基于深度卷积神经网络的自动标签

Keunwoo Choi, Gyo¨rgy Fazekas, Mark Sandler
Queen Mary University of London
{keunwoo.choi, g.fazekas, mark.sandler}@qmul.ac.uk

摘要

我们提出了一种使用完全卷积神经网络（FCN）的基于内容的自动音乐标记算法。我们仅评估由 2D 卷积层和降采样层组成的不同网络结构。在实验中，我们使用 MagnaTagATune 数据集测量了具有不同复杂性和输入类型的网络结构的 AUC-ROC 得分，其中采用了梅尔谱图输入的 4 层结构显示了最先进的性能。此外，我们通过在较大数据集（MSD）上改变网络层数来评估网络结构的性能，发现更深的模型优于 4 层结构。实验表明，梅尔谱图是自动标记的有效时频表示，以及更复杂的模型受益于更多的训练数据。

1. 引言

音乐标签是一组*描述性关键字*，可传达有关音乐片段的高级信息，例如情感（悲伤，愤怒，快乐），流派（爵士，古典）和乐器（吉他，弦乐，人声，器乐））。由于标签提供了来自听众角度的高级信息，因此它们可用于音乐发现和推荐。

自动标记是一项分类任务，旨在使用音频信号预测音乐标签。这要求首先提取声学特征，这些特征是我们感兴趣的标签类型的良好估计，随后进行单标签或多标签分类，或者在某些情况下进行回归。从特征提取的角度来看，文献中提出了两种主要类型的系统。传统上，特征提取依赖于信号处理前端，以便从时域或频域音频表示中计算相关特征。这些特征随后被用作机器学习阶段的输入。但是，很难知道哪些特征与当前任务相关。尽管特征选择已被广泛用于解决此问题[29]，但仍未出现关于特征与标签类别的良关联的明确建议。最新的方法将特征提取与机器学习结合在一起，以允许自动学习相关特征。这种方法称为特征学习，需要深度神经网络（DNN）。

在[25]中引入了聚合人工设计的特征以进行音乐标记的方法。 随后的几项工作依赖于“帧袋”方法——对每一帧计算一组特征，然后进行统计聚合。典型特征被设计代表声音的物理或感知特性，包括 MFCC，MFCC 的派生特征和谱特征（例如频谱滚降和频谱质心）。由于这些是帧级特征，因此可以计算它们的统计数据（例如均值和方差）[25]，或者将它们进行聚类和矢量量化[15]，以获得片段级特征。最后，将分类器（例如 k-NN 或支持向量机）应用于标签预测。

作为上述系统的替代方案，DNN 在计算机视觉，语音识别[19]和自动标记[6、8、18、28]中取得成功之后，最近已广泛用于音频分析。从工程的角度来看，DNN 避开了构建或寻找任务相关音频特征这一问题。它们的常见结构包括多个隐藏层，其中隐藏单元被训练以表示数据中的某些基础结构。

在计算机视觉领域中已经引入了深度卷积神经网络（CNN），因为它们可以模拟人类视觉系统的行为并学习层次特征，从而在模型中使对象具有对平移和变形的局部不变性和鲁棒性[14]。出于类似的原因，CNN 已被引入基于音频的问题中，并在语音识别[19]和音乐分割[26] 中显示出最佳性能。

针对自动音乐标记也已经提出了几种与 DNN 相关的算法。在[5]和[28]中，球形 k-均值算法和多层感知器被分别用作特征提取器和分类器。在[5]中使用多分辨率频谱图来利用音频信号在不同的时间尺度上的信息。在[28]中，多层感知器的预训练权重被迁移以预测其他数据集的标签。在[6]中使用了一个两层的卷积网络，并以梅尔频谱图和原始音频信号作为输入特征。[18]则是提取特征包并将其输入到堆叠受限玻尔兹曼机（RBM）中。

在本文中，我们提出了一种基于深度完全卷积网络（FCN）的自动标记算法。

FCN 是仅由卷积层（和降采样层）组成而不包含任何全连接层的深度卷积网络。FCN 最大化了卷积网络的优势。它通过权重共享来减少参数的数量，并使学习到的特征具有对频谱图时频平面上位置的不变性，也就是说，该网络对音频信号的时间和谐波结构建模，从而比人工设计和统计聚合的特征更具优势。在所提出的架构中，我们将三到七个卷积层与降采样层结合使用，从而将特征图的大小减小到 1×1，整个过程是完全卷积化的。随后我们采用 2D 卷积核来处理局部谐波关系。

