Abstract

In this paper we present an overview of Musical Genre Recognition (MGR) using Convolutional Neural Networks (CNNs). We discuss the background of MGR and CNNs, before going on to discuss and compare the use of spectrograms and raw audio as the input to these models. We also discuss deconvolution of the CNNs and auralisation of the resulting spectrograms, finding that CNNs are extracting meaningful features for the task of MGR.

We conclude by finding that though CNNs are capable of extracting musical features from raw audio, they perform significantly better when extracting features from spectrograms.

1. Introduction

Music Information Retrieval (MIR) is the process of extracting features from audio recordings of music. MIR is important in categorising music and helping recommender systems make more relevant recommendations. Since genre is one of the most important features of a musical piece Music Genre Recognition (MGR) is one of the most important topics in MIR.

It is becoming increasingly important with the advent of online streaming services such as Spotify, Tidal or YouTube to organise and provide recommendations on a large dataset of music. To manually label all these tracks would be impossible and so, to address this problem, automatic music genre recognition systems have been a popular area of research.

The process of MGR has traditionally involved 3 steps. Firstly, features are extracted from the raw audio, normally by hand. Secondly, the features believed to be important to genre classification were picked out. And finally, machine learning techniques were applied to the hand picked features, attempting to recognise the music's genre [1]. However, hand crafted features have disadvantages: these features may not generalise well, making it hard to apply the model to other classification tasks, and the success of the model relies heavily on the quality of the features extracted, not on the model itself.

Recently, off the back of their success in image classification [2] and speech recognition [3], researchers have been using Convolutional Neural Networks (CNNs) to perform MIR, including MGR. There have been two main approaches to this, the first is directly inspired by image classification and trains the CNN on spectrograms, which are visual representations of music. The second approach is to train the CNN directly on raw audio to create an end-to-end music genre recognition system.

In this paper we aim to show that, despite there being useful features in an end-to-end system, the best way to automatically perform MGR using CNNs is to train them on spectrograms. And so, in Section 2 we discuss the background of CNNs, MGR and available datasets. We then discuss the attempts to train CNNs to perform MGR using spectrograms in Section 3. In Section 4 we discuss the attempts to perform MGR on raw audio using CNNs followed by a discussion about deconvolving and auralising CNNs to learn more about their inner structure in Section 5. Finally, we close the paper with a conclusion of the topics discussed, followed by suggestions of future work.

2. Background

2.1 Music Genre Recognition

Genre is the main category that music dealers, archivers and online platforms use to organise their artists and songs. They have developed over time as a way of expressing the similarities between artists and their music and are therefore one of the most important features for performing music recommendation [4]. However, despite commonly using genres to categorise music, it has been shown that even humans can struggle in genre recognition, in some research achieving only a 76% accuracy [5]. Therefore the task of automatically recognising genres is an important and difficult challenge in which an AI system could potentially outperform humans.

2.2 Convolutional Neural Networks

A convolutional neural network is a neural network with constraints on the connections in certain layers, called convolutional layers. The constraints mean each unit in a convolutional layer only observes a small region of the input [6]. The kernel is shared across the feature map and so creates a pattern detector that achieves high activation when patterns appear in the input [6]. A CNNs structure can vary in a number of ways, such as the number of layers and the type activation functions, and so we discuss these further in following sections.

CNNs have found popularity in image recognition [2] and speech analysis [3]. One of their advantages is that the number of parameters can be reduced by exploiting the strong relationship between local pixels and the translation invariance of images. A similar technique can be applied to audio signals, where the correlation between local sounds is based on time rather than geometric position [7]. They perform feature extraction and classification where all parameters are learned together with back propagation [8] and are known to be particularly good at automatically extracting high level features from input data. However they aren't without their drawbacks, one of which is that they need a large amount of labelled data in order to learn [9].

2.3 Datasets

The main issue in providing a dataset of music is that copyright law restricts the unlicensed distribution of audio tracks. Therefore the majority of datasets usually consist of a clip of raw audio, around 30 seconds to 1 minute in length, alongside metadata such as artist, album and year of release, then a number of audio features such as tempo, timbre, key and genre.

