The square filter network (`SF-CNN`) contains four 2-D convolutional layers with 128 small-rectangular filters of size 3×3 and *same* padding. After each convolutional layer, batch normalization is applied with an ELU [22] activation function. Each convolutional layer is followed by strided $(2, 2)$ max-pooling, with the exception of the final convolutional layer, which also uses one of the two summarization pooling operations described in Section 2.1.4.

2.1.3. *Harmonic CNN*

In [18], AIRC was approached using a CNN with a data-driven harmonic filter-based front-end (`H-CNN`) [17]. We re-implement this architecture and use it as a baseline to test our proposed models. The input t is passed through a set of triangular band-pass filters to obtain a tensor representing it as six harmonics. Harmonic structure is captured by treating the harmonics as channels and processed by a 2-D CNN. The CNN consists of seven convolution layers and a fully connected layer. All but the final convolutional layer is followed by 2×2 max-pooling, batch normalization and a ReLU activation function. Global max-pooling is applied to the final convolutional layer. The output layer is a 5-way fully-connected layer with, a sigmoid activation function and a 50 % dropout.

2.1.4. *Summarization Pooling*

We consider two pooling operations for summarizing the information learned in the final convolutional layers: auto-pooling and standard global max-pooling.

Auto-pool is a trainable pooling operator capable of adapting to data characteristics by interpolating between min-, max-, or average-pooling [23]. For the configurations that use auto-pooling, the final convolutional layer uses as kernel size (4,1). This is followed by batch normalisation and a time-distributed dense layer with a sigmoid activation function and r output nodes, where r is equal to the number of classes. The output of the time-distributed dense layer is fed to the auto-pooling operation, which produces the final predictions.

For configurations that use GMP, the final convolutional layer is summarised with global max-pooling and then fed to a fully-connected output layer consisting of r output nodes, sigmoid activation functions and a 50% dropout.

2.1.5. *Loss function*

The loss function used for updating the parameters of each model is binary cross-entropy (BCE). BCE can be calculated as:

$$\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\, p(y_i)) + (1-y_i) \cdot \log\,(1-p(y_i)), \quad (1)$$

where *N* represents the number of examples in the training set and $p(y_i)$ s the predicted probability of the i^{th} example.

2.2. Network Training

Input audio is pre-processed through resampling and conversion to a spectrogram representation. Audio loops are resampled to 16kHz and the short-time Fourier transform (STFT) of each loop is calculated using a window size of 512 samples and a hop size of 256 samples. For the `H-CNN`, magnitudes of STFT are provided as input to the model. For the `SF-CNN` and `VF-CNN`, the inputs are log-scaled Mel spectrograms with 128 Mel-frequency bands.

All models are trained using the Adam optimiser [24] with a learning rate $1e$–4, where each iteration takes a mini-batch of 8 examples. All weights are initialized using He's constant [25] to promote equalized learning. Early stopping was used to complete the training once the model performance ceases to improve over 15 epochs. The epoch that achieves the best accuracy on the validation set is used for testing.

2.3. Loop Activation Transcription

Loop activation transcription involves predicting the loop activations of instrumentation roles as they occur over time. Taking ad-

vantage of the grid-based structure and consequently fixed tempo of loop-based EM, we are able to use the proposed AIRC system to analyse the loop structure of a given composition. The AIRC system enforces separation between roles by design and does not rely on loops being an exact repetition of themselves, thus making it robust to variation such as automation and resequencing.

Instrumental role predictions for full EM compositions are obtained by passing an audio file into the AIRC system in 4-bar segments and assessing output activations. By segmenting a full-length EM composition into 4-bar loops, on which we then perform AIRC, instrumentation role activations may be derived for each loop, resulting in a form of EM transcription.

3. EVALUATION

The AIRC model presented in Section 2 is assessed through two evaluations to determine: 1) AIRC performance using the various configurations and augmented version of the FSLD, and 2) performance for loop activation transcription.

