Representation Learning for Structural Music Similarity Measurements

Introduction

The purpose of the project is to determine whether learned audio features can perform better than engineered features when used as part of a structural music similarity measurement system. The performance of the features will be measured against widely used engineered audio features using the system proposed in [1] and [2].

The products that will be generated are neural network models and learned features which accomplish this purpose. The beneficiaries of the work comprise three groups who benefit from an improved ability to identify structural similarity in music:

- Members of the Music Information Retrieval (MIR) community who are interested in furthering the goals of MIR, one of these being measuring structural music similarity.
- Music copyright owners such as artists and record labels, who stand to gain financially from increased protection of their intellectual copyright.
- Users of online music distribution services who will benefit from better content-based recommendations.

The question posed for the research project is as follows:

> Can representation learning improve the performance of structural audio similarity measurements in comparison with engineered features?

Critical Context

The Need for Music Similarity Systems

Piracy and File Sharing

Efficient audio compression algorithms in combination with faster home internet speeds and increased numbers of personal devices have enabled more widespread piracy of music [3]. Many popular peer-to-peer file-sharing services such as Napster and Pirate Bay have now fallen foul of copyright law. However, online sharing of music still constitutes a large loss of revenue to the music industry [4]; one estimate is that "28.2% if the population participates in illegal file sharing" [5]. The ability to recognise music which has been slightly modified to evade piracy detection measures (for example by adding random low-level noise) would be a valuable tool in reducing lost revenues.

Broadcasting Royalties

Every time a song is broadcast on the radio, television or online, the broadcaster has a duty to provide a royalty payment to the copyright owner. Although most broadcasters and large online providers such as YouTube have agreements with performing rights organisations, an increase in the number of online broadcasters has the potential to overwhelm the organisations responsible for collecting these royalties and revenue may be lost.

Sampling and Accreditation

A number of genres of music rely heavily on sampling from other songs. This has increased with the advent of samplers and computer-based music programs. Other issues arise when the line between emulation and copying is not clear. A recent high-profile case involved a $7.4 million payment from to the family of Marvin Gaye when it was found that they had his work had been copied. Accurate structural music similarity systems would be useful for detecting sampling and copying by artists.

Representation Learning for Structural Music Similarity Measurements

Music Information Retrieval Systems

The field of MIR system research has myriad applications, including genre, composer and artist classification, and extraction of other metadata such as key, meter, tempo and instrumentation which can aid in processes such as automatic transcription, cover song identification and compression. One of the applications with the highest potential commercial value is music recommendation systems.

Recommender Systems

Probably the most famous recommender system problem was posed in 2006 by Netflix, who offered a prize of $1M to anyone who could increase the quality of their user recommendations by 10% [6]. Such latent factors models have also been applied in the domain of music-based recommendation [7]. However, the collaborative filtering approach to recommendation suffers from a number of drawbacks, such as the "cold-start" problem. For music recommendation systems, such as those used at online music distributors like Google Play, iTunes and Spotify, the cold-start problem occurs when an artist releases a new song and it has not yet been rated by any users. While tracks by popular artists are likely to be listened to anyway, this is not the case for new or obscure artists who release new tracks, due to the long-tail distribution in popularity of songs [8]. This is where content-based recommendation systems, using audio similarity measures are most useful.

Feature Extraction

Engineered Features

Audio features are the basic building block for measuring similarity between two pieces of music. Digital audio in its uncompressed format, consists of sampled or synthesised time histories of acoustic pressure waves which, when reproduced via a physical transducer such as a loudspeaker or headphones, reproduce the audio as sound. Once the sound enters the human ear, it undergoes complex processing by the human auditory system. Based on physical experiments on other mammalian subjects, and human perceptual tests, the details of this system are quite well understood. Using this understanding, a number of audio features have been developed, which are useful for analysis and representation of musical audio.

