On the Robustness of Deep Convolutional Neural Networks for Music Classification

Keunwoo Choi, György Fazekas, *Member, IEEE,* Kyunghyun Cho, and Mark Sandler, *Fellow, IEEE*

***Abstract*—Deep neural networks (DNN) have been successfully applied for music classification including music tagging. However, there are several open questions regarding generalisation and best practices in the choice of network architectures, hyper-parameters and input representations. In this article, we investigate specific aspects of neural networks to deepen our understanding of their properties. We analyse and (re-)validate a large music tagging dataset to investigate the reliability of training and evaluation. We perform comprehensive experiments involving audio preprocessing using different time-frequency representations, logarithmic magnitude compression, frequency weighting and scaling. Using a trained network, we compute label vector similarities which is compared to groundtruth similarity.**

The results highlight several import aspects of music tagging and neural networks. We show that networks can be effective despite of relatively large error rates in groundtruth datasets. We subsequently show that many commonly used input preprocessing techniques are redundant except magnitude compression. Lastly, the analysis of our trained network provides valuable insight into the relationships between music tags. These results highlight the benefit of using data-driven methods to address automatic music tagging.

***Index Terms*—Music tagging, convolutional neural networks**

I. Introduction

MUSIC tags are descriptive keywords that convey various types of high-level information about recordings such as mood (*sad, angry, happy*), genre (*jazz, classical*) and instrumentation (*guitar, strings, vocal, instrumental*) [1]. Tags may be associated with music in the context of folksonomy, i.e., user-defined metadata collections in online streaming services as well as personal music collection management tools. As opposed to expert annotation, these types of tags are deeply related to listeners' or communities' subjective perception of music. In the aforementioned tools and services, a range of activities including search, navigation, and recommendation may depend on the existence of tags associated with tracks. New and rarely accessed tracks however often miss the tags necessary to support them, which leads to well-known problems in music information management [2]. For instance, tracks or artists residing in the long tail of popularity distributions associated with large music catalogues may have insufficient tags, therefore they are rarely recommended or accessed and tagged in online communities. This leads to a circular problem. Expert annotation is notoriously expensive and intractable for large catalogues, therefore content-based annotation is highly valuable to bootstrap these systems. Music tag prediction is often called *music auto-tagging* [3]. Content-based music tagging algorithms aim to automate this task by learning the relationship between tags and the audio content.

A. The Importance of Music Tagging

Music tagging can be seen as a multi-label classification problem because music can be correctly associated with more than one true label, for example, {'rock', 'guitar', 'happy', and '90s'}. This example also highlights the fact that music tagging may be seen as multiple distinct tasks from the perspective of music informatics. Because tags may be related to genres, instrumentation, mood and era, the problem is a combination of genre classification, instrument recognition, mood and era detection, and possibly others. In the following, we highlight three aspects of the task that emphasise its importance in music informatics research (MIR).

First, collaboratively created tags reveal a lot of information about music consumption habits. Tag counts show how listeners label music in the real-world, which is very different from the decision of a limited number of experts (see Section III-A) [4]. The first study on automatic music tagging proposed the use of tags to enhance music recommendation [3] for this particular reason. Second, the diversity of tags and the size of tag datasets make them relevant to several MIR problems including genre classification and instrument recognition. In the context of deep learning, tags can particularly be considered a good *source task* for transfer learning [5], a method of reusing a trained neural network in a related task, after adapting to a smaller and more specific dataset. Since a music tagger can extract features that are relevant to different aspects of music, tasks with insufficient training data may benefit from this approach. Finally, investigating trained music tagging systems may contribute to our understanding of music perception and music itself. For example, analysing subjective tags such as mood and related adjectives can help building computational models for human perception on music (see Section V-A).

B. Contributions and organisation of this paper

This paper focuses on the robustness and generalisability of deep convolutional neural networks for automatic tagging of popular music. Particular attention is paid to audio preprocessing strategies and the choice of hyperparameters related to network architecture. We aim to dispel the myth and common fallacy that having large enough training data is sufficient to address every problem. Therefore we investigate

K. Choi, G. Fazekas, and M. Sandler are with the Centre for Digital Music, Electric Engineering and Computer Science, Queen Mary University of London, London, UK, e-mail: keunwoo.choi@qmul.ac.uk

K. Cho is with Center for Data Science, New York University, New York, USA.

Manuscript received June 2017

the behaviour of our models in the context of noisy datasets. The main contributions of this paper are: *i)* An analysis of the largest and most commonly used public dataset for music tagging including an assessment of the distribution of labels within this dataset. *ii)* We validate the groundtruth and discuss the effects of noise (e.g. mislabelling) on training and evaluation. *iii)* We compare audio input representations and preprocessing methods, and finally, *iv)* analyse the trained network to obtain valuable insight into how social tags are related to music tracks.

The rest of the paper is organised as follows. Section II outlines relevant problems and related works. Section III summarises our analyses of a large tag dataset and the label distribution of typical music tags. In Section IV, we introduce a comprehensive benchmark with different audio input preprocessing methods. In Section V, we assess the capacity of the trained network to represent musical knowledge in terms of similarity between predicted labels and co-occurrences between ground truth labels, utilising parts of the weights corresponding to the final classifications we termed *label vectors*. Finally, we draw overall conclusions and suggest some best practices for network training in Section VI.

II. Background and Related Work

Music tagging relates to common music classification and regression problems such as genre classification and emotion prediction. Using conventional machine learning systems, researches have been focusing on extracting relevant music features and applying a suitable classifier or regressor. For example, the first auto-tagging algorithm [3] proposed the use of mid-level audio descriptors such as Mel-Frequency Cepstral Coefficients (MFCCs) and an AdaBoost [6] classifier. Since most audio features are extracted frame-wise, aggregates such as mean, variance and percentiles are also commonly used to represent the distribution of features. Subsequently, vector quantisation and clustering was proposed in [7] as an alternative to parametric representations.

A recent trend in music tagging is the use of data-driven methods to *learn* features instead of designing them. These approaches are often called *representation learning* or *deep learning*, due to the use of multiple layers in neural networks that aim to learn both low-level features and higher-level semantic categories. Convolutional Neural Networks (denoted 'ConvNets' hereafter) have been providing state-of-the-art performance for music tagging in recent works [8], [1], [9]. In the rest of this section, we first review the datasets relevant to the tagging problem and highlight some issues associated with them. We then discuss the use of ConvNets in the context of music tagging.

A. Music tagging datasets and their properties

Training a music tagger requires examples, i.e., tracks labelled by listeners, constituting a groundtruth dataset. The size of the dataset needed for creating a good tagger depends on the number of parameters in its machine learning model. Using training examples, ConvNets can learn complex, non-linear relationships between patterns observed in the input audio and high-level semantic descriptors such as generic music tags. However, these networks have a very high number of parameters and therefore require large datasets and efficient training strategies. Creating sufficiently large datasets for the general music tagging problem is difficult for several reasons. Compared to genre classification for instance, which can rely mostly on metadata gathered from services such as MusicBrainz[1] or Last.fm[2], tagging often requires listening to the whole track for appropriate labelling because of the many different kinds of tags listeners may use or may be interested in while searching.

