AUTOMATIC MUSIC DETECTION IN TELEVISION PRODUCTIONS

Klaus Seyerlehner, Tim Pohle, Markus Schedl
Dept. of Computational Perception
Johannes Kepler University Linz, Austria
klaus.seyerlehner@jku.at

Gerhard Widmer
Dept. of Computational Perception
Johannes Kepler University Linz, Austria and
Austrian Research Institute for AI, Vienna
gerhard.widmer@jku.at

ABSTRACT

This paper presents methods for the automatic detection of music within audio streams, in the fore- or background. The problem occurs in the context of a real-world application, namely, the analysis of TV productions w.r.t. the use of music. In contrast to plain speech/music discrimination, the problem of detecting music in TV productions is extremely difficult, since music is often used to accentuate scenes while concurrently speech and any kind of noise signals might be present. We present results of extensive experiments with a set of standard machine learning algorithms and standard features, investigate the difference between frame-level and clip-level features, and demonstrate the importance of the application of smoothing functions as a post-processing step. Finally, we propose a new feature, called *Continuous Frequency Activation* (CFA), especially designed for music detection, and show experimentally that this feature is more precise than the other approaches in identifying segments with music in audio streams.

1. INTRODUCTION

Annotation and tagging of audio data have mainly been human tasks in the past. The growing amount of digital media, however, makes manual tagging impractical. One of these tiresome tasks is to determine whether or not music is present in an audio excerpt. This problem occurs in many application contexts. A particular variant of this problem was posed to us by the Austrian National Broadcasting Corporation (ORF): the task is to automatically determine where in the sound track of a TV production there is music being played, in the foreground or in the background. This is important for the calculation of royalty fees, which are paid to a national agency according to certain rules. Ideally, the production team would supply a precise list of all the music segments occurring in a TV production, but in reality these lists are often incorrect or simply empty, which requires the ORF to more or less guess the amount of music within a production, since manually annotating all productions is simply impossible. Thus, it would be desirable to have a system that automatically detects music segments and predicts, with high precision, the percentage of time where music is present within a production.

In this paper, we present our approach to this difficult music detection problem. First a brief literature review is given out in Section 2. Section 3 presents an overview of our overall approach and a detailed description of the features examined. In Section 4.1 we report on the ground truth data that were collected, on extensive machine learning experiments and the results obtained with them. We then introduce a new feature in Section 5 and show that this feature indeed yields further improvement. Finally, we present our conclusions and discuss future work.

2. RELATED WORK

There has been quite some research recently on the automatic discrimination between speech and music. Even if our problem is related, it must be pointed out that detecting music within TV productions is more complicated than simple music/speech discrimination. The major reason is that music and other sounds are generally mixed in TV, and in particular that the musical background of movies is typically rather soft compared to spoken words or scene-related sounds in the foreground. That is because music is normally used to create the atmosphere of a scene and should not attract the listener's attention. Interestingly, when people pay attention to the presence of music in movies, most of them are surprised at what a high percentage of music is present and how difficult it is, in many cases, to even tell whether or not music is being played at all.[1] Thus, in contrast to the typical datasets usually used in speech/music discrimination research, which mostly consist of relatively distinct cases of the classes *music* and *speech*, we mainly have to deal with soft music signals mixed with other sound signals.

There has been some previous work that is relevant to our problem, for example Santo et al. [1], who worked on automatic video segmentation based on audio track analysis. In contrast to our problem – deciding whether there is music present or not – they deal with seven different classes. When aggregating the results they report for their 7 classes to the two broad classes *music* and *no_music* only, we arrive at a classification accuracy of their system of approximately 75.86%, which we consider as a quite good and a useful baseline to evaluate our approach. Khan et al. [2] give an interesting overview of existing features and methods for movie audio classification, although the results of the various approaches are incomparable to each other, due to the lack of common test databases and different application areas. Minami et al. [3, 4] focus on the automatic indexing of videos by discriminating video scenes according to the classes *speech*, *music* and *music and speech*. Their system is composed of two expert systems, one for detecting music and the other one for detecting speech. They report an average detection rate of 90% for musical segments. However their ground truth database seems to be very unrepresentative – we re-implemented their approach and only achieved a low 55.78% on our real world dataset (see below). Mauclair and Pinquier [5] apply their speech/music classification system to recordings from radio stations, where they achieve a classification accuracy of 86.9% for music/non-music discrimination (which should

[1] To illustrate the difficulty of this problem we provide, on our homepage, some audio samples of the television productions we have been annotating — see http://www.cp.jku.at/people/seyerlehner/md.html.

be simpler than background music detection in TV shows). Altogether, speech/music discrimination seems to have broad application potential and attracted a lot of research attention, but to our knowledge there is no scientific work focusing specifically on music detection.

