A Deep Learning Approach for Mapping Music Genres

Sharaj Panwar, Arun Das, Mehdi Roopaei, Paul Rad
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, Texas, USA

Abstract*—Deep feature learning methods have been aggressively applied in the field of music tagging retrieval. Genre categorization, mood classification, and chord detection are the most common tags from local spectral to temporal structure. Convolutional Neural networks (CNNs) using kernels extract the local features that are in different levels of hierarchy while Recurrent Neural Networks (RNNs) discover the global features to understand the temporal context. CRNN architectures as a powerful music tagging utilize the benefits of the both CNN and RNN structures. In this article a CRNN structure on MagnaTagATune dataset is proposed. The AUC-ROC index for the proposed architecture is 0.893 which shows its superiority rather than traditional structures on the same database. The merging mechanism to obtain 50 tags from the whole 188 existing tags of this dataset and simple CRNN architecture designed for tag discovering are the main contribution of this paper.

Keywords*—Deep Learning; Cloud Computing; Recurrent Neural Network; Music Tagging; MagnaTagATune, Music Decomposition, Music Genre Recognition, Tag Retrieval

I. INTRODUCTION

CNNs have been used extensively in solving various complicated machine learning problems such as sentiment analysis, feature extraction, genre classification and prediction [1,2,3,4,29]. A hybrid models of CNNs and RNNs have been recently applied for temporal data like audio signals and word sequencing [5]. Convolution Recurrent Neural Networks (CRNN's) are complex neural networks formed by combining Convolutional CNN and RNN networks [6]. CRNN architecture as a modified model of CNN with a RNN structure placed over it. This architechture has the capability to be as a robust structure to ectract local feature using CNN layers and temporal summation by RNN networks.

CNN's have been very popular in music recognition in divers aspects such as automatic tagging [7], hybrid music recommender [8] and feature learning [9]. The key elements for a CNN network is: type of input signal [7, 10], learning rate, activation function [5-11], batches [12] and architecture [6,7]. Mel-spectogram is the prefered input type for music information retrieval [7]. Mel-spectrograms consists of widespread features for tagging, boundary and onset detection, latent feature learning and it has been proved that Mel-scale is similar to the human auditory system [13,30].To achieve mel-spectogam signal, STFT (short time Fourier transform), and Log-amplitude spectrogram are required as pre-processing phase [7].

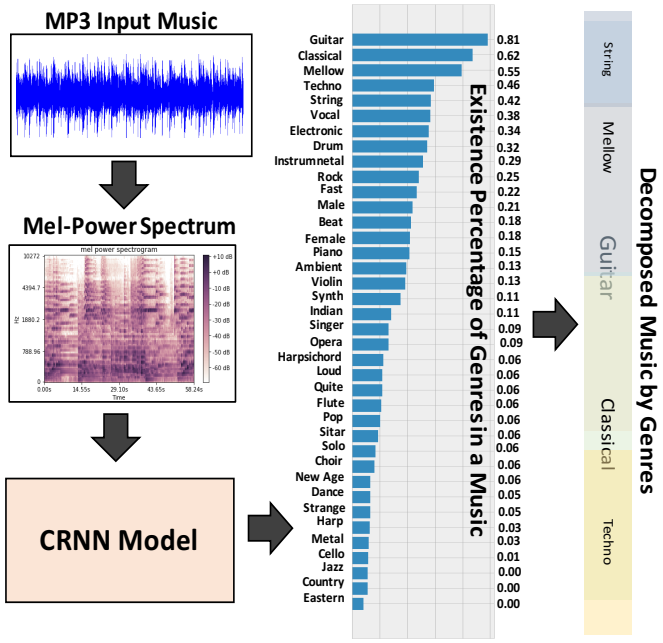

Fig.1- Deep Music-to-Genre Decomposition Using Deep Learning

Music feature learning with deep networks was improved with ReLu as activation funcytion [5]. Later this function is replaced with ELU (Exponential Linear Unit) to get fast and accurate learning [8]. Deep network training further accelerated by reducing internal covariate shift using batch normalization [12]. Recurrent neural networks also experienced significant improvement when gated recurrent neural network are applied [14]. Gated RNN's have gating units which limit the flow of information through them, allowing to capture critical information from different time scales [14].