3.1. Automatic Instrumentation Role Classification

3.1.1. Evaluation Methodology

The architectures (i.e., `VF-CNN`, `SF-CNN`, and `H-CNN`) and pooling strategies (i.e., auto-pooling and GMP) presented in Section 2 are evaluated in order to determine the optimal configuration for AIRC. Following [18, 26], we use two sets of performance measurements: Area under receiver operating characteristic curve (ROC-AUC) and area under precision-recall curve (PR-AUC). The metrics were calculated on the test set for each of the models under evaluation.

In [18], the authors also calculate the F1 score; however, we omit this evaluation metric as it depends on a decision threshold applied to the per-class output scores; whereas, ROC-AUC and PR-AUC measure model performance globally, integrating all possible thresholds.

3.1.2. Augmented Freesound Loop Dataset

To train and evaluate the different models we use the Freesound Loop Dataset (FSLD) [27], comprising of various loops uploaded to Freesound [28] under Creative Commons licensing. Of the various annotations present within the FSLD, we use tempo, key and loop instrumentation roles. The most important of which is the instrumentation role—a multi-label annotation for which the possible roles are: percussion, bass, chords, melody, sound Fx and vocals.

The original FSLD contains 2936 loops, of which 1531 have only one instrumentation role and 1405 have more than one. As can be seen from the class distribution in Table 1, the classes in this dataset are heavily imbalanced.

Percussion	54.95	Fx	24.80
Bass	19.10	Melody	21.31
Chords	11.90	Vocal	2.29

Table 1: Distribution (%) of instrumentation roles in FSLD.

In order to adapt this dataset to our task, we apply modifications to the data. We first remove all vocal loops as they do not provide sufficient training and testing material. All remaining loops are time-stretched to 120 beats per minute (BPM). Longer loops are cropped to a length of 4 bars (i.e., 8 secs), while loops shorter than 4 bars are cropped to either 1 or 2 bars and repeated to a length of 4 bars. We separate loops which have multiple instrumentation roles from those which only have one, and randomly select 70% of each for training and 30% for validation and testing. From the latter split, 60% are used for testing and 40% for validation.

Besides using the previously described training set of the FSLD, we applied a data augmentation procedure to handle the main imbalance issues on the dataset. These are 1) the lesser presence of loops with more than one instrumentation role (i.e., multi-label) compared to the ones with just one role (i.e., single-label) and; 2) the number of loops for each instrumentation role class, shown in Table 2.

The data augmentation procedure utilises common production techniques that are used in commercial music recordings including key matching, tempo matching and the use of audio Fx such as distortion, reverb and chorus.

Percussion	929	Fx	222
Bass	92	Melody	174
Chords	102		

Table 2: Distribution of the loops with only one instrumentation role in FSLD.

To balance the number of loops per class, we use an augmentation methodology similar to the one proposed in Ramires et al. [29]. The loops are processed through several effects, including delay, bitcrusher, chorus, flanger, reverb, tube saturation and pitch-shifting, resulting in 1000 loops for each of the r classes under observation ($r = 5$), totalling 5000 loops.

We create multi-role data by overlapping loops from each augmented single label class such that all single and combined classes contain the same number of loops. We start by calculating the possible combinations $\binom{r}{k}$, where k is the number of instrumentation roles in the combination ($2 \leq k \leq 5$). To balance the dataset in both the number of loops per instrumentation role and in k, the number of augmented loops (5000) is divided by $\binom{r}{k}$ to obtain the number of loops required for each combination (e.g., for k=2, $5000/\binom{r}{k} = 500$ for each combination of roles). The final loops are then created by harmonically combining the single instrumental role loops. We combine only loops with compatible modes (e.g., Major with Major), and pitch shift the selected loops to their average key.

Discarding the original multi-label loops of the training set, this process results in a total of 25000 loops that can can used for training. In order to evaluate the effect of this augmentation procedure (Aug), we compare the accuracy of the models trained with those trained with the original dataset (FSLD) on the same test and validation data.

Percussion	27.59	Fx	23.17
Bass	20.33	Melody	18.15
Chords	10.77		

Table 3: Distribution (%) of instrumentation roles in the test set.

