長岡技術科学大学大学院
工学研究科修士論文

題　　目

楽曲特徴を考慮した
DNN ベース楽器音分離に関する研究

指導教員

杉田 泰則　准教授

著　　者

電気電子情報工学専攻

17315183　　田野﨑 蓮

令和 3 年 2 月 9 日

ABSTRACT

A DNN-based music source separation method considering music characteristics

Author : Ren TANOZAKI
Supervisor : Yasunori SUGITA

In recent years, the way to enjoy music has been changing, such as not only by listening to it but also by arranging it by oneself. Music source separation, which extracts indivisual instrument sound from a mixture sound source, has become an inportant task, and it is expected to be used as an preprocessing method for automatic transcription, and an method for creating karaoke sound by removing vocals from live sound sources.

Recently, DNN(Deep Neural Network) have shown a improvement of performance for music source separation, especially, used for blind sound separation(BSS), which is to separate sound without using location information of sources. Usually, only amplitude spectrogram are used as input features for music source separation using DNNs. On the other hand, in the field of speech synthesis, it is known that the use of speaker codes in addition to speech features during training produces better results than the method using only speech features. Therefore, it is thought that combining amplitude spectrograms with other musical features may improve the separation accuracy in source separation as well.

This paper proposed a method of inserting music information, such as genre information, into DNN-based separation models in order to utilize information obtained from music to improve the performance of music source separation models. It is expected that the model will obtain the appropriate parameters for the input music by training the separation model simultaneously with the music-dependent information. The proposed method uses three types of music information: genre information, BPM, Flatness, and trains separation models with suitable network architecture for each of information. The genre information is manually labeled, and the BPM and Flatness are extracted from the input soruce.

Using a dataset for music source separation called DSD100, I compare the performance of separation of each proposed method with that of the conventional method, MMDenseLSTM. In the results, I found that the proposed method of inserting the genre information outperforms the conventional method for three instrument sounds: Bass, Drums, and Vocals. In addition, the separation performance of the BPM insertion method was lower than that of conventional method, while the Flatness insertion method showed a slight improvement in Vocal.

目次

第 1 章

はじめに

1.1 研究背景

近年，音楽をただ聴くだけでなくユーザー自身の手でアレンジを施すなど音楽の楽しみ方が変化してきている．そうした中で，様々な楽器音が混ざった音源から各楽器音を抽出する楽器音分離は重要な課題となっている．ライブ音源からボーカルを取り除くことによるカラオケ音源の作成や，音楽の音源から各楽器の譜面を作成する自動採譜技術のための前処理としての活用などが期待されている．

楽器音分離の手法としては，ビームフォーミングや非負値行列因子分解 [1–3]，多チャンネルウィナーフィルタ [4] 等が使用されてきた．最近では DNN(Deep Neural Network) を用いたモデルが高い分離性能を誇っており，特に劣決定ブラインド音源分離と呼ばれる位置情報が不明かつ音源の数がマイク数より少ない信号を分離することが目的のときによく使われる．これは，DNN は音源情報を扱うことに長けており [5]，位置情報や多数のマイク入力による空間的情報を必要としなくとも高精度の楽器音分離が可能なためである．DNN を用いた楽器音分離の例として，Nugraha ら及び Uhlich らの研究 [6, 7] では FNN(Feed-forward Neural Network) を用いて楽器音分離を行っている．その際，時間的コンテキストを考慮するために連続したスペクトルを数フレームまとめて入力としている．LSTM(Long-Short Term Memory) を用いて楽器音分離を行っている手法 [8] もあり，STFT(Short Time Fourier Translation) より得られる振幅スペクトログラムを時系列順に入力することで FNN よりも長い時間的情報を扱うことが可能となっている．また，Uhlich らの研究では FNN と LSTM の出力を線形結合することで分離精度を向上させており，異なる DNN を組み合わせることで楽器音分離の結果がより良くなることを示している [8]．他に，MMDenseLSTM [9] は DenseNet [10] と BLSTM(Bidirectional LSTM) を組み合わせた手法であり，Multi-scale と Multi-band といった特徴により Uhlich らの研究 [8] よりも少ないパラメータでより高精度な分離

結果を出している．MMDenseLSTM の入力も振幅スペクトログラムであり、連続した複数フレームを１つの入力としている．

一般的に DNN を使用した楽器音分離の入力特徴量には振幅スペクトログラムのみを用いることが多いが，音声合成の分野においては学習の際に音声特徴量に加えて話者コードと呼ばれる話者固有の符号を用いることで従来手法よりも良い生成結果を出すことが知られている [11]．また，文献 [8] の結果では FNN のみの分離結果，BLSM のみの分離結果，FNN+BLSTM の分離結果でそれぞれ得意とする楽曲のジャンルが異なることが示されており，そこから，ジャンル情報が楽器音分離の精度に影響を与える可能性が考えられる．

1.2　研究目的

本論文では，楽器音分離モデルの性能向上のためにジャンル情報などの楽曲から得られる情報を活用することを目的とし，楽曲情報を DNN ベースの分離モデルに挿入する手法を提案する．話者コードを用いた音声合成では生成モデル内の話者依存パラメータが話者コードにより当該話者に適した形に変更されるためにモデルの性能が向上すると考えられており，楽器音分離においても楽曲固有の情報と同時に分離モデルを学習することで入力される楽曲の特徴に適したパラメータとなることを期待する．提案手法の有用性を確認するため，DSD100 と呼ばれるデータセットを用いて従来手法との比較実験を行う．

1.3　本論文の構成

本論文の構成は以下の通りである．第 1 章では本論文の背景及び目的について述べた．第 2 章では楽器音分離の従来手法について述べる．第 3 章では提案手法である楽曲特徴を挿入した DNN ベースの分離モデルについて述べる．第 4 章では従来手法と提案手法の比較実験を行い，提案手法の有用性について示す．第 5 章では本論文を通してのまとめと今後の課題について述べる．

第 2 章

関連研究

2.1 MMDenseNet

この手法 [12] は画像認識タスクで良い成果を出している DenseNet [10] を拡張した分離モデルである．DenseNet の基本はある層の出力にそれ以前の層の出力がつなげられることであり，これによって前の層で計算された特徴量を再利用できる：

$$x^{(l)} = H([x^{(l-1)}, x^{(l-2)}, ..., x^{(0)}]) \tag{2.1}$$

ここで，$x^{(l)}$は l 層目のネットワークの出力，$H(\cdot)$ は非線形変換，$x^{(0)}$はネットワークへの最初の入力であり，[...] は変数の結合を意味する．楽器音分離の目的は干渉音に埋もれた楽器のスペクトログラムを推定することであるため，入力音源や以前の層の出力を参考にして各層の特徴量をブラッシュアップできる DenseNet の性質は楽器音分離に適していると示されている [12]．画像認識タスクで良く使われている ResNet [13] は 1 つ前の層の出力だけしか参考にすることができず，この点が DenseNet と異なる．ただし，DenseNet は層の深さが増すにつれて層間結合の数が指数関数的に増加するため，多くのメモリを要求するという欠点がある．

高橋らが提案した MMDenseNet はこの問題を解決するために Multi-scale を導入し，加えて，モデルの性能向上のために Multi-band を導入している．それぞれの特徴について以下の小節で説明する．

2.1.1 Multi-Scale DenseNet

Multi-Scale DenseNet は図 2.1 で示される dense block とダウン (アップ) サンプリング層で構成される．comp. layer はバッチ正規化, 活性化関数, 畳み込み層で構成されている．ダウンサンプリングを行い低解像度の特徴マップを作成することで，計算コストを低減しつつ層を深くすることが可能となり，より長時間のコンテキストと広範囲

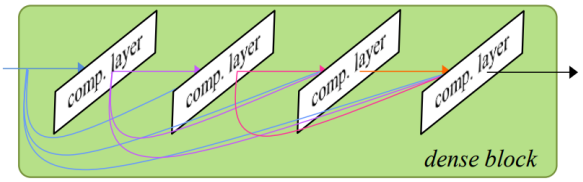


図 2.1　dense block の構成 (文献 [12] より引用)

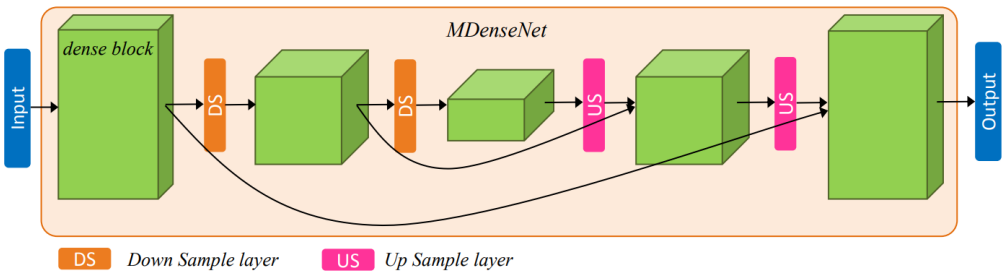


図 2.2　MDenseNet の構成 (文献 [12] より引用)

の周波数特性をモデル化することができる．作成された低解像度の特徴マップはアップサンプリングによって高解像度に復元される．このとき，低解像度に圧縮されてない特徴量を復元に使用するために，同じスケールのブロック同士を結合するスキップ接続を導入している．このアーキテクチャ (図 2.2) を MDenseNet と呼ぶ.

2.1.2　Multi-Band DenseNet

高橋らはモデル機能を向上させるため、全体的な周波数スペクトラムとともに特定の周波数帯域専用の MDenseNet を導入している。周波数軸に沿った畳み込みは音響ドメインにおいて効果的であるが、スペクトログラムの局所的なパターンは異なる周波数域で異なることが良くあるとされている：

- 低周波数域には、高いエネルギー、調整、長時間持続する音が含まれる可能性が高い
- 高周波数域には低いエネルギー、ノイズ、急速に減衰する音を持つ傾向がある

多くの畳み込みカーネルは高エネルギー帯域に焦点を合わせ，より低いエネルギー帯域を無視するために結果として復元に失敗するとされている．そのため，各帯域専用のネットワークを提供している．このアイデアを使用したアーキテクチャ (図 2.3) を MMDenseNet と呼ぶ.

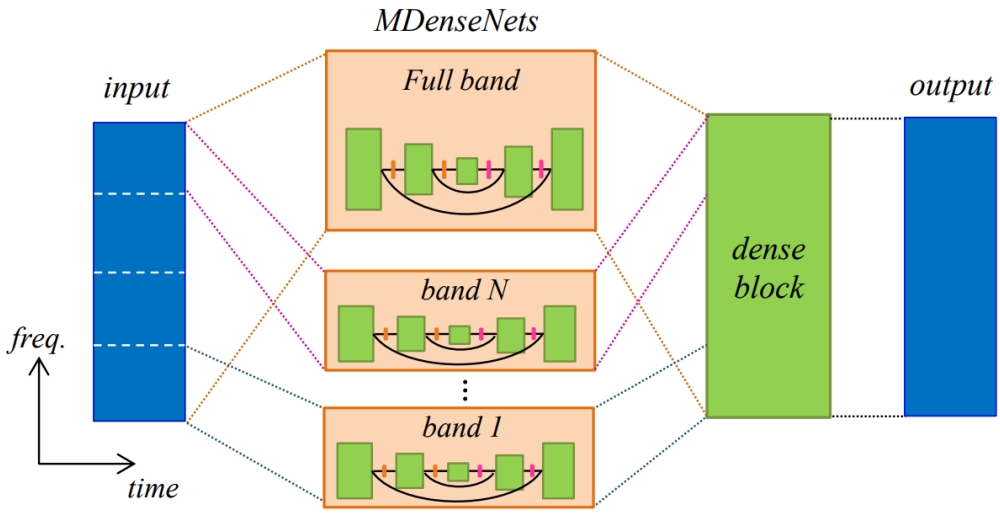


図 2.3　MMDenseNet の構成 (文献 [12] より引用)

2.2　MMDenseLSTM

高橋らは音源分離でよい成果を上げた MMDenseNet [12] を強化し、音響コンテキスト内の長期的構造を効率的にモデル化する新しいアーキテクチャとして MMDenseLSTM を提案している [9]．MMDenseLSTM は，CNN ベースのネットワーク構造である MMDenseNet と，RNN 的な構造である BLSTM をスキップ接続で連結したモデルであり，歌声分離タスクにおいて ideal binary mask よりも良い結果を出している．

Uhlich らの研究によると異なる DNN アーキテクチャを融合することでより良い性能を出すことが示されている [8] が，異なるアーキテクチャの出力の融合はモデルサイズ及び計算時間を増加させるため，MMDenseLSTM は 1 つのネットワーク構造の中で DenseNet と LSTM ブロックを組み合わせている．ここで，LSTM ブロックは以下の要素で構成される．

- 1x1 畳み込み: 特徴マップの数を 1 に減らす
- BLSTM: 時間軸の連続データとして特徴マップを扱う
- 全結合層: BLSTM の出力から周波数次元のデータに戻す

LSTM ブロックは MMDenseNet 内の各 DenseNet の後ろに接続され，またモデルサイズを増大させないために高解像度を扱う階層の DenseNet とは接続されない．ネットワークの全体構造を図 2.4 に示す．

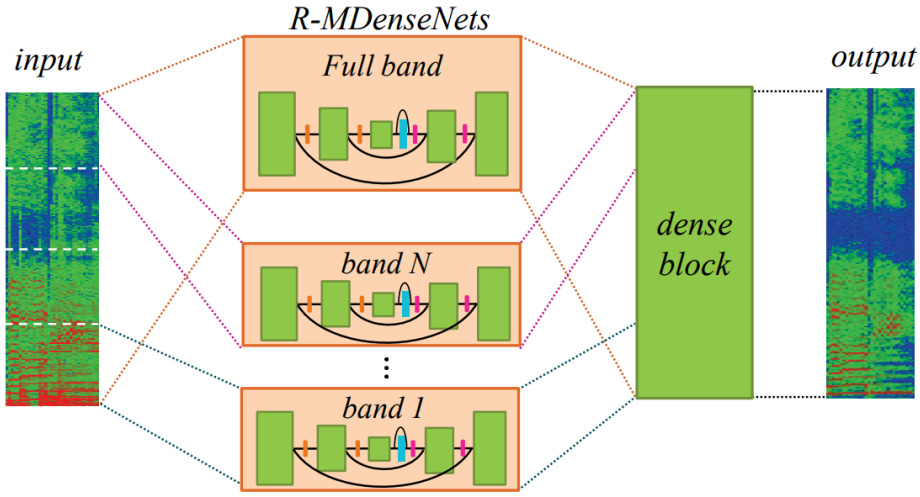


図 2.4　MMDenseLSTM の構成 (文献 [9] より引用)