Spectrograms

One of the most basic features is the spectrogram. This feature is obtained by performing a Fourier Transform $\mathcal{F}$ on a short window x_t of the full audio time history x_T so that it is represented in the frequency domain (a Short Term Fourier Transform or STFT), and squaring the magnitude of the result. The frequency resolution of the spectrogram is determined by the duration of the time window used as input to the Fourier Transform. Longer time windows give higher resolutions in the frequency domain, but the time resolution is lower. This is the well-known time vs. frequency trade-off. Figure 1 illustrates the calculation of a spectrogram $s(f)_t$ from a window of a signal x_T. In each case, the $\diamond$ symbol represents the results of the previous stage in the calculation.

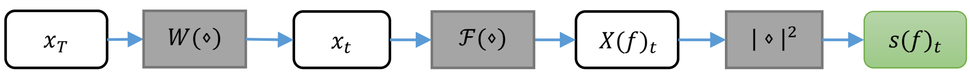


Figure 1 - Calculation of a spectrogram from an audio signal

Mel-Frequency Cepstral Coefficients (MFCCs)

A cepstrum [9] can be thought of as the spectrum of a spectrum. After calculation of the STFT, the spectrum is mapped onto the mel scale [10], which is logarithmic to the frequencies of the spectral coefficients, and is subjectively equally spaced, according to curves produced by Stevens and Volkmann. Then after taking logarithms of the mel values, the Discrete Cosine Transform $\mathbb{C}$ of the log values is taken, returning the MFCCs. Figure 2 illustrates the calculation procedure for MFCCs from an STFT of a windowed signal.

Figure 2 - Calculation of MFCCs from a Short-Time Fourier Transform of a windowed signal

MFCCs are widely used in speech processing and telecommunications. They have also been used in music applications in the identification of musical instruments, as the lower MFCCs are useful for detecting the timbre of musical instruments [11].

Chroma

Chroma features, also known as Pitch Class Profiles, are a family of features which are designed primarily for western music. They derive from Shepard's work in [12], where he quoted earlier work by Drobisch in noting that a helical representation of the continuum of pitch has "the advantage of brining tones an octave apart into closer spatial proximity", and separated the concepts of tonal height (overall pitch) from tonal chroma (pitch class). Each chroma can be represented as a 12-dimensional vector, with each member representing the energy in all octaves of each of the semitones of the equal-tempered scale {C, C#, D … A, A#, B}. Each chroma vector represents an equal duration of time which is much shorter than the durations of individual musical notes [13].

Each member of the chroma family of features is calculated in a slightly different way, but they all share some basic characteristics. Figure 3 shows the basic calculation method, where the STFT is passed through a logarithmic-frequency filter bank $\mathcal{B}$, and then folded into bins of the chroma vector by summing the log-frequency magnitude spectrum over Z octaves for each pitch class b, where there are β classes per octave.

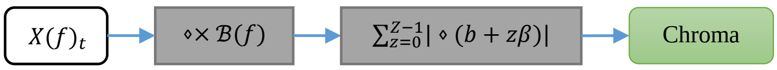


Figure 3 - Calculation of a chroma vector from an STFT

More sophisticated members of the Chroma family include the Harmonic Pitch Class Profile (HPCP) [14], Chroma Energy Normalised Statistics (CENS) [15], Chroma DCT-Reduced log Pitch (CRP) [11] and Beat Synchronous Chroma features [16]. Each of these is designed to overcome various shortcomings of the basic feature for certain applications, such as polyphonic music transcription or structural similarity, for example reducing the effects of transients or the use of different instrumentation [1].