Model	Dataset	Pooling	Param.	PR-AUC	ROC-AUC	Bass	Fx	Perc.	Chords	Melody
H-CNN	Aug	GMP	3619986	59.18	77.34	40.30	57.05	94.60	47.92	56.01
H-CNN	Pure	GMP	3619986	61.83	80.39	53.65	42.21	94.10	60.30	58.89
VF-CNN	Aug	GMP	1098869	65.60	80.99	47.11	64.92	97.62	63.11	55.22
VF-CNN	Pure	Auto	1102394	66.98	82.52	57.59	66.43	95.75	50.37	64.77
VF-CNN	Aug	Auto	1102394	67.47	81.40	46.18	67.10	97.05	56.13	**70.89**
SF-CNN	Aug	Auto	**313674**	68.15	82.19	55.12	68.30	**98.03**	58.81	60.49
SF-CNN	Aug	GMP	445701	68.40	81.59	**62.21**	59.80	95.93	62.39	61.68
SF-CNN	Pure	GMP	445701	68.74	83.83	58.72	63.11	95.97	64.74	61.14
VF-CNN	Pure	GMP	1098869	70.62	**85.72**	53.83	**71.73**	97.84	64.90	64.78
SF-CNN	Pure	Auto	**313674**	**71.28**	85.12	57.76	59.18	95.98	**73.20**	70.30

Table 4: AIRC performance (%) and model size for each configuration, where bold indicates highest scores.

3.2. Loop Activation Transcription

3.2.1. Evaluation Methodology

To investigate the capacity of the AIRC system for transcribing loop activations in EM compositions, we compare all the AIRC configurations (Section 2). The best performing configurations are then compared with the results of the previous approach to loop activation transcription by Smith et al. [12].

As in [30, 12], we evaluate the loop activation predictions against a ground truth in terms of accuracy. As accuracy expects a binarised transcription, we use a repeated k-fold cross validation together with a grid search to identify the best threshold for binarising the predictions of each role. In order to investigate the generalization of the proposed models, we use 2-fold cross validation repeated 10 times, where one fold is used as a validation set to identify thresholds and the other is reserved for computing accuracy against the ground truth. Thresholds for each class are identified by performing a grid search over a range between 0.01 and 1 with a step size of 0.01, then selecting the thresholds which provide highest accuracy on the validation set.

In [12], approaches which require the downbeat tracking are considered *guided*. As our proposed approach requires BPM annotations for time-stretching, we only compare our models with the *guided* algorithms.

3.2.2. Dataset

We apply our proposed models to the dataset used in [30, 12]. The dataset consists of simplified EM compositions built by generating templates similar to the ones in Figure 1 with 4-bar loops. We refer to this as the *Artificial* dataset for the reason that the the loops are repeated without variation, which would usually be achieved in professional music through DAW techniques, such as automation and resequencing.

The automatic arrangement method provided in [12] is used to build 21 music compositions with seven genres and three templates–*composed*, *factorial* and *shuffled factorial*. For the *composed* template, loops are introduced and removed in an iterative manner. The *factorial* template contains all possible combinations of loops, arranged iteratively. The *shuffled factorial* template contains the same loop combinations, with shuffled ordering. *Factorial* and *shuffled factorial* datasets are useful for seeing how the models perform on all of the loop combination possibilities for the *Artificial* dataset, whereas *composed* layout is more representative of typical EM compositions in regards to the way that loops are iteratively introduced and removed throughout the composition.

Following the AIRC procedure, compositions are time-stretched from their annotated tempo to 120BPM and divided into 4-bar loops, which are provided as input to the AIRC systems.

4. RESULTS & DISCUSSION

4.1. Automatic Instrumentation Role Classification

The models are evaluated using use the marco-average (MA) of the PR-AUC and of the ROC-AUC as a global metric. For individual instrumentation roles, we only show the PR-AUC. Due to the imbalance of the FSLD, which also affects the test set (Table 3), MA is used to provide an average accuracy over each class.