A CRNN architecture for tag recognition in music application is proposed in the current article. The dataset utilize in this paper is MagnaTagATune which has 188 different tags and raw music in the mp3 format. To have better visibility on tag discovering the tags are reduced to 50 with merging similar features such as genre and instrumnet. The mp3 format is tranfer to mel-spectrogram signal in pre-processing stage which has rich information around the music. The prepared information as a audio metrix is fed to the proposed CRNN architecture which is designed to understanf local and temporal features with inside CNN and RNN networks respectively. The rest of the paper is organized as follow: Section II explains about the proposed approch, experimnet and results are discussed in section III and finally the conclusion is expressed in section 4.

II. PROPOSED DESIGN AND APPROACH

A. Dataset

MagnaTagATune dataset is a widely-used dataset for different applications such as genre classification, auto tagging etc [15]. The dataset is highly unbalanced with some tags having most of the data. The dataset has 188 different features in categories such as music genre's, instrument used, gender of singer, mood etc. The dataset is skewed with some tags having much more content associated with them than the rest. For our problem, first we took the top 50 tags after merging together very similar tags. These 50 tags are a subset of the dataset with categories including the ones mentioned above. The merging mechanism to the rest of 50 tags is chosen based on the similarity in an specific category like as genre or instrument which are illustrated in the figure 2.

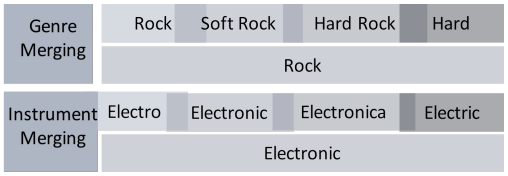

Fig.2. Merging of similar tags

The MagnaTagATune dataset was split into 19773 training, 4566 testing and 1521 validation datasets.

B. Preprocessing

Log amplitude Mel-Spectrogram is applied as an input to the proposed model. An mp3 sound audio with 29.14 second is fed for preprocessing phase. This audio signal is quantizied with 12k samples per second. Fast Fourier Transform (FFT) is performed over a frame with size of 512 which is associated witht a short period of time therefore, this process called as Short Time Fourier transform (STFT), [16].

A hop size is defined as an overlapping area between two frames which is considered as 256. It means each sample is analyzed twice in consecutive FFT frames. Spectrogram is calculated by mapping amplitude with frequency for each frame after FFT and merging it based on FFT frame id and hop size.

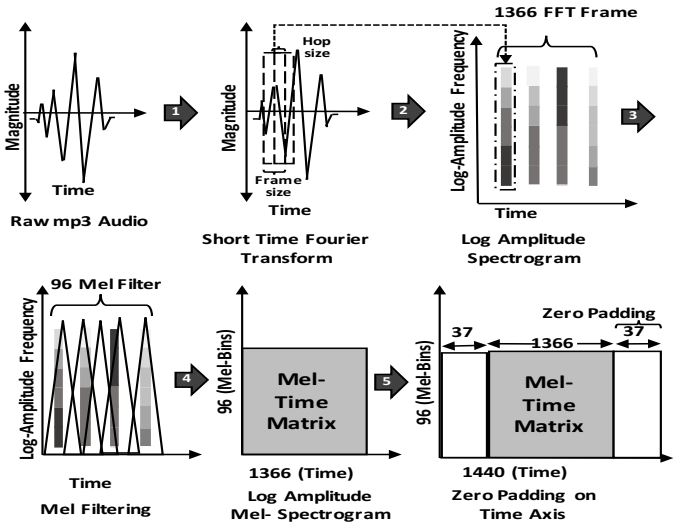

Fig. 3. Preprocessing- mp3-to-melspectrum

This spectrogram represents information of amplitude-frequency variation of 29.12 second audio clip with time. The magnitude of the spectrogram is made logarithmic to obtain log amplitude spectrogram. Since human perception and recognition to sound is based on certain mel frequencies, 96 numbers of mel filters are placed non-uniformly with less felters at low frequencies and more filters are cosidered at high frequencies to obtain log amplitude mel spectrogram. The parameters used for preprocessing stage are summarized in Table 1.

Table 1. Pre-processing parameter s

Time Frame	29.12s
Sampling rate/s	12000
FFT Frame size	512
Hop Size	256
No. of Mel filters	96
Mel-Time matrix size	1×96×1366
Frames added in Zero Padding	74
Mel-Time Matrix after Zero Padding	1×96×1400

This preprocessing is performed by Librosa [17]. The output of the preprocessing step is an array with the size of $1\times96\times1366$ where 96 represents number of mel bands also known as mel bins and 1366 are the total number of frames used for FFT. Zero padding is the last step of preparocessing to make better frequency resulution of input signal by adding 74 frames on time axis to infuse more data with same information. The block diagram of the whole pre-processing steps are provided in the figure 3.

