Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 151

Automatic speech/music discrimination for broadcast signals

Anna Kruspe[1] and Dominik Zapf[2] and Hanna Lukashevich[3]

Abstract: Automatic speech/music discrimination describes the task of automatically detecting speech and music audio within a recording. This is useful for a great number of tasks in both research and industry. In particular, this approach can be used for broadcast signals (e.g. from TV or radio stations) in order to determine the amount of music played. The results can then be used for various reporting purposes (e.g. for royalty collection societies such as the German GEMA).
Speech/music discrimination is commonly performed by using machine learning technologies, where models are first trained on manually annotated data, and can then be used to classify previously unseen audio data.
In this paper, we give an overview over the applications and the state of the art of speech/music discrimination. Afterwards, we present our approaches based on a set of audio features, Gaussian Mixture Models and Deep Learning. Finally, we give suggestions for the direction of new research into this topic.

Keywords: speech/music discrimination; music/speech classification; music detection; music analysis

1 Introduction and motivation

Speech/music discrimination describes the task of automatically detecting whether an audio signal contains speech or music content. In recent years, this has become especially relevant for TV and radio broadcasters who are broadcasting both types of signals, and need to find out details about this content for monitoring purposes. This applies to audio catalogues, media archives, and live broadcasting. Ideally, such broadcasters could analyze their playout protocols, but this may not always produce realistic results - for example, jingles, syndicated contributions, and production music will usually not be included in these lists. Even when playout protocols are being used, this information may not be synchronized with the databases of copyright holders. When metadata is available, it is often incomplete or contains historical gaps, and the rights situation is in constant flux. Historically, broadcasters have partially performed this task manually, but this is a very expensive and tedious task. For these reasons, manual music detection can only be done punctually. In addition to this, the results are highly subjective.
Automatic speech/music discrimination can help solve these tasks. The results can then again be used for various applications. First, they can assist in finding music-related metadata

[1] Fraunhofer IDMT, Semantic Music Technologies, Ehrenbergstr. 31, 98693 Ilmenau, kpe@idmt.fraunhofer.de
[2] Fraunhofer IDMT, Semantic Music Technologies, Ehrenbergstr. 31, 98693 Ilmenau, zapfdk@idmt.fraunhofer.de
[3] Fraunhofer IDMT, Semantic Music Technologies, Ehrenbergstr. 31, 98693 Ilmenau, lkh@idmt.fraunhofer.de

doi:10.18420/in2017_10

about the played music, which is necessary for reporting to copyright holders and royalty collection societies. They are also useful for ensuring that no additional copyrights are being violated (e.g. for online distribution). For royalty collection societies, the percentage of music played is important for various reasons - as an example, it determines the media format by which royalties are calculated. Finally, those results can be used internally to optimize programming and the various contributions. Broadcasting schedules can be tailored to the station's needs, and contributions can be adapted to customers' expectations.
Related technologies include plagiarism detection of musical pieces, cover version detection, mix recognition, and music similarity search.

2 State of the art

The problem of speech/music discrimination has been addressed in literature for several decades [Sa96, SS97, CPLT99]. Those systems already show promising results. However, they are usually trained and tested using clean, single-labeled recordings without noisy or mixed signals. The succeeding approaches address the development on noise robust features for speech/music discrimination [WGY03, FWX09]. Other works develop musically motivated features like Continuous Frequency Activation (CFA) [Se07]. This feature builds upon the fact that the musical signals tend to have more time stationary elements than the speech signals. CFA can be used as a standalone feature for threshold-base music detection [Se07] or in the combination with other acoustical features [HC13].
Speech/music discrimination was a task featured in the 2015 edition of the Music Information Retrieval Evaluation eXchange (MIREX)[4], which poses challenges in several MIR tasks every year and thus provides an annual overview of currently used methods. In this challenge, there were five methods that achieved an accuracy of over 99%. Three of them were using Neural Networks. One was trained in an unsupervised way by using Restricted Boltzmann Machines (RBM) and using the weights for initializing a feed-forward neural network [Sc13]. The other two implemented Convolutional Neural Networks, one using Constant Q Transform spectrograms (CQT-grams) [RLHM15], and the second one using Mel-frequency bands [Li15]. The two remaining works both implemented Random Forest classifiers for their method, one using a cent filterbank [So15] and the other one using RMS Energy, Zero Crossing Rate, Spectral Rolloff, Spectral Flux, Spectral Flatness, Spectral Flatness Per Band, and MFCCs, which were scaled and pre-processed using Principal Component Analysis (PCA) to reduce the number of dimensions down to 8 [Ts15].
Other works focus on choosing the optimal feature descriptors for speech and music. As the frequency range of human speech is different from that of music, choosing timbral features which take spectral differences into account is an obvious step. While there are several other timbral features used in audio-related machine learning tasks, which are presented below, Mel-Frequency Cepstral Coefficients (MFCC) [Lo00] have proven to perform well in speech/music discrimination [MKG16].