第 3 章

提案手法

3.1 ジャンル情報の挿入

MMDenseLSTM 内にジャンル情報を挿入する方法として，まずは話者コードを用いた音声合成の研究 [11] と同様にネットワークの各層に one-hot vector 形式のジャンル情報を結合する方法を考えた．しかし，MMDenseLSTM の各層には BN(Batch Normalization) [14] が組み込まれており，ジャンル情報の挿入を行う際に one-hot vector が BN の作用に悪影響を与えるという問題が発生する．BN がネットワークの各層において 1 ミニバッチ内の同一チャネルに含まれる潜在変数を平均 0, 分散 1 に正規化し学習の効率化を図るものであるのに対し，one-hot vector はある個所が 1, その他の箇所が 0 であるベクトルデータであるため，潜在変数を正規化するためのパラメータがジャンル情報に大きく依存してしまい，本来行いたい正規化が出来ない可能性があると考えられる．そこで，one-hot vector と BN が互いに干渉しないネットワーク構造を 4 種類提案する．

3.1.1 ネットワークの一部にジャンル情報を挿入

MMDenseLSTM のネットワーク構造における DenseNet と BLSTM の結合部分に楽曲のジャンル情報を挿入し楽器音分離の学習を行うモデル (図 3.1) を提案する．この個所には元から BN が採用されていないため，one-hot vector を挿入しても悪影響を及ぼさないと考える．ジャンル情報は one-hot vector の形で与えられ，DenseNet と BLSTM の出力と結合された上で以下の式で非線形変換が行われる．

$$\boldsymbol{h}(t,f) = \sigma(\boldsymbol{W}(t,f) \cdot \boldsymbol{h}_{cat}(t,f)) \tag{3.1}$$

$$\boldsymbol{h}_{cat}(t,f) = [\boldsymbol{h}_{dense}(t,f), \boldsymbol{h}_{lstm}(t,f), \boldsymbol{c}_{genre}] \tag{3.2}$$

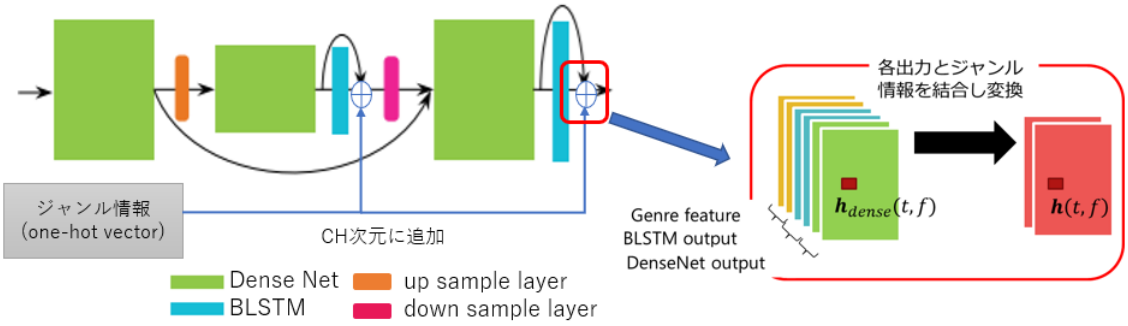


図 3.1　ジャンル情報を挿入するネットワーク構成

ここで，t はある時間フレーム，f はある周波数ビン，$\boldsymbol{h}$ は結合後の特徴量，$\boldsymbol{h}_{dense}$はDenseNetの出力，$\boldsymbol{h}_{lstm}$はBLSTMの出力，$\boldsymbol{c}_{genre}$はジャンル情報，$\boldsymbol{W}$ は結合重み行列，$\sigma(\cdot)$ は活性化関数である．

また，短期的コンテクストを扱う DenseNet と長期的コンテクストを扱う BLSTM の結合が MMDenseLSTM の重要な箇所であるため，そこにジャンル情報を挿入することでより効果的な学習が行われることを期待する．

3.1.2　DenseNet 構造の変更

ジャンル情報を one-hot vector で表現しながら MMDenseLSTM の各層に挿入する場合，ネットワーク構成を変更する必要がある．図 3.2 に示す DenseNet 内のネットワーク構成を見ると，DenseNet を構成する comp. layer の一番上の層が BN 層であるため，one-hot vector を含む潜在変数が直接バッチ正規化されてしまう．そこで，comp. layer に使われている BN 層の位置を図 3.3 に示す通り変更することを提案する．これにより，one-hot vector が結合された潜在変数が一度変換された後に BN 層に入力されるため，バッチ正規化のパラメータを大きく歪めることはないと考えられる．ここで，conv 層と BN 層が連続すると，学習時に conv 層のバイアスパラメータと BN 層のパラメータが互いに干渉しあうため，conv 層のバイアス項を削除することに注意する．

3.1.3　ジャンル情報挿入前に Linear 層を追加

ジャンル情報といった数値でないデータを DNN に入力する際の埋め込み表現として one-hot vector が用いられることは一般的であるのだが，0 と 1 のみで構成されるベクトルを含む潜在変数が BN に入力されることは好ましくないため，ジャンル情報を実数ベクトルで表現する必要がある．そこで，one-hot vector をそのままネットワーク

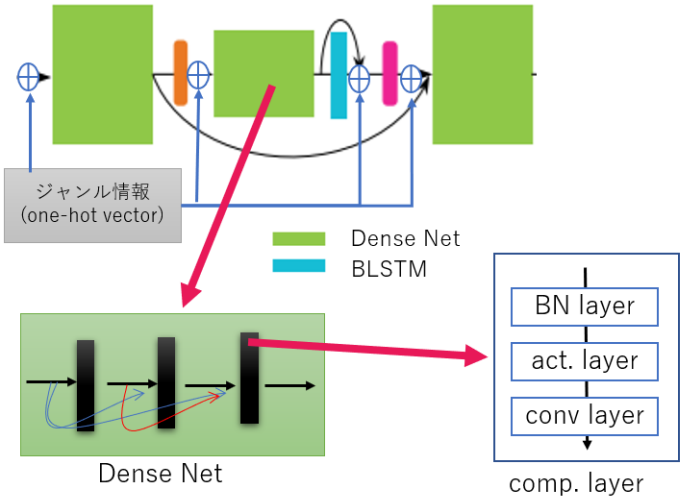


図 3.2　DenseNet の構成

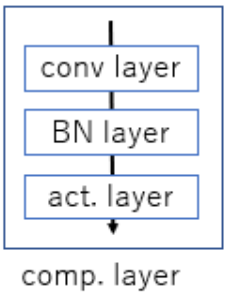


図 3.3　変更後の comp.layer

に挿入するのではなく，図 3.4 に示すように Linear 層を挟むことでジャンル情報を実数ベクトルである分散表現に変換し，その後ネットワークへの挿入を行う．このとき，MMDenseLSTM の l 番目の階層における分散表現 $\boldsymbol{v}_{genre}^{(l)}$ を出力するための Linear 層は以下の式で表す非線形変換を行う．

$$\boldsymbol{v}_{genre}^{(l)} = \sigma(\boldsymbol{W}_{genre}^{(l)} \cdot \boldsymbol{c}_{genre} + \boldsymbol{b}_{genre}^{(l)}) \tag{3.3}$$

ここで，$\boldsymbol{b}_{genre}^{(l)}$ はバイアスベクトルである．また，得られた分散表現 $\boldsymbol{v}_{genre}^{(l)}$ は以下の式でネットワークの潜在変数と結合される．

$$\boldsymbol{h}(t,f) = \sigma(\boldsymbol{W}^{(l)}(t,f) \cdot [\boldsymbol{h}_{dense}^{(l-1)}(t,f), \boldsymbol{v}_{genre}^{(l)}]) \tag{3.4}$$

このときの $\boldsymbol{h}(t,f)$ は l 層目の DenseNet の入力となる．

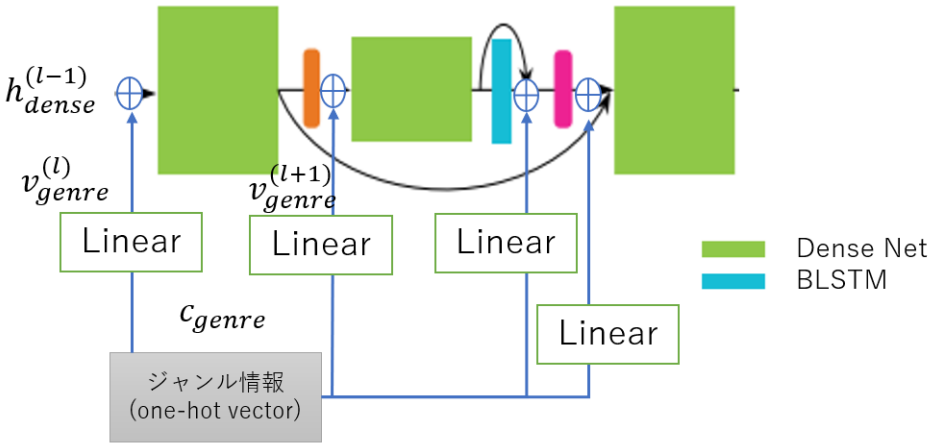


図 3.4　Linear 層を追加したネットワーク構成

3.1.4　BN の削除

one-hot vector 形式に限らず，ネットワークの途中にデータを挿入すること自体が BN との相性が悪い可能性を考え，ネットワーク構成から BN を削除する．しかし，BN はモデルの正則化を行っているため，削除してしまうと分離モデルが過学習してしまう可能性がある．そこで，過学習を抑制する方法として，MMDenseLSTM の出力層を除いた各 conv 層の後に dropout を追加する．

3.2　BPM の挿入

3.1 節で説明したジャンル情報は楽曲の特徴を表すラベルの一つではあるが，楽器音分離をする際に必ず取得できるとは限らないため，3.1 節の手法は事前にジャンルがラベリングされている楽曲だけにしか適用できないという問題がある．そこで，楽曲の音源から抽出した特徴量を楽器音分離モデルに挿入することを提案する．本節で BPM [15] の挿入について，次節で Flatness [16] の挿入について説明する．

BPM(Beat Per Minute) は一分間の拍数のことを言い，音源がどの程度の速さであるかを表す指標である．BPM はジャンル分類等に用いられることもあり [17]，また打楽器が発する音とも関係があるため，楽器音を分離する際に有効ではないかと考えた．

本論文では，入力されるスペクトログラムから抽出 [15] した BPM を楽器音分離に挿入する際に以下の 2 つの方法を用いた．

1. BPM を [0, 1] に正規化したのちに分離モデルに挿入する
2. BPM を one-hot vector に変換したのちに分離モデルに挿入する

1. の方法では，BPM は 300 で割られることで [0, 1] の範囲に正規化されることとなる．音楽における BPM の数値は一般的に 300 を超えることは無いと考え，この数字を使用した．

2. の方法では BPM の範囲を [〜80, 81〜90, ..., 171〜180, 181〜] と振り分け，入力音源の BPM が対応する部分が 1，それ以外が 0 である 11 次元の one-hot vector を作成する．

それぞれ作成した BPM 特徴量は 3.1.1 節と同様に DenseNet と BLSTM の結合部分に以下の式の通り挿入される．

$$\boldsymbol{h}(t,f) = \sigma(\boldsymbol{W}(t,f) \cdot \boldsymbol{h}_{cat}(t,f)) \tag{3.5}$$

$$\boldsymbol{h}_{cat}(t,f) = [\boldsymbol{h}_{dense}(t,f), \boldsymbol{h}_{lstm}(t,f), BPM] \tag{3.6}$$

ここで，BPM は [0, 1] に正規化された数値か，11 次元の one-hot vector のいずれかとなる．

3.3　Flatness の挿入

Flatness はパワースペクトルの平坦度を表す指標であり，ホワイトノイズのような平坦な場合には 1 に近づく．Flatness を算出するためには，ある時刻におけるパワースペクトルを任意のサブバンド数に分け，各サブバンドにおけるパワーの値を合計したあと，全サブバンド数の相乗平均を相加平均で除算することで求めることができる [18]．算出式を以下に示す．

$$Fla = \frac{\sqrt[\lambda]{\prod_{\lambda=0}^{\Lambda} \rho(\lambda)}}{\frac{1}{\Lambda}\sum_{\lambda=0}^{\Lambda} \rho(\lambda)} \tag{3.7}$$

ここで，Λ はサブバンド数，ρ_λ は λ 番目のサブバンドにおけるパワーの合計値を表す．Flatness は楽曲のジャンル分類やクラスタリングを行う際に特徴量として用いる研究 [19,20] があり，その中で有意差のある結果を出している場合もあるため，楽曲を特徴づける情報であると考えた．

本論文では，入力されるスペクトログラムから抽出した Flatness を楽器音分離モデルに挿入する際に以下の 2 つの方法を用いた．このとき，入力音源がステレオ音源であるため，Flatness も L と R の両方が作成される．

1. Flatness の数値を振幅スペクトログラムと同時に入力する
2. FNN で次元圧縮を行った後に分離モデルに挿入する

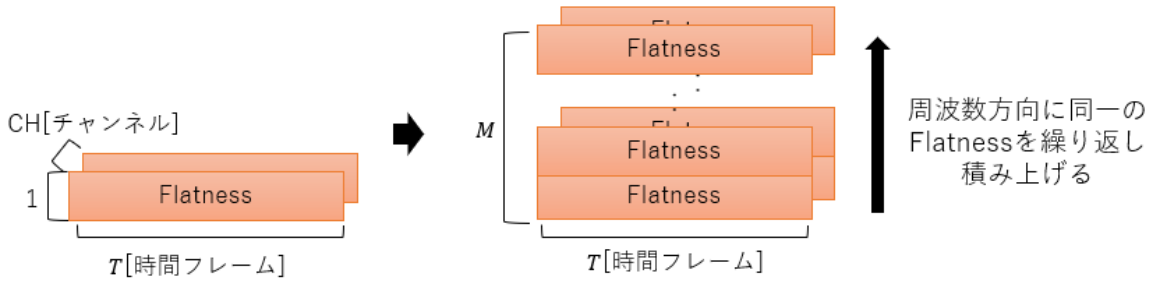


図 3.5　Flatness の拡張

1. の方法では，Flatness と振幅スペクトログラムをチャンネル次元で結合したものを分離モデルの入力とする．振幅スペクトログラムの大きさが (CH, T, M) であるのに対し，Flatness の大きさは (CH, T) であるため，チャンネル次元で結合するには Flatness を図 3.5 のように周波数軸の方向に拡張する必要がある．