Tagging is often seen as an inherently ill-defined problem since it is subjective and there is an almost unconstrained number of meaningful ways to describe music. For instance, in the Million Song Dataset (MSD) [10], one of the largest and most commonly used groundtruth sets for music tagging, there are 522,366 tags outnumbering 505,216 tracks. In fact, there is no theoretical limit on the number of labels in a tag dataset, since users often 'invent' specific labels of cultural significance that cannot be found in a dictionary, yet become widely associated with niche artistic movements, styles, artists or genres. Peculiar misspellings also become commonplace and gain specific meaning, for instance, using 'nu' in place of 'new' in certain genre names (nu jazz, nu metal) suggests music with attention to pop-culture references or particular fusion styles, or the use of 'grrrl' refers to bands associated with the underground feminist punk movement of the 90s. Given a degree of familiarity with the music, listeners are routinely able to associate songs with such tags, even if the metadata related to the artist or the broader cultural context surrounding a particular track is not known to them. Therefore, we can hypothesise that audio can be sufficient to assign a broad range of tags to music automatically.

Tags are also of different kinds and a single tag may often convey only a small part of what constitutes a good description. Tagging therefore is a multi-label classification problem. Consequently, the number of unique output vectors in a set increases exponentially with the number of items, while that of single-label classification only increases linearly. Given K binary labels, the size of the output vector set can increase up to 2^K. In practice, this problem is often alleviated by limiting the number of tags, usually to the top-N tags given the number of music tracks a tag is associated with, or the number of users who applied them.

The prevalence of music tags is also worth paying attention to, because datasets typically exhibit an unbalanced distribution with a long-tail of rarely used tags. From the training perspective, there is an issue with non-uniform genre distributions too. In the MSD for example, the most popular tag is *rock* which is associated with 101,071 tracks. However, *jazz*, the 12nd most popular tag is used for only 30,152 tracks and *classical*, the 71st popular tag is used 11,913 times, despite these three genres are on the same hierarchical level.

[1] See https://musicbrainz.org: a crowd-sourced music meta-database.

[2] See http://www.last.fm: a personalised online radio.

B. Labelling strategies

We finally have to consider a number of labelling strategies. Audio items in a dataset may be 'strongly' or 'weakly' labelled, which may refer to several different aspects of tagging. First, there is a questions of whether only positive labels are used. A particular form of weak labelling means that only positive associations between tags and tracks are provided. This means, given a finite set of tags, a listener (or annotator) applies a tag in case s/he recognises a relation between the tag and the music. In this scenario, no negative relations are provided, and as a result, a tag being positive means it is 'true', but negative means 'unknown'. The most common tags are about positiveness – labels usually explain the existence of features, not the non-existence of them. Exceptions that describe negativeness include 'instrumental', which may indicate the lack of vocals. Typical crowd-sourced datasets are weakly labelled, because it is the only practical solution to create a large dataset. Strong labelling in this particular context would mean that disassociation between a tag and a track confirms negation, i.e., a zero element in a tag-track matrix would signify that the tag does not apply. To the best of our knowledge, *CAL500* [11] is the biggest music tag dataset (500 songs) that is strongly labelled. Most recent research has been relying on collaboratively created, and therefore weakly-labelled datasets such as *MagnaTagATune* [12] (5,405 songs) and the *MSD* [10] containing 505,216 songs if only tagged items are counted.

The second aspect of labelling relates to whether tags describe the whole track or they are only associated with a segment where a tag is considered to be true. Time-varying annotation is particularly difficult and error prone for human listeners, therefore it does not scale. Multiple tags may be applied on a fixed-length segment basis, as it is done in smaller datasets such as MagnaTagATune for 30s segments. The MSD uses only track-level annotation, which can be considered another form of weak labelling. From the perspective of training, this strategy is less adverse for some tags than others. Genre or era tags are certainly more likely to apply to the whole track consistently than instrument tags for instance. This discrepancy may constitute noise in the training data. Additionally, often only preview clips are available to researchers. This forces them to assume that tags are correct within the preview clip too, which constitutes another source of groundtruth noise. In this work, we train ConvNets to learn the association between track-level labels and audio recordings using preview clips associated with the MSD.

C. Convolutional neural networks

ConvNets are a special type of neural network introduced in computer vision to simulate the behaviour of the human vision system [13]. ConvNets have convolutional layers, each of which consist of convolutional kernels. The convolutional kernels sweep over the inputs, resulting in weight sharing that greatly reduces the number of parameters compared to conventional layers that do not sweep and are fully-connected instead. Kernels are trained to find and capture local patterns that are relevant to the task using error backpropagation and gradient descent algorithms. Researchers in music informatics are increasingly taking advantage of deep learning techniques. ConvNets have already been used for chord recognition [14], genre classification [15], onset detection [16], music recommendation [17], instrument recognition [18] and music tagging [1], [8], [9].

A ConvNet is suitable when the input data is a multidimensional array. In MIR, the majority of works use two dimensional time-frequency representations as inputs. Recently, several works proposed learning 2D representations by applying one dimensional convolution to the raw audio signal [8]. It is possible to improve performances by learning more effective representations, although the approach requires increasingly more data, which is not always available. Moreover, these approaches have been shown to learn representations that are similar to conventional time-frequency representations that are cheaper to compute [8], [19].

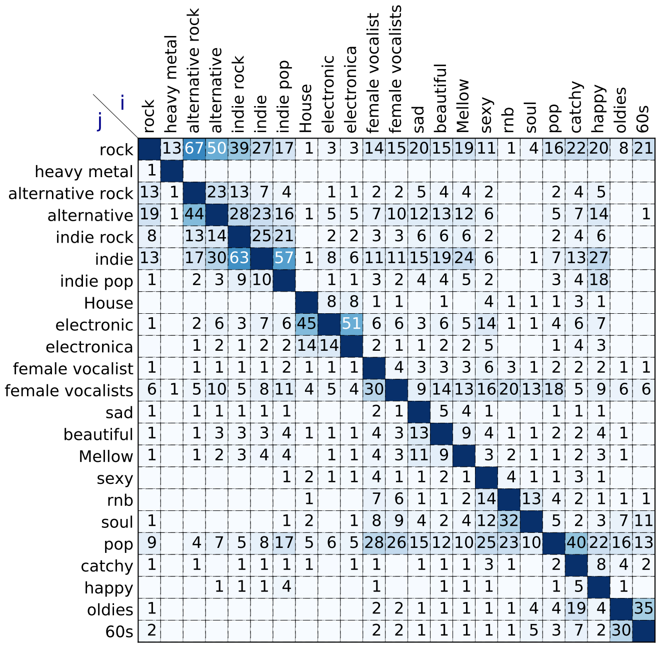
ConvNets have been applied to various music (and audio) related tasks, assuming that certain relevant patterns can be detected or recognised by cascaded one- or two dimensional convolutions. They provide state-of-the-art performances in several music information retrieval tasks including music segmentation [20], beat detection [21] and tagging [9], as well as in non-music tasks such as acoustic event detection [22]. There are several reasons which justify the use of ConvNets in music tagging. First, music tags are often considered among the topmost high-level features representing song-level information above intermediate features such as chords, beats, and tonality. This hierarchy fits well with ConvNets as it can learn hierarchical features over multilayer structures. Second, the properties of ConvNets such as translation, distortion and local invariances can be useful for learning musical features when the relevant feature can appear at any time or frequency range with small time and frequency variances.