[4] http://www.music-ir.org/mirex/wiki/2015:Music/Speech_Classification_and_Detection

As shown in [SC14], using pitch-related features like chroma features can additionally contribute to improving classification results. Chroma features are obtained by separating the audio signal into 88 frequency bands, centered at the pitches of the equal-tempered scale, summing up the octaves resulting in 12 bins and calculating the short-time mean-square-power of each bin [MKC05]. Additionally, short-time statistics are applied to these bins, resulting in Chroma Energy Normalized Statistics (CENS).
[SC14] suggest combining such chroma-based features with other spectral features. While comparing several audio features, [MKG16] found combining MFCCs and CENS features to yield the best results in comparison to other timbral features like Spectral Flux, Spectral Centroid, andinformatik Zero Crossing Rate.

3 Proposed approach

An overview of the general processing chain is given in Figure 1. As shown, the algorithm uses audio data as its input, which is then pre-processed for the following feature extraction and classification. The produced results can then be post-processed for various purposes. We will describe these steps in more details in the following paragraphs.

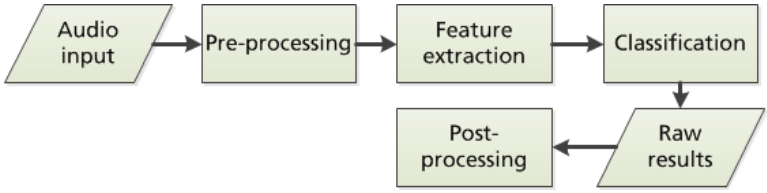

Fig. 1: Overview of the speech/music discrimination processing chain.

3.1 Pre-Processing

Audio data can be input in a variety of formats. Commonly, it comes from the multimodal broadcast signal, which contains audio and video and therefore must first be de-multiplexed to obtain the audio part. The signal is then pre-processed to shape it into a uniform format, since the following algorithms require the same audio format for which they were developed. Common pre-processing steps are normalization, re-sampling to a fixed sampling rate, and stereo-to-mono conversion.
Which steps are necessary and how they are implemented depends on what datasets are used. In our approach, two training datasets are used. The first one consists of recordings of public German TV stations (ARD, BR and SWR), which are stored in 44.1 kHz and 16-bit stereo .mp3 files. The data set consists of 171 files, with each file containing 10 minutes of audio, adding up to 28.5 hours.

The second one is the GTZAN music/speech dataset[5], which contains 60 files for both music and speech, each with a length of 30 seconds, sampled at 22.05 kHz, 16-bit, and saved as mono .wav files. In pre-processing, the broadcast recordings were decoded, resampled to 22.05 kHz, and converted to mono in order to have all files represented in the same format.

3.2 Feature Extraction

Pure audio contains a large amount of data, but is in its original shape not very conducive to determining its content. It is also highly redundant and "noisy" in the sense that it contains a high percentage of information that provides no evidence over whether or not music is contained. For this reason, we extract so-called audio features from the audio. In our approach, we use a large number of audio features that are descriptive of musical and speech content [BP05, Pe04]. To facilitate an overview, the audio features are subdivided into three categories covering the timbral, rhythmic and tonal aspects of sound. Depending on the required time resolution, audio features are then aggregated over longer time frames to smooth the representation.

Timbral features Although the concept of timbre is still not clearly defined with respect to music signals, it has proved to be very useful for automatic music signal classification. To capture timbral information, we use Mel-Frequency Cepstral Coefficients, the Audio Spectrum Centroid, the Spectral Flatness Measure, the Spectral Crest Factor, and the Zero-Crossing Rate. In addition, modulation spectral features [AS03] are extracted from the aforementioned features to capture their short term dynamics. We applied a cepstral low-pass filtering to the modulation coefficients to reduce their dimensionality and decorrelate them as described in [DBG07].