Learned Features

The common element of all the features described above is that they have been designed by humans. These designs are informed by empirical measurements of the mammalian auditory system, subjective tests on human perception and domain knowledge of the western music system. The strengths of each of these features are their specificity to particular applications. However this can also manifest as a weakness in generalisation to new applications. A feature designed for music similarity may not apply very well to speech recognition and vice versa; as discussed in [17], it "would be useful to have a system that can automatically extract relevant features from the audio, without having to depend on ad-hoc domain dependent signal processing strategies"

Classical Neural Networks

Neural networks with several layers are notoriously difficult to train using standard algorithms such as backpropagation due to the problem of vanishing gradients [18], which tends to concentrate most of the learning in the upper layers. This limits the level of abstraction and consequent sophistication of features that can be leveraged from raw data. Networks trained in this way are also limited in what they can learn because they require labelled training data, which can be difficult to obtain in sufficient quantity and quality.

Unsupervised Learning and Deep Networks

Over the last ten years, significant progress has been made which has shifted the focus in machine learning from engineered features to learned features. Much of this progress is due to the ability to harness increased and parallel computation power, and unsupervised training methods.

So-called "deep" networks are typically pre-trained in pairs of unsupervised layers, using unlabelled data, which is easier to obtain in larger quantities. Research papers have shown that these deep networks produce more and more sophisticated features as the number of levels increases [19]. One limitation identified in using deep belief networks for genre recognition, is that using features learned from short clips of audio cannot capture "long-term time dependencies" [17].

Restricted Boltzmann Machines

The restricted Boltzmann machine (RBM) is a two-layer unsupervised network that is trained to learn a probability distribution over the training data, using neurons in the hidden layer. It is trained by an algorithm known as contrastive divergence.

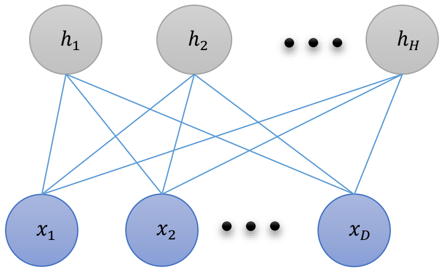


Figure 4 - Illustration of an RBM

Auto-Encoders

The auto-encoder (AE) is a three-layer unsupervised feed-forward network which is trained to reproduce its input at the output layer. The part of the model that computes the latent representation of the data is known as the encoder, and the decoder transforms this back into the output. If the hidden layer is made smaller than the input and output layers, then the auto-encoder can be used to compress the data using its distribution.

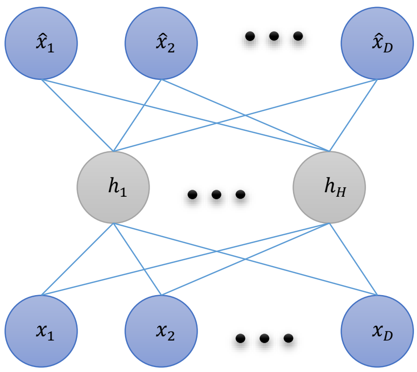


Figure 5 - Illustration of an AutoEncoder

Convolutional Neural Networks

One of the most successful recent types of deep network is the Convolutional Neural Network (CNN), which by imposes a degree of sparsity on neuron activations, and encourages localisation of feature learning. While the majority of effort has been concentrated in the more popular field of image recognition, audio convolutional networks have been successfully used in both research [20] and industry [21] for musical feature learning. They were also identified in [17] as a potential solution to learning long-term time dependencies.

Unsupervised Learning for Feature Extraction

While unsupervised networks are typically used in deep learning to extract more and more abstract types of features as the number of layers increases, shallow networks such as RBMs have been shown to improve results of music similarity learning [22]. Mid- level features from stacked denoising autoencoders have also been used in the place of conventional chroma for the task of chord estimation [23]. Therefore there is a sound basis for believing that such learned features might also prove to be useful representations for other MIR applications, such as structural music similarity measurements.

Structural Similarity

Music is a highly structured representation of audio information – musical pieces are constructed from successively smaller units of composition, from the full piece, to sections (verses, bridges and choruses in popular music), to bars, and down to the level of beats and fractional notes. All of this is governed by an overall tempo and time signature. Therefore, even if the melodic information between two given pieces of music is completely different, the pieces still might exhibit a significant amount of similarity in their structure, which would not be captured using short-time harmonic features alone.