3. SYSTEM OVERVIEW

The architecture of our music detection system resembles a classical machine learning process extended by a post-processing stage. In a first step the audio stream is cut into small frames and features are extracted for each frame. Second, a classifier is trained on a distinct training set and learns to distinguish the two classes *music* and *no_music*. It is then used to predict the class labels for all the frames in new TV shows. In a final post-processing stage the classification results are smoothed in such a way that we obtain a plausible label sequence for longer continuous segments of audio.

Since the choice of features is very critical, we first decided to test some promising features known from recent work in the field of speech/music discrimination. The next section gives a detailed description of the features we have chosen to investigate.

3.1. Features

We focused on four sets of features. A major aspect during the decision process was that a feature must still be able to capture musical properties, even if speech or any kind of sounds are present. Thus, for example, we did not consider *4 Hz modulation energy* and *zero-crossing rate* related features, since they are merely useful in speech/music discrimination to detect speech segments, but not in the case of music detection alone.

3.1.1. Spectral Entropy (SE)

In general the entropy measures the uncertainty or unpredictability of a probability mass function (PMF). The entropy of the spectrum of an audio frame is a well-known feature for speech/music discrimination [6]. To be able to compute the entropy, the power spectrum is converted into a probability mass function:

$$x_i = \frac{X_i}{\sum_{j=1}^{N} X_j} \tag{1}$$

where X_j denotes the energy of j-th frequency component of the STFT spectrum of the current frame. For each frame the entropy is computed from $\vec{x}$ as:

$$H = \sum_{i=1}^{N} -x_i \log_2 x_i \tag{2}$$

In general the *spectral entropy* should be higher for speech frames than for music frames.

3.1.2. Chromatic Spectral Entropy (CSE)

The Chromatic Spectral Entropy, as defined in [7], is a variant of the *Spectral Entropy*. Instead of computing the entropy directly based on the normalized power spectrum, the power spectrum is first mapped onto the Mel-frequency scale and divided into 12 sub-bands, where the center frequency f_i of a band coincides with one of the 12 semitones of the chromatic scale. For a fixed center frequency f_0 of the lowest band, the center frequencies of the other sub-bands correspond to:

$$f_i = 1127.01048 * \log(\frac{f_0 * 2^{\frac{k}{12}}}{700} + 1) \tag{3}$$

As for the *Spectral Entropy* the energies of the sub-bands X_i are normalized according to equation (1), and the entropy of the chromatic representation of a frame is again computed as in equation (2).

3.1.3. Mel Frequency Cepstrum Coefficients (MFCC)

Mel Frequency Cepstrum Coefficients are a compact representation of the spectral envelope of a frame. After a non-linear mapping onto the Mel-frequency scale, to better approximate the frequency resolution of the human ear, the envelope of the log-spectrum is compactly represented by the first few coefficients after a DCT compression. MFCCs are well-known for capturing timbral aspects of short audio frames. Ezzaidi et al. [8] show the successful application of ΔMFCCs in the area of speech/music classification.

3.1.4. Linear predictive Coefficients (LPC)

Linear prediction is used to predict the current value $\hat{s}(n)$ of the real-valued time series $s(n)$ based on past p samples [9].

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{4}$$

The filter coefficients a_i define the p-th order linear predictor (FIR filter). The optimal filter coefficients are determined by minimizing the prediction error in the least squares sense. The prediction error, or residual error, is given by

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i). \tag{5}$$

For the compact representation of an audio frame the time-series $s(n)$ is the time-domain sample sequence of the current frame. The prediction error is expected to be significantly higher for impulsive speech compared to steady notes played by instruments.

Additionally, ΔMFCC, ΔLPC, ΔSE and Δ CSE were also added to the feature set.

3.2. From frame-level to "clip-level" features

Short-term frame-level features capture essential information about the sound of an audio frame. Such an audio frame commonly lasts 10-40 ms, which means that it contains little if any temporal information. Our machine learning approach does not assume any specific ordering of the training or test examples either and thus most of the temporal information is lost. For speech/music discrimination temporal information might be useful, because speech segments tend to be more impulsive than music segments, leading to a higher variance of the frame-level feature values over time. To capture some of this temporal information we summarize a sequence of consecutive feature vectors by computing mean and standard deviation over a fixed number of frames. The resulting features – now representing an audio clip of several seconds of audio – are called *clip-level* features in accordance with [9]. We performed dedicated machine learning experiments to investigate if these clip-level features yield any improvement over frame-level features. We will report on the results in section 4.1.

3.3. Smoothing

The result of the classification process is a sequence of class labels *music* or *no_music*, where each label is associated with a short excerpt of audio. Depending on the type of feature, either frame-level or clip-level, the labels might change every few milliseconds or every few seconds. An analysis of our annotated ground truth material (see section 4.1) shows that there are no music segments shorter than 3 seconds, and only 14 out of 324 music segments are shorter than 7 seconds. This indicates that music as used in TV productions lasts at least several seconds. Consequently, frame-based class labels should be aggregated into larger continuous segments of *music* or *no_music* in order to get a plausible segmentation of an audio stream. To come up with a smoothed version of the label sequence we iteratively apply (twice) a majority filter with a sliding window length corresponding to 5 seconds. In a final step, if there are any continuous label segments left that are shorter than 5 seconds, we remove them by swapping first of all the *no_music* segments shorter than 5 seconds to *music* and then the *music* segments shorter than 5 seconds to *no_music*. Altogether we smooth the label sequence in a fist step and filter out all remaining segments shorter than 5 seconds by swapping their class label. It is important to note that smoothing functions might introduce a bias by slightly favoring one of the classes.