Rhythmic features All rhythmic features used in the current setup are derived from the energy slope in excerpts of the different frequency-bands of the Audio Spectrum Envelope feature. These comprise the Percussiveness [UH03] and the Envelope Cross-Correlation (ECC). Further mid-level features [DBG07] are derived from the Auto-Correlation Function (ACF). In the ACF, rhythmic periodicities are emphasized and phase differences are annulled. Thus, we also compute the ACF Cross-Correlation (ACFCC). The difference to ECC again captures useful information about the phase differences between the different rhythmic pulses. In addition, the log-lag ACF and its descriptive statistics are extracted according to [GDG09].

Tonal features Tonality descriptors are computed from a Chromagram based on Enhanced Pitch Class Profiles (EPCP) [Le06]. The EPCP undergoes a statistical tuning estimation and correction to account for tunings deviating from the equal-tempered scale. Pitch-space representations as described in [Ga07] are derived from the Chromagram as mid-level features. Their usefulness for audio description has been shown in [GD09].

[5] http://marsyasweb.appspot.com/download/data_sets/

Feature extraction is one of the major adjustment points to adapt the algorithm to specific use cases. As already presented in section 2, MFCC and Chroma features have been particularly successful in current speech/music discrimination approaches. Therefore, we reduced the feature set for our newer approaches using Neural Networks. As described in [MKG16], we calculate 13 MFCC features and 12 CENS features leading to a total of 25 features computed at a hop length of 20ms. Also each feature was normalized to have zero mean and unit variance over all training samples.

3.3 Classification

For the actual classification into the speech or music classes, we employ machine learning algorithms. These require a training procedure, which consists of feeding manually annotated data into an algorithm, which then "learns" a mapping (model) from the feature data to the annotations. Manual annotations come from human listeners, who have determined which audio segments contain music or speech.
We use realistic data for this purpose - i.e. actual broadcast recordings, together with additional music or speech recordings from other sources. In its most basic shape, we train models for mapping the features to the classes "Music" and "No music", but we have also tested other configurations. One of those distinguishes between four classes: "Music", "Speech", "Music+Speech", and "Silence/Noise". From experience, it is best to train a separate model for the task "Silence/Noise" versus "Non-Silence", and then combine the results by overwriting the output from the speech/music classifier when silence is detected. We have also started to add additional classes of audio categories, most notable "Applause". An overview of the classification process on the example of the four class variant is given in Figure 2.

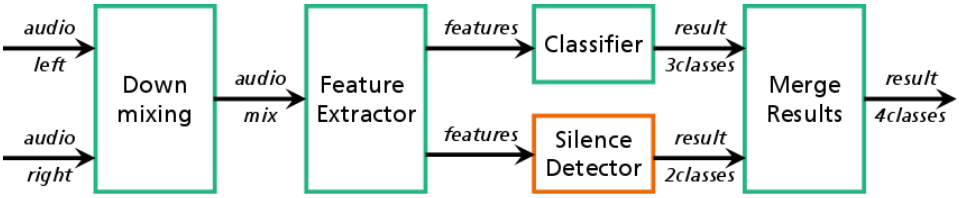

Fig. 2: Classification process on the example of 4 classes, where a separate silence detector is added.

3.3.1 Gaussian Mixture Model

Our basic approach employs so-called Gaussian Mixture Model (GMM) classifier as its machine learning algorithm. GMM is commonly used generative classifier. Single data samples of the class are thought of as generated from various sources and each source is modeled by a single multivariate Gaussian. The probability density function (PDF) of the feature frames is estimated as a weighted sum of the multivariate normal distributions.

Each single i-th mixture is characterized by its mean vector μ_i and covariance matrix Σ_i. Thus, a GMM is parametrized in $\Theta = \{\omega_i, \mu_i, \Sigma_i\}$, $i = \overline{1, M}$, where ω_i is the weight of the i-th mixtures and $\sum_i \omega_i = 1$. The generalization properties of the model can be adjusted by choosing the number of Gaussian mixtures M. The parameters of the GMM can be estimated using the Expectation-Maximization algorithm [DLR77].