In [1], a method was proposed for combining engineered audio features with structural similarity techniques for measuring the overall level of similarity between two pieces of music, both in a harmonic and structural context. The outline of the method is as follows:

1. Extract short-time audio features from the pieces under consideration.
2. Represent the structure of each piece using a recurrence plot, using a specified time-delay embedding.
3. Measure the difference between the recurrence plots of each piece using the Normalised Compression Distance (NCD).

Various configurations of each stage were tested as shown in Table 1.

Stage	Component	Variations
Feature Selection	Type of Feature	Chroma / CENS / CRP
	Feature Rate	Beat-synchronous Features per second {0.333, 0.5, 1, 1.25, 2.5, 5, 1}
	Feature Sequence Length	Proportional to signal length Fixed # Frames {300, 500, 700, 900, 1100}
Recurrence Plot	Time Delay Embedding	Embedding Dimension [1, 7] Time Delay [1, 5]
	Recurrence Plot Threshold	Normalised Euclidean Fixed Amount of Nearest Neighbours Fixed Recurrence Rate
NCD	Compression Method[1]	bzip2 PPMd

Table 1 - Configurations tested for Structural Music Similarity Tests

[1] Only *bzip2* is used in the 2011 paper [7], since it was found to be superior to *PPMd* during earlier work in [8]

Representation Learning for Structural Music Similarity Measurements

Approaches: Methods & Tools for Analysis & Evaluation

Literature Survey

The literature search and review was performed using Google Scholar and the City University Library to determine the research question and to plan the approach. However, an ongoing review will be made of other papers relating to the topic as the need arises throughout the project.

Gather Data

Source Audio

The source material used will be music from the British Library Music Project. The music selected will be similar in nature to the music used in [1] to ensure that experimental results can be compared. If possible, the same source audio will be used. There will be half to one day required for data curation and annotation. Once the source material has been selected, primary features such as spectrograms and chroma features will be extracted using Sonic Annotator with VAMP Plug-Ins. These features will be stored on the City University Lewes server, with a backup copy kept on a secure external hard-drive.

Software

All of the necessary network installation is understood to be complete. However, this period will also be used to ensure this is the case, and complete installation of any missing elements.

Develop Model

Replication of Existing Results for Comparison

The reported experiments in [1] took 3 weeks of computation time. Therefore it is not considered possible in the time available for the project to test many configurations of the parameters varied in the paper. Therefore the settings which were found to perform best, both in terms of experimental precision as well as generalisation (in terms of the sensitivity to the settings of other parameters) will be used to ensure that sufficient computational time can be allocated to the deep network training. All the software used for the techniques in [1] is known to be freely available. Most of the implementation uses Matlab toolboxes. The toolboxes have been downloaded (see Table 2) and will be tested at the start of the project. The author of [1] has been contacted to obtain the software.

Technique	Software used by Bello	Platform	Status	Other Options
Chroma and CENS	Chroma Toolbox	Matlab	Downloaded	Python - BregmanToolkit?
Beat Tracking	http://labrosa.ee.columbia.edu/projects/coversongs/	Matlab	Not using beat tracking	
Recurrence Plots	CRP Toolbox (requires registration)	Matlab	Downloaded	pyunicorn?
NCD	CompLearn Toolkit	Windows / Linux	Downloaded	

Table 2 - Software necessary to replicate structural similarity and back-ups

Feature Learning

New features will be learned from spectrograms of the source audio using unsupervised learning. The most likely software currently under consideration is Theano. Theano has many advantages: it is already installed on the City University computing cluster; there are example scripts available for download which could be quickly modified for use; it is easier to configure on a cluster than systems such as Matlab; and it can use GPU hardware acceleration for performing many of the matrix calculations involved in neural network training.