D. Evaluation of tagging algorithms

There are several methods to evaluate tagging algorithms. Since the target is binarised to represent if a i^{th} tag is true ($y_i \in \{0, 1\}$), classification evaluation metrics such as 'precision' and 'recall' can be used if the prediction is also binarised. However, optimal thresholding is an additional challenge and discards information. Instead, the *area under curve - receiver operating characteristic* (AUC-ROC, or simply AUC) is used often as an evaluation metric. A ROC curve is created by plotting the true positive rate against the false positive rate. As both rates range between $[0, 1]$, the area under the curve also ranges between $[0, 1]$. However, the effective range of AUC is $[0.5, 1]$ since random classification yields 0.5 of AUC when the true positive rate increases at the exact same rate of false positives.

III. Experiment Set I - Preparation: Tag Dataset Analysis

In this section, we presents results of experiments that analyse the Million Song Dataset. Section III-A is concerned with mutual relationships between tags. In Section III-B, we re-validate the groundtruth of the dataset to ascertain the



i j	rock	heavy metal	alternative rock	alternative	indie rock	indie	indie pop	House	electronic	electronica	female vocalist	female vocalists	sad	beautiful	Mellow	sexy	rnb	soul	pop	catchy	happy	oldies	60s
rock		13	67	50	39	27	17	1	3	3	14	15	20	15	19	11	1	4	16	22	20	8	21
heavy metal	1																						
alternative rock	13	1		23	13	7	4		1	1	2	2	5	4	4	2			2	4	5		
alternative	19	1	44		28	23	16	1	5	5	7	10	12	13	12	6			5	7	14		1
indie rock	8		13	14		25	21		2	2	3	3	6	6	6	2			2	4	6		
indie	13		17	30	63		57	1	8	6	11	11	15	19	24	6		1	7	13	27		
indie pop	1		2	3	9	10			1	1	3	2	4	4	5	2			3	4	18		
House									8	8	1	1		1		4	1	1	1	3	1		
electronic	1		2	6	3	7	6	45		51	6	6	3	6	5	14	1	1	4	6	7		
electronica			1	2	1	2	2	14	14		2	1	1	2	2	5			1	4	3		
female vocalist	1		1	1	1	1	2	1	1	1		4	3	3	3	6	3	1	2	2	2	1	1
female vocalists	6	1	5	10	5	8	11	4	5	4	30		9	14	13	16	20	13	18	5	9	6	6
sad	1		1	1	1	1	1				2	1		5	4	1			1	1	1		
beautiful	1		1	3	3	3	4	1	1	1	4	3	13		9	4	1	1	2	2	4	1	
Mellow	1		1	2	3	4	4		1	1	4	3	11	9		3	2	1	1	2	3	1	
sexy							1	2	1	1	4	1	1	2	1		4	1	1	3	1		
rnb								1			7	6	1	1	2	14		13	4	2	1	1	1
soul	1						1	2		1	8	9	4	2	4	12	32		5	2	3	7	11
pop	9		4	7	5	8	17	5	6	5	28	26	15	12	10	25	23	10		40	22	16	13
catchy	1		1		1	1	1	1		1	1		1	1	1	3	1		2		8	4	2
happy				1	1	1	4				1			1	1	1			1	5		1	
oldies	1										2	2	1	1	1	1	1	4	4	19	4		35
60s	2										2	2	1	1	1	1	1	5	3	7	2	30	

Fig. 1: Normalised tag co-occurrence pattern of the selected 23 tags from the training data. For the sake of visualisation, we selected 23 tags out of 50 that have high co-occurrences and represent different categories; genres, instruments and moods. The values computed using Eq. 1 (and are divided by 100, i.e., shown in percentage), where y_i and y_j respectively indicate the labels on the x-axis and y-axis.

reliability of research that uses it. We select the MSD as it is the largest public dataset available for training music taggers. It also provides crawlable track identifiers for audio signals, which enables us to access the audio and re-validate the tags manually by listening.

The tags in the MSD are collected using the Last.fm API which provides access to crowd-sourced music tags. We use the top 50 tags sorted by popularity (occurrence counts) in the dataset. The tags include genres (*rock, pop, jazz, funk*), eras (*60s – 00s*) and moods (*sad, happy, chill*). The number of clips are 201,672/12,633/28,537 for train/validation/test sets respectively, following the set splits provided by the MSD. The tag counts range from 52,944 (*rock*) to 1,257 (*happy*) and there are 12,348 unique tag vectors represented as a joint variable of 50 binary values.

A. Tag co-occurrences in the MSD

We investigate the distribution and mutual relationships of tags in the dataset. This procedure helps understanding the task. Furthermore, our analysis represents information embedded in the training data. This will be compared to knowledge we can extract from the trained network (see Section V-A).

Here, we investigate the tuple-wise[3] relations between tags and plot the resulting normalised co-occurrence matrix (NCO) denoted $\mathbf{C}$. Let us define $\#y_i := |\{(x, y) \in D | y = y_i\}|$, the total number of the data points with i^{th} label being *True* given the dataset D. In the same manner, $\#(y_i \text{ and } y_j)$ is defined as the number of data points with both i^{th} and j^{th} labels being *True*, i.e., those two tags *co-occur*. NCO is computed using Eq. 1 and illustrated in Fig. 1.

$$C(i, j) = \#(y_i \text{ and } y_j)/\#y_i. \tag{1}$$

In Fig.1, the x and y-axes correspond to i, j respectively. Note that $C(i, j)$ is not symmetric, which is the same when we use the tag words instead of y, e.g., *(alternative rock, rock)* $=$ #(alternative rock and rock)/#alternative rock.

These patterns reveal mutual tag relationships which we categorise into three types: (i) tags belonging to a genre hierarchy, (ii) synonyms, i.e., semantically similar words and (iii) musical similarity. Genre hierarchy tuples include for instance (*alternative rock*, *rock*), (*house*, *electronic*), and (*heavy metal*, *metal*). All first labels are sub-genres of the second. Naturally, we can observe that similar tags such as (*electronica*, *electronic*) are highly likely to co-occur. Lastly, we notice tuples with similar meaning from a musical perspective including (*catchy*, *pop*), (*60s*, *oldies*), and (*rnb*, *soul*). Interestingly, $C(i, j)$ values with highly similar tag pairs y_i and y_j are not close to 100%. For example (*female vocalist*, *female vocalists*)[4] and (*alternative*, *alternative rock*) reach only 30% and 44%. This is because the items are weakly labelled and there are often more preferred tags to describe a certain aspect of a track than others, e.g., *female vocalists* is preferred over *female vocalist*.

B. Validation of the MSD as groundtruth for auto-tagging

Next, we analyse the noise of groundtruth in the MSD and examine the effect of this on training and evaluation. There are many sources of noise including incorrect annotation as well as information loss due to the typical trimming of full tracks to preview clips. Some of these factors may be assumed to be less adverse than others. In large-scale tag datasets, the often used weak labelling strategy (see SectionII-B) may introduce a significant amount of noise. This is because by the definition of weak labelling, for several tags, a large portion of items are simply unlabelled, but then are misinterpreted as negative during training.