我们将在第 2 节中详细介绍 CNN，并在第 3 节中定义问题。我们将在第 4 节中将提出我们的架构，在第 5 节中给出对它的评估，最后在第 6 节中得出结论。

2. 将 CNN 用于音乐信号分析

2.1 将 CNN 用于音频分析的动机

在本节中，我们将针对音乐信号回顾 CNN 的性质。CNN 的发展是由生物视觉系统推动的，在该系统中，局部区域的信息被许多感觉细胞反复捕获并用于提取更高级别的信息。因此，CNN 提供了一种学习鲁棒特征的方法，学习到的特征对应于局部、平移、变形不变视觉对象。这些优点通常也适用于音频信号，尽管音频信号（或其 2D 表示）的拓扑与视觉图像的拓扑不同。

CNN 已被应用于各种音频分析任务，主要是假设可以通过*观察（see）*其时频表示来检测或识别听觉事件。尽管深度学习的优点是学习特征，但应仔细考虑网络的架构，并考虑在多大程度上需要某种属性（例如不变性）。

下述几个理由表明在自动标签任务中使用 CNN 是合理的。首先，音乐标签通常被认为是表示歌曲级信息的最高级特征之一，位于中级特征（例如和弦，节拍，音调和时间包络，它们随时间和频率而变化）之上。这种层次结构非常适合 CNN，因为它能借助多层结构学习层次结构特征。其次，如果与标签相关的目标音乐事件可以出现在任何时间或频率范围，那么 CNN 的特性（例如平移，失真和局部不变性）对于学习音乐特征很有用。

2.2 CNN 的架构设计

将 CNN 应用于音频信号有许多变体。它们的区别在于输入表示的类型，卷积轴，卷积核或降采样的大小和数量以及隐藏层的数量。

2.2.1 时频表示

梅尔谱图已成为标签[5]，边界检测[26]，起始检测[21]和潜在特征学习[27]中应用最广泛的特征之一。梅尔尺度的使用得到人类听觉系统领域知识[17]的支持，并已通过各种任务[6、18、21、26、27]中的性能提升得到了经验证明。 常数 Q 变换（CQT）主要用于应精确识别音符基本频率，例如和弦识别[10]和转录[22]。

当需要逆变换时，最好直接使用短时傅立叶变换（STFT）系数[3，23]，例如边界检测[7]。但与 STFT 在数字信号处理中的普遍使用相比，它不那么受欢迎。与 CQT 相比，STFT 的频率分辨率在低频范围内不足以识别基本频率。相反，在相同的频带数量下，STFT 在大于 2kHz 的频段上提供比梅尔谱图更好的分辨率（对于某些任务更合适）。但是，到目前为止，它并不是最受青睐的选择。

最近，有研究专注于从给定任务的原始音频中学习某种优化变换。这些方法被称为端到端模型，并被应用于音乐[6]和语音[20]。在语音识别中其性能可以媲美梅尔谱图[20]。另外值得注意的是，在[6]和[20]中学习到的滤波器组都与梅尔尺度相似，从而支持使用人类听觉系统中的非线性。

2.2.2 卷积——核尺寸与轴

每个大小为 H×W×D 的卷积层都学习 H×W 的 D 个特征，其中 H 和 W 分别表示学习的内核的高度和宽度。 内核大小决定了它可以精确捕获的组件的最大大小。 如果内核大小太小，则该层将无法学习有意义的数据形状（或分布）表示。 因此，在[10]中提出了相对较大的内核，例如 17×5。 任务（和弦识别）也证明了这一点，在该任务中，沿频率轴的分布发生细微变化应会产生不同的结果，因此不应允许频率不变。Each convolution layer of size $H\times W \times D$ learns D features of $H\times W$, where H and W refer to the height and the width of the learned kernels respectively. The kernel size determines the maximum size of a component it can precisely capture. If the kernel size is too small, the layer would fail to learn a meaningful representation of shape (or distribution) of the data. For this reason, relatively large-sized kernels such as 17×5 are proposed in [10]. This is also justified by the task (chord recognition) where a small change in the distribution along the frequency axis should yield different results and therefore frequency invariance shouldn’t be allowed.