4. EXPERIMENTS AND RESULTS

4.1. Data and Ground Truth

To be able to train our classifiers on real world situations representative for the later operation at the television station, we recorded a number of real TV telecasts. Recording was done using the DVB-T standard, and the digital digital video streams were encoded as MPEG-2. In a second stage the sound tracks of the 13 TV shows (approximately 545 minutes of audio) were converted to PCM mono at 22 kHz with a precision of 16 Bit/sample. Thereafter all audio files were annotated manually according to the class labels *music* and *no_music*, which turned out to be quite challenging, because it is often hard to tell when precisely some background music starts or stops playing. Consequently, assuming an imprecision for each change of the label of just one second, we get an upper bound on the overall classification accuracy of 98%, which is still a very optimistic estimate. Table 1 shows the distributions of the two classes for each of the 13 recorded ORF TV productions. Obviously, the amount of music present in a show depends heavily on the type of show. The baseline for the overall classification accuracy is 58.01%, which is the percentage of the more frequent class, *no_music*.

To yield a clear separation between training and test data, all the frames of an entire show must either be in the training or the test set. The first three shows, which are separated from the others in table 1, constituted the training set. Such a separation prevents a bias of learning algorithms towards specific characteristics of a single broadcast, e.g. the voice of the moderator, which would lead to too optimistic results.

4.2. Prediction Experiments

One of our interests was to find out if we can achieve any improvement by using clip-level features generated out of frame level features instead of using the frame-level features themselves (see section 3.2). To do so, we extracted both frame-level (with a window size of 24ms) and clip-level features (with a window size of 1172ms) for all 13 audio streams. Our current framework makes use of the `WEKA`[10] machine learning library. We used five (very) different `WEKA` classifiers to evaluate the features via machine learning experiments. The simple nearest-neighbor classifier *IBk* was chosen as a representative of instance-based learning methods, *Support Vector Machines (SMO)* for kernel-based machine learning methods, *MultilayerPerceptron* as the most popular representative of the neural network family of classifiers, and *REPTree* and *RandomForest* for decision tree learners. For each of these classifiers we computed the overall classification accuracy on the test set (approximately 372 minutes of audio) after learning from the independent training set.

title	type	% music	min
Der Volksanwalt	law show	1.48 %	35
Starmania	music show	50.18 %	89
Sturm der Liebe	soap opera	70.52 %	49
Alpen Donau Adria	documentary	57.08 %	30
Barbara Karlich Show	talk show	7.51 %	57
Da wo es noch Treue gibt	soap opera	62.90 %	89
Frisch gekocht	cooking show	10.01 %	24
Gut beraten Österreich	talk show	8.76 %	18
Heilige Orte	documentary	54.34 %	44
Heimat fremde Heimat	documentary	29.72 %	30
Hohes Haus	parliament show	17.50 %	30
Julia	soap opera	80.36 %	43
ZIB	news show	4.91 %	7
total	–	41.99 %	545

Table 1: *The ground truth data.*

classifiers	frame level	clip level
IBk	**69.94** %	66.47 %
MultilayerPerceptron	**69.67** %	65.99 %
SMO	69.48 %	**73.27** %
REPTree	64.07 %	**64.48** %
RandomForest	70.66 %	**73.19** %

Table 2: *Frame level versus clip level features.*

Table 2 shows the results. They seem to strongly depend on the type of classifier. No general advantage of clip-level features compared to frame-level features could be shown by our experiments. All further experiments are based on frame-level features.

The second experiment investigated the benefits of smoothing. The classification results before and after the application of the smoothing function are compared in table 3. For all of the five classifiers a substantial improvement of the classification result could be shown.

In general, applying smoothing functions increases the accuracy, but tests with various smoothing functions indicate that more sophisticated smoothing does not seem to improve the classification results any further. Figure 4 shows the classification results of *"Julia"* before and after the application of the smoothing function. Even visually it is quite obvious that the aggregation of the frame level classifications makes sense.

classifier	no smoothing	smoothed
IBk	69.94 %	**79.82** %
MultilayerPerceptron	69.67 %	**81.21** %
SMO	69.48 %	**76.19** %
REPTree	64.07 %	**73.48** %
RandomForest	70.66 %	**77.20** %

Table 3: *Smoothed versus original results.*

The best overall result using the machine learning approach and various standard features was achieved with the smoothing function applied to the class predictions of the *MultilayerPerceptron*. A total accuracy of **81.21 %** can be reached with this configuration. Table 4 shows the classification accuracy for each recorded show of the test set separately.