C. Model Architecture

The CRNN architechter is proposed for Music event recognition.The details for the structure of proposed network are: 4 convolutaional and 2 Recurrent neural netwok layers. RNN layers with gated recurrent unit (GRU) are used for temporal pattern summation over features learned in CNNs. It is assumed that RNNs are best fit in temporal summation and aggregating the features.

Log-amplitude mel-spectrogram input after applying zeropading and with the size of $(1\times96\times1400)$ is infused to first 2d convolution layer of size $(63\times3\times3)$. Extracted features are max pooled by max pooling strides of size (2×2) which caused time dimension to reduce to 720 and frequency to 48. Hence input for second convolution layer becomes $(63\times48\times720)$. Second convolution layer with size $(128\times3\times3)$ followed by max pooling size (3×3) gives an output size $(28\times16\times240)$. An output of size $(128\times4\times60)$ from third convolution layer is given as an input to fourth and last convolution layer of size $(128\times3\times3)$. These convolution operations give an output size of $(128\times1\times15)$. This output is further processed to make it compatible to RNN network. Permute operation is used to change dimension ordering from [(channel) × (mel-bins) × (time)] to [(time) × (channel) × (mel-bin)]and mold output size to $(15 \times 128 \times 1)$. Reshaping is performed to reduce dimensionality as it merges time and frequency against channel axis. This results the input size to 15×128 (mel-bins×time) × (channel) to the recurrent neural network. The block diagram of the proposed architechture are depicted in the figure 4.

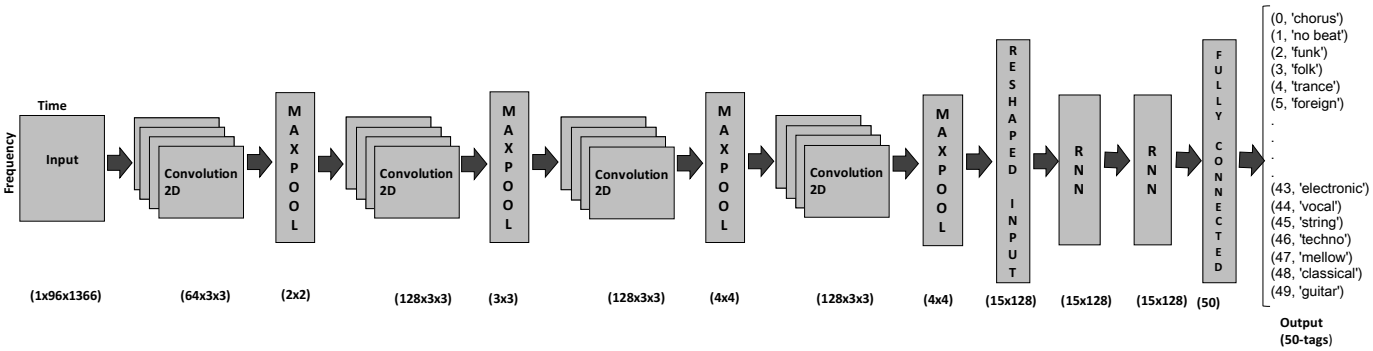

Fig. 4. CRNN architecture

The specification of the CRNN models for CNN and RNN layers are summarized in Table 2.

Table 2- CRNN Specifications

Input shape	1×96×1440
1st convolution layer	64×3×3
Max pooling	2×2
Input to 2nd convolution	64×48×720
2nd convolution layer	128×3×3
Max pooling	3×3
Input to 3rd convolution	128×16×240
3rd convolution layer	128×3×3
Max pooling	4×4
Input to 4th convolution	128×4×60
4th convolution layer	128×3×3
Max pooling	4×4
CNN output	128×1×15
Reshaped input to RNN	15×128)
RNN	GRU(32)
RNN	GRU(32)
FC layer	50
Model Output	50 different Tags

III. Experiment and Results

The experiment is performed on Chameleon cloud testbed [18]. Each bare-metal cloud servers used in this project has Dual-Xeon processors with NVIDIA M40 GPU's [19]. The compute nodes are connected to each other through high-speed interconnects and have access to petabytes of object storage. The dataset is stored in the object storage and is fetched in time of training.