2. の方法では，3.1.1 節と同様に DenseNet と BLSTM の結合部分に Flatness を挿入する．しかし，Flatness の次元数はジャンル情報や BPM と比べて大きいため，DNN を使用して次元削減を行う．具体的には，図 3.6 に示す 3 層の FNN を用いて Flatness の次元を削減した後，以下の式の通りに分離モデルに挿入される．

$$\boldsymbol{h}(t,f) = \sigma(\boldsymbol{W}(t,f) \cdot \boldsymbol{h}_{cat}(t,f)) \tag{3.8}$$

$$\boldsymbol{h}_{cat}(t,f) = [\boldsymbol{h}_{dense}(t,f), \boldsymbol{h}_{lstm}(t,f), H_{FNN}(Fla)] \tag{3.9}$$

ここで，$H_{FNN}(Fla)$ は Flatness の値を入力したときの FNN の出力である

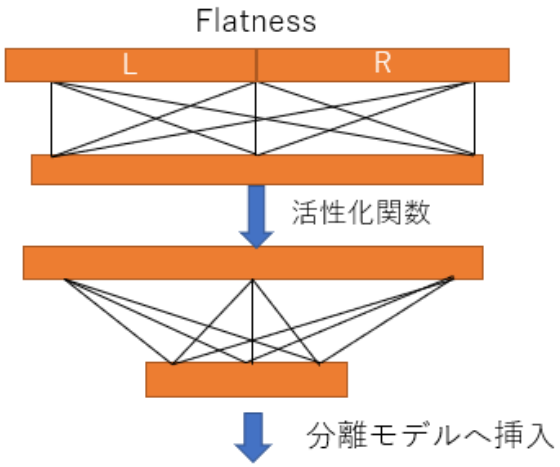


図 3.6　Flatness の次元削減

第 4 章

実験

4.1 実験条件

4.1.1 モデルの各種パラメータ

本論文で提案した楽器音分離モデルのネットワーク構成のうち，3 章で説明した楽曲情報を挿入する部分以外はベース手法である MMDenseLSTM を参考にしており，詳細なパラメータは文献 [9, 12] に記載されているものと同一である．また，入力される振幅スペクトログラムは 4.1[kHz] と 11[kHz] を境に 3 つの周波数帯に分割されており，時間フレーム長は 352 フレームとしている．活性化関数として Mish 関数 [21] を用いており，ドロップアウトを使用した際のドロップアウト率は 0.2 である。図 3.6 の FNN の次元数は上から順に 704, 352, 7 である．

4.1.2 学習条件

学習に使用した最適化関数は RAdam 関数 [22] であり，学習率を 0.0001 とした．また損失関数は平均二乗誤差を用いた．ミニバッチ数は 1 であり，学習回数は 40[epoch] である．

4.1.3 データセット

本論文では，楽器音分離モデルの学習のために DSD100 を用いた．DSD100 は全 100 曲からなる楽器音分離用データセットであり，各楽曲は Mixture, Vocal, Bass, Drums, Other の 5 つの音源で構成される．そのうち 50 曲を学習用，50 曲を評価用として使用した．各音源はステレオ音源であり，サンプリング周波数は 44.1[kHz] である．また，楽曲を Country, Heavy Metal, Hip-hop, Jazz, Pop, Rock, Techno, Other の 8 つのジャンルに分類した．楽曲によっては Pop かつ Rock など、複数のジャンルに分

表 4.1　学習に使用した各ジャンルに分類された楽曲数

Country	Heavy Metal	Hip-hop	Jazz	Pop	Rock	Techno	Other
5	4	3	1	9	18	8	5

類されるものも存在する．学習に使用した各ジャンルの楽曲数を表 4.1 に示す．

4.1.4　前処理

標準偏差が 1 になるように正規化した時間波形に対して短時間フーリエ変換 (STFT) を施し，振幅スペクトログラムを作成した．短時間フーリエ変換の窓サイズは 4096[sample]，オーバーラップは 75% であり，窓関数は hanning 窓を使用した．また，スペクトログラムの周波数ビン数は 2049，時間フレーム数は 352 フレームである．

4.1.5　後処理

音源分離モデルの出力である振幅スペクトログラムに逆短時間フーリエ変換 (iSTFT) を施し，時間波形に復元する．この時，復元のために使用する位相スペクトログラムは分離前の音源と同じものを使用する．その後，復元した時間波形に多チャンネル音源が持つ空間的情報を利用して楽器音分離を行う手法である Multi-channel Wiener Filter(MWF) [8] を施す．DNN を用いた楽器音分離では空間情報を扱うことを苦手とするため，後処理として MWF を使用することで分離精度の向上を図る．

4.1.6　評価指標

客観的評価指標として以下の式で表される SDR(Signal to Distorsion Ratio) を用いた．

$$\mathrm{SDR} =: 10\log\frac{\sum|\mathrm{S}|^2}{\sum|\mathrm{S}-\hat{\mathrm{S}}|^2} \tag{4.1}$$

推定信号 $\hat{S}$ が教師信号 S と比べどの程度波形が歪んでいるかを表し，数値が高いほど波形の類似度が高いことを示す．この時，推定信号と教師信号はどちらも L2 正規化された上で計算されている．

4.2　実験結果

4.2.1　ジャンル情報を挿入した楽器音分離モデルの性能評価

MMDenseLSTM をベースとした楽器音分離モデルにおけるジャンル情報を挿入するネットワーク構成として，

- PART: 3.1.1 節で説明した，DenseNet と BLSTM の結合部分のみに挿入する構成
- ALL(ONE_HOT_VECTOR): **??**節で説明した，全層に one-hot vector 形式で挿入し，comp. layer を変更する構成
- ALL(DISTRIBUTED): 3.1.3 節で説明した，全層に分散表現形式で挿入する構成
- ALL(DROPOUT): 3.1.4 節で説明した，BN を dropout に変更した構成

を提案した．各々の構成と既存手法との比較結果について表 4.2 に示す．表中の ALL はバッチ正規化に対する対策をせずに one-hot vector を全層に挿入した場合の結果である．まず，Other 以外の楽器音について PART の結果が最もよく，ベース手法と比べて Bass, Drums, Vocal において SDR が 0.1 から 0.2dB 程度上回っている．ALL(ONE_HOT_VECTOR) が Other において 0.05dB 程 PART を上回ったものの，その他の楽器音については PART の結果が一番良く，ALL(DISTRIBUTED) では全楽器音において 0.1 から 0.2dB 程度 PART の結果を下回っている．これについては one-hot vector を分散表現に変換する際にジャンル情報が劣化してしまったと考えられる．また，dropout では BN ほどの正則化が出来ずに過学習が起きてしまったのか，ALL(DROPOUT) では音源分離の精度が大きく落ちてしまった．ALL(ONE_HOT_VECTOR) では BN 層に直接 one-hot vector が入力されないようにネットワーク構成の一部を変更したが，BN 層の前に conv 層を配置したとしても one-hot vector がバッチ正規化に与える悪影響を抑えることはできなかったと考えられる．以上のことから，DenseNet と BLSTM の結合部分にジャンル情報を挿入することで分離精度が向上することがわかる．

また，各ジャンルごとにおける客観的評価の結果を図 4.1 に示す．これにより，Heavy metal，Techno の 2 つのジャンルでは全楽器音を通して提案手法の結果がベース手法を上回っていることがわかる．特に Techno は生の楽器音ではなく電子音であるシンセサイザー等の楽器音を使用しており，ジャンルに依存した情報が多いためにジャンル情報を用いることでより精度の良い分離が出来たのではないかと考えられる．

表 4.2　ジャンル情報を挿入する楽器音分離手法の比較結果

	SDR in dB			
Method	Bass	Drums	Other	Vocal
BL(mixture)	-0.19	0.005	0.49	-0.28
baseline(MMDenseLSTM [9])	3.53	4.86	3.46	5.26
ALL	3.40	4.78	3.13	4.93
PART	3.64	4.94	3.43	5.45
ALL(ONE_HOT_VECTOR)	3.57	4.86	3.48	5.32
ALL(DISTRIBUTED)	3.54	4.82	3.40	5.22
ALL(DROPOUT)	3.23	4.32	2.96	3.97
BU(ideal binary mask)	7.84	8.30	8.90	11.1

次に，ベース手法 (MMDenseLSTM) 及び提案手法 (PART) を用いた楽器音分離の生成結果を図 4.2〜4.17 に示す．それぞれ，5 秒間の振幅スペクトログラムであり，mixture は分離前の混合音源，soruce は各楽器音の教師信号，baseline はベース手法の出力結果，proposed は提案手法 (PART) の出力結果である．図 4.2〜4.5 はある Techno の楽曲の分離結果を示している．図 4.2, 4.3, 4.5 を見ると，baseline よりも proposed の方が分離したい音以外の背景音が小さくなっており，分離が進んでいることがわかる．図 4.6〜4.9 はある Heavy Metal の楽曲の分離結果を示している．図 4.6 と図 4.8 では，Thecno の楽曲と同様に proposed の方が背景音が小さくなっていることがわかる．また，図 4.9(c) を見ると，本来分離したい Vocal の楽器音のうち 4[kHz] より上の部分がうまく分離出来ていないことがわかる．Heavy Metal 特有の Vocal 音であるシャウトが上手く Vocal として認識できず，他の楽器音に含まれてしまったのではないかと考えられる．一方で，図 4.9(d) では比較的上手く分離できているため，ジャンル情報の挿入によってシャウトの分離を上手く学習できたのではないかと考えられる．図 4.10〜4.13 はある Rock の楽曲の分離結果を示している．図 4.10 と図 4.13 では proposed の方が背景音が小さくなっており上手く分離出来てるが，図 4.11 と図 4.12 ではそこまで変わらない．学習データに含まれる楽曲のうち Rock に分類される楽曲が多く，ベース手法で学習した分離モデルと Rock のジャンル情報を挿入した分離モデルのパラメータが似たものになっている可能性があるため，分離性能があまり向上しなかったのではないかと考えらえる．図 4.14〜図 4.17 はある Hip-hop の楽曲の分離結果を示している．図 4.14, 4.17 を見ると Bass と Vocal の分離は proposed の方が比較的出来ていると言えるが，図 4.16 では baseline の方が良く分離できているように

表 4.3　楽曲特徴を挿入した楽器音分離手法の比較結果

	SDR in dB			
Method	Bass	Drums	Other	Vocal
baseline(MMDenseLSTM [9])	3.53	4.86	3.46	5.26
BPM(正規化)	3.42	4.78	3.31	5.11
BPM(ONE_HOT_VECTOR)	3.54	4.56	3.25	4.77
Flatness(INPUT)	3.58	4.82	3.47	5.31
Flatness(FNN)	3.45	4.80	3.39	5.39

見える．Hip-hop の Vocal 分離は上手く学習できたが，それ以外の楽器音についてはあまり上手く学習できなかったと考えられる．学習に使用したジャンルのうち，Jazz と Hip-hop の楽曲数が少なかったため，それぞれのジャンルに使われている楽器の特徴を学習しきれなかったのではないかと考えられる．

4.2.2　楽曲から抽出した情報を挿入した楽器音分離モデルの性能評価

楽曲の音源から抽出した特徴量を楽器音分離モデルに挿入する手法として，

- BPM(正規化): 3.2 節で説明した，BPM を正規化して挿入する手法
- BPM(ONE_HOT_VECTOR): 3.2 節で説明した，BPM を one-hot vector 形式で挿入する手法
- Flatness(INPUT): 3.3 節で説明した，Flatness を混合音源と同時に入力する手法
- Flatness(FNN): 3.3 節で説明した，Flatness を FNN に通した後に挿入する手法

を提案した．各々の手法における楽器音分離結果について表 4.3 に示す．BPM を用いた手法は両方ともベース手法より SDR が低い結果となった．楽曲の速さが楽曲の特徴であると考えたが，実際は予想と異なる結果となった．実際にジャンルごとの BPM の平均値を測定したところ，Jazz, Techno, Hip-hop の BPM が高い傾向にあることが分かったが，その差は約 20[BPM] 程度であり，あまり大きい差ではないことが分かった．少なくとも，今回用いた正規化手法及び one-hot vector 作成手法では大きな差が感じられないものであったため，BPM が楽曲を特徴づける数値足りえなかったと考える．Flatness を用いた手法では，一部楽器音で Flatness(INPUT) がベース手法を 0.05〜0.1dB 程度上回る結果，Flatness(FNN) が Vocal でベース手法を 0.15dB 程度上回る結果となった．そのため，ジャンル情報ほどではないが Flatness は楽曲特徴として活用可能であり，特に Vocal に有効であると考えられる．