title	% real	% est.	diff.
Alpen Donau Adria	57.08 %	19.00 %	**38.08**
Barbara Karlich Show	7.51 %	12.33 %	4.82
Da wo es noch Treue gibt	62.90 %	63.47 %	0.57
Frisch gekocht	10.01 %	22.73 %	**12.72**
Gut beraten Österreich	8.76 %	6.42 %	2.34
Heilige Orte	54.34 %	49.82 %	4.52
Heimat fremde Heimat	29.72 %	52.17 %	**22.45**
Hohes Haus	17.50 %	15.84 %	1.66
Julia	80.36 %	68.01 %	**12.35**
ZIB	4.91 %	2.84 %	2.07

Table 4: *The percentage of music really present versus the percentage estimated.*

Since in our project we have to determine the *percentage* of time where music is present within a production, the difference in percentage points is our real quality measure. Even if the machine learning approach yields an overall classification accuracy of more than 80%, the error, in terms of the difference in percentage points, is too high for some TV shows to be useful for the ORF. In general, a maximal prediction error of 5 percentage points would be desirable for the planned application, and a prediction error of 10 percentage points may be the maximum that is still considered acceptable. In table 4 all results exceeding this maximum error of 10 percentage points are highlighted. Consequently, to further improve the obtained results, a new feature especially designed for the detection of music was developed and will be introduced in the next section.

5. CONTINUOUS FREQUENCY ACTIVATION (CFA) - A NEW FEATURE FOR MUSIC DETECTION

Our experiments show that standard speech/music discrimination features work reasonably well overall, but produce rather large errors in some cases. On the other hand most of these features were not designed for this particular type of music detection task we are working on. None of these features accounts for the special characteristics of music signals. In essence, what makes music different from other sounds are structural properties. Examples of higher-level structural properties are rhythm and harmony. Music

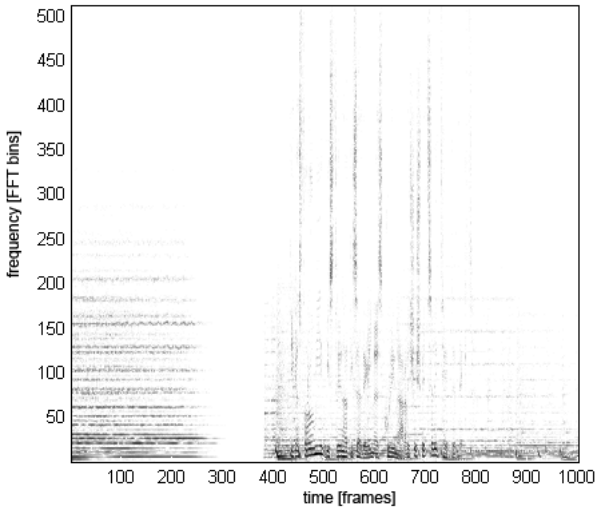

Figure 1: *Spectogram of an audio excerpt containing music and speech.*

detection might benefit from focusing on such structural properties, at various levels of the signal.

Consider for example the audio track of a movie containing some sort of background music. Because of the music being played in the background the music signal itself will be embedded very softly in the audio signal, and the global characteristics of the audio signal will more strongly resemble the characteristics of speech or noise. Features characterizing for example the frequency distribution of an audio frame will tend to model the properties of the sounds belonging to the foreground and are therefore not useful for music detection in such a case. However, we still might be able to reveal structural properties of background music, e.g. the rhythmic structure, because it is unlikly that all rhythmic events are completely masked by the foreground signal. Consequently, features focusing on the extraction of structural properties especially attributable to music might be more successful in separating *music* from *no_music* segments. In the following, we develop an intuitive feature that is meant to capture a kind of low-level structural property of musical sounds.

5.1. The basic Idea

In general music tends to have more stationary parts than speech, resulting in horizontal perceivable bars within the spectrogram representation of an audio signal (see figure 1). This property was already investigated by Hawley, who was interested in the structure of music [11] and who was the first to propose a simple *music detector* based on this. The horizontal bars in the spectrogram are continuous activations of specific frequencies and are usually the consequence of sustained musical tones. Minami et al.[3, 4] tried to construct an improved feature based on this observation. Their feature seems to work quite well for clearly distinct examples of *music* and *no_music*, but tends to fail when it comes to reliably detecting music within mixed segments containing for example speech and music. (We checked that by reimplementing their feature in our framework.) A deeper analysis of the feature led to the conclusion that concentrating on absolut energy values of the

spectogram has a counter-productive effect, because the horizontal bars might be rather soft and the absolute values of foreground sounds will have a stronger impact. Keum et al. [12] recently introduced a feature that relies on a binarization step to neglect the absolute strength of an activation. However, their binarization threshold is chosen so as to remove the small magnitudes, which is equivalent to removing all the soft activations corresponding to the soft musical tones we want to detect. In the next section a new feature is proposed to make the detection of continuous frequency activations more reliable, even if other audio signals are present simultaneously.