Validation of the annotation requires re-annotating the tags after listening to the excerpts, which is not a trivial task for several reasons. First, manual annotation does not scale and requires significant time and effort. Second, there is no single correct answer for many labels – music genre is an ambiguous and controversial concept, emotion is highly subjective as well as labels such as ‘beautiful’ or ‘catchy’. Only instrumentation labels can be objective to some extent, assuming the annotators have expertise in music. Therefore, we re-annotate items in two subsets and using four labels as described below.

- **Labels:** ‘female vocalists’, ‘male vocalists’, ‘instrumental’, ‘guitar’.
- **Subsets:**
  - ■ *Subset100:* randomly selected 100 items from the training set, 50/50 positive/negative labels respectively,
  - ■ *Subset400:* randomly selected 400 items from the test set.

[3] These are not pairwise relations since there is no commutativity due to the normalisation term.

[4] Music by a female vocalist is more often tagged as *female vocalists* than *female vocalist* [2]

TABLE I: The scores of groundtruth with respect to strongly-labelled manual annotation (subset100) in (a)-(d) and occurrence counts by the groundtruth (e), estimation (f), and on Subset400

	Scores				Occurrence counts		
	(a) Error rate, Positive label [%]	(b) Error rate, Negative label [%]	(c) Precision [%]	(d) Recall [%]	(e) In groundtruth (for all items)	(f) Estimate by Eq.2 (on Subset100)	(g) By our re-annotation (on Subset400)
female vocalists	4.0	24.0	96.0	80.0	17,840 (7.3%)	71,127 (29.3%)	94 (23.5%)
male vocalists	2.0	64.0	98.0	60.5	3,026 (1.2%)	156,448 (64.4%)	252 (64.0%)
instrumental	6.0	12.0	94.0	88.7	8,424 (3.5%)	36,048 (14.9%)	85 (21.3%)
guitar	2.0	70.0	98.0	58.3	3,311 (1.4%)	170,916 (70.4%)	266 (66.5%)

1) Groundtruth validation: Table I column (a)-(d) summarises the statistics of Subset100. The average error rate of negative labels is 42.5%, which is very high, while that of positive labels is 3.5%. As a result, the precision of the groundtruth is high (96.5% on average) while the recall is much lower (71.9% on average). This suggests that the tagging problem should be considered weakly supervised learning to some extent. We expect this problem exists in other weakly-labelled datasets as well, since annotators do not tag using all the possible labels.

Such a high error rate for negative labels suggests that the tag occurrence counts in the groundtruth are under-represented. This can be related to the *tagability* of labels, a notion which may be defined as: *how likely it is that a track will be tagged as positive for a label when it really is positive*. If the likelihood is replaced with the portion of items, tagability is measured by recall, as presented in Table I. For example, bass guitar is one of the most widely used instruments in modern popular music, but it is only the 238th most popular tag in the MSD since tagging music with 'bass guitar' does not provide much information from the perspective of the average listener of modern popular music. According to the scores, we can assume that 'female vocalists' (88.7% of recall) and 'instrumental' (80.0%) are more taggable than 'male vocalists' (60.5%) and 'guitar' (58.3%), which presumably indicates the latters are considered to be less peculiar.

The correct number of *positive/negative* items can be estimated by applying Baye's rule with the error rate. The estimated positive label count $\hat{N}^+$ is calculated using Eq.2 as follows:

$$\hat{N}^+ = N^+(1-p^+) + (T-N^+)p^-, \qquad (2)$$

where N^+ is the tag occurrence, T is the number of total items ($T = 242,842$ in our case), and p^+, p^- refers to the error rates of positive and negative labels respectively. Column (f) of Table I presents the estimated occurrence counts using Equation 2. This estimation is validated using Subset400. Comparing the percentages in columns (f) and (g) confirms that the estimated counts are more correct than the tag occurrences of the groundtruth. In short, the correct occurrence count is not correlated with the occurrence in the dataset, which shows the bias by tagability. 'Male vocalists' is more likely to occur in music than 'female vocalists', which means it has lower tagability, and therefore it ends up having fewer occurrences in the groundtruth.

Despite such inaccuracies, it is possible to train networks for tagging with good performances using MSD, achieving AUC between 0.85 [1] and 0.90 [9]. This may be because even with such noise, the network is weakly supervised by stochastically correct feedbacks, where the noise is alleviated by a large number of training examples [23]. In other words, given $\mathbf{x}$ is the input and $\mathbf{y}_{true}$, $\mathbf{y}_{noisy}$ are the correct and noisy labels respectively, the network can approximate the relationship $f : \mathbf{x} \rightarrow \mathbf{y}_{true}$ when training using $(\mathbf{x}, \mathbf{y}_{noisy})$.

2) Validation of the evaluation: Another problem with using a noisy dataset is evaluation. In the previous section, we assumed that the system can learn a *denoised* relationship between music pieces and tags, $f : \mathbf{x} \rightarrow \mathbf{y}_{true}$. However, the evaluation of a network with respect to $\mathbf{y}_{noisy}$ includes errors due to noisy groundtruth. This raises the question about the reliability of the results. We use our strongly-labelled annotation of Subset400 to assess this reliability.

Figure 2 illustrates three AUC scores: *i)* the score of all tags (red) while *ii* and *iii)* are the scores of the four instrumentation tags, where one is evaluated using the dataset groundtruth (blue) and the other is using our annotation (yellow). The reliability of evaluation can be measured by ρ_1, the Pearson correlation coefficient between AUCs using our annotation and the MSD shown in red (dotted) and blue (dashed) curves respectively. The correlation between the four tags and all other tags using the MSD (shown in blue and yellow), which is denoted ρ_2, is a measure of how can we generalise this re-annotation result to the results of all tags. We selected three sets of tagging results. The first two sets, left and centre in Figure 2, are from the experiments described in Section IV-B. These are the scores with varying training data size using melspectrogram and Short-time Fourier transform (STFT) magnitudes respectively. The third diagram compares the results with different settings with varying input time-frequency representations, training data size and input pre-processing techniques. For this assessment, we selected six results that range in [0.786, 0.845]. This is small compared to the first two sets of results.

First, by looking at ρ_1, the results suggest that noisy groundtruth provides reasonably stable evaluation. On the first two sets, the scores of four tags using the MSD groundtruth (in blue) are highly correlated ($\rho_1 = 0.905$ and 0.833) to the scores using our groundtruth set (red). Therefore we will use these for all experiments discussed in subsequent sections. However, on the third set, the scores are in a smaller range while ρ_1 decreases to 0.543. The overall results imply that the evaluation on the groundtruth may be significantly distorted when the performance difference is small. Finally, large ρ_2 indicates that our validation can be generalised to all tags. The correlation coefficients ρ_2 is stable and reasonably high in all three sets. It is 0.856 on average.

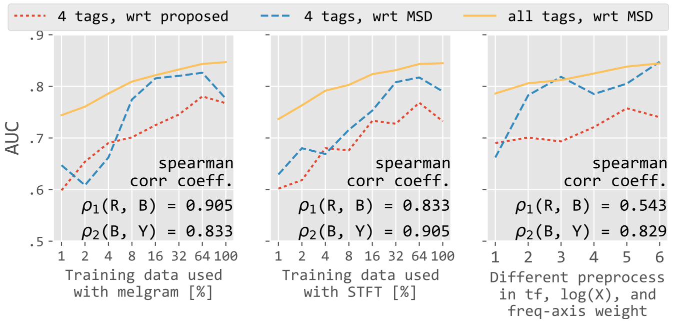


Fig. 2: AUC scores of all tags (yellow, solid) and four instrumentation tags. Instrumentation tags are evaluated using *i)* dataset groundtruth (blue, dashed) and *ii)* our strong-labelling annotation (red, dotted). Pearson correlation coefficients between {red vs. blue} and {blue vs. yellow} is annotated on each chart as ρ.