Evaluation

The suitability of the neural network features will be assessed in terms of their accuracy on a training dataset, and their generalisation ability using a test dataset. Based on the results from the assessment, one or more models will be selected to use for evaluation in the full structural similarity system. These model(s) will be used to learn features from the full dataset. To evaluate results, direct A/B comparisons will be made between the best performing configuration of [1] and configurations using the learned features. A one week period has been built in to the end of the evaluation phase to run more models in case the analysis of the results leads to relevant questions or clarifications which could be answered in the report. Evaluation of results will be done using Python as I have significant ability and experience with this language.

Complete Report

The report will begin to be drafted as soon as the project commences. However, given that much of it cannot be written until some results have been obtained, a five week period has been included at the end of the project to complete the first draft and address any comments which arise in the review.

Additional Time Allowance

A two week allowance has been built in to the end of the project in order to allow for certain aspects of the work taking longer than planned at this early stage, and to investigate any unexpected findings which arise during the analysis. If there is sufficient time then a deep network using semi-supervised learning may be investigated.

Project Meetings

Short, ad-hoc meetings will take place as and when issues affecting the project progress arise. Four formal meetings have been scheduled at the kick off and at various key stages in the project to ensure that the project is fully on track and update the project plan.

Risks

The following risk register outlines the envisaged risks, their associated likelihoods, consequences and impacts, and what has been done or will be done to mitigate them.

Description	Likelihood (1 – 3)	Consequence (1 – 5)	Impact (L x C)	Mitigation
Original source audio cannot be obtained	3	4	**12**	Use British Library audio
Original code cannot be obtained	2	4	**8**	Make early enquiries to obtain code Review toolkits and
Extracting spectrograms takes a long time	1	3	**3**	Start this early Ask Daniel Wolff to help
Other students are using the computing cluster when I want to run experiments	3	4	**12**	Meet with supervisor at kick-off meeting to review other students' project timelines
Neural network models are hard to write	3	5	**15**	Use Theano example scripts and modify Test small models on home computer
Neural network models take too long to run	2	5	**10**	Reduce number of models Consider renting cluster space
City University cluster crashes for a long time	1	5	**5**	Consider renting cluster space
Existing results cannot be replicated exactly	3	3	**9**	Resort to relative comparison
Code is lost or accidentally written over	2	5	**10**	Use version control software e.g. Git