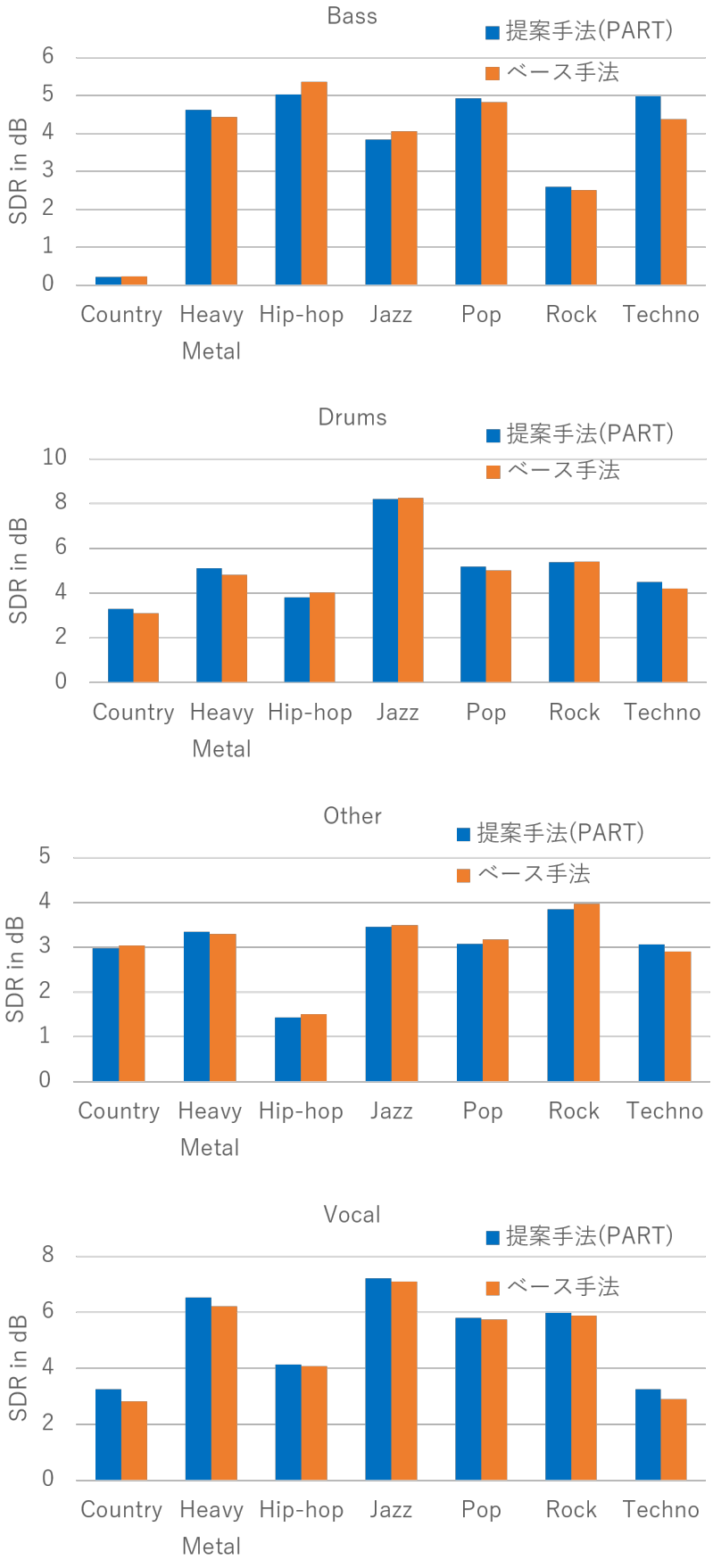
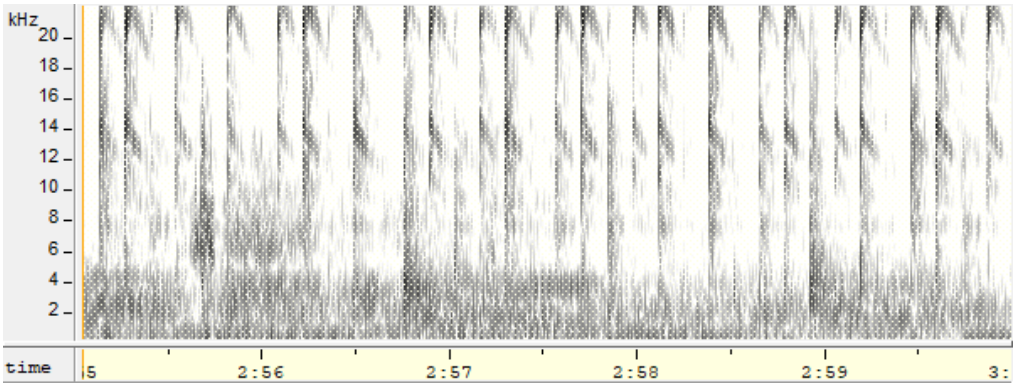
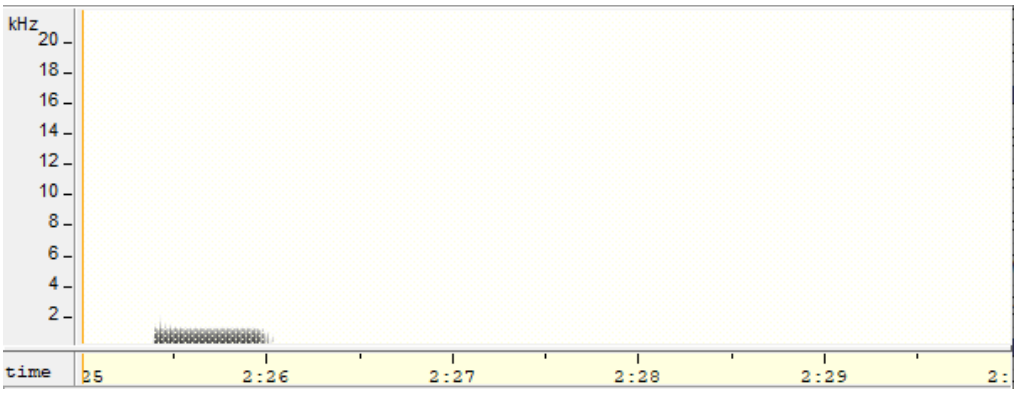