The use of large kernels may have two drawbacks however. First, it is known that the number of parameters per representation capacity increases as the size of kernel increases. For example, 5×5 convolution can be replaced with two stacked 3×3 convolutions, resulting in a fewer number of parameters. Second, large kernels do not allow invariance within its range.

The convolution axes are another important aspect of convolution layers. For tagging, 1D convolution along the time axis is used in [6] to learn the temporal distribution, assuming that different spectral band have different distributions and therefore features should be learned per frequency band. In this case, the global harmonic relationship is considered at the end of the convolution layers and fully-connected layers follow to capture it. In contrast, 2D convolution can learn both temporal and spectral structures and has already been used in music transcription [22], onset detection [21], boundary detection [26] and chord recognition [10].

2.2.3 *Pooling - sizes and axes*

Pooling reduces the size of feature map with an operation, usually a *max* function. It has been adopted by the majority of works that are relying on CNN structures. Essentially, pooling employs subsampling to reduce the size of feature map while preserving the information of *an activation* in the region, rather than information about the whole input signal.

This non-linear behaviour of subsampling also provides distortion and translation invariances by discarding the original location of the selected values. As a result, pooling size determines the *tolerance* of the location variance within each layer and presents a trade-off between two aspects that affect network performance. If the pooling size is too small, the network does not have enough distortion invariance, if it is too large, the location of features may be missed when they are needed. In general, the pooling axes match the convolution axes, although it is not necessarily the case. What is more important to consider is the axis in which we need invariance. For example, time-axis pooling can be helpful for chord recognition, but it would hurt time-resolution in boundary detection methods.

3. PROBLEM DEFINITION

Automatic tagging is a *multi-label classification task*, i.e., a clip can be tagged with multiple tags. It is different from other audio classification problems such as genre classification, which are often formalised as a single-label classification problem. Given the same number of labels, the output space of multi-label classification can exponentially increase compared to single-label classification. Accordingly, multi-label classification tasks require more data, a model with larger capacity and efficient optimisation methods to solve. If there are K exclusive labels, the classifier only needs to be able to predict one among K different vectors, which are *one-hot vectors*. With multiple labels however, the number of cases increases up to 2^K.

In crowd-sourced music tag datasets [2,13], most of the tags are *false*(0) for most of the clips, which makes accuracy or mean square error inappropriate as a measure. Therefore we use the Area Under an ROC (Receiver Operating Characteristic) Curve abbreviated as AUC. This measure has two advantages. It is robust to unbalanced datasets and it provides a simple statistical summary of the performance in a single value. It is worth noting that a random guess is expected to score an AUC of 0.5 while a perfect classification 1.0, i.e., the effective range of AUC spans between [0.5, 1.0].

FCN-4
Mel-spectrogram *(input: 96×1366×1)*
Conv 3×3×128
MP (2, 4) *(output: 48×341×128)*
Conv 3×3×384
MP (4, 5) *(output: 24×85×384)*
Conv 3×3×768
MP (3, 8) *(output: 12×21×768)*
Conv 3×3×2048
MP (4, 8) *(output: 1×1×2048)*
Output 50×1 (sigmoid)

Table 1. The configuration of FCN-4

4. PROPOSED ARCHITECTURE

Table 1 and Figure 1 show one of the proposed architectures, a 4-layer FCN (*FCN-4*) which consists of 4 convolutional layers and 4 max-pooling layers. This network takes a log-amplitude mel-spectrogram sized 96×1366 as input and predicts a 50 dimensional tag vector. The input shape follows the size of the mel-spectrograms as explained in Section 5.1.

The architecture is extended to deeper ones with 5, 6 and 7 layers (*FCN-{5, 6, 7}*). The number of feature maps and subsampling sizes are summarised in Table 2. The number of feature maps of FCN-5 are adjusted based on FCN-4, making the hierarchy of the learned features deeper. FCN-6 and FCN-7 however have additional 1×1 convolutional layers(s) on the top of FCN-5. Here, the motivation of 1×1 is to take advantage of increased nonlinearity [16] in the final layer, assuming that the five layers of FCN-5 are sufficient to learn hierarchical features. An architecture with 3 layers (FCN-3) is also tested as a baseline with a pooling strategy of [(3,5),(4,16),(8,17)] and [256, 768, 2048] feature maps. The number of feature maps are adjusted based on FCN-4 while the pooling sizes are set to increase in each layer so that low-level features can have sufficient resolutions.