5.2. The feature extraction process

The computation of the *Continuous Frequency Activation* (CFA) of an audio stream can be subdivided into several steps:

- **Conversion of the input audio stream**
  The input stream is converted to 11 kHz and mono.
- **Computation of the power spectrum**
  We compute the power spectrum using a Hanning window function and a window size of 1024 samples, corresponding to approximately 100ms of audio. A hop-size of 256 samples is used, resulting in an overlap of 75% percent. After the conversion to decibel, we obtain a standard spectrogram representation.
- **Emphasize local peaks**
  To emphasize local energy peaks within each frame of the STFT we subtract from the power spectrum of each frame the running average using a window size of $N = 21$ frequency bins:

$$X_i^{emph} = X_i - \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} X_{\min(\max(k,1),N)} \qquad (6)$$

  where X_i denotes the energy of the i-th frequency component of the current frame. This step is useful to emphasize very soft tones, belonging to background music: if a soft tone is not masked by another signal over its entire duration (which is unlikely, as non-music signals tend to be less stationary), the perceivable horizontal bars in the spectogram are compositions of consecutive local maxima. Thus, we try to emphasize these soft bars by emphasizing all local maxima in the spectrum of a frame.
- **Binarization**
  To neglect the absolute strength of activation (energy) in a given frame j, we binarize each frequency component X_{ij}^{emph} by comparing to a fixed binarization threshold. The binarization threshold $t = 0.1$ was chosen in such a way that even soft activations could be kept in the binarized spectogram. Only frequency bins which are obviously not active at all, will be set to 0 using this low threshold. This is an important difference to Keum et al. [12], who apply a threshold to remove small magnitudes.

$$B_{ij} = \begin{cases} 1 & X_{ij}^{emph} > t \\ 0 & X_{ij}^{emph} \leq t \end{cases} \qquad (7)$$

  Neglecting the actual strength of the activation allows us to focus on structural aspects of the emphasized spectrogram only.
- **Computation of the frequency activation**
  We further process the binarized power spectrum in terms of *blocks*. Each block consists of $F = 100$ frames and blocks overlap by 50%, which means that a block is an excerpt of the binarized spectrogram corresponding to 2.6 seconds of audio. For each block we compute the frequency activation function $Activation(i)$. For each frequency bin i, the frequency activation function measures how often a frequency component is active in a block. We obtain the frequency activation function for a block by simply summing up the binarized values for each frequency bin i:

$$Activation(i) = \frac{1}{F} \sum_{j=1}^{F} B_{ij} \qquad (8)$$

  Normalizing the frequency activation by the length of the block is not necessary, but would make it possible to compare results from different block lengths. Figure 2 shows the binarized emphasized power spectra of two blocks and the resulting frequency activation functions. Subplot **(b)** is typical of blocks containing music, whereas subplot **(a)** is representative for blocks without any musical elements.
- **Detect strong peaks**
  Strongs peaks in the frequency activation function of a given bock indicate steady activations of narrow frequency bands. The "spikier" the activation function, the more likely horizontal bars, which are characteristic of sustained musical tones, are present. Even one large peak is quite a good indicator for the presence of a tone. The peakiness of the frequency activation function is consequently a good indicator for the presence of music. To extract the peaks we use the following simple peak picking algorithm.

  1. Collect all local peaks, starting from the lowest frequency. Each local maximum of the activation function is a potential peak (and there are many of them – cf. Figure 2).
  2. For each peak x_p, compute its height-to-width index or *peak value* $pv(x_p) = h(x_p)/w(x_p)$, where the height $h(x_p)$ is defined as $\min[f(p) - f(x_l), f(p) - f(x_r)]$, with $f(x)$ the value of the activation function at point (frequency bin) x and x_l and x_r are closest local minima of f to the left and right of x_p, respectively. The width $w(x_p)$ of the peak is given by:

$$w(x_p) = \begin{cases} p - x_l & f(p) - f(x_l) < f(p) - f(x_r) \\ x_r - p & otherwise \end{cases}$$

  Steps 1 and 2 can be done in one left-to-right scan of the activation function.
- **Quantify the Continuous Frequency Activation**
  To quantify the *Continuous Frequency Activation* of the activation function of a block, the pv values of all detected peaks are sorted in descending order, and the sum of the five largest peak values is taken to characterize the overall "peakiness" of the activation function.

As a result of this lengthy extraction process we obtain exactly one numeric value for each block of frames, which quantifies the presence of steady frequency components within the current audio segment. For blocks containing music the resulting value should be higher than for blocks where no music is present.