Table 3 - Risk Register

Representation Learning for Structural Music Similarity Measurements

Plan of Work

Work Breakdown Structure

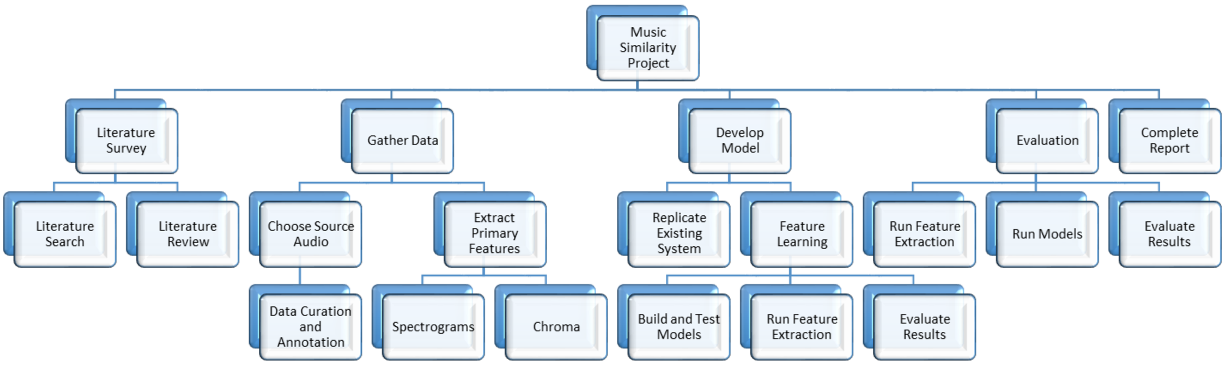
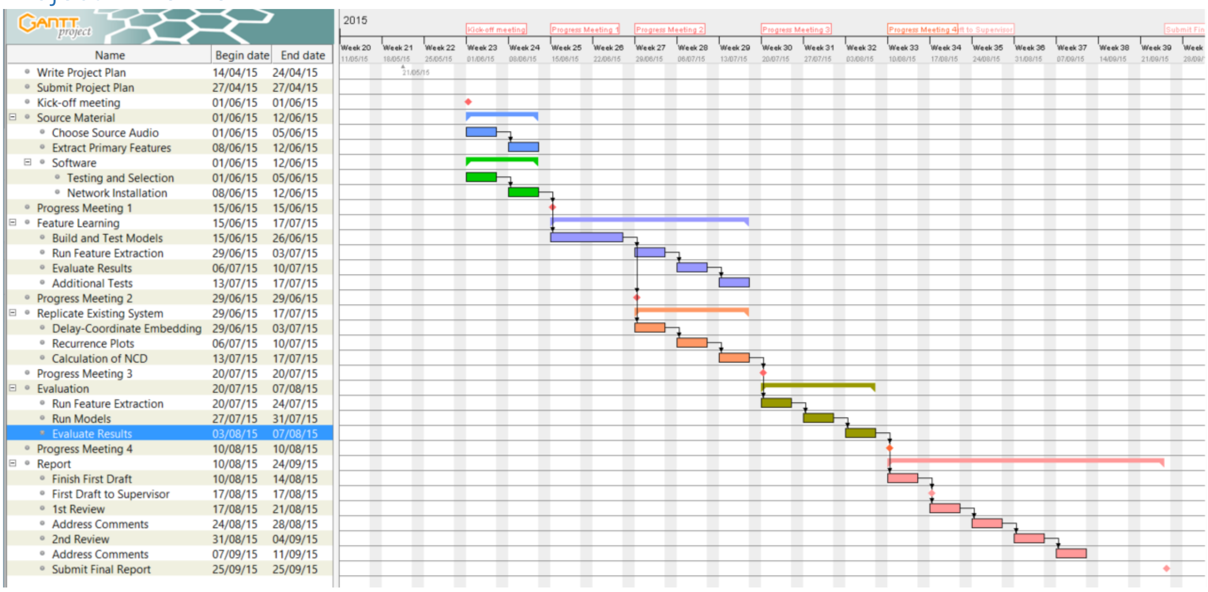


Figure 6 - Work Breakdown Structure

Project Timeline



Name	Begin date	End date
Write Project Plan	14/04/15	24/04/15
Submit Project Plan	27/04/15	27/04/15
Kick-off meeting	01/06/15	01/06/15
Source Material	01/06/15	12/06/15
Choose Source Audio	01/06/15	05/06/15
Extract Primary Features	08/06/15	12/06/15
Software	01/06/15	12/06/15
Testing and Selection	01/06/15	05/06/15
Network Installation	08/06/15	12/06/15
Progress Meeting 1	15/06/15	15/06/15
Feature Learning	15/06/15	17/07/15
Build and Test Models	15/06/15	26/06/15
Run Feature Extraction	29/06/15	03/07/15
Evaluate Results	06/07/15	10/07/15
Additional Tests	13/07/15	17/07/15
Progress Meeting 2	29/06/15	29/06/15
Replicate Existing System	29/06/15	17/07/15
Delay-Coordinate Embedding	29/06/15	03/07/15
Recurrence Plots	06/07/15	10/07/15
Calculation of NCD	13/07/15	17/07/15
Progress Meeting 3	20/07/15	20/07/15
Evaluation	20/07/15	07/08/15
Run Feature Extraction	20/07/15	24/07/15
Run Models	27/07/15	31/07/15
Evaluate Results	03/08/15	07/08/15
Progress Meeting 4	10/08/15	10/08/15
Report	10/08/15	24/09/15
Finish First Draft	10/08/15	14/08/15
First Draft to Supervisor	17/08/15	17/08/15
1st Review	17/08/15	21/08/15
Address Comments	24/08/15	28/08/15
2nd Review	31/08/15	04/09/15
Address Comments	07/09/15	11/09/15
Submit Final Report	25/09/15	25/09/15