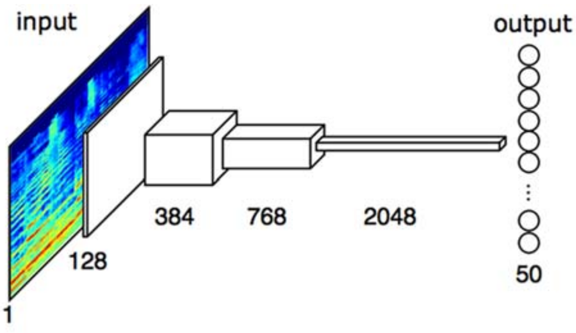

Figure 1. A block diagram of the proposed 4-layer architecture, *FCN-4*. The numbers indicate the number of feature maps (i.e. channels) in each layer. The subsampling layers decrease the size of feature maps to 1×1 while the convolutional layers increase the depth to 2048.

Other configurations follow the current generic optimisation methods in CNNs. Rectified Linear Unit (ReLU) is used as an activation function in every convolutional layer except the output layer, which uses Sigmoid to squeeze the output within [0, 1]. Batch Normalisation is added after every convolution and before activation [11]. Dropout of 0.5 is added after every max-pooling layer [24]. This accelerates the convergence while dropout prevents the network from overfitting.

Homogeneous 2D (3 × 3) convolutional kernels are used in every convolutional layers except the final 1 × 1 convolution. 2D kernels are adopted in order to encourage the system to learn the *local* spectral structures. The kernels at the first convolutional layer cover 64 ms × 72 Hz. The coverage increases to 7s × 692 Hz at the final 3 × 3 convolutional layer when the kernel is at the low-frequency. The time and frequency resolutions of feature maps become coarser as the max-pooling layer reduces their sizes, and finally a single value (in a 1 × 1 feature map) represents a feature of the whole signal.

Several features in the proposed architecture are distinct from previous studies. Compared to [28] and [18], the proposed system takes advantages of convolutional networks, which do not require any pre-training but fully trained in a supervised fashion. The architecture of [6] may be the most similar to ours. It takes mel-spectrogram as input, uses two 1D convolutional layers and two (1D) max-pooling layers as feature extractor, and employs one fully-connected layer as classifier. The proposed architectures however consist of 2D convolution and pooling layers, to take the potential local harmonic structure into account. Results from many 3s clips are averaged in [6] to obtain the final prediction. The proposed model however takes the whole 29.1s signal as input, incorporating a temporal nonlinear aggregation into the model.

The proposed architectures can be described as fully-convolutional networks (FCN) since they only consist of convolutional and subsampling layers. Conventional CNNs have been equipped with fully-connected layers at the end of convolutional layers, expecting each of them to perform as a feature extractor and classifier respectively. In general however, the fully connected layers account for the majority of parameters and therefore make the system prone to overfitting. This problem can be resolved by using FCNs with average-pooling at the final convolutional layer. For instance in [16], the authors assume that the target visual objects may show large activations globally in the corresponding images. Our systems resemble the architecture in [16] except the pooling method, where we only use max-pooling because some of the features are found to be local, e.g. the voice may be active only for the last few seconds of a clip.

FCN-5	FCN-6	FCN-7
Mel-spectrogram *(input: 96 × 1366 × 1)*		
Conv 3 × 3 × 128		
MP (2, 4) *(output: 48 × 341 × 128)*		
Conv 3 × 3 × 256		
MP (2, 4) *(output: 24 × 85 × 256)*		
Conv 3 × 3 × 512		
MP (2, 4) *(output: 12 × 21 × 512)*		
Conv 3 × 3 × 1024		
MP (3, 5) *(output: 4 × 4 × 1024)*		
Conv 3 × 3 × 2048		
MP (4, 4) *(output: 1 × 1 × 2048)*		
	Conv 1 × 1 × 1024	Conv 1 × 1 × 1024
		Conv 1 × 1 × 1024
Output 50 × 1 (sigmoid)		

Table 2. The configurations of 5, 6, and 7-layer architectures. The only differences are the number of additional 1 × 1 convolution layers.