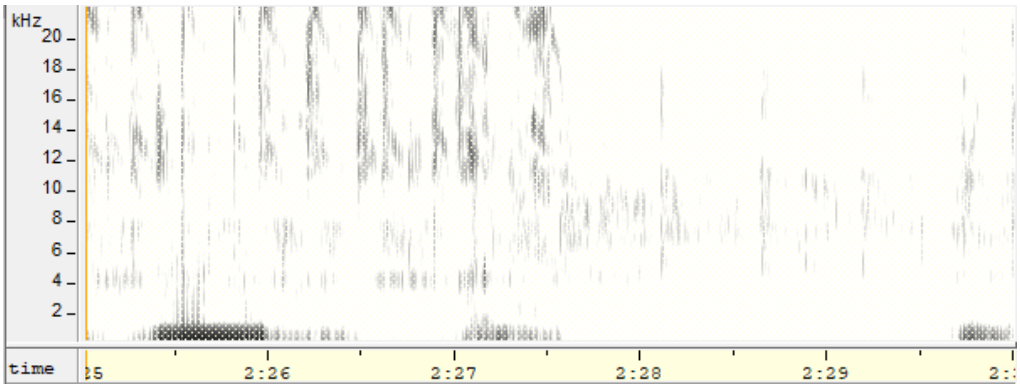
図 4.1　ジャンルごとの客観的評価



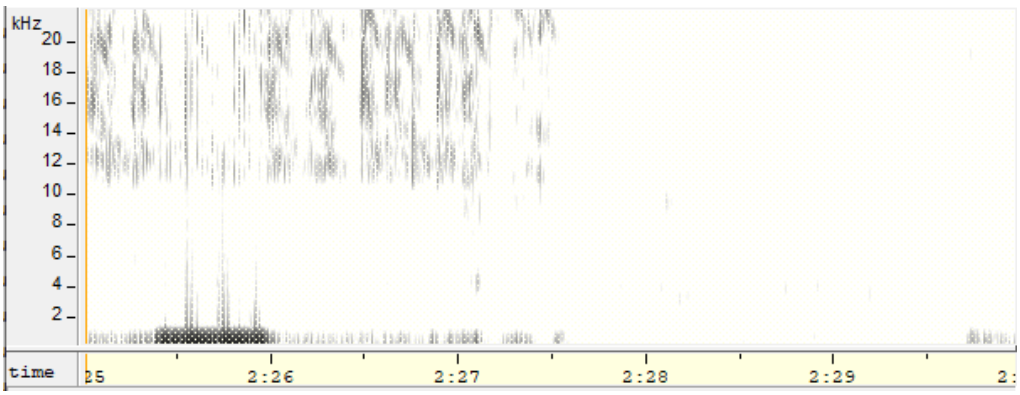
(a) mixture



(b) source

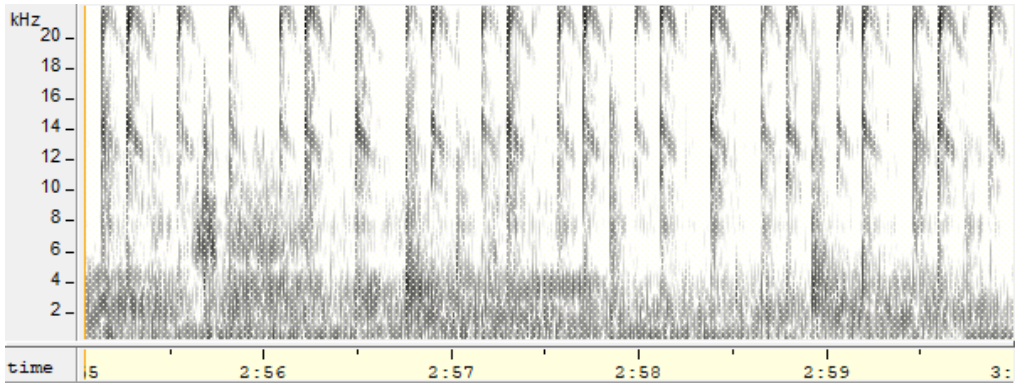


(c) baseline

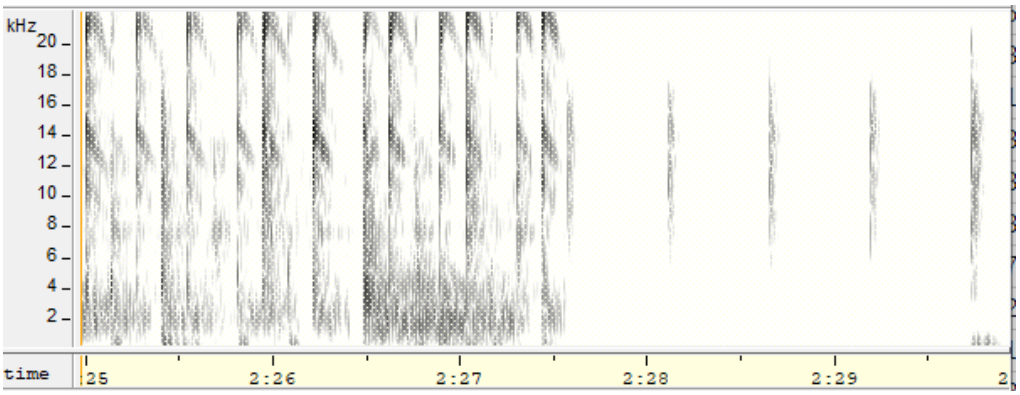


(d) proposed

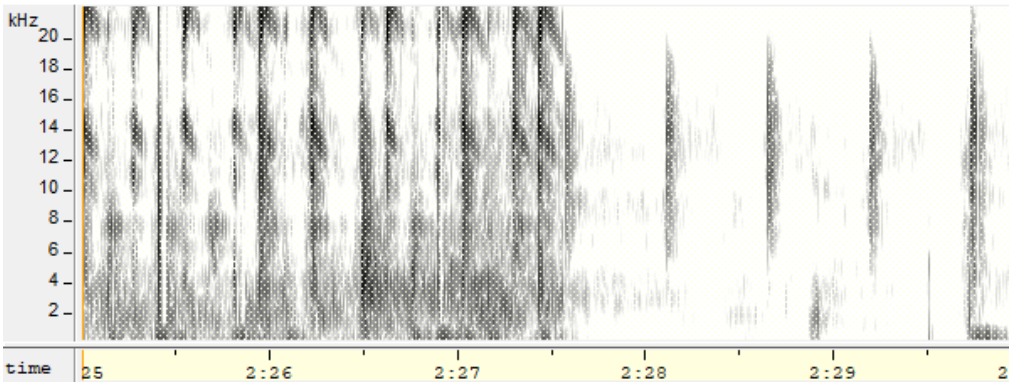
図 4.2　ジャンル：Techno　楽器音：Bass



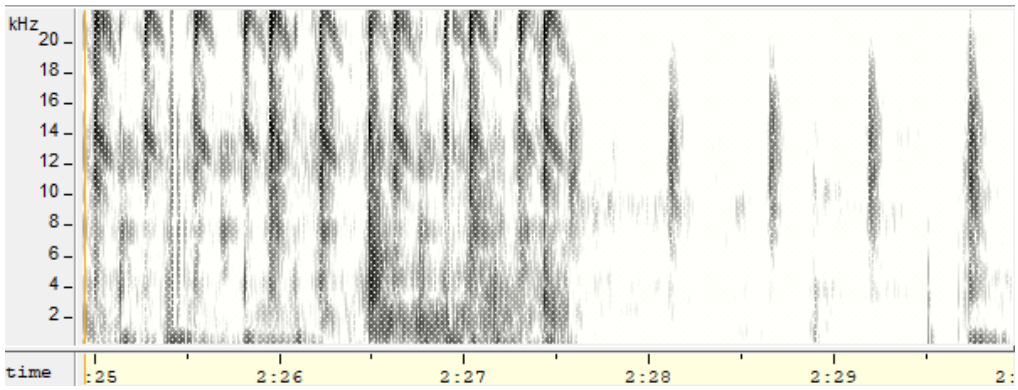
(a) mixture



(b) source

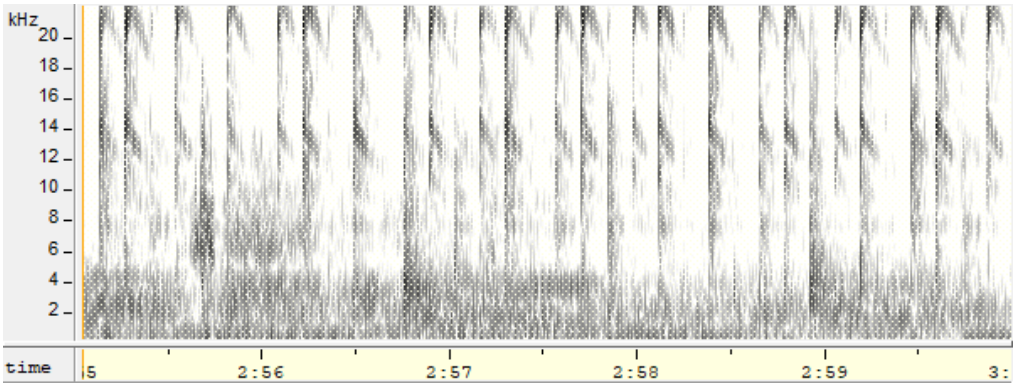


(c) baseline

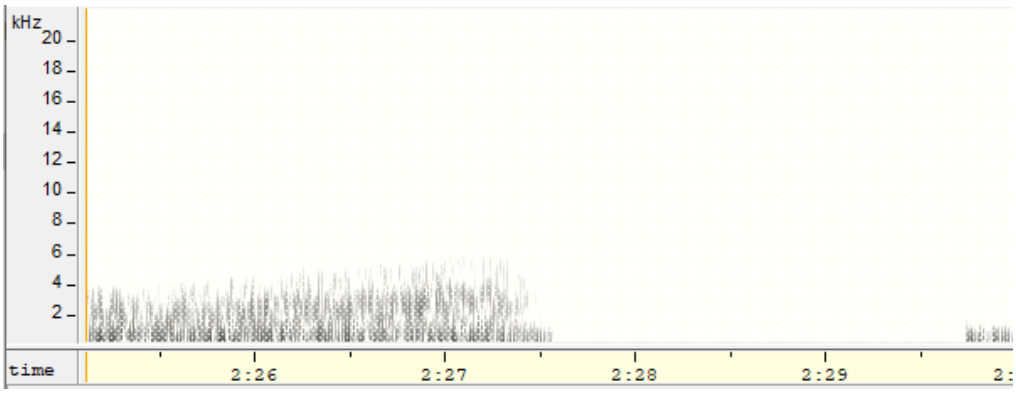


(d) proposed

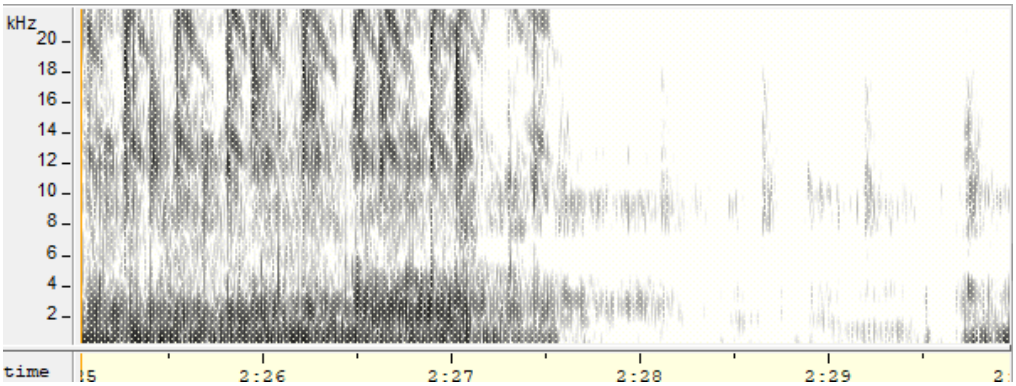
図 4.3　ジャンル：Techno　楽器音：Drums



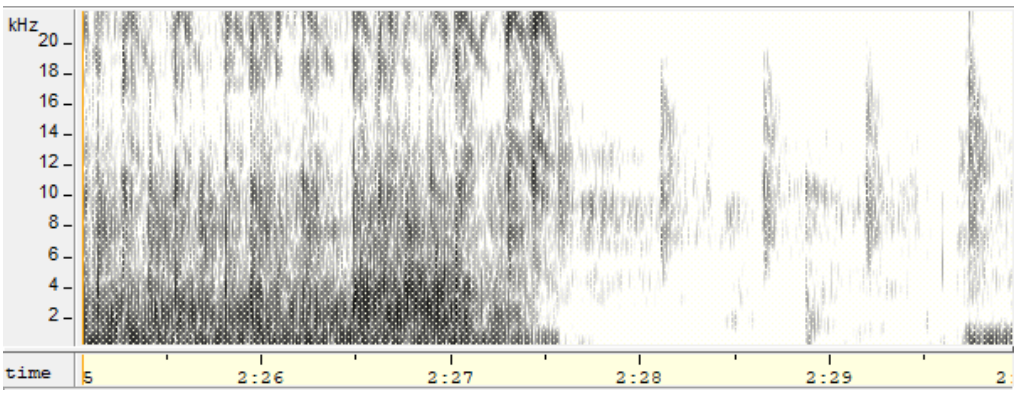
(a) mixture



(b) source

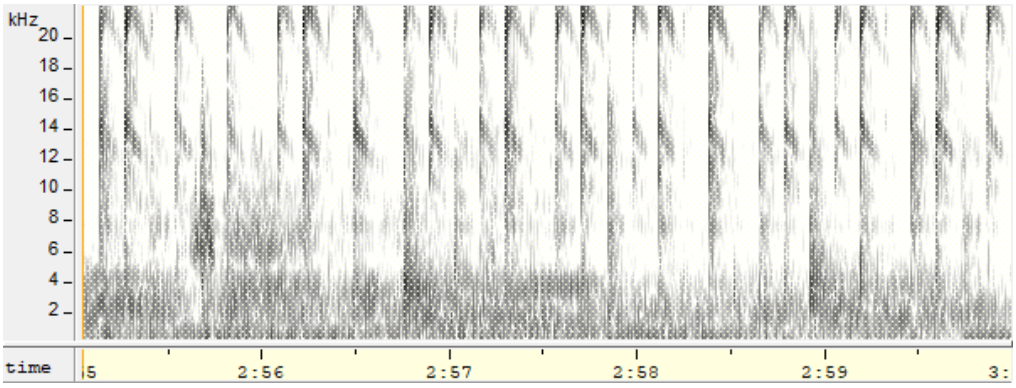


(c) baseline

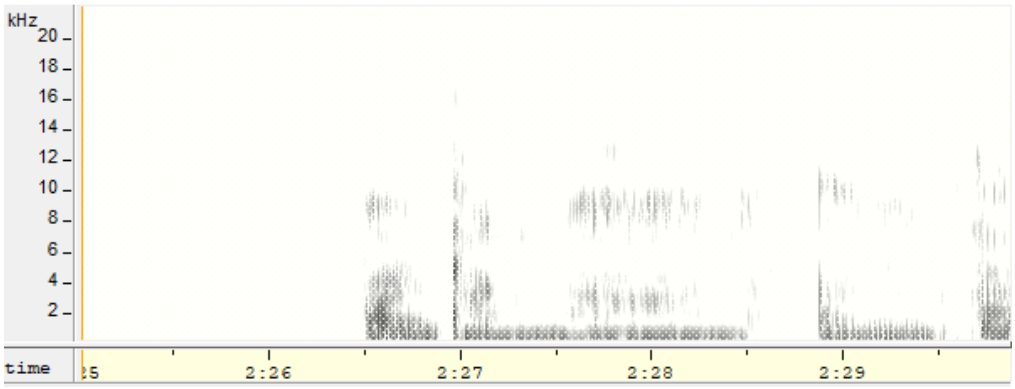


(d) proposed

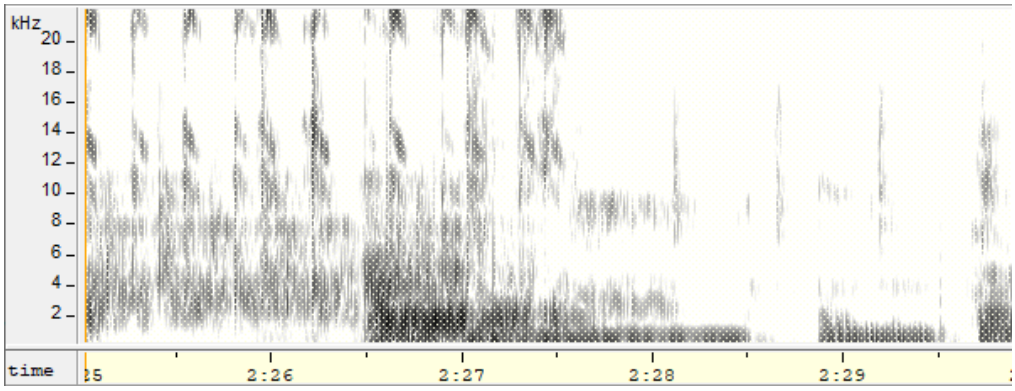
図 4.4　ジャンル：Techno　楽器音：Other