Two popular datasets are the GTZan dataset which contains 1,000 labelled audio samples [10]. There is also the MagnaTagATune dataset, which consists of 25,000 audio samples [11]. However, the obvious problem with these datasets is that they contain a small number of samples and may not be sufficiently large for training a CNN. This problem is addressed by the Million Song Dataset (MSD) [11] which is the largest dataset by far. It is a dataset of precomputed features and metadata for a million songs that can be linked with the LastFM's tag database and with 7digital's audio samples database to obtain tags and 30 second clips for 99% of the songs [12].

3. Spectrograms

In the previous section we discussed MGR, the structure of CNNs and available datasets. We will now discuss and evaluate how researchers have used CNNs and spectrograms to automatically perform MGR.

3.1 Creating Spectrograms

There are several different types of spectrograms with the most common being a graph of two geometric dimensions, with time on the x-axis and frequency on the y-axis. The third axis is the amplitude of a frequency at a point in time and is represented by the intensity of colour at a point in the graph as shown in Fig 1.

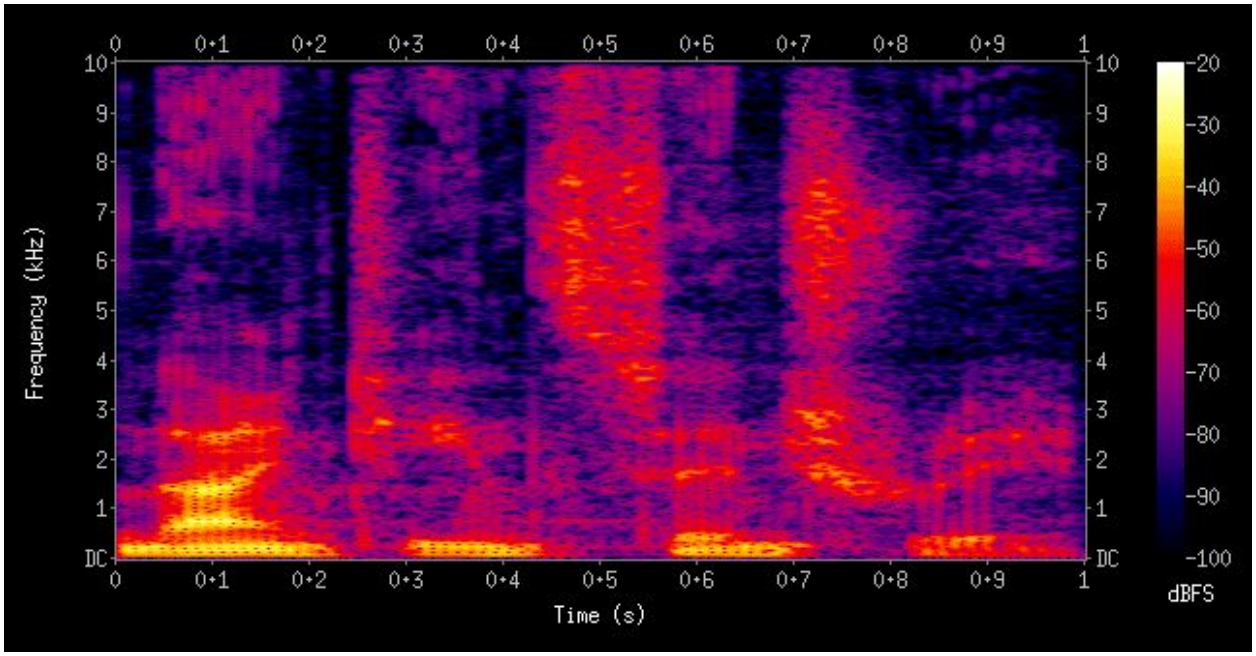

Figure 1.

Since extracting features from a spectrogram requires a large amount of processing time, audio samples of tracks are typically split into segments of a few seconds long, with each segment having an overlap of 50% [8]. Although this step is required to make training the model more efficient, it could potentially result in the loss of valuable information therefore making spectrograms an unsuitable data source for training CNNs in MGR. For example, there may be certain global features of a track, that correlate with genre, that are lost when the input is reduced to shorter clips. Researchers then apply mel-scaling to their audio samples by adjusting the frequency bands in the spectrogram to more accurately represent how humans perceive sound. Once audio clips have been rescaled they can be converted into spectrograms.