Figure 7 - Project Timeline

References

[1] J. P. Bello, “Measuring Structural Similarity in Music,” *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, no. 7, pp. 2013 - 2025, 2011.

[2] J. P. Bello, “Grouping Recorded Music by Structural Similarity,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, Kobe, 2009.

[3] J. Popham, “Factors influencing music piracy,” *Criminal Justice Studies,* vol. 24, no. 2, pp. 199 - 209, 2011.

[4] S. Butler, “Piracy Losses,” *Billboard,* p. 18, 8 September 2007.

[5] O. F. Bustinza, F. Vendrell-Herrero, G. Parry and V. Myrthianos, “Music business models and piracy,” *Industrial Management & Data Systems,* vol. 113, no. 1, pp. 4 - 22, 2013.

[6] Y. Koren, “http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf,” 1 April 2015. [Online]. Available: http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf. [Accessed 1 April 2015].

[7] A. van den Oord, S. Dieleman and B. Schrauwen, “Deep content-based music recommendation,” in *Advances in Neural Information Processing Systems 26*, Lake Tahoe, 2013.

[8] O. C. Herrada, *Ph.D. Thesis: Music Discovery and Recommendation in the Long Tail,* Barcelona: Herrada, Oscar Celma;, 2008.

[9] B. P. Bogert, M. J. R. Healy and J. W. Tukey, “The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-Cepstrum, and Saphe Cracking,” in *Proceedings of the Symposium on Time Series Analysis*, New York, 1963.

[10] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, “Chapter 11: Noise, Signal Detection, Hearing, and Speech,” in *Fundamentals of Acoustics, Third Edition*, New York, John Wiley and Sons, 1998, p. 273.

[11] M. Muller and S. Ewert, “Towards Timbre-Invariant Audio Features for Harmony-Based Music,” *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, no. 3, pp. 649 - 662, 2010.

[12] R. N. Shepard, “Circularity in Judgments of Relative Pitch,” *Journal of the Acoustical Society of America,* vol. 36, no. 12, pp. 2346 - 2353, 1964.

[13] R. B. Dannenberg and N. Hu, “Pattern Discovery Techniques for Music Audio,” *Journal of New Music Research,* vol. 32, no. 2, pp. 153 - 163, 2003.

[14] E. G. Gutierrez, “Ph.D. Dissertation: Tonal Description of Music Audio Signals,” Universtat Pompeu Fabra, Barcelona, 2006.

[15] M. Muller, “Pitch- and Chroma-Based Audio Features,” in *Information Retrieval for Music and Motion*, Berlin, Springer Berlin Heidelberg, 2007, pp. 51 - 67.

[16] M. A. Bartsch and G. H. Wakefield, “To catch a chorus: using chroma-based representations for audio thumbnailing,” in *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, New York, 2001.

[17] P. Hamel and D. Eck, “Learning features from music audio with deep belief networks,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, Utrecht, 2010.

[18] S. Hochreiter, *Ph.D. Thesis: Untersuchungen zu dynamischen neuronalen Netzen,* Munich: Josef Hochreiter Institut fur Informatik, 1991.

[19] Y. Bengio, “Learning Deep Architectures for AI,” *Foundations and Trends in Machine Learning,* vol. 2, no. 1, pp. 1 - 127, 2009.

[20] H. Lee, P. Pham, Y. Largman and A. N. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Electronic Proceedings of the Neural Information Processing Systems Conference*, Vancouver, 2009.