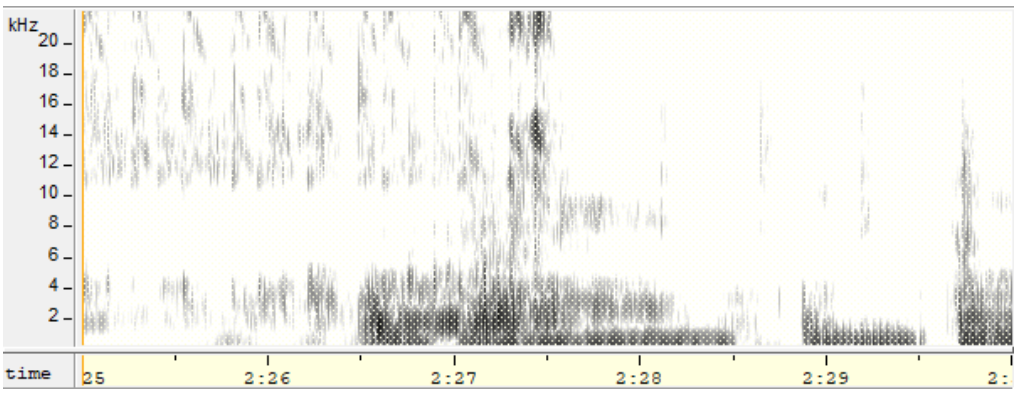
(a) mixture



(b) source

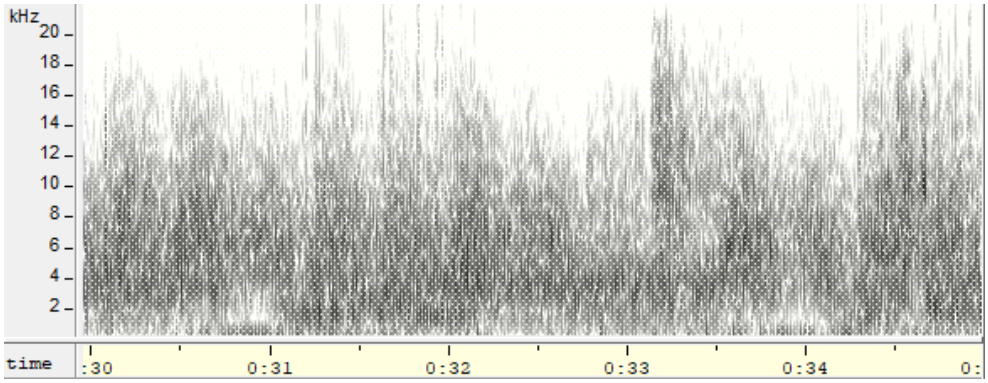


(c) baseline

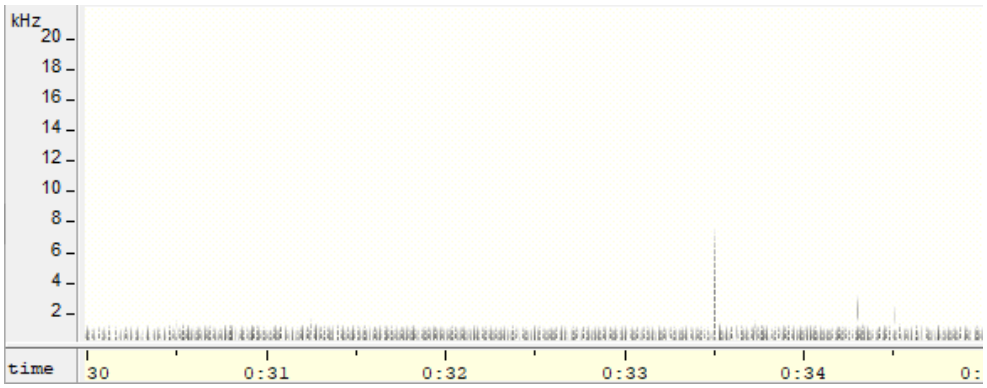


(d) proposed

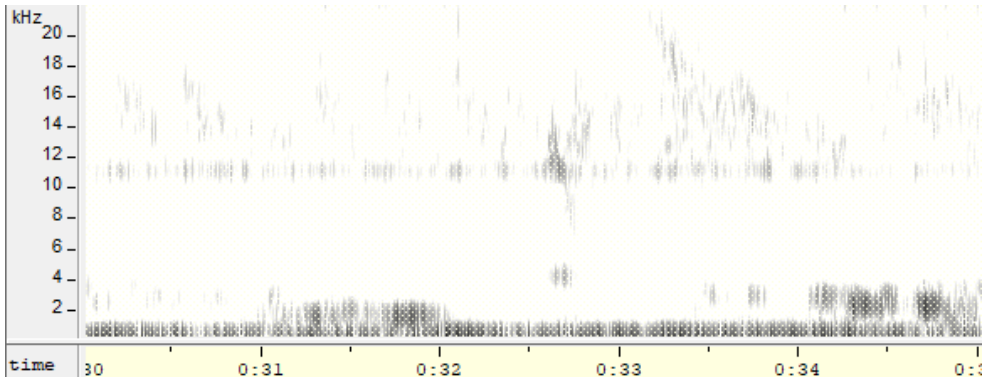
図 4.5　ジャンル：Techno　楽器音：Vocal



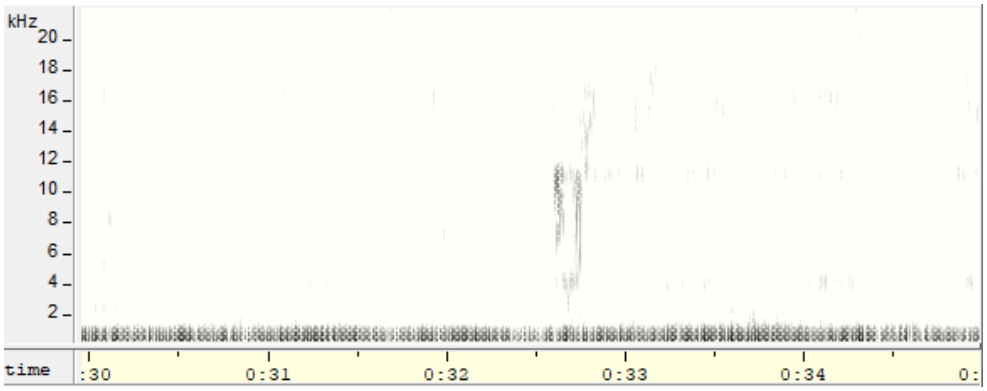
(a) mixture



(b) source

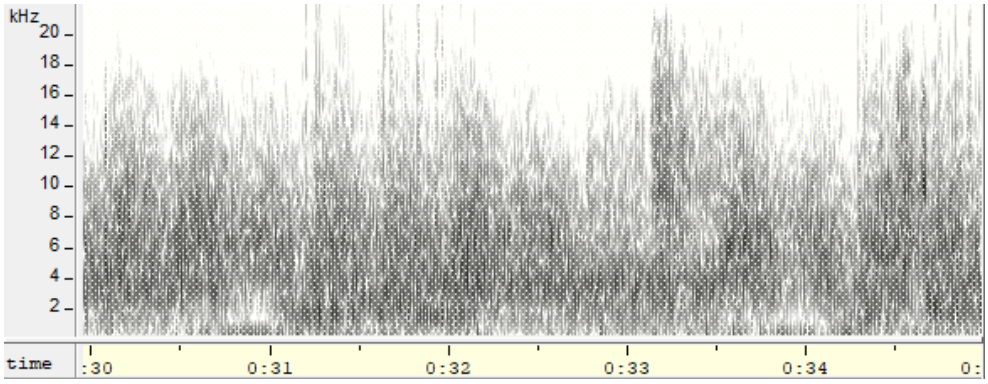


(c) baseline

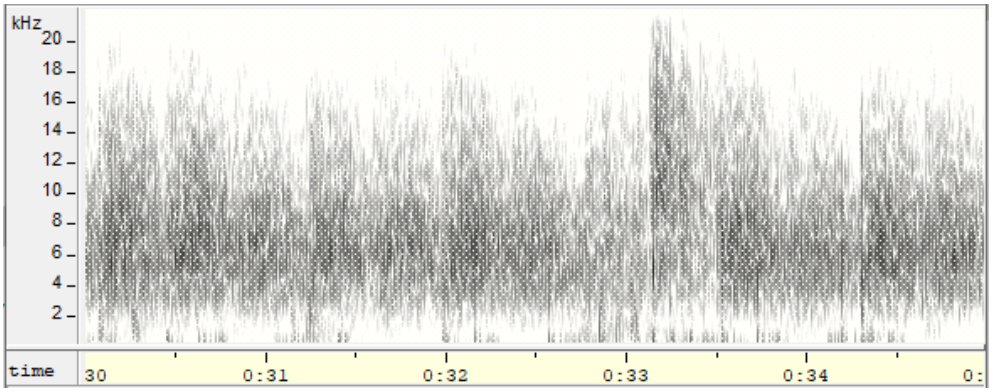


(d) proposed

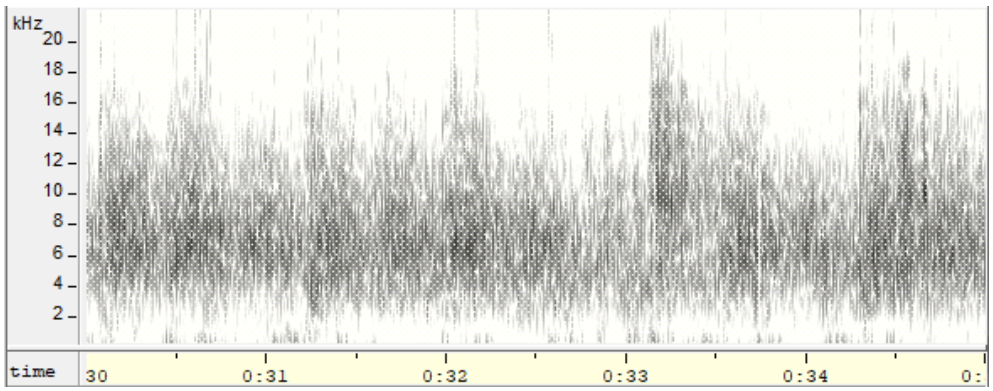
図 4.6　ジャンル：Heavy Metal　楽器音：Bass



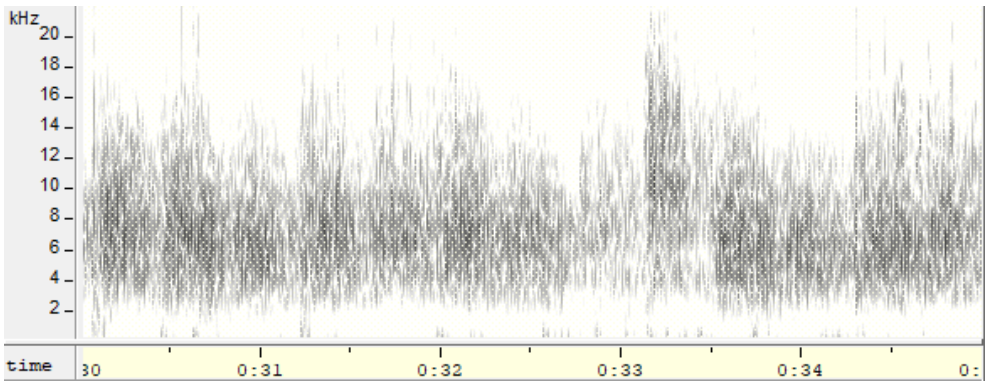
(a) mixture



(b) source

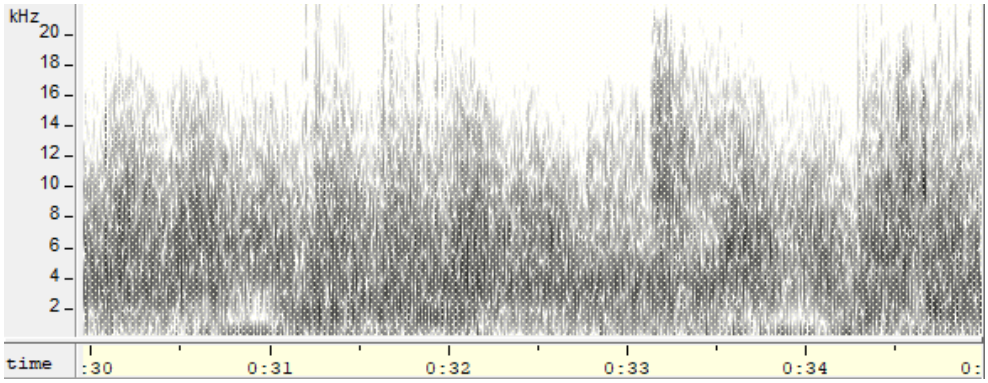


(c) baseline

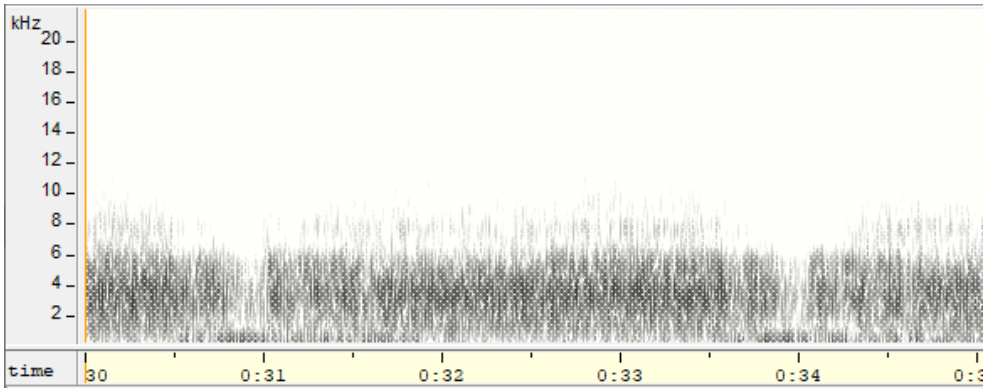


(d) proposed

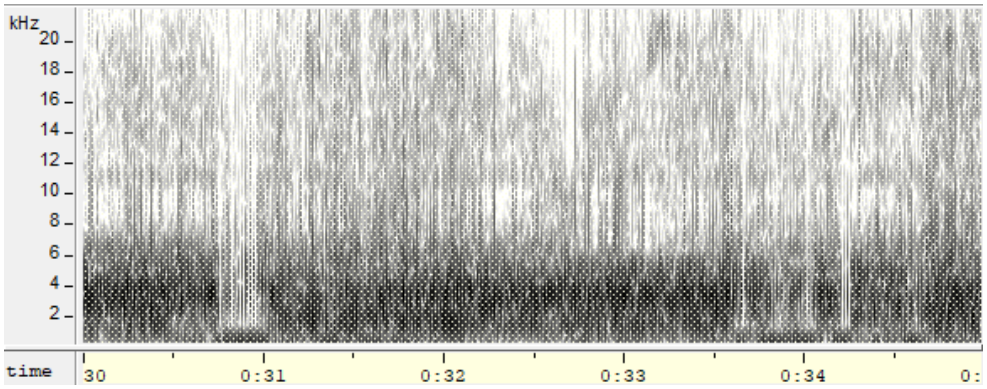
図 4.7　ジャンル：Heavy Metal　楽器音：Drums



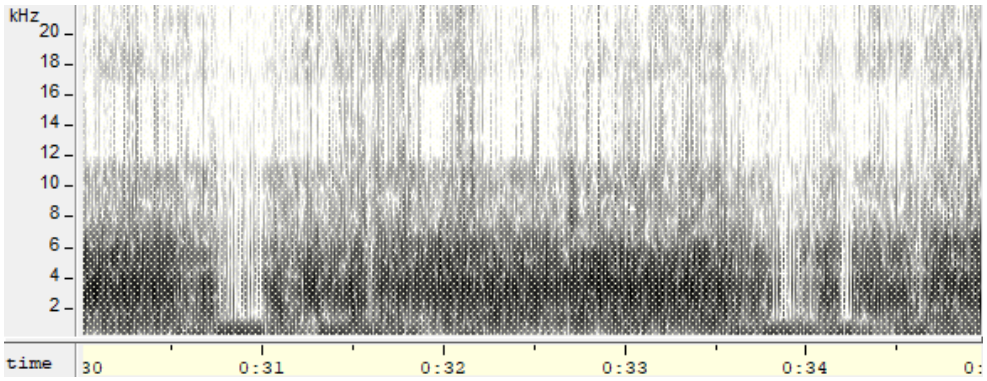