One of the main visual aspects of spectrograms is their texture [9] and it is therefore common to use a descriptor of the texture as the input to the CNN. This can be done a number of ways including Local Binary Patterns (LBP) [9] and Grey-Level Co-Occurence Matrices (GLCM) [8]. LBP have been used extensively in facial recognition [18] and involves transforming the image into an array of integers describing the texture of the image. GLCM is another popular method for describing the texture of an image and it works by encoding the occurence of different combinations of gray levels between pairs of pixels in an image. For their experiments, Nakashika et al. [8] adjusted their spectrograms to have only 16 grey levels to represent amplitude of a frequency at a point in time.

To address the issue that CNNs don't perform well on high resolution images the spectrograms are usually split again into several patches which are each individually fed into the CNN [9]. Though this helps with training the CNN it could exacerbate the loss of global features within the sample.

3.2 Architecture

A variety of models are used to extract genres from the spectrograms. Costa et al. [9] creates a CNN inspired by a model that has performed well on previous pattern recognition tasks. It contains repeating convolutional layers, followed by max-pooling layers. A max-pooling layer involves grouping units into small non-overlapping blocks. These blocks are passed onto the next layer and aggregated to form one unit whose activation is the maximum activation found in the units that formed the block [7]. The genre prediction was calculated by the final layer, which was a fully connected layer.

A similar architecture is adopted by Nakashika et al. [8] where the CNN is designed to take a number of different GLCM maps as input. This input layer is followed by convolutional layers, with non-linear activation functions distributed throughout. The final layer outputs probabilities for each of the 10 genres they are trying to recognise. The genre with the highest probability is therefore chosen as the prediction.

For Zhang et al. [1] the CNN contains 8 layers as well as an input and a softmax output layer. The CNN alternates between convolutional layers and max-pooling operations. This structure means the CNN can review non-overlapping regions of the spectrograms and return the maximum value which will help mitigate potential loss of global features. This also means that the CNN has a form of translation invariance. The final 3 layers are dense with the final layer providing probabilities for the 10 genres they are attempting to predict.

3.3 Performance

In general, the performance of CNNs on spectrograms has been promising. The CNN created by Cost et al. [9], when evaluated alone, achieved an accuracy of 83% and when combined with current recognition models, it performed better than the state of the art by achieving a 92%

accuracy. Similarly promising results were also achieved by Nakashika et al. [8] with an accuracy of 72% and by Zhang et al. [1] who's best model achieved an accuracy of 87.4%. Zhang et al. [1] also suggested that improvements could be made by combining maximum and average pooling layers and by adding connections that skip a number of layers as is used in residual learning. It was found that more layers were better as long as the amount of data was sufficient to train the model. This is where the MSD would provide the best results.

3.4 Evaluation

We can see that CNNs trained on spectrograms are achieving high accuracies and in the best cases, even beating humans. Therefore the process of converting raw audio into spectrograms is most likely not resulting in the loss of important musical features, and the CNNs are adapting well to the task of feature extraction on spectrograms. Therefore we can see that spectrograms are a viable source with which to train CNNs in MGR.

However there are a number of criticisms of the use of spectrograms. Local receptive fields, used in CNNs, mean that the strongest correlations are learned in local regions of an image [13]. However, this idea does not apply to spectrograms where there may be musical features that span the entire spectrogram that are important to genre classification. For example, two frequencies occurring at the same time, but on the opposite ends of the spectrum (opposite ends of the y-axis) will not be local to each other in the spectrogram. It is also claimed by Dieleman et al. [14], that the use of spectrograms is still a form of manual feature extraction which requires prior knowledge and is therefore not desirable. Therefore in the next section we will discuss the pursuit to build an end-to-end system for recognising music genre.

4. End-to-End Approach

When CNNs are used in image recognition, they are trained on the raw pixels of an image. However in the previous section we discussed a process of converting raw audio into spectrograms to then train CNNs. One of the disadvantages of converting audio tracks to spectrograms is that it requires prior knowledge and expertise in the area [14]. Whereas in an end-to-end system, expertise is only required for tuning the models hyperparameters. Learning features directly from audio could also result in a more accurate performance since features are automatically learned for the task [14]. In this section we therefore discuss an end-to-end approach, also used in speech recognition [15], to perform MGR on raw audio.