Table 4 presents the results of our AIRC experiment for the models discussed in Section 2, in which each model is presented in ascending order of their average PR-AUC. The ROC-AUC performance measure is consistently higher than PR-AUC; however, this metric can lead to over-optimistic scores when the dataset is unbalanced [31].

The best performing models *w.r.t* PR-AUC, are the SF-CNN with auto-pooling (71.28%) followed by the VF-CNN with GMP (70.61%). Both models surpass the current state-of-the-art, H-CNN trained on FSLD (61.82%) by a substantial margin. The SF-CNN mostly performs better than its VF-CNN counterpart. Vertical filters have been demonstrated to produce comparatively better results with tonal musical audio [32]; however, the results of our evaluations suggest that square filters generalise better to the non-standard types of audio associated with EM.

The overall best performing model in terms of PR-AUC is the SF-CNN with auto-pooling trained on the Pure dataset. However, by closely inspecting the results achieved for individual instrumentation roles, it can be seen that it surpasses by almost 10% the PR-AUC achieved by other models in the *Chords* class, while not achieving such a high result in *Bass*, *Fx* and *Percussion*.

The highest performing instrumentation role for all models is *Percussion*, which was expected due to this role having the largest number of examples in the FSLD dataset. The roles that generally perform worst are *Bass* and *Chords*, which have the smallest number of examples in the FSLD. The performance of *Bass* has a considerable increase when using a combination of the SF-CNN with GMP and augmented data. Additionally, *Chords* performs significantly better when using the SF-CNN and auto-pooling configuration trained with the Pure dataset.

The best three performing models in terms of PR-AUC are trained on the Pure dataset, followed by the Augmented one. However, it can be seen that the *Bass*, *Percussion* and *Melody* roles tend

to benefit from training with the Augmented dataset. As the configurations perform better for different classes, it is possible to use a combination of the models for classifying individual instrumentation roles. This combination would lead to an average PR-AUC of 75,213%, substantially surpassing each model.

4.2. Loop Activation Transcription

Table 5 presents the loop activation transcription results using the AIRC configurations (Section 2) to transcribe the compositions in the *Artifical* dataset. Each model is presented in ascending order of their mean classification accuracy over the instrumentation roles. Additionally, Table 5 provides the classification accuracy for each individual role (*Bass*, *Drums*, *Fx* and *Melody*).

Model	Data	Pooling	Mean	Bass	Drums	Fx	Melody
H-CNN	Pure	GMP	75.1	71.8	96.1	55.6	76.7
H-CNN	Aug	GMP	79.5	53.1	95.8	81.6	87.6
VF-CNN	Pure	Auto	80.2	63.7	98.6	63.1	**95.3**
VF-CNN	Pure	GMP	80.9	69.0	99.3	65.8	89.4
SF-CNN	Pure	Auto	81.0	66.9	97.3	71.6	88.4
SF-CNN	Pure	GMP	81.8	69.2	**100.0**	63.4	94.6
VF-CNN	Aug	Auto	82.5	**74.2**	99.7	79.7	76.6
VF-CNN	Aug	GMP	84.7	71.7	**100.0**	75.7	91.4
SF-CNN	Aug	GMP	86.2	71.7	**100.0**	79.6	93.2
SF-CNN	Aug	Auto	**86.9**	68.3	**100.0**	**85.7**	93.4

Table 5: Loop activation transcription accuracy (%) results for AIRC configurations, where bold indicates highest scores.

The overall best performing model uses the SF-CNN with auto-pooling configuration trained using the augmented dataset (86.9%) followed by the SF-CNN with GMP (86.2%). For this task, models trained with the augmented dataset generally appear to outperform those trained with Pure dataset, which could be due to the fact that the augmentation process ensures there is a balanced distribution of all possible role combinations and it is common in the compositions for several roles to be active in a single loop. Drums are classified most accurately for all model configurations with four models achieving 100% accuracy. This is expected as *percussion* has the largest number of samples in the FSLD dataset, and is usually the most prominent element in EM compositions. In some cases, the VF-CNN configuration seems to improve performance of *Melody* and *Bass* roles, which could suggest that the classification of roles containing melodic instruments benefit from using vertical filters at the front end of the system.