(a) mixture



(b) source

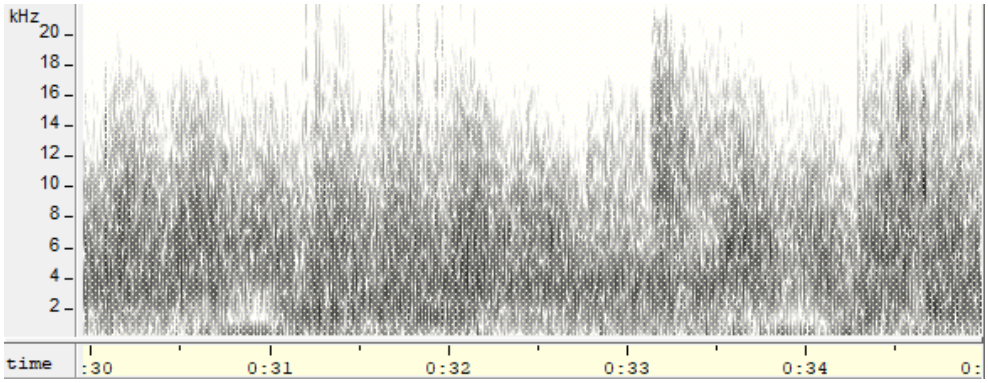


(c) baseline

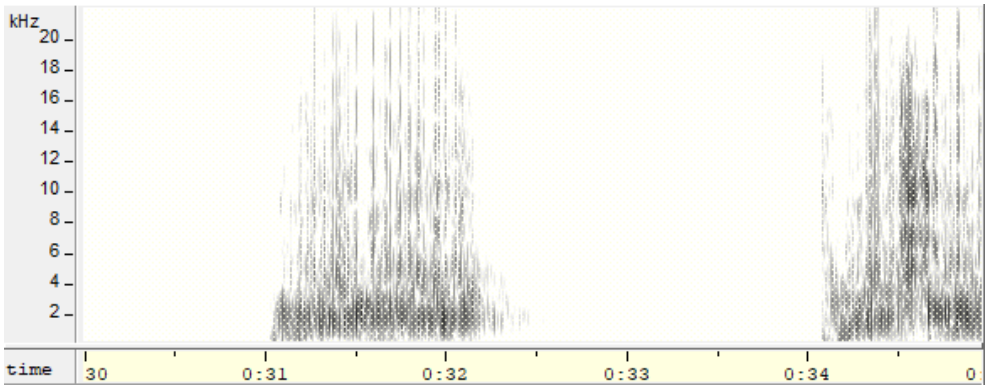


(d) proposed

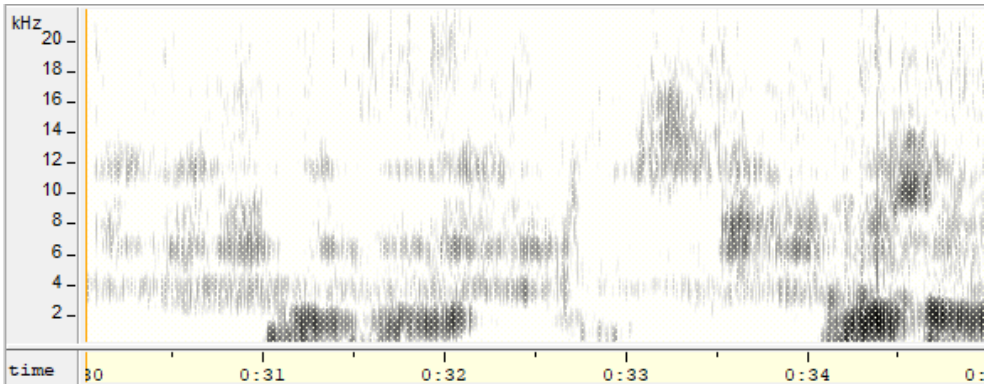
図 4.8　ジャンル：Heavy Metal　楽器音：Other



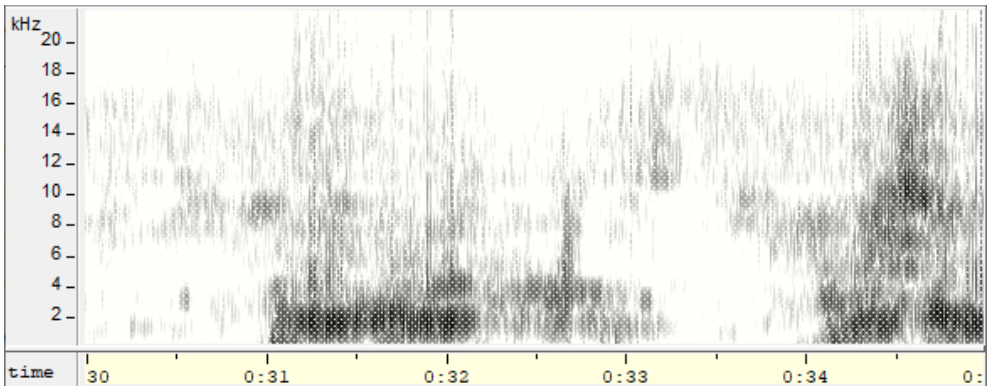
(a) mixture



(b) source

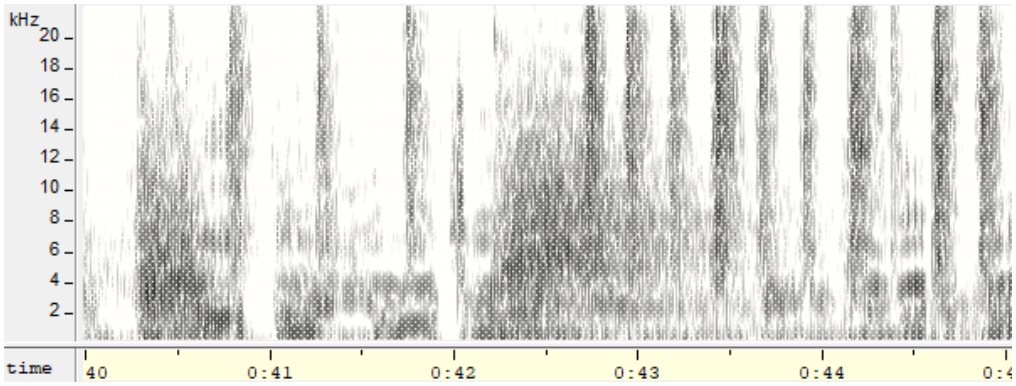


(c) baseline

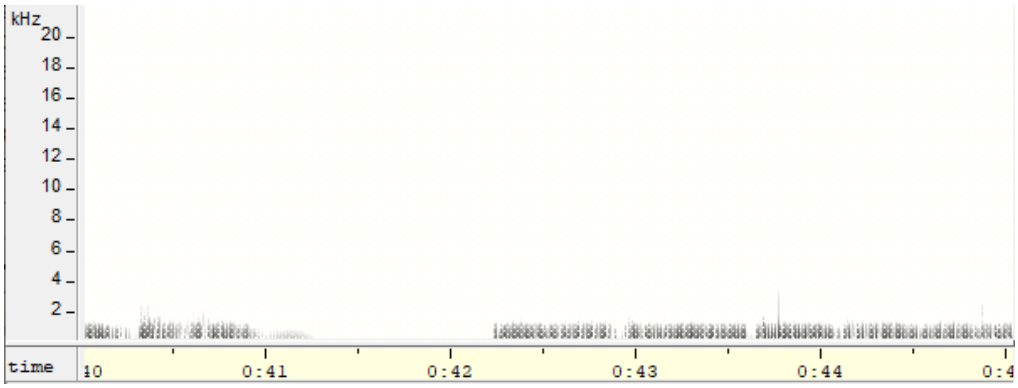


(d) proposed

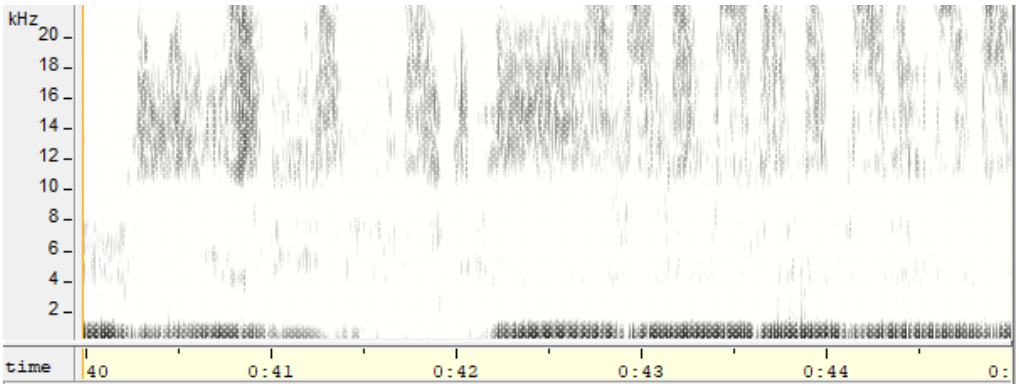
図 4.9　ジャンル：Heavy Metal　楽器音：Vocal



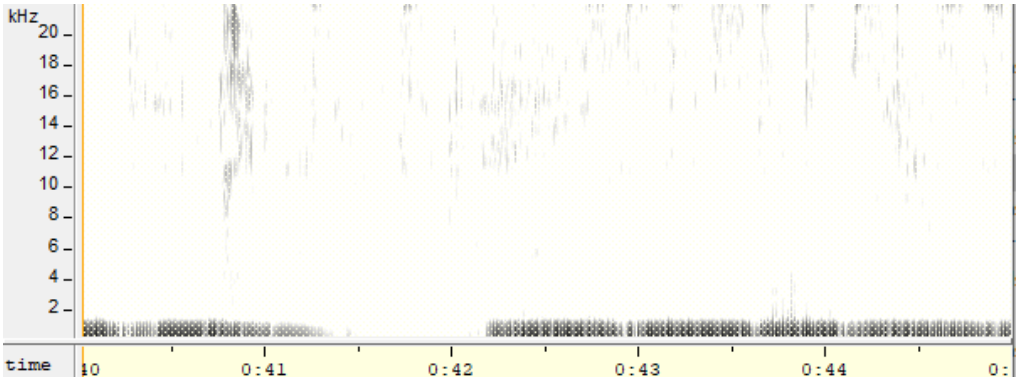
(a) mixture



(b) source

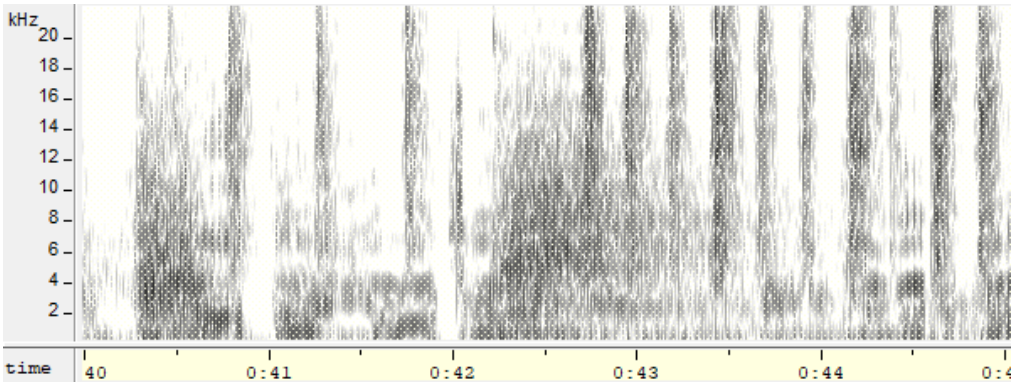


(c) baseline

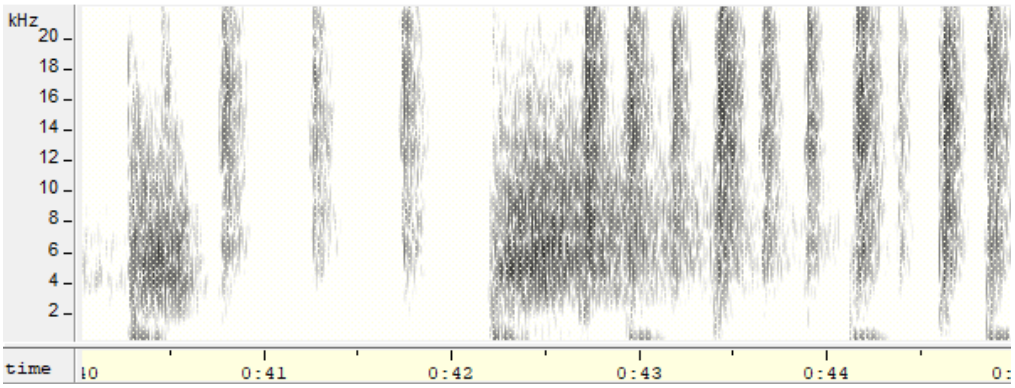


(d) proposed

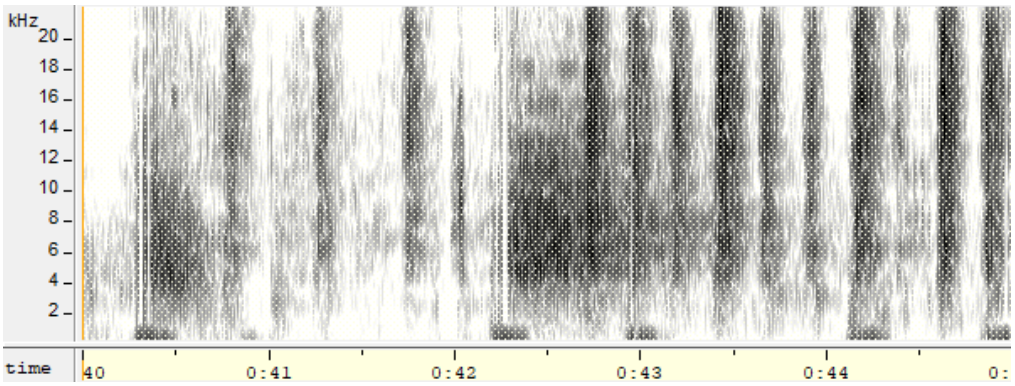
図 4.10　ジャンル：Rock 楽器音：Bass



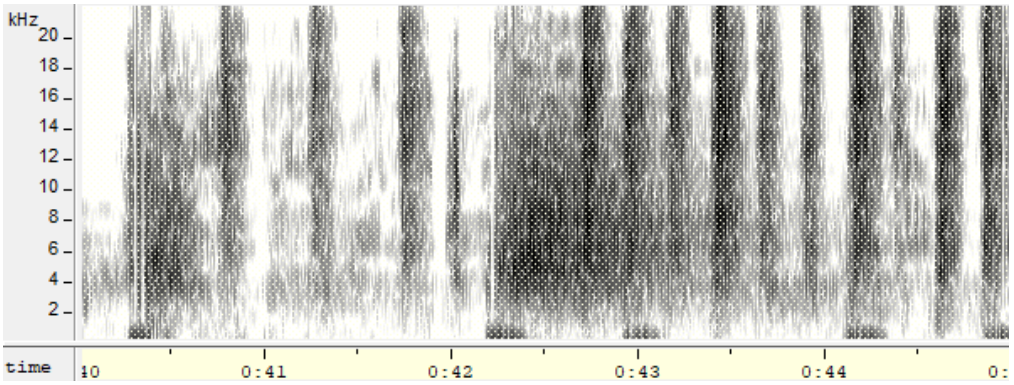
(a) mixture



(b) source