The MagnaTagATune dataset was first preprocessed and shuffled before creating the training, test and validation splits. In the preprocessing stage, all mp3 tracks are converted into their respective mel-spectrograms. Librosa is a library for sound processing is used for preprocessing a raw audio mp3 input to get log-amplitude mel-spectrograms, [17]. An array of these spectrograms is then fed to the proposed CRNN model [6]. Keras is another library provides high-level building blocks to handle low-level operations such as tensor multiples, convolutions [20].

The model predicts top 50 tags from the MagnaTagATune dataset. The tags have different genre of music, mood and other information. The aim of the model is to correctly predict whether or not a specific tag is there in the music file. Randomly shuffled dataset is used for experiment as the dataset is skewed. The model is trained for more than 60000 iterations with a batch size of 32, completing a total of 100 epochs.

The distribution of each layer's input changes during training, as the parameters of the previous layer changes. This slows down the learning by requiring lower learning rates and makes it difficult to train the network. This phenomenon is called internal covariant shift. Batch normalization [12] is performed after each convolutional layer and before activation function to reduce internal covariant shift.

Exponential linear unit (ELU) function is used as an activation function after each convolutional layer [11]. ELU speeds up learning in deep neural networks and leads to higher classification accuracies. Being negative valued, Elu pushes mean unit activation closer to zero like batch normalization but with lower computation complexity. The exponential unit (ELU) with $\alpha > 0$ is

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases}$$

ADAM (Adaptive moment estimation) optimizer is used for learning rate control [21]. The algorithm is a gradient-based optimization of an objective functions, based on adaptive estimates of lower-order moments. ADAM is well designed for problems that are large in terms of data and/or parameters.

Binary cross entropy is used for loss calculation between predictions and targets [22]. The typical loss function that we use in cross entropy is computed by taking the average of all cross-entropies in the sample. The loss function for N samples is given by:

$$L(w) = -\frac{1}{N}[y_n \, log\hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)]$$

Where N is the size of test set, y_n is the actual labeled value and $\hat{y}_n$ is the predicted value. Since the tags in dataset are binary vectors, binary cross-entropy is preferred as loss function. Early stopping is done by interrupting our training if we don't find any considerable improvement in learning, figure 5.

A threshold value of 0.5 is selected for probability value of predicted tags. These probabilities are then converted to binary values based on the threshold; '1' for probability greater than 0.5 and '0' for probability less than 0.5. Binary value 1 is considered

as positive and 0 as negative. True positive and False positives prediction can then be computed by comparing with existing dataset tags.

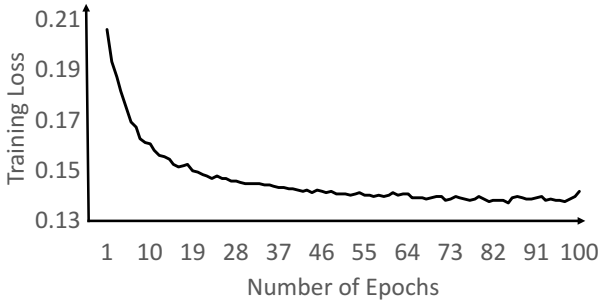

Fig.5. Training Loss-Epoch Graph

The accuracy of 94% on the test dataset with a loss of the 0.15 are achieved. However, in MagnaTagATune dataset most of the tags are false (0) for most of the clips, figure 6, which makes accuracy calculation inappropriate as a measure [7]. Therefore, Area Under an ROC (Receiver Operating Characteristic) Curve is used for determine the accuracy [23,24]. 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.