[21] S. Dieleman, “http://benanne.github.io/2014/08/05/spotify-cnns.html,” 5 August 2014. [Online]. Available: http://benanne.github.io/2014/08/05/spotify-cnns.html. [Accessed 21 April 2015].

[22] S. Tran, D. Wolff, T. Weyde and A. d. Garcez, “Feature Preprocessing with Restricted Boltzmann Machines for Music Similarity Learning,” in *AES 53rd International Conference on Semantic Audio*, London, 2014.

[23] N. Glazyrin, “Mid-level Features for Audio Chord Estimation using Stacked Denoising Autoencoders,” Ural Federal University, Ekaterinburg, 2013.

Representation Learning for Structural Music Similarity Measurements

Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with ethical guidelines. In some cases ethics approval will have to be obtained from an ethics committee before the project can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that due consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

Part A: Ethics Checklist. All students must complete this part. The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

Part B: Ethics Proportionate Review Form. This part is an application for ethical approval of low-risk research. Students who have answered "no" to questions 1 – 18 and "yes" to question 19 in the checklist must complete this part. The project supervisor has delegated authority to approve this application.

Part A: Ethics Checklist

If your answer to any of the following questions (1 – 3) is YES, you must apply to an appropriate external ethics committee for approval:		*Delete as appropriate*
1.	Does your project require approval from the National Research Ethics Service (NRES)? (E.g. because you are recruiting current NHS patients or staff? If you are unsure, please check at http://www.hra.nhs.uk/research-community/before-you-apply/determine-which-review-body-approvals-are-required/)	**No**
2.	Will you recruit any participants who fall under the auspices of the Mental Capacity Act? (Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee http://www.scie.org.uk/research/ethics-committee/)	**No**
3.	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation? (Such research needs to be authorised by the ethics approval system of the National Offender Management Service.)	**No**

If your answer to any of the following questions (4 – 11) is YES, you must apply to the Senate Research Ethics Committee for approval (unless you are applying to an external ethics committee):		*Delete as appropriate*
4.	Does your project involve participants who are unable to give informed consent, for example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf?	**No**
5.	Is there a risk that your project might lead to disclosures from participants concerning their involvement in illegal activities?	**No**

6.	Is there a risk that obscene and or illegal material may need to be accessed for your project (including online content and other material)?	**No**
7.	Does your project involve participants disclosing information about sensitive subjects?	**No**
8.	Does your project involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning? (http://www.fco.gov.uk/en/)	**No**
9.	Does your project involve invasive or intrusive procedures? For example, these may include, but are not limited to, electrical stimulation, heat, cold or bruising.	**No**
10.	Does your project involve animals?	**No**
11.	Does your project involve the administration of drugs, placebos or other substances to study participants?	**No**

If your answer to any of the following questions (12 – 18) is YES, you must submit a full application to the Computer Science Research Ethics Committee (CSREC) for approval (unless you are applying to an external ethics committee or the Senate Research Ethics Committee). Your application may be referred to the Senate Research Ethics Committee.		*Delete as appropriate*
12.	Does your project involve participants who are under the age of 18?	**No**
13.	Does your project involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.	**No**
14.	Does your project involve participants who are recruited because they are staff or students of City University London? For example, students studying on a particular course or module. (If yes, approval is also required from the Head of Department or Programme Director.)	**No**
15.	Does your project involve intentional deception of participants?	**No**
16.	Does your project involve participants taking part without their informed consent?	**No**
17.	Does your project pose a risk to participants greater than that in normal working life?	**No**
18.	Does your project pose a risk to you, the researcher, greater than that in normal working life?	**No**

If your answer to the following question (19) is YES and your answer to all questions 1 – 18 is NO, you must complete part B of this form.		
19.	Does your project involve human participants? For example, as interviewees, respondents to a questionnaire or participants in evaluation or testing.	**No**