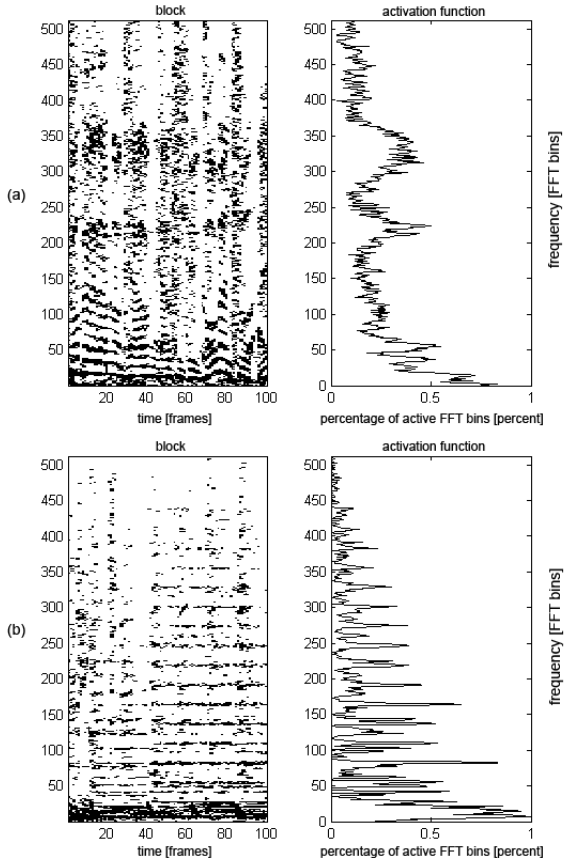

Figure 2: *Binarized spectogram of a block and the corresponding activation function. Block **(a)** contains no music, whereas in block **(b)** music is present.*

5.3. Results using the new feature

Returning just a single numeric value, the newly proposed feature simplifies the classification process a lot. The separation of the two classes *music* and *no_music* can be done by a simple comparison with a threshold t. Optimising the threshold on our training set (the top 3 shows in Table 1) yielded a value of $t = 1.24$.

Table 5 shows the percentage predictions on the test set with this threshold value, after the application of the smoothing function introduced in section 3.3. Only two estimates, highlighted in bold face, exceed the error level of 10 percentage points. To illustrate the effectiveness of the CFA, Figure 5 once more shows the automatic segmentation of *"Julia"*. It is clearly visible that the CFA feature makes far fewer mistakes even before smoothing. The classification accuracy of **81.21%** obtained with the machine learning approach improves to **89.93%**, although now just one feature and simple thresholding is used. This compares favorably to the 75.86% we reconstructed from the results reported by [1] on a related problem (see section 2 above).

Figure 3 compares the real percentages of music present, the percentages predicted by the machine learning approach, and the percentages estimated using CFA alone. There are still some cases where the CFA feature fails. Especially when the continuous frequency activations are a consequence of continuous noise signals, such as e.g. helicopter noise, the CFA feature wrongly detects music segments. Again, some examples of those misclassifications can be found at (`http://www.cp.jku.at/people/seyerlehner/md.html`).

title	% real	% est.	diff.
Alpen Donau Adria	57.08 %	48.61 %	8.47
Barbara Karlich Show	7.51 %	6.64 %	0.87
Da wo es noch Treue gibt	62.90 %	63.50 %	0.60
Frisch gekocht	10.01 %	6.69 %	3.32
Gut beraten Österreich	8.76 %	5.74 %	3.02
Heilige Orte	54.34 %	42.70 %	**11.64**
Heimat fremde Heimat	29.72 %	17.33 %	**12.39**
Hohes Haus	17.50 %	9.26 %	8.24
Julia	80.36 %	76.88 %	3.48
ZIB	4.91 %	0 %	4.91

Table 5: *The percentage of music really present versus the percentage estimated using Continuous Frequency Activation.*

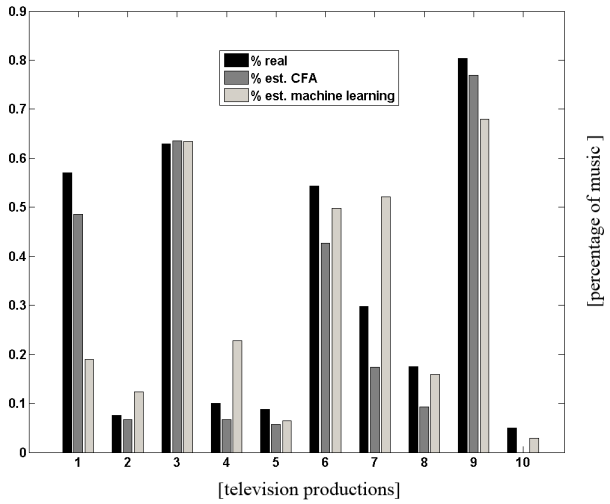

Figure 3: *Comparison of the real percentage of music, the machine learning estimate and the CFA estimate (see tables 4 and 5; The numbering of the television productions corresponds to the rows in these tables.)*

We also tested the CFA feature on a different set of reference data, namely, the Scheirer-Slaney database [13], which consists of 245 samples of radio recordings (which are presumably easier to classify than our data). To our knowledge, this is currently the only dataset of this kind that is publicly available.[2] The dataset was split into a training and a test set by Dan Ellis and is described in detail in [15]. The only change we made was to reduce the classes to *music* and *no_music* only. Based on this training set of 184 examples, we found an empirical threshold of $t = 1.05$. Using this threshold, 60 out of the 61 examples of the test set were classified correctly – a classification accuracy of 98.36%, which is

[2] We hope to get the permission by the Austrian National Broadcasting Corporation (ORF) to make our ground truth data available online.

roughly comparable to the results reported in [14].