IV. Experiment Set II - Training and Effective Preprocessing

In this section, we discuss the effects of pre-processing the input audio on the performance of neural networks. Although deep neural networks are known to be universal function approximators [24], training efficiency and performance may vary significantly with different training methods as well as generic techniques including preprocessing the input data [25]. Both empirical decisions and domain knowledge are crucial since choosing between various preprocessing methods can be seen as a non-differentiable choice function, therefore it cannot be optimised using gradient-based learning methods.

As in the previous section, we used the MSD with preview audio clips. The training data are 30-60s stereo mp3 files with a sampling rate of 22,050Hz and 64kbps constant bit-rate encoding. For efficient training in our experiments, we downmix and downsample the signals to 16kHz after decoding and trim the audio duration to 29s to ensure equal-sized input signals. The preprocessing is performed using Librosa [26]. To compare different preprocessing approaches, a ConvNet with 2D kernels and 2D convolution axes was chosen. This showed state-of-the-art performance with efficient training in a prior benchmark [27], where the model we selected was denoted *k2c2*, indicating 2D kernels and convolution axes. As illustrated in Figure 3, homogeneous 2D (3×3) convolutional kernels are used in every convolutional layer. The input has a single channel, 96 mel bins, and 1,360 temporal frames (1, 96, 1360). The figures in Table II denote the number of channels (N), kernel height and kernel width for convolutional layers and subsampling height, subsampling width for max-pooling layers. Here, the hight and width corresponds to the frequency- and time-axes respectively. Exponential linear unit (ELU) is used as an activation function in all convolutional layers [28].

During training, the binary cross-entropy function is used as a loss function. For the acceleration of stochastic gradient descent, we use adaptive optimisation based on ADAM [29]. The experiment is implemented in Python with *Keras* [30] and *Theano* [31] as deep learning frameworks and *Kapre* [32], developed for real-time time-frequency conversion and normalisations on the GPU. The STFT and melspectrogram are computed using a hop size of 256 samples (16ms) with a 512 point DFT aggregated to yield 96 mel bins per frame.

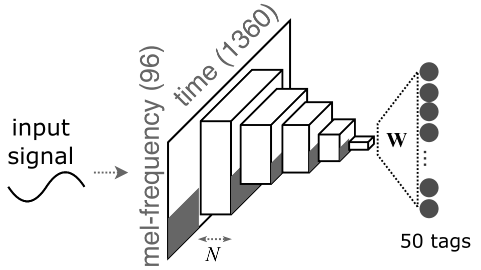


Fig. 3: Network structure of the 5-layer ConvNet. N refers to the number of feature maps while $\mathbf{W}$ refers to the weights matrix of the fully-connected output layer.)

TABLE II: Details of the ConvNet architecture shown in Figure 3. 2-dimensional convolutional layer is specified by (channel, (kernel lengths in frequency, time)). Pooling layer is specified by (pooling length in frequency and time)

input (1, 96, 1360)	
Conv2d and Max-Poolng	(32, (3, 3)) and (2, 4)
Conv2d and Max-Poolng	(32, (3, 3)) and (4, 4)
Conv2d and Max-Poolng	(32, (3, 3)) and (4, 5)
Conv2d and Max-Poolng	(32, (3, 3)) and (2, 4)
Conv2d and Max-Poolng	(32, (3, 3)) and (4, 4)
Fully-connected layer	(50)
output	

A. Variance by different initialisation

In deep learning, using K-fold cross-validation is not a standard practice for two reasons. First, with large enough data and a good split of train, validation and test sets, the model can be trained with small variance. Second, the cost of hyperparameter search is very high and it makes repeating experiments too expensive in practice. For these reasons, we do not cross-validate the ConvNet in this study. Instead, we present the results of repeated experiments with fixed network and training hyperparameters, such as training example sequences and batch size. This experiment therefore measures the variance of the model introduced by different weight initialisations of the convolutional layers. For this, a normal distribution is used following He et al. [33], which has been shown to yield a stable training procedure.

The results are summarised in Figure 4. This shows the AUC scores of 15 repeated experiments on the left, as well as error-bars of their 95% confident interval (CI, 0.00069), mean average error (MAE, 0.00103) and standard deviation (Std, 0.00136) on the right. Small MAE and standard deviation indicate that we can obtain a reliable, precise score by repeating the same experiments for a sufficient number of times. The two largest differences observed between the average AUC score and that of experiment 4 and 8 (AUC differences of 0.0028 and 0.0026 respectively) indicate that we may obtain up to ~ 0.005 AUC difference among experiment instances. Based on this, we can assume that an AUC difference of < 0.005 is non-significant in this paper.

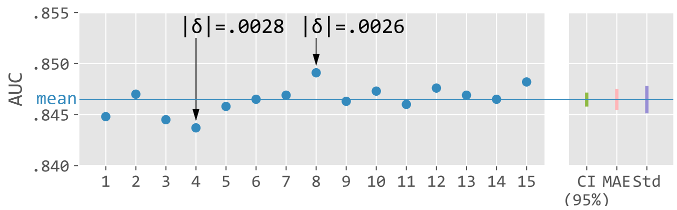


Fig. 4: Performances and their mean (left), as well as the 95% confidence interval (CI), mean absolute error (MAE), and standard deviation (Std) shown on the right. The two deltas on the plot indicate the difference between the average AUC and the scores of experiments 4 and 8.

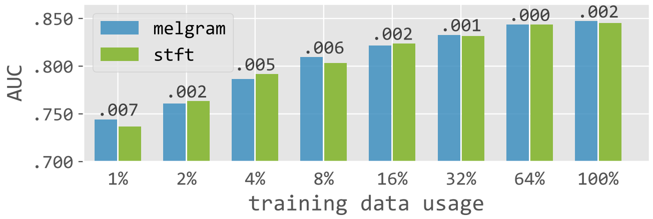


Fig. 5: Performances of predictions with melspectrogram and STFT with varying training data sizes. The numbers above bars indicate the absolute performance differences between melspectrograms and STFTs.

B. *Time-frequency representations*

STFT and melspectrogram have been the most popular input representations for deep learning for music. Melspectrograms provide an efficient and perceptually relevant representation compared to STFT [34] and have been shown to perform well in various tasks [1], [8], [17], [20], [35], [36]. However, an STFT is closer to the original signal and neural networks may be able to learn a representation that is more optimal to the task. This requires large amounts of training data however, as reported in [1] where using melspectrograms outperformed STFT with a smaller dataset.

Figure 5 shows the AUC scores obtained using melspectrogram vs. STFT while varying the size of the utilised training data with logarithmic scaling. Although there are small differences on AUC scores up to 0.007, neither of them outperforms the other, especially when enough data is provided. This rebuts a previous result in [1] because melspectrograms did not have a clear advantage here even with a small training data size. This may be due to the difference in frequency resolution of the representations used and summarised as follows.