Predicted Output	Actual labels	T-F Predicted	T-F Actual
techno : 0.818	**'techno'**	**techno : T**	**techno : T**
fast : 0.624	**'beat'**	fast : T	fast : F
electro : 0.453	**'dance'**	electro : F	electro : F
beat : 0.434	**'synth'**	**beat : F**	**beat : T**
dance : 0.386	**'piano'**	**dance : F**	**dance : T**
drum : 0.176	**'violin'**	drum : F	drum : F
synth : 0.173		**synth : F**	**synth : T**
trance : 0.171		trance : F	trance : F
loud : 0.156		loud : F	loud : F
no singer : 0.113		no singer: F	no singer: F
rock : 0.060		rock : F	rock : F
pop : 0.024		pop : F	pop : F
modern : 0.020		modern : F	modern : F
bass : 0.017		bass : F	bass : F
strange : 0.016		strange : F	strange : F
vocal : 0.013		vocal : F	vocal : F
male : 0.008		male : F	male : F
ambient : 0.008		ambient : F	ambient : F
heavy : 0.006		heavy : F	heavy : F
guitar : 0.006		guitar : F	guitar : F
new age : 0.005		new age : F	new age : F
no piano : 0.004		no piano : F	no piano : F
female : 0.004		female : F	female : F
slow : 0.004		slow : F	slow : F
hard : 0.004		hard : F	hard : F
piano : 0.003		**piano : F**	**piano : T**
india : 0.003		india : F	india : F
jazz : 0.002		jazz : F	jazz : F
singer : 0.001		singer : F	singer : F
no beat : 0.001		no beat : F	no beat : F
violin : 0.001		**violin : F**	**violin : T**
string : 0.001		string : F	string : F
sitar : 0.001		sitar : F	sitar : F
quiet : 0.000		quiet : F	quiet : F
solo : 0.000		solo : F	solo : F
foreign : 0.000		foreign : F	foreign : F
eastern : 0.000		eastern : F	eastern : F
classical : 0.000		classical :F	classical :F
flute : 0.000		flute : F	flute : F
soft : 0.000		soft : F	soft : F
country : 0.000		country : F	country : F
folk : 0.000		folk : F	folk : F
chant : 0.000		chant : F	chant : F
opera : 0.000		opera : F	opera : F
choir : 0.000		choir : F	choir : F
harpsichord : 0.000		Harpsichord:F	Harpsichord:F
orchestra : 0.000		orchestra: F	orchestra: F
baroque : 0.000		baroque : F	baroque : F
harp : 0.000		harp : F	harp : F
cello : 0.000		cello : F	cello : F

Accuracy=[True Positive + True Negative]/ALL=[1+43]/50=0.88

Fig. 6. True-False for an actual and predicted sample

An ROC (Receiver Operating Characteristics) curve is a two-dimensional depiction of classifier performance [24]. ROC curve is plotted between true positive (sensitivity) and false positive rates (1-specificity).

$$\text{ROC} = \frac{\text{TP}/(\text{TP}+\text{FN})}{\text{FP}/(\text{FP}+\text{TN})} = \frac{\text{sensitivity}}{1-\text{specificity}}$$

The area under the ROC curve called AUC-ROC which demonstrates the performance of the model for the given training set. The effective range of AUC-ROC is in the interval between 0.5 and 1.0. If AUC-ROC value falls way below 0.5, it indicates a conflict between the predictions and the actual labels. This means that the network is not learning properly and that the problem has been set up wrongly. This measure is calculated for each epoch and the resulted for the proposed model is plotted in the figure 7.

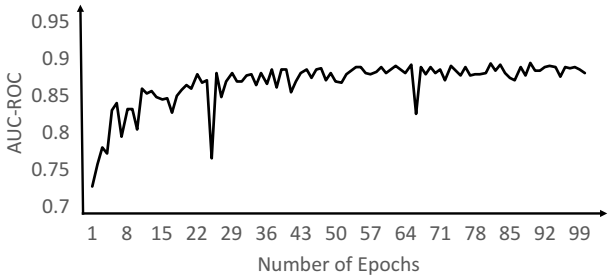

Fig. 7. AUC-ROC Curve

The AUC-ROC for the proposed model is 0.893 which is compeared with the exsiting models on the same dataset, Table 4.

Table 4. AUC-ROC-The current model and previous studies.

Methods	AUC-ROC
2017, CRNN, Our Proposed Method	0.893
2016, FCN 4 [7]	0.894
2015, Bag of features and RBM [25]	0.888
2014 ID convolutions [10]	0.882
2014, Transferred learning [26]	0.88
2012, Multi-scale approach [27]	0.898
2011, pooling MFCC [28]	0.861

The performance of the proposed model is comparable with the fully connected method represented in [3], however our model has more effective in temporal tags such as moods (happy, sad, and etc.) and with less number of parameters and more simple and practical structure.