4.1 Architecture

Li et al. [6] used a CNN originally designed for image information retrieval which saved them 66.8% on computational requirements. There were 5 layers, including the input and output, and the CNN was trained by using stochastic gradient descent. Once a classification was given for

each clip (in the GTZan dataset), the results were then aggregated using a majority voting process [6].

Dieleman et al. [14] made direct comparisons between CNNs trained on raw audio and on mel-scale spectrograms. The architecture of the CNN for spectrograms was a network containing 6 layers that alternated between convolutional layers and max-pooling layers. They used Rectified Linear Units in all layers, apart from the final layer where they used a sigmoidal function. To adapt this CNN to work with raw audio, another convolutional layer was added before the input which would perform a strided convolution.

An unsupervised approach can also be used, such as where the MSD was used to pretrain the CNN in music feature extraction [7]. Once pretrained the CNN was trained in a supervised manner, where only a small dataset of relevant samples were provided. The structure of the CNN consisted of alternating convolutional layers that were then max-pooled, with a pool size of 4 [7]. The final layer performed logistic regression, providing a probability of the genre of each clip.

4.2 Performance

The models we examined got varying results, however the general outcome was that they performed worse on raw audio than they did on spectrograms. When reviewing how well their model would generalize Chan et al. [6] found they had an accuracy of below 30% on the test set, therefore showing that the model was not viable for genre recognition. Upon further study they believe that the poor generalization was due to the wide variety of musical data and that their model became sensitive to variations in timbre, tempo and key which are not relevant to genre. Similar issues were encountered by Dieleman et al. [7] where they found that pre-training made their model perform worse on genre recognition where it only achieved accuracies of 30%.

4.3 Evaluation

However, despite the poor test error, the experiments did show that the CNNs could perform some form of automatic feature extraction on raw audio [6]. To improve their accuracy it was suggested that some manipulations could be performed on the input so that the models are not affected by variations in timbre, tempo or key, for example a mel-frequency spectrum as was discussed in Section 3 [6].

Dieleman et al. [14] directly compared CNNs trained on spectrograms and on raw audio and also found that spectrograms consistently outperformed raw audio in MGR tasks. They believe the sizeable gap in performance was due to the CNNs architectures are already well established for feature extraction from images in general, but not well suited to feature extraction on raw audio. Despite the disappointing results on raw audio, they did find that the CNNs were automatically discovering some features in the audio signals, such as frequency

decompositions and phase or translation invariant features. In [14] they found that the features being extracted in the first layer of the CNN were all frequency based features, and that they were predominantly extracted from the lower frequencies. This is in keeping with our understanding of the melodic structure of music, where the melody mostly resides in the lower frequencies. Upon further inspection it was seen that the frequency range being extracted was similar to the mel-scale [14].

Though we can see here that CNNs trained on raw audio can automatically extract features, they are not as accurate as spectrograms in recognising a music's genre. This suggests that CNNs trained on raw audio are not a viable model for performing MGR.

5. Auralisation of Convolutional Neural Networks

Though CNNs have been achieving promising results, one disadvantage is that they can be somewhat of a black box. Therefore researchers have attempted to deconvolve CNNs hoping to glean insights on their inner workings. One of the advantages of deconvolving a CNN is it helps us fit the hyperparameters (such as number of layers) of the model by allowing us to inspect the learnt weights [16]. Another advantage is that it can give us insight into the task that is being modelled. However, deconvolving a CNN to produce a spectrogram doesn't give us much insight into how the model is working. This is where researchers begin using auralisation, which is the process of creating audio signals from the spectrograms [16].

Since the first layer weights take the raw input, with no max-pooling, we can inspect the weights independent of input [16]. Choi et al. [17] found that the first layer of the CNN was a crude onset note detector, which is in line with image recognition CNNs where the first layers are edge detectors. This is since, in a spectrogram, a note onset is shown by a vertical line (or edge) on the spectrogram. Horizontal lines express harmonic features as well as percussive instruments, however diagonal lines are uncommon in spectrograms and refer to frequency modulation [16]. Choi et al. [16] found that their second layer started to capture more frequency based features such as bass notes and harmonic components. The third layer was similar to the second layer but started to distinguish between different instruments based on their harmonic structure and the sustain and release of their notes. For example, the CNN was learning the difference between voices and piano, and percussion instruments such as snare drums [16]. As we move up through the layers, the features extracted are becoming more high level and therefore harder to distinguish, however in layer 5 some features for genre classification are clear. There was one feature that was only activated by hip hop tracks and others that were activated when melodies and percussion were highly synchronised. One interesting finding by Choi et al. [16] was that for a CNN trained to perform MGR, the music's key had very little effect on how the CNN classified genre which is in keeping with the fact that genre is often independent of key.