3.3.2 Deep Neural Network

Our most recent approach implements an feed-forward deep neural network (DNN). While artificial neural networks were already developed in the 40s and 50s [MP43], they gained a lot of attention in the machine learning community in the last few years by increasing performance in a lot of research fields like image classification or artificial intelligence [GBC16, Ha16]. DNNs are inspired by the processes in the human brain. They typically consist of multiple layers of neurons that are activated by weighted connections to the neurons in the previous layer, the first layer representing the features used for training, and the last layer representing the results, depending on the task the DNN is designed to perform.
DNNs are often trained using a supervised learning method, which usually integrates the backpropagation algorithm - i.e., for each training example the result of the current network and the error of the result is calculated. The connection weights are then adapted using the error of the result. The performance of an DNN depends heavily on the selection of the parameters and architecture of the network [GBC16].
While the number of layers and number of neurons in each layer can be adapted to influence the overall performance of the network, the activation function (i.e. the way a neuron sums up its previous neurons) and the backpropagation algorithm can be adapted to optimize the learning process, e.g. the duration until the model converges.
We train three binary classification DNNs individually for the following three detection tasks: (a) music vs. non-music, (b) speech vs. non-speech, and (c) silence/noise vs. music/speech. With these separate models the results can be calculated and interpreted as necessary by combining the individual results. Thus, the results of the music and speech detection networks are combined with the silence detection, i.e. music and speech will be only detected when the frame was not classified as silence or noise. The presence of music and speech together is then simply given by the combination of the results of both networks.
In order to find a suitable architecture for the DNN for the discrimination task, a grid search was conducted over the number of layers and the number of nodes per layer, and the best-performing architecture was chosen to be trained with more data. In this case, a network with five fully connected layers with 100, 500, 500, 500, and 100 nodes respectively was selected which leads to about 600k adjustable parameters. These parameters were initialized randomly using glorot initializing[GB10]. Additionally, dropout was introduced to each of the five hidden layers. This has been shown to prevent the network from overfitting [Sr14]. A dropout rate of 0.3 gave us the best results. An overview of the architecture is given in 1. We chose the ReLU (Rectified Linear Unit) [NH10] as the activation function and Adam [KB14] with a learning rate of 0.001 as the optimization algorithm, which speeds up the learning process. The error of the network was calculated using binary cross entropy as

the loss function.
The DNN was implemented in Python 3 using the Keras[6] framework with the Tensorflow[7] backend. Each training epoch took roughly 3 minutes to finish, resulting in about 5 hours of total training for each network.

Layer Type	Layer Size	Activation	Dropout Rate
Input	25 x 1	-	-
Hidden	100 x 25	ReLU	0.3
Hidden	500 x 100	ReLU	0.3
Hidden	500 x 500	ReLU	0.3
Hidden	500 x 500	ReLU	0.3
Hidden	100 x 500	ReLU	0.3
Output	1 x 100	Sigmoid	-

Tab. 1: Architecture of the neural network used for the speech/music discrimination task.

3.4 Post-processing of results

Since audio features are fed frame-wise into the classification model, the results come in the shape of class likelihoods per frame. These likelihoods can then be post-processed to generate meaningful final results. One of the first steps is commonly a majority vote to determine one class per frame. In cases where the likelihoods are too similar, a decision threshold may be integrated to remove uncertain results.
Then, multiple frames are usually grouped into segments. The underlying assumption is that the class will not change rapidly over time (e.g. there will not be a segment of $10ms$ of music playing). For this reason, results are smoothed over time. The result is a list of segments with a start time, end time, and a class result ("speech", "music", etc.). Finally, such segment results can be aggregated to provide statistics over longer time frames - e.g. over whole days or weeks.

4 Results and current applications

4.1 Traditional approach using GMMs

Our traditional approach is the one using GMMs described above. We employ it for broadcasting applications, mainly those where broadcasters need to report their music percentages over longer timeframes.
In order to evaluate the validity of this approach, we compared our results to manual annotations by a German TV broadcaster. The evaluation was performed over the course of four full days of TV program, and was done for the two-class case (music present vs. no

[6] https://keras.io
[7] https://www.tensorflow.org

music present). This data comprised a mix of TV shows typically seen on German TV, such as news programs, documentations, talk shows, game shows etc.
The results over 6 hour timeframes are shown in Figure 3. The blue bars show the manually annotated amounts of music played over these timeframes (which are assumed to be ground truth), while the red bars show the results using the GMM approach. On average, the difference over these timeframes is 2.11%. This evaluation was chosen because this is usually the format requested by TV broadcasters.