IV. Conclusion

Music information retrieval could be a help to understand the context in audio signals and categorize them based on various feature such as genre, instrumnet, mood and etc. This classification make the audience to have better visibilty in selecting their favorate music. More accurate to retrieval the existed tags in a music causes more chance to attract larger amount of audience with enough satisfication to select their music intelligently.Nowadays precice tagging retrieval mechanisms are provided by deep feature extrating and learning methods. They attemp to understand almost all information

inside a music from local to temporal features. Comnination of CNN and RNN networks is an architecture which has potential to explore the required information for tag classification. In this paper, a CRNN network is propsed for tag recognition on MagnaTagATune database. The entire dataset has 188 tages however just 50 of them are chosen from the most population and then rest of them are merged into those 50 tags. The AUC-ROC measured for the proposed architectur was 0.893 which shows the supriority of the simple struchture with a very less amonut of parameters in comparison to traditional methods on the same dataset with different approches.

REFERENCE

[1] Mehdi Roopaei, Paul Rad, Mo Jamshidi, Deep Learning Control for Complex and Large Scale Cloud Systems, Intelligent Automation and Soft Computing (AUTOSOFT), DOI: 10.1080/10798587.2017.1329245.3)

[2] Li, Tom LH, Antoni B. Chan, and A. Chun. "Automatic musical pattern feature extraction using convolutional neural network." Proc. Int. Conf. Data Mining and Applications. 2010.

[3] Nakashika, Toru, Christophe Garcia, and Tetsuya Takiguchi. "Local-feature-map Integration Using Convolutional Neural Networks for Music Genre Classification." Interspeech. 2012.

[4] Sigtia, Siddharth, and Simon Dixon. "Improved music feature learning with deep neural networks." Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014.

[5] Lai, Siwei, et al. "Recurrent Convolutional Neural Networks for Text Classification." AAAI. Vol. 333. 2015.

[6] Choi, Keunwoo, et al. "Convolutional Recurrent Neural Networks for Music Classification." arXiv preprint arXiv:1609.04243 (2016).

[7] Choi, Keunwoo, George Fazekas, and Mark Sandler. "Automatic tagging using deep convolutional neural networks." arXiv preprint arXiv:1606.00298 (2016).

[8] Chiliguano, Paulo, and Gyorgy Fazekas. "Hybrid music recommender using content-based and social information." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.

[9] Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).

[10] Dieleman, Sander, and Benjamin Schrauwen. "End-to-end learning for music audio." Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014.

[11] Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289 (2015).

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

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

[14] Tang, Duyu, Bing Qin, and Ting Liu. "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification." EMNLP. 2015.

[15] Law, Edith, et al. "Evaluation of Algorithms Using Games: The Case of Music Tagging." ISMIR. 2009.

[16] Welch, Peter. "The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms." IEEE Transactions on audio and electroacoustics 15.2 (1967): 70-73

[17] McFee, Brian, et al. "librosa: Audio and music signal analysis in python." Proceedings of the 14th python in science conference. 2015

[18] Chameleon Cloud: A configurable experimental environment for large-scale cloud research. Chameleoncloud.org. Retrieved, from https://www.chameleoncloud.org/. 2017.

[19] Lindholm, Erik, et al. "NVIDIA Tesla: A unified graphics and computing architecture." IEEE micro 28.2 (2008).

[20] Chollet, François, “Keras: Deep Learning library for Python: Convnets, recurrent neural networks, and more, Runs on Theano or TensorFlow.” Github.com. Retrieved, from http://github.com/fchollet/keras. 2017

[21] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[22] Buja, Andreas, Werner Stuetzle, and Yi Shen. "Loss functions for binary class probability estimation and classification: Structure and applications." Working draft, November (2005).

[23] Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.

[24] Bradley, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." Pattern recognition 30.7 (1997): 1145-1159.

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

[26] Aa¨ron Van Den Oord, Sander Dieleman, and Ben- jamin Schrauwen. "Transfer learning by supervised pre-training 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.

[27] Sander Dieleman and Benjamin Schrauwen. "Multi- scale approaches to music audio feature learning". In Proceedings of the 14th International Society for Mu- sic Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013.

[28] 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 Soci- ety for Music Information Retrieval Conference, IS- MIR 2011, Miami, Florida, USA, October 24-28, 2011, pages 729–734, 2011.

[29] R. Polishetty, M. Roopaei and P. Rad, "A Next-Generation Secure Cloud-Based Deep Learning License Plate Recognition for Smart Cities," 2016 15th IEEE-ICMLA, Anaheim, CA, 2016, pp. 286-293. doi: 10.1109/ICMLA.2016.00542)

[30] 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.