Figure 3 presents loop activation transcription results for the three template variations using our two best performing AIRC configurations (i.e., SF-CNN-AUTO and SF-CNN-GMP) compared with the NTF [12] and NMFD [10] methods previously proposed for this task.

On a glance, we can see our architectures out perform the previous methods in regards to accuracy for the *composed* layout, with SF-CNN-GMP (red) achieving the highest score. NTF (blue) achieves the best performance for the *factorial* layouts closely followed by our SF-CNN-AUTO architecture. Furthermore, the AIRC system has a considerably faster runtime than NTF (∼30 secs per composition) and NMFD (∼10 mins per composition). Predictions for a full EM composition are calculated in under a second using AIRC, which could be beneficial when analysing large collections of music in DJ software. As mentioned in [12], an additional shortcoming of the NTF and NMFD approaches is that the algorithms depend on loop roles not co-occurring throughout the composition. The proposed AIRC approach enforces independence between the different roles, thus making it more suitable for transcribing loop activations of real-world EM compositions, in which loops often vary through automation and resequencing.

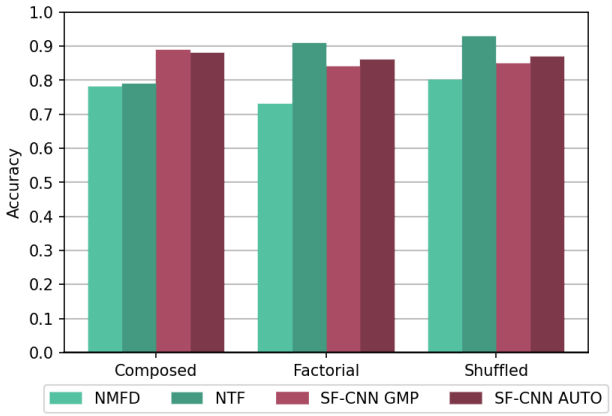

Figure 3: Loop activation transcription accuracy scores.

4.3. Real-world Scenario

Our approach to loop activation transcription with AIRC can be applied to full-length, professionally produced EM, which has not been explored in previous literature.

An *instrumentation role activation map* (IRAM) of the EM composition *Joyspark* (2020) by Om Unit[2] using the proposed method for loop-based EM structure analysis (Section 3.2) is presented in Figure 4. For visualisation and comparison, we show a log-scaled STFT power spectrum of the EM composition above the IRAM. The IRAM allows us to visualise activations for each role over the duration of the EM composition, where each square is a measurement of four bars. Furthermore, we can see how each role develops throughout the EM composition. For example, the melody role activations progressively increase between bars 1–41, which corresponds with a synthesizer arpeggio that is gradually introduced by automating the cut-off frequency of a low-pass filter. Additionally, the chord role activations increase between bars 1–49 in correlation with the chords in this section that gradually increase in volume. Activations for the percussion role also correlate well with the composition as can be seen between bars 49–81 and 97–129—the only sections that contain percussion. Finally, the key structural sections of the composition are easily identifiable. For example, the introduction to the composition (bars 1–49) begins relatively sparse in the composition and IRAM; whereas, bars 49–81 and 97–129 are quite clearly the *core* of the piece—that is, the most energetic sections of the composition typically established by the *drop* [33].

Additionally, the transcription enabled by our system could help EM producers identify sections of music that contain specific roles. For example, this would be useful for finding breakbeats (i.e., percussion-only passages) in digital music recordings [3].