6. CONCLUSIONS

In this paper we introduced a new music detection application, namely music detection in TV productions, and pointed out that this application differs from common speech/music classification problems. Our experiments show that standard speech/music discrimination features in combination with standard machine learning algorithms yield interesting, but highly varying results. Therefore we focused on the development of more reliable features.

Extending standard frame-level features to clip-level features, thus incorporating some rudimentary temporal information, seems not to be a successful strategy. On the other hand, our experiments show that the application of an appropriate smoothing function results in plausible segmentations and improves the overall accuracy considerably.

We then introduced a new feature which was especially designed to detect music in an accurate and robust way. This raised the total accuracy on our highly non-trivial test set to **89.93%**. Surprisingly, a simple thresholding approach based on this new feature alone outperforms the machine learning approach. This supports the thesis that music detection can be further improved if one makes use of the structural aspects of music, which even allows the detection of background music.

We have plans to further optimize the parameter settings of the *Continuous Frequency Activation* and to develop other features exploiting structural properties of music signals. One interesting direction might be to focus on rhythmic properties, as Scheirer and Slaney [13] and Jarina et al. [16] have already tried. With respect to the current application, we will also investigate combinations of speech/music discrimination features and the CFA feature and hope to deploy an operational music detection system at the Austrian National Broadcasting Corporation (ORF) in the near future.

7. ACKNOWLEDGMENTS

We would like to thank Harald Frostel and Richard Vogl, who implemented large parts of the experimentation framework built around the `WEKA`[10] machine learning environment. Thanks to Dan Ellis for making his dataset available. This research was supported in part by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant L112-N04.

8. REFERENCES

[1] M.D. Santo, G. Percannella, C. Sansone, and M. Vento, "Classifying audio of movies by a multi-expert system," in *Proc. of the 11th International Conference on Image Analysis and Processing (ICIAO'01), Palermo, Italy*, 2001, pp. 386–392.

[2] M.K.S. Khan, W.G. AI-Khatib, and M. Moinuddin, "Classifying audio of movies by a multi-expert system," in *Proc. Proceedings of the 2nd ACM international workshop on Multimedia databases (MMDB'04), Washington, DC, USA*, 2001, pp. 386–392.

[3] K. Minami, A. Akutsu, H. Hamada, and Y. Tonomura, "Video handling with music and speech detection," *IEEE MultiMedia*, vol. 5, no. 3, pp. 17–25, 1998.

[4] K. Minami, A. Akutsu, H. Hamada, and Y. Tonomura, "Enhanced video handling based on audio analysis," in *Proc. of the 1997 International Conference on Multimedia Computing and Systems (ICMCS'97), Washington, DC, USA*, 1997, pp. 219–226.

[5] J. Mauclair and J. Pinquier, "Fusion of descriptors for speech / music classification," in *Proc. of the 12th European Signal Processing Conference (EUSIPCO'04), Vienna, Austria*, 2004.

[6] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky, "Spectral entropy based feature for robust asr," in *Proc. of of the 29th International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04), Montreal, Canada*, 2004, pp. 193–196.

[7] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis, "A computationally efficient speech/music discriminator for radio recordings," in *Proc. of of the 7th International Conference on Music Information Retrieval (ISMIR'06), Victoria, Canada*, 2006, pp. 107–110.

[8] H. Ezzaidi and J. Rouat, "Speech, music and songs discrimination in the context of handsets variability," in *Proc. of of the 7th International Conference on Spoken Language Processing (ICSLP'02), Denver, USA*, 2002, pp. 16–20.

[9] M.K.S. Khan and W.G. Al-Khatib, "Machine-learning based classification of speech and music," *Multimedia Systems*, vol. 12, no. 1, pp. 55–67, 2006.

[10] I.H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, 2nd Edition*, Morgan Kaufmann, 2005.

[11] M. Hawley, *Structure Out of Sound*, Ph.D. thesis, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences, 1993.

[12] J. Keum and H. Lee, "Speech/music discrimination based on spectral peak analysis and multi-layer perceptron," in *Proc. of the 9th International Conference on Hybrid Information Technology (ICHIT'06), Cheju Island, Korea*, 2006, pp. 56–61.

[13] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. of the 22th International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97), Munich, Germany*, 1997, pp. 1331–1334.

[14] G. Williams and D. Ellis, "Speech/music discrimination based on posterior probability features," in *Proc. of the 6th European Conference on Speech Communication and Technology (EUROSPEECH'99), Budapest, Hungary*, 1999, pp. 687–690.