- STFT in [1]: $6000/129{=}46.5$ Hz (256-point FFT with 12 kHz sampling rate)
- STFT in our work: $6000/257{=}23.3$ Hz (512-point FFT with 12 kHz sampling rate)
- Melspectrogram in [1] and our work: 35.9 Hz for frequency < 1 kHz (96 mel-bins and by [37] and [26])

In [1], the frequency resolution of the STFT was lower than that of the melspectrogram to enable comparing them with similar number of frequency bins. On the contrary, STFT of higher frequency resolution is used in our experiment and it is found to be as good as melspectrogram in terms of performance. The model using STFT does not take advantage of finer input however. This means overall that melspectrogram may be preferred since its smaller size leads to reduced computation in training and prediction. The figure also illustrates how much the data size affects the performance. Exponentially increasing data size merely results in a linear AUC improvement. AUC starts to converge at 64% and 100%, after which a network with bigger capacity can still make an improvement.

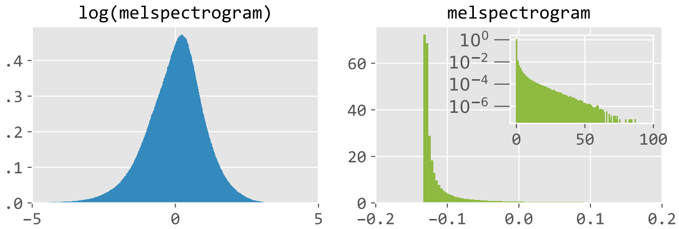


Fig. 6: Normalised histogram of the magnitude of melspectrogram time-frequency bins with (left) and without (right) logarithmic scaling. Log compression significantly affects the histogram making the distribution Gaussian.

C. *Log-scaling of magnitudes*

In this section, we discuss how logarithmic scaling of magnitudes, i.e. decibel scaling, affects performance. This is considered standard preprocessing in music information retrieval. The procedure is motivated by the human perception of loudness [34] which has logarithmic relationship with the physical energy of sound. Although learning a logarithmic function is a trivial task for neural networks, it can be difficult to implicitly learn an optimal nonlinear mapping when it is embedded in a complicated task. A nonlinear mapping was also shown to affect the performance in visual image recognition using neural networks [38]. Figure 6 compares the histograms of the magnitudes of time-frequency bins after zero-mean unit-variance standardisation. On the left, a logarithmically compressed melspectrogram shows an approximately Gaussian distribution without any extreme values. The bins of linear melspectrogram on the right however is extremely condensed in a very small range while they range in wider region (see the small zoomed-out histogram in Figure 6 where y-axis is exponentially spaced to illustrate the long-tail distribution). This means the network should be trained with higher numerical precision to use STFT, hence more vulnerable to noise.

As a result, decibel-scaled melspectrograms always outperform the linear versions as shown in Fig 7, where the same-coloured bars should be compared across within {1 vs. 2} and {1s vs. 2s}. Colours indicate normalization schemes while {1 vs. 1s} and {2 vs. 2s} compare scaling effect, both of which are explained in Section IV-D[5]. Compared to the performance differences while controlling the training set size (the pink bar charts on the right of Figure 7) the additional work introduced by not using decibel scaling can be roughly estimated by comparing these scores to those networks when the training data size is limited. While this also depends on other configurations of the task, seemingly twice the data is required to compensate for the disadvantage of not using a decibel scaled representation.

[5] Decibel-scaled STFT also outperformed linear STFT in our unreported experiments.

TABLE III: Top-20 Similar tag tuples by two analysis approaches. The first row is by analysing co-occurrence of tags in groundtruth (see III-A for details). The second row is by similarity of trained label vector (see V-A for details). Common tuples are annotated with matching symbols.

Similar tags by groundtruth labels	(alternative rock, rock)[§] (indie rock, indie)[#] (House, dance)[‡‡] (indie pop, indie) (classic rock, rock) (electronica, electronic)[*] (alternative, rock) (hard rock, rock) (electro, electronic)[**] (House, electronic) (alternative rock, alternative)[¶] (catchy, pop) (indie rock, rock) (60s, oldies)[††] (heavy metal, metal)[§§] (rnb, soul) (ambient, electronic) (90s, rock) (heavy metal, hard rock)[‡] (alternative, indie)[‖]
Similar tags by label vectors	(electronica, electronic)[*] (indie rock, indie)[#] (female vocalist, female vocalists) (heavy metal, hard rock)[‡] (indie, indie pop) (sad, beautiful) (alternative rock, rock)[§] (alternative rock, alternative)[¶] (happy, catchy) (indie rock, alternative) (alternative, indie)[‖] (rnb, sexy) (electro, electronic)[**] (sad, Mellow) (Mellow, beautiful) (60s, oldies)[††] (House, dance)[‡‡] (heavy metal, metal)[§§] (chillout, chill) (electro, electronica)

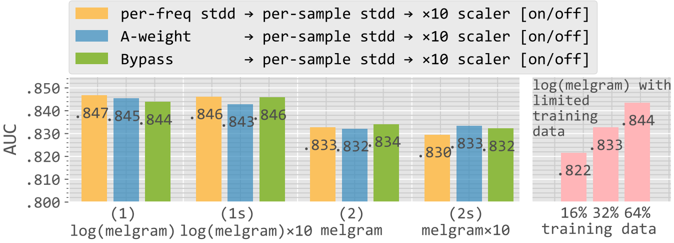


Fig. 7: Performance comparisons of different preprocessing procedures. From left to right, four groups of scores are from different magnitude processing (melspectrogram in decibel scale and linear scale), with additional ×10scaler turned on/off. In each group, yellow/blue/green bars indicates different frequency-axis processing as annotated in the legend.

D. Analysis of scaling effects and frequency-axis weights

Lastly, we discuss the effects of magnitude manipulation. Preliminary experiments suggested that there might be two independent aspects to investigate; *i)* frequency-axis weights and *ii)* magnitude scaling of each item in the training set. Our experiment is designed to isolate these two effects. We tested two input representations *log-melspectrogram* vs. *melspectrogram*, with three frequency weighting schemes *per-frequency*, *A-weighting* and *bypass*, as well as two scaling methods ×10 (`on`) and ×1 (`off`), yielding 2×3×2=12 configurations in total. We summarise the mechanism of each block as follows.

- `per-frequency stdd`: Compute means and standard deviations across time, i.e., per-frequency, and standardise each frequency band using these values. The average frequency response becomes flat (equalised). This method has been used in the literature for tagging [1], singing voice detection [39] and transcription [40].
- `A-weighted`: Apply the international standard IEC 61672:2003 A-weighting curve, which approximates human perception of loudness as a function of frequency.
- `Bypass`: Do not apply any processing, i.e., $f: \mathbf{X} \rightarrow \mathbf{X}$
- `per-sample stdd`: Excerpt-wise normalisation with its overall mean and standard deviation, i.e., using statistics across time and frequency of each spectrogram.
- `×10 scaler`: Multiply the input spectrogram by 10, i.e., $f: \mathbf{X} \rightarrow 10\mathbf{X}$.