5. EXPERIMENTS AND DISCUSSION

5.1 Overview

Two datasets were used to evaluate the proposed system, the MagnaTagATune dataset [13] and the Million Song Dataset (MSD) [2]. The MagnaTagATune dataset has been relatively popular for content-based tagging, but similar performances from recent works [5, 6, 18, 28] seem to suggest that performances are saturated, i.e. a glass-ceiling has been reached due to noise in the annotation. The MSD contains more songs than MagnaTagATune, it has various types of annotations up to 1M songs. There have not been many works to compare our approach with, partly because audio signals do not come with the dataset. Consequently, we use the MagnaTagATune dataset to compare the proposed system with previous methods and evaluate the variants of the system using the MSD.

In Experiment I, we evaluate three architectures (FCN-{3,4,5}) with mel-spectrogram input as proposed in Section 4. Furthermore, we evaluated STFT, MFCC, and mel-spectrogram representations as input of FCN-4. The architecture of STFT input is equivalent to that of mel-spectrograms with small differences in pooling sizes in the frequency axis due to the different number of spectral bands. For the architecture of MFCCs, we propose a frame-based 4-layer feed-forward networks with time-axis pooling (instead of 2D convolutions and poolings) because relevant information is represented by each MFCC rather than its local relationships. In Experiment II, we evaluate five architectures (FCN-{3,4,5,6,7}) with mel-spectrogram input.

Computational cost is heavily affected by the size of the input layers which depends on basic signal parameters of the input data. A pilot experiment demonstrated similar performances with 12 and 16 kHz sampling rates and mel-bins of 96 and 128 respectively. As a result, the audio in both datasets was trimmed as 29.1s clips (the shortest signal in the dataset) and was downsampled to 12 kHz. The hop size was fixed at 256 samples (21 ms) during time-frequency transformation, yielding 1,366 frames in total. STFT was performed using 256-point FFT while the number of mel-bands was set as 96. For each frame, 30 MFCCs and their first and second derivatives were computed and concatenated.

We used ADAM adaptive optimisation [12] on Keras [4] and Theano [1] framework during the experiments. Binary cross-entropy function is used since it shows faster convergence and better performance than distance-based functions such as mean squared error and mean absolute error.

5.2 Experiment I: MagnaTagATune

The MagnaTagATune dataset consists of 25,856 clips of 29.1-s, 16 kHz-sampled mp3 files with 188 tags. We only uses Top-50 tags, which includes genres (*classical, rock*), instruments (*piano, guitar, vocal, drums*), moods (*soft, ambient*) and other descriptions (*slow, Indian*). The dataset is not balanced, the most frequent tag is used 4,851 times while the 50-th most frequent one used 490 times in the training set. The labels of the dataset consist of 7,644 unique vectors in a 50-dimensional binary vector space.

The results of the proposed architecture and its variants are summarised in Table 3. There is little performance difference between FCN-4 and FCN-5. It is a common phenomenon that an additional layer does not necessarily lead to an improved performance if, *i*) the gradient may not flow well through the layers or *ii*) the additional layer is simply not necessary in the task but only adds more parameters. This results in overfitting or hindering the optimisation. In our case, the most likely reason is the latter of the two. First, the scores are only slightly different, second, both FCN-4 and FCN-5 showed similar performances compared to previous research as shown in Table 4. Similar results were found in the comparison of FCN-5, FCN-6, and FCN-7 in Experiment II. These are discussed in Section 5.3.

	AUC
FCN-3, mel-spectrogram	.852
FCN-4, mel-spectrogram	**.894**
FCN-5, mel-spectrogram	.890
FCN-4, STFT	.846
FCN-4, MFCC	.862

Table 3. The results of the proposed architectures and input types on the MagnaTagATune Dataset

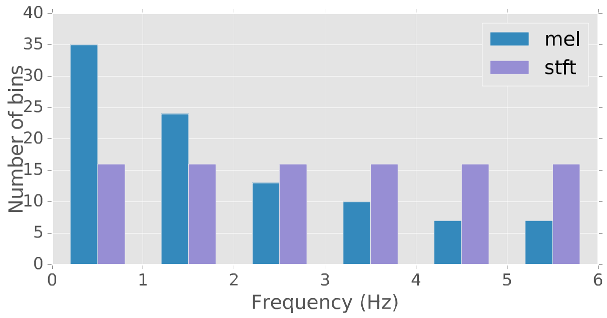

Figure 2. The numbers of bins per 1kHz bandwidth in mel-spectrograms and STFTs .