[2] https://omunit.bandcamp.com

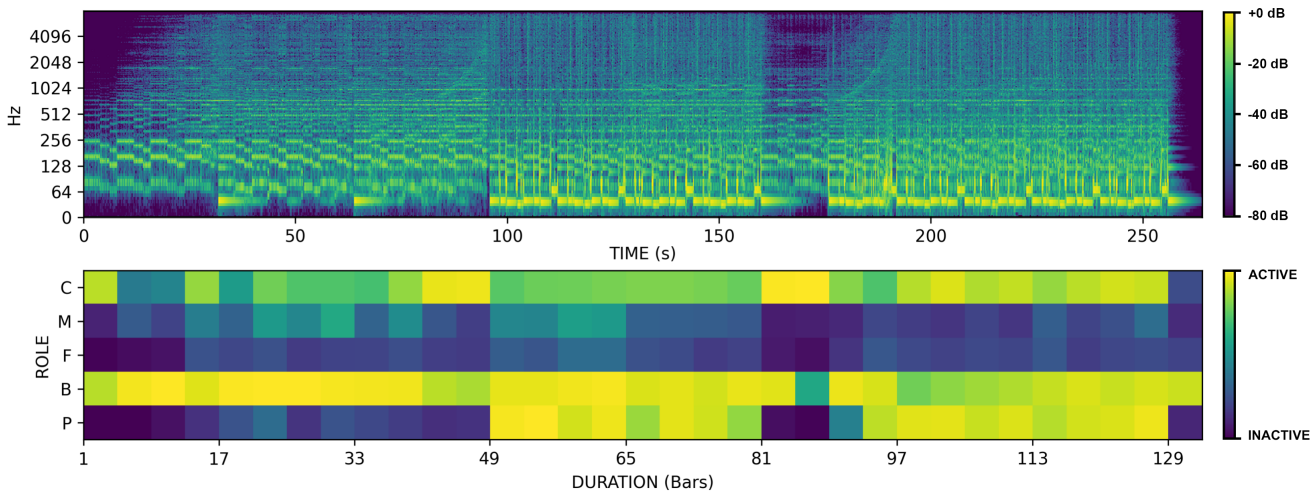

Figure 4: Estimated loop activation structure of *Joyspark* (2020) by Om Unit using our proposed model. Log-scaled STFT power spectrogram of the EM composition (*top*) and estimated templates corresponding to the loop activations showing predictions for each class: Chords (`C`), Melody (`M`), Sound Fx (`F`), Bass (`B`), and Percussion (`P`) at 4-bar intervals (*bottom*).

5. CONCLUSIONS

In this study, we have introduced a system for automatic instrument role classification of loops that utilises a novel data augmentation method and CNN-based architecture with auto-pooling. The evaluation results show that we outperform previous state-of-the-art performance in AIRC, allowing for a more reliable transcription of loops in unstructured collections of audio. Furthermore, we have introduced a deep learning approach for estimating the structure of loop-based electronic music and compared it with previous loop activation detection methods. Our approach achieves comparable results while achieving a considerably faster computation time.

The IRAM derived from our system has many potential use cases in music production and performance. MIR tasks that rely on structural information could benefit from this transcription (e.g., automatic DJing [19], music mashups [20]). The IRAM could be used as a visual aid for DJs to anticipate upcoming sounds (e.g., drums, bass) or to help to identify key structural events in EM [33].

A possible direction for future research in this area would be to train the system using a smaller timescale (e.g., 1-bar measures) to achieve higher resolution transcription of instrumentation role activations. Additionally, as no annotations for ground-truth instrumentation roles exist for real-world EM compositions, future work could involve annotating a corpus of these recordings for the evaluation of this task.

6. ACKNOWLEDGMENTS

The authors would like to kindly thank Patricio López-Serrano and Jordan B. L. Smith for the fruitful discussions and access to the loop activation transcription datasets and Eduardo Foncesca for the guidance on the implementation of auto-pooling.

7. REFERENCES

[1] Glenn Stillar, "Loops as genre resources," *Folia Linguistica*, vol. 39, no. 1-2, pp. 197 – 212, 2005.

[2] Olivier Gillet and Gaël Richard, "Drum loops retrieval from spoken queries," *Journal of Intelligent Information Systems*, vol. 24, no. 2, pp. 159–177, 2005.

[3] Patricio López-Serrano, Christian Dittmar, and Meinard Müller, "Finding drum breaks in digital music recordings," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, 2017, pp. 111–122.

[4] Zhengshan Shi and Gautham J Mysore, "Loopmaker: Automatic creation of music loops from pre-recorded music," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–6.