The above insights therefore show us that CNNs trained on spectrograms are extracting relevant and meaningful features for the task of genre recognition. This shows that representing

audio as spectrograms is not resulting in the loss of important information nor hindering the performance of CNNs in MGR.

6. Conclusion

In this paper we are aiming to show that the best technique for performing MGR using CNNs is to represent the input as a spectrogram. We described models trained on spectrograms as well as raw audio and also discussed the process of auralising CNNs, which showed us that the CNNs were extracting relevant features from the spectrograms for the task of MGR.

CNNs trained on raw audio were pursued as the researcher's belief was that spectrograms were a form of manual feature extraction and so an end-to-end system would be more desirable and potentially perform better. However, despite showing the ability to extract features from the audio and even perform a crude mel-scaling, these models performed significantly worse than the spectrograms. Though training CNNs with spectrograms does require a degree of expertise for initial feature extraction, the performances achieved outweighs these negatives. Once the method of initial feature extraction has been established, the process will become automated. And once the models have been trained they consistently outperform raw audio input by achieving high accuracies such as 92%. Therefore the best way to perform MGR using CNNs is by using spectrograms as input.

To advance performance, future work could make use of larger datasets, such as the MSD, in order to better train their CNNs, since it has been shown that CNNs perform better when they have access to more training data. One reason that the spectrograms performed significantly better is due to the larger amount of previous research on CNNs and image recognition. Therefore if future research concentrated on developing CNNs trained on raw audio we may see a large increase in accuracy using this method.

Bibliography

[1] Zhang, W., Lei, W., Xu, X., & Xing, X. (2016). Improved Music Genre Classification with Convolutional Neural Networks. *Interspeech, 2016*, 3304–3308.
[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9.
[3] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (July 2015), 4277–4280.
[4] Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, *23*(2), 133–141.
[5] Lippens, S., Martens, J.-P., & De Mulder, T. (2004). A comparison of human and automatic musical genre classification. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, *4*, iv-233-iv-236 vol.4.
[6] Chan, A. B., & Li, T. L. H. (2010). Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network.
[7] Dieleman, S., Brakel, P., & Schrauwen, B. (2011). Audio-based Music Classification with a Pretrained Convolutional Network. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir), 669–674.
[8] Nakashika, T., Garcia, C., Takiguchi, T., & Lyon, I. De. (2012). Local-feature-map Integration Using Convolutional Neural Networks for Music Genre Classification. *Interspeech*, 1–4.
[9] Costa, Y. M. G., Oliveira, L. S., & Silla, C. N. (2017). An evaluation of Convolutional Neural Networks for music classification using spectrograms. *Applied Soft Computing*, *52*, 28–38.
[10] Sturm, B. L. (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use, (11), 1–29.
[11] McFee, B., Bertin-Mahieux, T., Ellis, D. P. W., & Lanckriet, G. R. G. (2012). The million song dataset challenge. *Proceedings of the 21st International Conference Companion on World Wide Web - WWW '12 Companion*, 909.
[12] Van Den Oord, A., Dieleman, S., & Schrauwen, B. (n.d.). Deep content-based music recommendation, 1–9.
[13] Humphrey, E. J., Bello, J. P., & Lecun, Y. (2013). Feature learning and deep architectures: New directions for music informatics. *Journal of Intelligent Information Systems*, *41*(3), 461–481.
[14] Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 6964–6968.
[15] Lee, H., Pham, P. T., Largman, Y., & Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. *Nips*, *9*, 1096–1104.
[16] Choi, K., Fazekas, G., & Sandler, M. (2016). Explaining Deep Convolutional Neural Networks on Music Classification.
[17] Choi, K., Fazekas, G., Sandler, M., & Kim, J. (2015). Auralisation of Deep Convolutional Neural Networks : Listening To Learned Features, 4–5.

[18] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Inlligence*, *28*(12), 2037–2041.