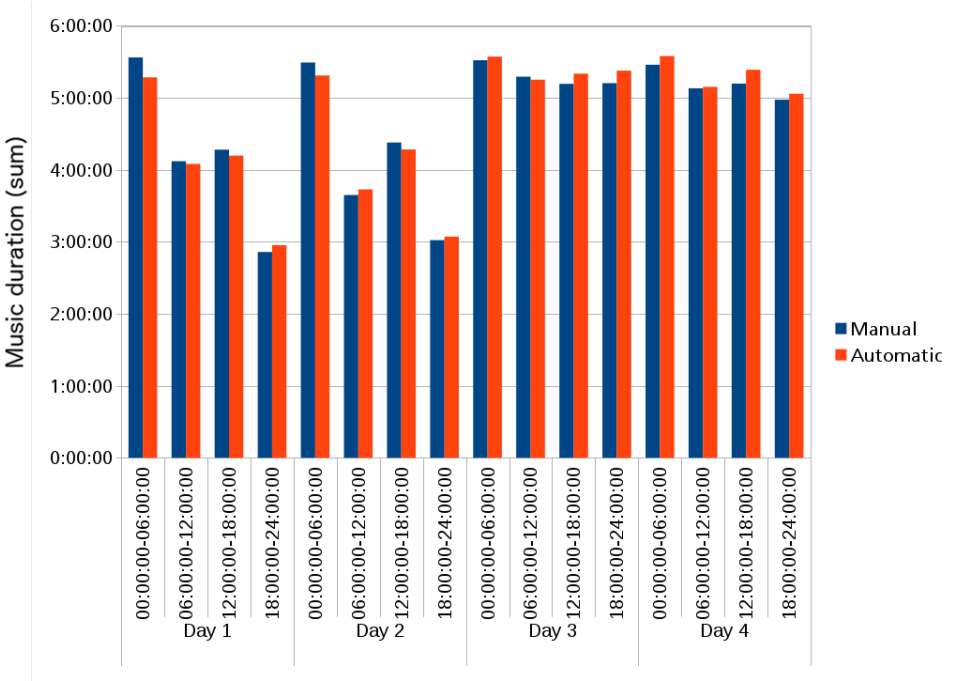

Fig. 3: Music durations in 6-hour timeframes over four days of TV broadcast audio: Manual ground truth annotations in blue vs. automatic results in red.

4.2 Current approach using an DNN

The experiments with DNNs were split up into different parts. As already stated above, it usually makes sense to separate the speech/music discrimination process into different tasks. In every training phase, a randomly selected 20% split of the training data was taken from the dataset and used as validation data to verify training results on unknown data. The results were calculated frame-wise.
In the first step, the network was trained to discriminate silence and noise from non-silence. For this task, another set of artificially generated 63 audio samples containing 30 second long segments of silence and/or noise was used in combination with the GTZAN dataset. After training for 100 epochs, the neural network was able to confidently discriminate between silence/noise and non-silence with >99% accuracy. As the broadcast recordings do not

contain any segments labeled as silence, we were not able to test this network on this dataset, but testing manually chosen frames of the dataset confirmed above results. In all of these models the classification was done on the features extracted for single 20ms long frames. In the next step, two neural networks were trained, one classifying speech/non-speech, the other classifying music/non-music. Both, the broadcast recordings and the GTZAN dataset, were used for training. Classifying speech, the neural network was able to achieve precision, recall and f-measure values of 0.99 and an accuracy of 98%. On the other hand, the network classifying music obtained a precision of 0.93, a recall of 0.89, an f-measure of 0.91 and an accuracy of 93%. One reason for the music network performing worse than the speech network could be the composition of the training data. While 84.7% of the training samples contain speech, only 35.2% of the samples contain music, so the variance of music in the training data is lower as well compared to the variance of speech.

4.3 Sources of error

Following an in-depth manual analysis of the results, we were able to identify sources of error that occur across the various approaches. The following is a non-exhaustive list, which offers starting points for improving the algorithm.