[5] Jordan BL Smith, Yuta Kawasaki, and Masataka Goto, "Unmixer: An interface for extracting and remixing loops.," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 824–831.

[6] Bo-Yu Chen, Jordan BL Smith, and Yi-Hsuan Yang, "Neural loop combiner: Neural network models for assessing the compatibility of loops," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020.

[7] Diogo Cocharro, George Sioros, Marcelo F. Caetano, and Matthew E. P. Davies, "Real-time manipulation of syncopation in audio loops," in *Music Technology meets Philosophy - From Digital Echos to Virtual Ethos: Joint Proceedings of the 40th International Computer Music Conference and the 11th Sound and Music Computing Conference*, 2014.

[8] Guillaume Alain, Maxime Chevalier-Boisvert, Frederic Osterrath, and Remi Piche-Taillefer, "Deepdrummer : Generating drum loops using deep learning and a human in the loop," in *Proceedings of The 2020 Joint Conference on AI Music Creativity*, 2020, pp. 81–91.

[9] Pritish Chandna, Antonio Ramires, Xavier Serra, and Emilia Gómez, "Loopnet: Musical loop synthesis conditioned on intuitive musical parameters," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings*, 2021.

[10] Patricio López-Serrano, Christian Dittmar, Jonathan Driedger, and Meinard Müller, "Towards modeling and decomposing loop-based electronic music.," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 502–508.

[11] Paris Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the International Conference on Independent Component Analysis and Signal Separation*, 2004, pp. 494–499.

[12] Jordan BL Smith and Masataka Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings*, 2018, pp. 171–175.

[13] D. FitzGerald, M. Cranitch, and E. Coyle, "Sound source separation using shifted non-negative tensor factorisation," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[14] Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[15] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings*, 2014, pp. 6964–6968.

[16] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *Proceedings of the 25th European Signal Processing Conference*. IEEE, 2017, pp. 2744–2748.

[17] Minz Won, Sanghyuk Chun, Oriol Nieto, and X. Serra, "Data-driven harmonic filters for audio representation learning," *2020 IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings*, pp. 536–540, 2020.

[18] Joann Ching, António Ramires, and Y. Yang, "Instrument role classification: Auto-tagging for loop based music," in *Proceedings of The 2020 Joint Conference on AI Music Creativity*, 2020, pp. 196–202.

[19] Len Vande Veire and Tijl De Bie, "From raw audio to a seamless mix: creating an automated dj system for drum and bass," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, pp. 1–21, 2018.

[20] Matthew EP Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto, "Automashupper: An automatic multi-song mashup system.," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[21] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[22] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *4th International Conference on Learning Representations*, 2016.

[23] Brian McFee, Justin Salamon, and Juan Pablo Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[26] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra, "End-to-end learning for music audio tagging at scale," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[27] Antonio Ramires, Frederic Font, Dmitry Bogdanov, Jordan B. L. Smith, Yi-Hsuan Yang, Joann Ching, Bo-Yu Chen, Yueh-Kao Wu, Hsu Wei-Han, and Xavier Serra, "The freesound loop dataset and annotation tool," in *Proceedings of the 21st International Society for Music Information Retrieval*, 2020.

[28] Frederic Font, Gerard Roma, and Xavier Serra, "Freesound technical demo," in *ACM International Conference on Multimedia*, 2013, pp. 411–412.

[29] António Ramires and Xavier Serra, "Data augmentation for instrument classification robust to audio effects," in *Proceedings of the International Conference on Digital Audio Effects*, 2019.

[30] Patricio López-Serrano, Christian Dittmar, Jonathan Driedger, and Meinard Müller, "Towards modeling and decomposing loop-based electronic music," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[31] Jesse Davis and Mark Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 233–240.

[32] Jordi Pons, *Deep neural networks for music and audio tagging*, Ph.D. thesis, Universitat Pompeu Fabra, 2019.

[33] Karthik Yadati, Martha A Larson, Cynthia CS Liem, and Alan Hanjalic, "Detecting drops in electronic dance music: Content based approaches to a socially significant music event.," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 143–148.