Methods	AUC
The proposed system, FCN-4	.894
2015, Bag of features and RBM [18]	.888
2014, 1D convolutions [6]	.882
2014, Transferred learning [28]	.88
2012, Multi-scale approach [5]	.898
2011, Pooling MFCC [8]	.861

Table 4. The comparison of results from the proposed and the previous systems on the MagnaTagATune Dataset

The degradations with other types of input signals–STFT and MFCC–are rather significant. This result is aligned with the preferences of mel-spectrograms over STFT on automatic tagging [5,6,18,27]. However, this claim is limited to this or very similar tasks where the system is trained on labels such as genres, instruments, and moods. Figure 2 shows how 96 frequency bins are allocated by mel-spectrograms and STFT in every 1kHz bandwidth. This figure, combined with the result in Table 3 shows that high-resolution in the low-frequency range helps automatic tagging. It also supports the use of downsampling for automatic tagging. Focusing on low-frequency can be more efficient.

Table 4 shows the performance of FCN-4 in comparison to the previous algorithms. The proposed algorithm performs competitively against the other approaches. However, many different algorithms only show small differences in the range of an AUC score of 0.88 – 0.89, making their performances difficult to compare. This inspired the authors to execute a second experiment discussed in the next section. In summary, the mel-spectrograms showed better performance than other types of inputs while FCN-4 and FCN-5 outperformed many previously reported architectures and configurations.

5.3 Experiment II: Million Song Dataset

We further evaluated the proposed structures using the Million Song Dataset (MSD) with *last.fm* tags. We select the top 50 tags which include genres (*rock, pop, jazz, funk*), eras (*60s – 00s*) and moods (*sad, happy, chill*). 214,284 (201,680 for training and 12,605 for validation) and 25,940 clips are selected from the provided training/test sets by filtering out items without any top-50 tags. The number of tags ranges from 52,944 (*rock*) to 1,257 (*happy*) and there are 12,348 unique tag vectors. Note that the size of the MSD is more than 9 times larger than the MagnaTagATune dataset.

The results of the proposed architectures with different numbers of layers are summarised in Table 5. Unlike the result from Experiment I, where FCN-4 and FCN-5 showed a slight difference of the performance (AUC difference of 0.008), FCN-5,6,7 resulted in significant improvements compared to FCN-4, showing that deeper structures benefit more from sufficient data. However, FCN-6 outperformed FCN-5 only by AUC 0.003 while FCN-7 even

Methods	AUC
FCN-3, mel-spectrogram	.786
FCN-4, —	.808
FCN-5, —	.848
FCN-6, —	**.851**
FCN-7, —	.845

Table 5. The results from different architectures of the proposed system on the Million Song Dataset

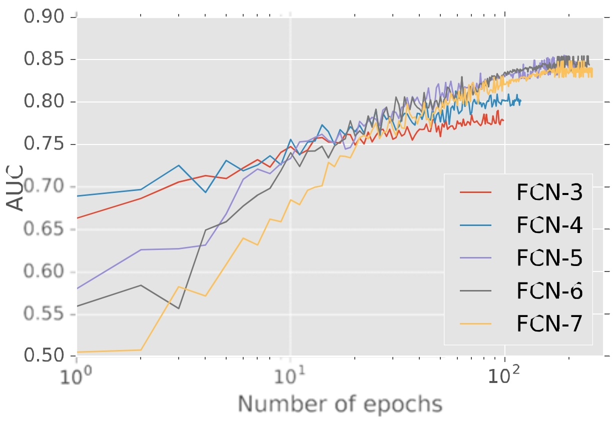

Figure 3. The learning curves of the AUC scores measured on the validation set (on the Million Song Dataset)

showed a slightly worse performance than FCN-6. This result agrees with a known insight in using deep neural networks. The structures of DNNs need to be designed for easier training when there are a larger number of layers [9]. In theory, more complex structures can perform at least equal to simple ones by learning an identity mapping. Our results supports this. In the experiment, the performances of FCN-6 and FCN-7 were still making small improvements at the end of the training, implying it may perform equal to or even outperform FCN-5. In practice, this approach is limited by computational resources and therefore *very deep* structures may need to be designed to motivate efficient training, for instance, using deep residual networks [9].