1) Frequency weighting: This process is related to the loudness, i.e., human perception of sound energy [34], which is a function of frequency. The sensitivity of the human auditory system drops substantially below a few hundred Hz[6], hence music signals typically exhibit higher energy in the lower range to compensate for the attenuation. This is illustrated in Figure 8, where uncompensated average energy measurements corresponding to the *Bypass* curve (shown in green) yield a peak at low frequencies. This imbalance affects neural network activations in the first layer which may influence performance. To assess this effect, we tested three frequency weighting approaches. Their typical profiles are shown in Figure 8. In all three strategies, excerpt-wise standardisation is used to alleviate scaling effects (see Section IV-D2).

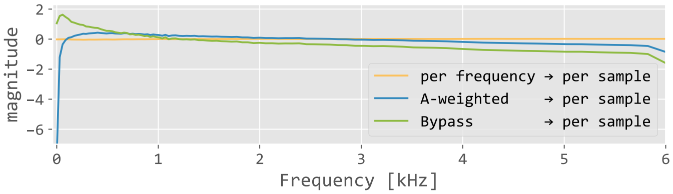


Fig. 8: Average frequency magnitude of randomly selected 200 excerpts with three frequency-axis normalisation. A per-sample (excerpt) standardisation follows to remove the effect of different overall average magnitude.

Our test results show that networks using the three strategies all achieve similar AUC scores. The performance differences within four groups, {`1`, `1s`, `2`, `2s`} in Figure 7 are small and none of them are governing the others. The curves in Figure 8 show the average input magnitudes over frequency. These offsets change along frequency, but the change does not seem large enough to corrupt the local patterns due to the locality of ConvNets, and therefore the network is learning useful representations without significant performance differences within each group.

2) Analysis of scaling effects: We may assume a performance increase if we scale the overall magnitudes for a number of reasons. During training using gradient descent, the gradient of error with respect to weights $\frac{\partial E}{\partial W}$ is proportional to $\frac{\partial}{\partial W} f(W^\top X)$ where f is the activation function. This means that the learning rate of a layer is proportional to the magnitude of input X. In particular, the first layer usually receives the weakest error backpropagation, hence scaling of the input may affect the overall performance.

We tested the effect of this with the results shown in Fig. 7. To this end, consider comparing the same-coloured bars of {`1` vs. `1s`} and {`2` vs. `2s`}. Here, the scaling factor is set to 10 for illustration, however many possible values <100 were tested and showed similar results. In summary, this hypothesis is rebutted – scaling did not affect the ma performance. The analysis of trained weights revealed that different magnitudes of the input only affects the bias of the first convolutional layer. Training with scaling set to ×10 results in 3.4 times larger mean absolute value of the biases in the first layer.

[6] See equal-loudness contours e.g. in ISO 226:2003.

This is due to batch normalization [41] which compensates for the different magnitudes by normalizing the activations of convolutional layers.

V. Experiment Set III - Application: Utilising Trained Network

A trained network for music tagging can provide information beyond the particular task. In this section, the trained weights are used to analyse how the network 'understands' music content by its label in Section V-A. This analysis also provides a way to discover unidentified relationships between labels and music contents.

A. Analysis of predicted label vectors

The goal of label vector analysis is to better understand network training as well as assess its capacity to represent domain knowledge, i.e., relationships between music tags that are not explicitly shown in the data.

In this part of the study, we use the trained ConvNets described in Section IV with the optimal hyperparameters, i.e., melspectrogram input, decibel scaling and per-sample standardisation normalisation (using bypass as described in Section IV-D). In the ConvNet, the output layer has a dense connection from the last convolutional layer. The weights are represented as matrix $\mathbf{W} \in \mathbb{R}^{N \times 50}$, where N is the number of feature maps (N=32 for our case) and the number of predicted labels is 50. After training, the columns of $\mathbf{W}$ can be interpreted as N-dimensional latent vectors since they represent how the networks combine the information in the last convolutional layer to make the final prediction. We call these *label vector* in this paper. We compute pairwise label vector similarity (LVS) using the dot product, i.e., $S(i,j) = w(i) \cdot w(j)$ where $i, j \leq 50$ or equivalently:

$$\mathbf{S} = \mathbf{W}^\top \cdot \mathbf{W}, \tag{3}$$

which yields a 50×50 symmetric matrix.

LVS is illustrated in Figure 9. The pattern is similar to the values in NCO (normalised co-occurrence) shown in Figure 1 (see Sec.II-B). On average, the similarities in $S(i,j)$ are higher than those in $C(i,j)$. In $\mathbf{S}$, only four pairs show negative values, 'classic rock' – 'female vocalists', and 'mellow' – {'heavy metal', 'hard rock', and 'dance'}. In other words, label vectors are distributed in a limited space correponding to a 32 dimensional vector space, where $w(i) \cdot w(j) < \pi/2$ for most of the label vector pairs.

This result can be explained in several different ways. If LVS is high for the pairs of obviously different tags, it means the network fails to learn how to distinguish between the tags. This conclusion is based on our musical knowledge to assess whether they are obviously different or not. In the same manner, we can validate that the network is trained well, by looking at whether high NCO results in high LVS. The Pearson correlation coefficient of the rankings by LVS and NCO is 0.580.[7] Additionally, if pairs with high LVS include pairs of tags of unknown mutual relationships, we may be able to conclude that the audio contents of those tags are similar, once we confirmed that the trained network already contains the information in NCO.

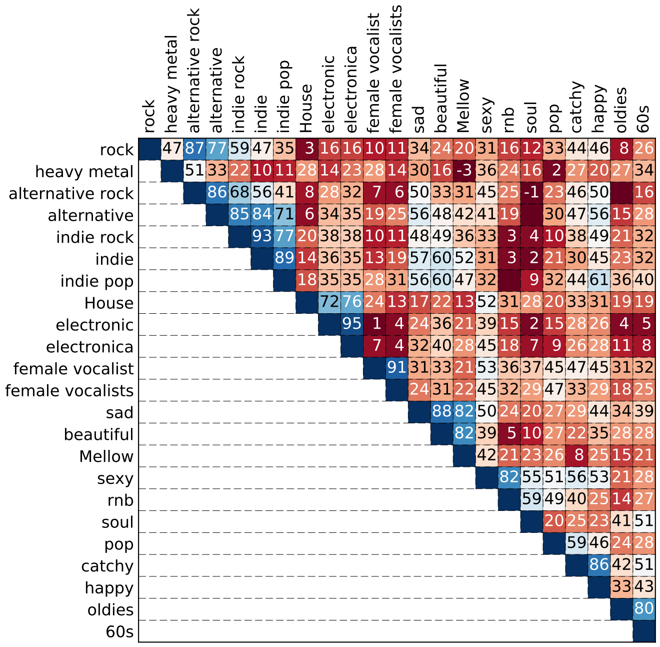


Fig. 9: Label vector similarity matrix by Eq. 3 (of manually selected 23 tags, same in Figure 1, where symmetric components are omitted and numbers are ×100 after dot product for visual clarity.

The top 20 most similar label pairs are sorted and described in Table III. The second row of the table shows similar pairs according to the label vectors estimated by the network. Eleven out of 20 pairs overlap with the top 20 NCO tuples shown in the top row of the table. Most of these relations can be explained by considering the genre hierarchy. Besides, pairs such as ('female vocalists', 'female vocalists') and ('chillout', 'chill') correspond to semantically similar words. Overall, tag pairs showing high similarity (LVS) reasonably represent musical knowledge and correspond to high NCO values computed from the ground truth. This confirms the effectiveness of the network to predict subjective and high-level semantic descriptors from audio only.