Very quiet music Due to masking effects, signals can sometimes contain music which is barely audible. Consequently, our approach often cannot detect such music either. As a human listener, the presence of music might be obvious from context (e.g., music is playing, then becomes very quiet in the background, then picks back up). Our approach could be improved by taking this context information into account.

Harmonic sounds or "chaotic" music There are environmental sounds that can disturb the algorithm by sounding like music from a technical perspective. This is especially true for highly harmonic sounds (e.g. a motor whirring) or rhythmic, percussive sounds (e.g. footsteps). In contrast, there is music that employs non-musical sounds frequently, in particular in the electronic genres. The line between those sources of audio is blurry, and distinguishing between sounds like these is difficult even for human listeners.

What is music? There is an on-going discussion between broadcasters, royalty collection societies, and research about the question what even constitutes music. For example, what about very short segments? Can music be playing for one second? What about sound effects or production sounds? What about crowds chanting?

Silence and other dramatic elements Coming from a signal perspective, silence is clearly not music. However, silence is frequently employed as a dramaturgical element in music, as are other effects. This is mainly a question of time resolution - if a very small resolution is required, how can we detect "meaningful" silence like this?

Reverb and echoes Adding reverb or echo to audio technically adds harmonic and/or rhythmic elements to it. For this reason, the algorithm may become confused by audio processed in this way.

5 Future work

While current methods of speech/music discrimination already have reached a high level of performance, there are still some machine learning approaches to be tested. A promising possibility would be using Recurrent Neural Networks (RNN), a variation of DNNs which add the ability to memorize previous data samples by extending the neurons in a recurrent layer with a hidden state. Variants of this architecture using Long Short Term Memory (LSTM) [HS97] or Gated Recurrent Units (GRU) [Ch14] cells to calculate the current hidden state were already successfully tested in other applications where some kind of memory is useful to model sequential and/or temporal properties of the data. As this also applies to speech/music discrimination, this could be a possibility to increase classification results.

Another important point usually heavily determining the performance of DNNs is the quality of the dataset. This not only means the pure audio quality, but also the variety and size of training data. Gathering large amounts of reliable data can be a difficult and time-consuming task for itself as the data has to be labeled manually. To avoid the lengthy process of manual annotation, there have been efforts to generate labels automatically through metadata. An example for this approach is the AudioSet [Ge17], which was released by a Google Research group in March 2017 and contains audio features and labels generated by metadata of videos uploaded to YouTube. This approach of data generation could contribute to improving machine learning models through extending data variety by a huge amount.

References

[AS03] Atlas, Les; Shamma, Shihaba A.: Joint acoustic and modulation frequency. EURASIP Journal on Applied Signal Processing, 7:668–675, 2003.

[BP05] Bello, Juan Pablo; Pickens, Jeremy: A Robust Mid-Level Representation for Harmonic Content in Music Signals. In: Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR). London, UK, 2005.

[Ch14] Chung, Junyoung; Gülçehre, Çaglar; Cho, KyungHyun; Bengio, Yoshua: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR, abs/1412.3555, 2014.

[CPLT99] Carey, Michael J.; Paris, Eluned S.; Lloyd-Thomas, Harvey: A Comparison of Features for Speech, Music Discrimination. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 149–152, 1999.

[DBG07] Dittmar, Christian; Bastuck, Christoph; Gruhne, Matthias: Novel Mid-Level Audio Features for Music Similarity. In: Proceedings of the International Conference on Music Communication Science (ICOMCS). Sydney, Australia, pp. 38–41, 2007.

[DLR77] Dempster, Arthur P.; Laird, Nan M.; Rubin, Donald B.: Maximum Likelihood From Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B, 39:1–38, 1977.

[FWX09] Fu, Zhong-Hua; Wang, Jhing-Fa; Xie, Lei: Noise Robust Features for Speech/Music Discrimination in Real-time Telecommunication. In: 2009 IEEE International Conference on Multimedia and Expo (ICME). pp. 574–577, 2009.

[Ga07] Gatzsche, Gabriel; Mehnert, Markus; Gatzsche, David; Brandenburg, Karlheinz: A Symmetry Based Approach for Musical Tonality Analysis. In: Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR). Vienna, Austria, pp. 207–210, 2007.

[GB10] Glorot, Xavier; Bengio, Yoshua: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256, 2010.

[GBC16] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron: Deep learning. MIT press, 2016.