[15] D.P.W. Ellis, "The Music-Speech Corpus," Available at http://labrosa.ee.columbia.edu/sounds/musp/scheislan.html, Accessed March 21, 2007.

[16] R. Jarina, N.O. Connor, S. Marlow, and N. Murphy, "Rhythm detection for speech-music discrimination in mpeg compressed domain," in *Proc. of the 14th International Conference on Digital Signal Processing (DSP'02), Hellas, Greece*, 2002, pp. 129–132.

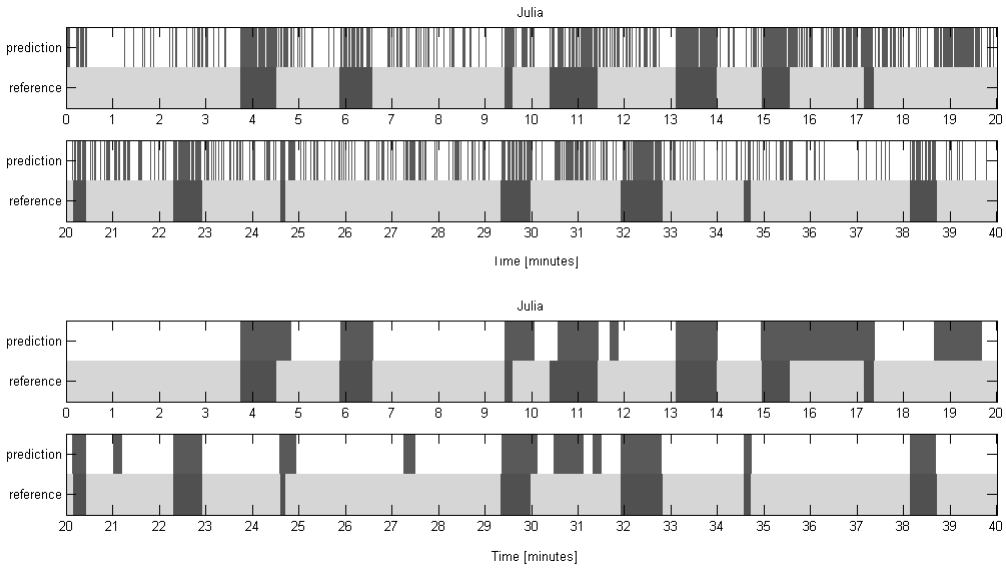

Figure 4: *Visualization of the classification results for the **machine learning approach** of 40 minutes of the soap opera "Julia". Each line represents 20 minutes of audio and is split to compare the class prediction with the true class. The class "music" is represented by a lighter color, whereas "no_music" is in the form of dark regions. The lower subplot illustrates the results after the application of the smoothing function.*

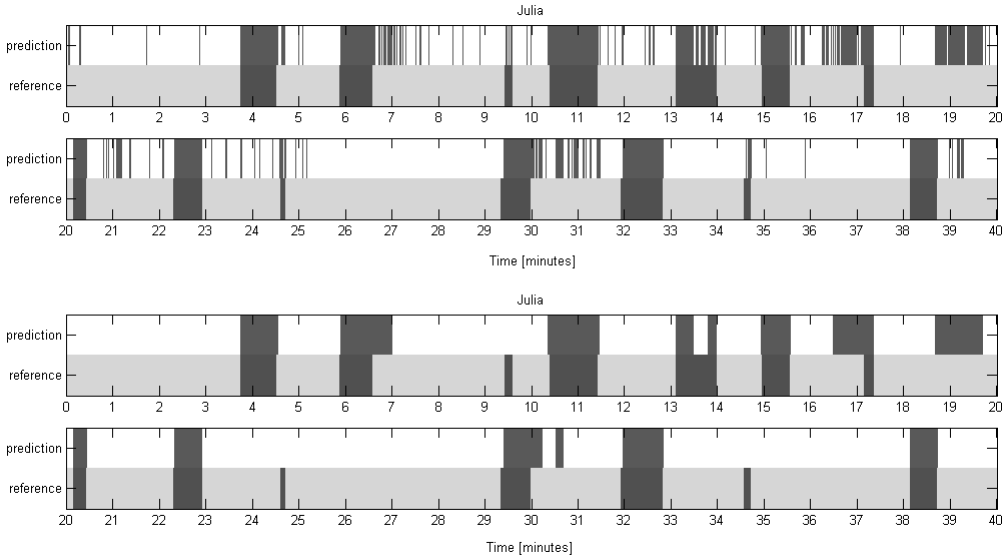

Figure 5: *Visualization the classification results for the new **CFA** feature of 40 minutes of the soap opera "Julia". Each line represents 20 minutes of audio and is split to compare the class prediction with the true class. The class "music" is represented by a lighter color, whereas "no_music" is shown in the form of dark regions. The lower subplot illustrates the results after the application of the smoothing function.*