There are several pairs that show the extracted representations of the trained network can be used to measure tag-level musical similarities even if they are not explicitly shown in the groundtruth. These can be specified as *i)* high in LVS, *ii)* low in NCO, and *iii)* the assumption that music listeners would reasonably agree with their high similarity. For example, pairs such as ('sad', 'beautiful'), ('happy', 'catchy') and ('rnb', 'sexy') are in the top 20 of LVS (6th, 9th, and 12th similarities with 0.88, 0.86, and 0.82 of similarity values respectively). On the contrary, according to the ground truth, they are only 129th, 232nd, 111th co-occurring with 0.13, 0.08, and 0.14 of co-occurrence likelihood respectively. In summary, the analysis based on LVS indirectly validates that the network learned meaningful representations that correspond to the groundtruth. Moreover, we found several pairs that are considered similar by the network which may help extending our understanding

[7] Because of the asymmetry of $C(i,j)$, rankings of $\max(C(i,j), C(y,y))$ are used.

of the relation between music and tags.

VI. Conclusions

In this article, we investigated several aspects of deep convolutional neural networks for music tagging. We analysed the MSD, the largest dataset available for training a music tagger from a novel perspective, reporting on a study aiming at validating the MSD as groundtruth for this task. We found that the dataset is reliable overall, despite several noise sources affecting training and evaluation. We have shown that input preprocessing can affect the performance. We quantify this in terms of the size of the training data required to achieve similar performances. Among several preprocessing techniques tested in this study, only logarithmic scaling of the magnitude resulted in significant improvement. Finally, we have used *label vectors* to analyse the capacity of the network to explain similarity relations between semantic tags. We found relationship between tags that are not shown in the training data.

Overall, the network in our study was shown to be robust to several potential problems. Training proved robust against groundtruth noise (Section III) as well as sparsity, i.e., a lack of co-occurring labels in sufficient number (Section V-A). The evaluation of the prediction was positively correlated with the evaluation using the groundtruth. During training, the network was resilient to most modifications of the input data, except logarithmic compression of magnitudes in various time-frequency representations. Although we focused on the music tagging task, our results provide general knowledge applicable in several other tasks. The analysis method presented here and the result of the tagging dataset help the analysis or generalise to similar tasks in other domains. For instance, where only weakly labelled datasets are available that are large enough (e.g. image tagging). The analysis of the effects of input preprocessing is applicable in many similar machine-listening problems, e.g., the prediction of environmental sound descriptors.

References

[1] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” in *The 17th International Society of Music Information Retrieval Conference, New York, USA*. International Society of Music Information Retrieval, 2016.

[2] P. Lamere, “Social tagging and music information retrieval,” *Journal of new music research*, vol. 37, no. 2, pp. 101–114, 2008.

[3] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, “Automatic generation of social tags for music recommendation,” in *Advances in neural information processing systems*, 2008, pp. 385–392.

[4] P. Saari, M. Barthet, G. Fazekas, T. Eerola, and M. Sandler, “Semantic models of musical mood: Comparison between crowd-sourced and curated editorial tags,” in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[5] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Transfer learning for music classification and regression tasks,” *https://arxiv.org/abs/1703.09179*, 2017.

[6] Y. Freund, R. E. Schapire *et al.*, “Experiments with a new boosting algorithm,” in *icml*, vol. 96, 1996, pp. 148–156.

[7] M. D. Hoffman, D. M. Blei, and P. R. Cook, “Content-based musical similarity computation using the hierarchical dirichlet process.” in *ISMIR*, 2008, pp. 349–354.

[8] S. Dieleman and B. Schrauwen, “End-to-end learning for music audio,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6964–6968.

[9] J. Lee and J. Nam, “Multi-level and multi-scale feature aggregation using pre-trained convolutional neural networks for music auto-tagging,” *arXiv preprint arXiv:1703.01793*, 2017.

[10] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*. University of Miami, 2011, pp. 591–596.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Towards musical query-by-semantic-description using the cal500 data set,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 439–446.

[12] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging.” in *ISMIR*, 2009, pp. 387–392.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] E. J. Humphrey and J. P. Bello, “Rethinking automatic chord recognition with convolutional neural networks,” in *Machine Learning and Applications, 11th International Conference on*, vol. 2. IEEE, 2012, pp. 357–362.

[15] L. Li, “Audio musical genre classification using convolutional neural networks and pitch and tempo transformations,” 2010.

[16] J. Schlüter and S. Böck, “Musical onset detection with convolutional neural networks,” in *6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic*, 2013.

[17] A. Van den Oord, S. Dieleman, and B. Schrauwen, “Deep content-based music recommendation,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[18] Y. Han, J. Kim, and K. Lee, “Deep convolutional neural networks for predominant instrument recognition in polyphonic music,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[19] J. Lee, J. Park, K. L. Kim, and J. Nam, “Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms,” *arXiv preprint arXiv:1703.01789*, 2017.

[20] K. Ullrich, J. Schlüter, and T. Grill, “Boundary detection in music structure analysis using convolutional neural networks,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Taipei, Taiwan*, 2014.

[21] S. Böck, F. Krebs, and G. Widmer, “Joint beat and downbeat tracking with recurrent neural networks,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[22] A. Jansen, D. Ellis, D. Freedman, J. F. Gemmeke, W. Lawrence, and X. Liu, “Large-scale audio event discovery in one million youtube videos,” in *Proceedings of ICASSP*, 2017.

[23] L. Torresani, “Weakly supervised learning,” in *Computer Vision*. Springer, 2014, pp. 883–885.

[24] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[25] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[26] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, E. Battenberg, J. Moore, D. Ellis, R. YAMAMOTO, R. Bittner, D. Repetto, P. Viktorin, J. F. Santos, and A. Holovaty, “librosa: 0.4.1,” Oct. 2015. [Online]. Available: https://doi.org/10.5281/zenodo.32193

[27] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2016.

[28] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[30] F. Chollet, “Keras: Deep learning library for theano and tensorflow,” https://github.com/fchollet/keras, 2015.

[31] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, “Theano: new features and speed improvements,” *arXiv preprint arXiv:1211.5590*, 2012.

[32] K. Choi, “kapre: Keras audio preprocessors,” *GitHub repository: https://github.com/keunwoochoi/kapre*, 2016.

[33] K. He, X. Zhang, S. Ren, and J. 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.

[34] B. C. Moore, *An introduction to the psychology of hearing*. Brill, 2012.

[35] J. Nam, J. Herrera, and K. Lee, "A deep bag-of-features model for music auto-tagging," *arXiv preprint arXiv:1508.04999*, 2015.

[36] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*. IEEE, 2014.

[37] M. Slaney, "Auditory toolbox," *Interval Research Corporation, Tech. Rep*, vol. 10, p. 1998, 1998.

[38] S. F. Dodge and L. J. Karam, "Understanding how image quality affects deep neural networks," *CoRR*, vol. abs/1604.04004, 2016. [Online]. Available: http://arxiv.org/abs/1604.04004

[39] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 44–50.

[40] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic music transcription," *arXiv preprint arXiv:1508.01774*, 2015.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.