[GD09] Gruhne, Matthias; Dittmar, Christian: Comparison of harmonic mid-level representations for genre recognition. In: Proceedings of the 3rd Internation Workshop on Learning Semantics of Audio Signals (LSAS). Graz, Austria, pp. 91–102, 2009.

[GDG09] Gruhne, Matthias; Dittmar, Christian; Gärtner, Daniel: Improving rhythmic similarity computation by beat histogram transformations. In: Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR). Kobe, Japan, pp. 177–182, 2009.

[Ge17] Gemmeke, Jort F.; Ellis, Daniel P. W.; Freedman, Dylan; Jansen, Aren; Lawrence, Wade; Moore, R. Channing; Plakal, Manoj; Ritter, Marvin: Audio Set: An ontology and human-labeled dataset for audio events. In: Proc. IEEE ICASSP 2017. New Orleans, LA, 2017.

[Ha16] AlphaGo: using machine learning to master the ancient game of Go. `https://blog.google/topics/machine-learning/alphago-machine-learning-game-go/`.

[HC13] Han, Jinyu; Coover, Bob: Leveraging structural information in music-speech dectection. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). pp. 1–6, 2013.

[HS97] Hochreiter, Sepp; Schmidhuber, Jürgen: Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[KB14] Kingma, Diederik P.; Ba, Jimmy: Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980, 2014.

[Le06] Lee, Kyogu: Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile. In: Proceedings of the International Computer Music Conference (ICMC). 2006.

[Li15] Lidy, Thomas: Spectral Convolutional Neural Network for Music Classification. MIREX, 2015.

[Lo00] Logan, Beth et al.: Mel Frequency Cepstral Coefficients for Music Modeling. In: ISMIR. 2000.

[MKC05] Müller, Meinard; Kurth, Frank; Clausen, Michael: Audio Matching via Chroma-Based Statistical Features. In: ISMIR. volume 2005, p. 6th, 2005.

[MKG16] Malviya, Yash; Kaul, Shiv; Goyal, Kushaagra: Music Speech Discrimination. CS229 Final Project, December 2016.

[MP43] McCulloch, Warren S; Pitts, Walter: A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biology, 5(4):115–133, 1943.

[NH10] Nair, Vinod; Hinton, Geoffrey E: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814, 2010.

[Pe04] Peeters, Geoffroy: , A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project: Technical Report, 2004.

[RLHM15] Royo-Letelier, Jimena; Hennequin, Romain; Moussallam, Manuel: MIREX 2015 Music/Speech Classification. MIREX, 2015.

[Sa96] Saunders, J.: Real-time discrimination of broadcast speech/music. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Atlanta, GA, USA, pp. 993–996, 1996.

[Sc13] Schlüter, Jan: Learning binary codes for efficient large-scale music similarity search. In: Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR). 2013.

[SC14] Sell, Gregory; Clark, Pascal: Music tonality features for speech/music discrimination. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014. pp. 2489–2493, 2014.

[Se07] Seyerlehner, Klaus; Pohle, Tim; Schedl, Markus; Widmer, Gerhard: Automatic Music Detection in Television productions. In: Proceedings of the 10th International Conference on Digital Audio Effects (DAFx). Bordeaux. France, 2007.

[So15] Sonnleitner, Reinhard: Speech Music Detection and Classification. MIREX, 2015.

[Sr14] Srivastava, Nitish; Hinton, Geoffrey E; Krizhevsky, Alex; Sutskever, Ilya; Salakhutdinov, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[SS97] Scheirer, E.; Slaney, M.: Construction and evaluation of a robust multifeature speech/music discriminator. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). volume 2, pp. 1331–1334, 1997.

[Ts15] Tsipas, Nikolaos; Vrysis, Lazaros; Dimoulas, Charalampos; Papanikolaou, George: MIREX 2015: Methods for Speech / Music Detection and Classification. MIREX, 2015.

[UH03] Uhle, Christian; Herre, Jürgen: Estimation of tempo, micro time and time signature from percussive music. In: Proceedings of the 6th International Conference on Digital Audio Effects (DAFx). London, UK, 2003.

[WGY03] Wang, W. Q.; Gao, W.; Ying, D. W.: A fast and robust speech/music discrimination approach. In: Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Proceedings of the Fourth Pacific Rim Conference on Multimedia. volume 3, pp. 1325–1329, 2003.