Figure 3 illustrates the learning curves of the AUC scores on the validation set. At the beginning of the training, there is a tendency that simpler networks show better performance because there is a fewer number of parameters to learn. FCN-4 and FCN-5 show similar performance between around 20–40 epochs. Based on this, it can be assumed that learning on the MagnaTagATune dataset stayed within this region and failed to make more progress due to the scarcity of training data. To summarise, FCN-5, FCN-6, and FCN-7 significantly outperformed FCN-3 and FCN-4. The results imply that more complex models benefit from more training data. The similar results obtained using FCN-5, FCN-6 and FCN-7 indicate the need for more advanced design methodologies and training of deep neural networks.

6. CONCLUSION

We presented an automatic tagging algorithm based on deep fully convolutional neural networks (FCN). It was shown that deep FCN with 2D convolutions can be effectively used for automatic music tagging and classification tasks. In Experiment I (Section 5.2), the proposed architectures with different input representations and numbers of layers were compared using the MagnaTagATune dataset against the results reported in previous works showing competitive performance. With respect to audio input representations, using mel-spectrograms resulted in better performance compared to STFTs and MFCCs. In Experiments II (Section 5.3), different number of layers were evaluated using the Million Song Dataset which contains nine times as many music clips. The optimal number of layers were found to be different in this experiment indicating deeper networks benefit most from the availability of large training data. In the future, automatic tagging algorithms with variable input lengths will be investigated.

7. ACKNOWLEDGEMENTS

This work was part funded by the FAST IMPACt EPSRC Grant EP/L019981/1 and the European Commission H2020 research and innovation grant AudioCommons (688382). Sandler acknowledges the support of the Royal Society as a recipient of a Wolfson Research Merit Award.

8. REFERENCES

[1] Fre´de´ric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[2] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, pages 591–596, 2011.

[3] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Ma´laga, Spain, October 26-30, 2015*, 2015.

[4] Franc¸ois Chollet. Keras: Deep learning library for theano and tensorflow. https://github.com/fchollet/keras, 2015.

[5] Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, 2013.

[6] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014.

[7] Thomas Grill and Jan Schlu¨ter. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *Proceedings of the 23rd European Signal Processing Conference (EUSPICO 2015), Nice, France*, 2015.

[8] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, pages 729–734, 2011.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[10] Eric J Humphrey and Juan P Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Machine Learning and Applications, 11th International Conference on*, volume 2, pages 357–362. IEEE, 2012.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[13] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, Japan, October 26-30, 2009*, pages 387–392. ISMIR, 2009.

[14] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[15] Dawen Liang, Minshu Zhan, and Daniel PW Ellis. Content-aware collaborative music recommendation using pre-trained neural networks. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Ma´laga, Spain, October 26-30, 2015*, 2015.

[16] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[17] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.

[18] Juhan Nam, Jorge Herrera, and Kyogu Lee. A deep bag-of-features model for music auto-tagging. *arXiv preprint arXiv:1508.04999*, 2015.

[19] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for lvcsr. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8614–8618. IEEE, 2013.

[20] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *Proc. Interspeech*, 2015.

[21] Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. IEEE, 2014.

[22] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic music transcription. *arXiv preprint arXiv:1508.01774*, 2015.

[23] Andrew JR Simpson, Gerard Roma, and Mark D Plumbley. Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. *arXiv preprint arXiv:1504.04658*, 2015.

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

[25] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

[26] Karen Ullrich, Jan Schlu¨ter, and Thomas 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.

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

[28] Aa¨ron Van Den Oord, Sander Dieleman, and Ben- jamin Schrauwen. Transfer learning by supervised pretraining for audio-based music classification. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, 2014.

[29] Yusuf Yaslan and Zehra Cataltepe. Audio music genre classification using different classifiers and feature selection methods. In *18th ICPR 2006*, volume 2, pages 573–576. IEEE, 2006.