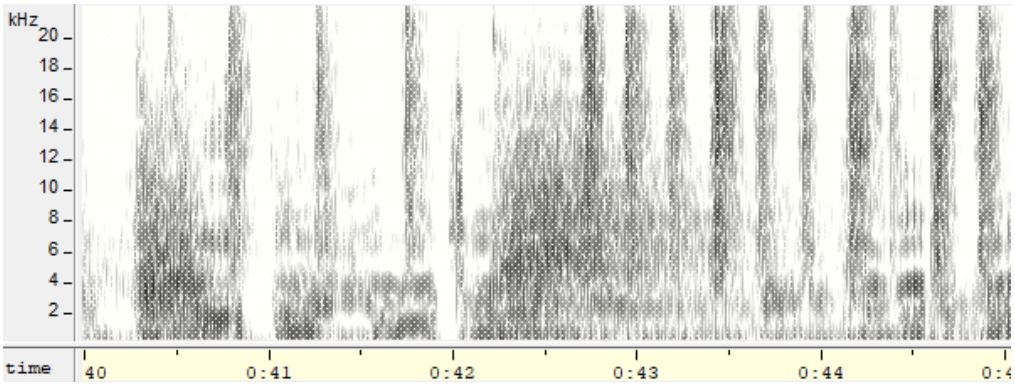


(c) baseline

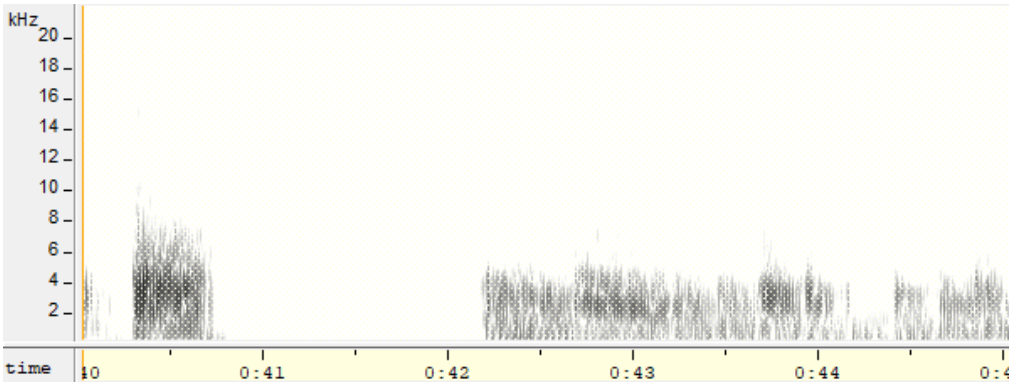


(d) proposed

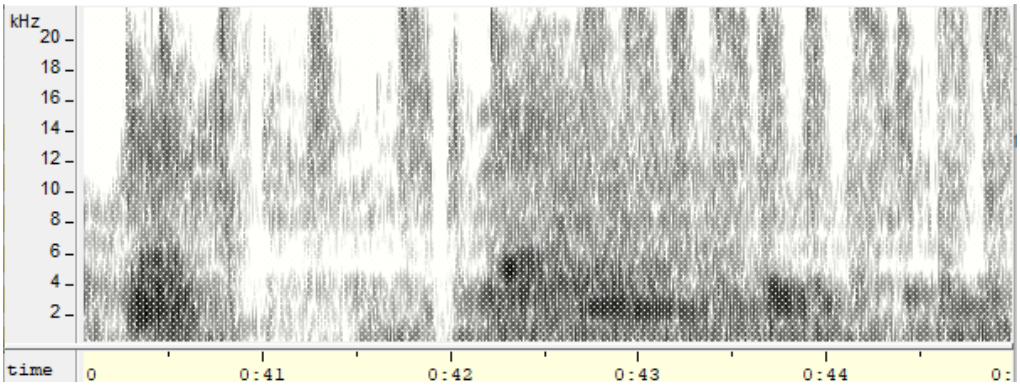
図 4.11　ジャンル：Rock　楽器音：Drums



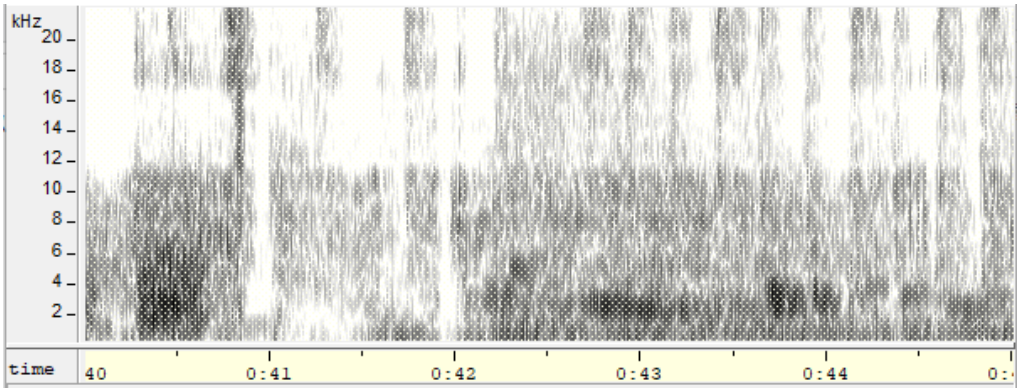
(a) mixture



(b) source

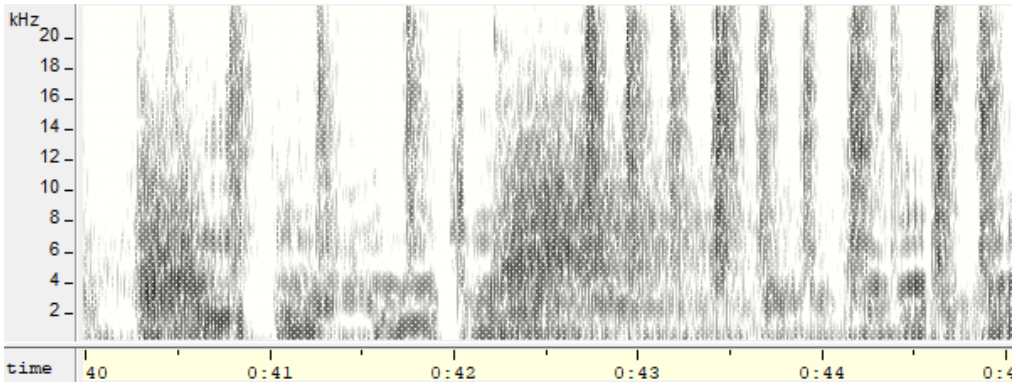


(c) baseline

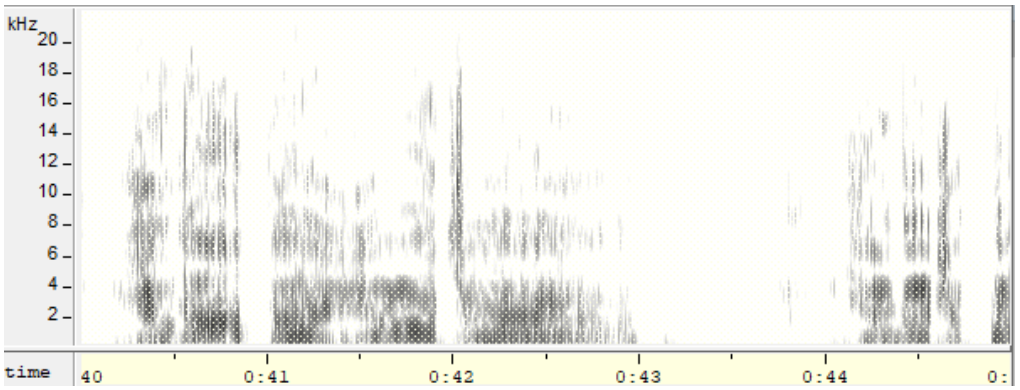


(d) proposed

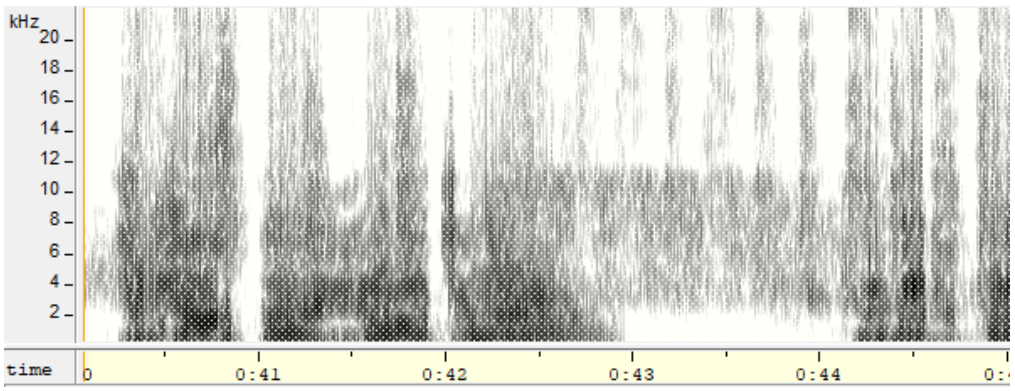
図 4.12　ジャンル：Rock　楽器音：Other



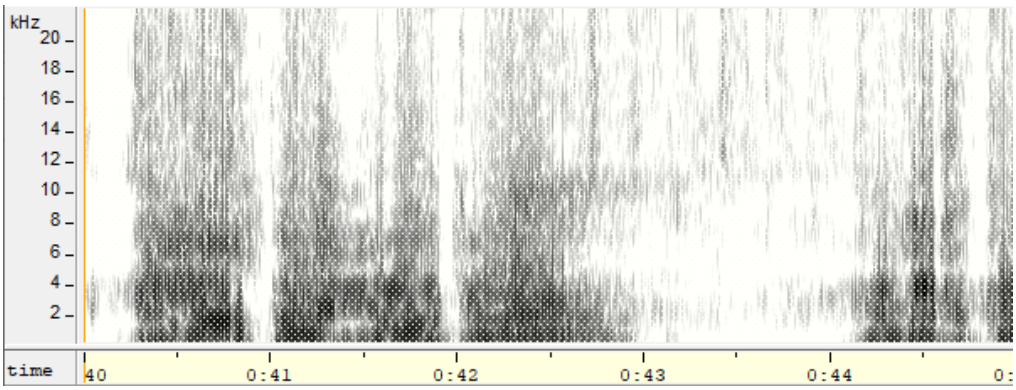
(a) mixture



(b) source

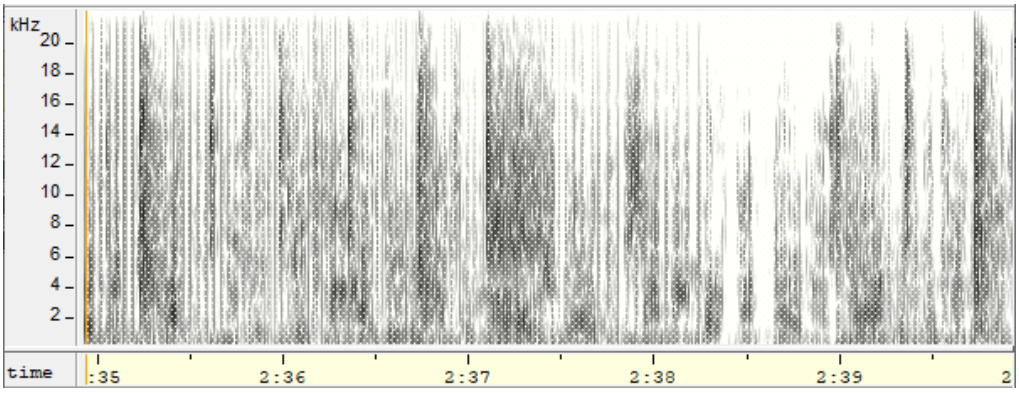


(c) baseline

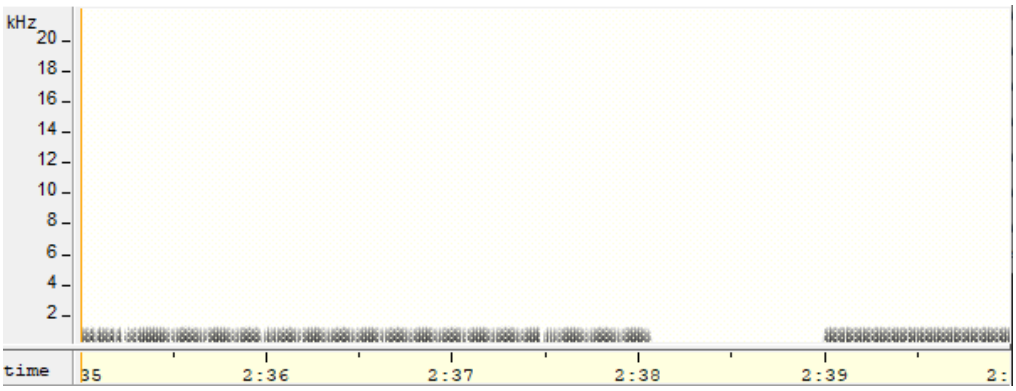


(d) proposed

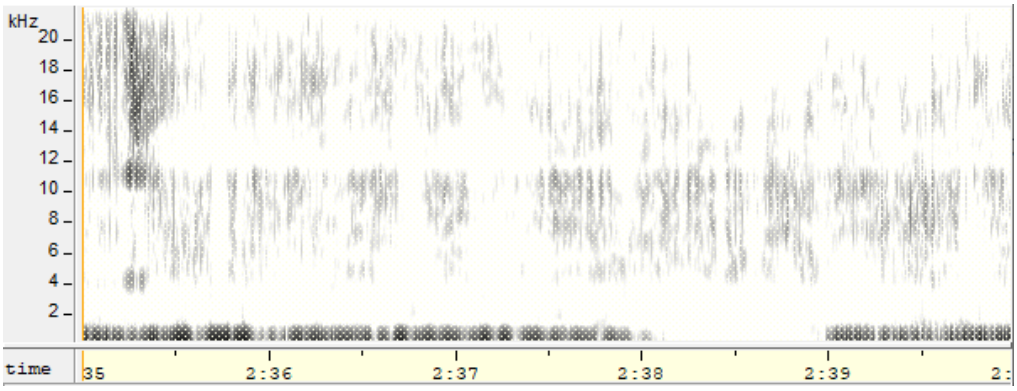
図 4.13　ジャンル：Rock　楽器音：Vocals



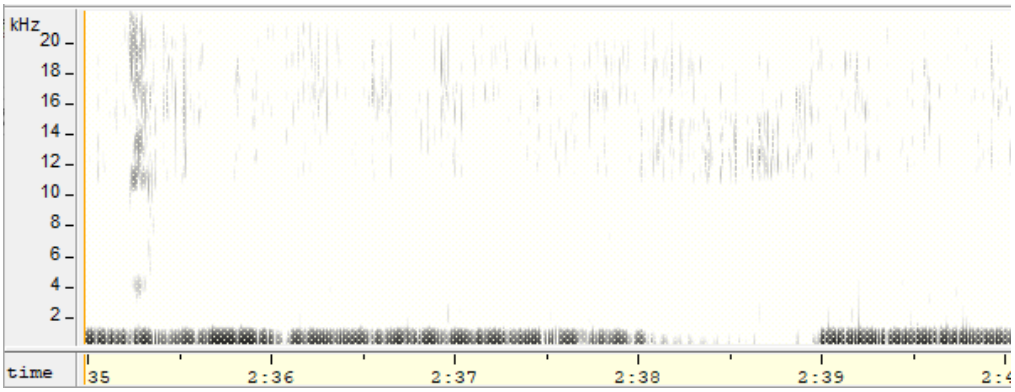
(a) mixture



(b) source

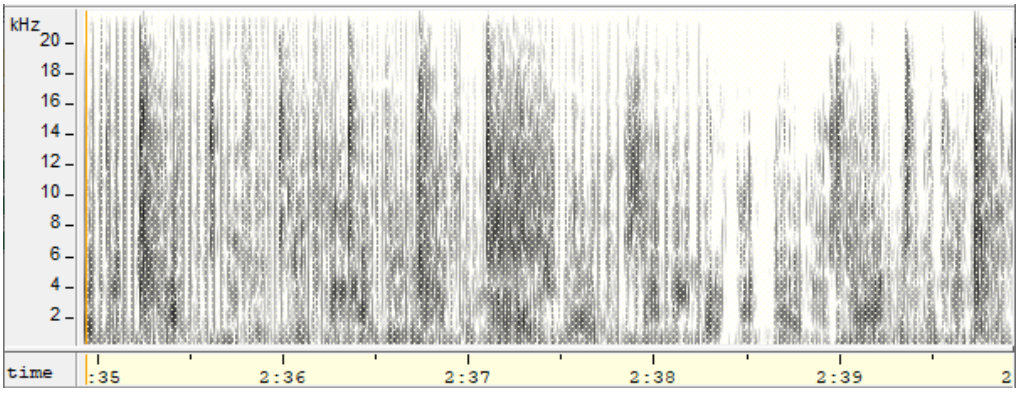


(c) baseline

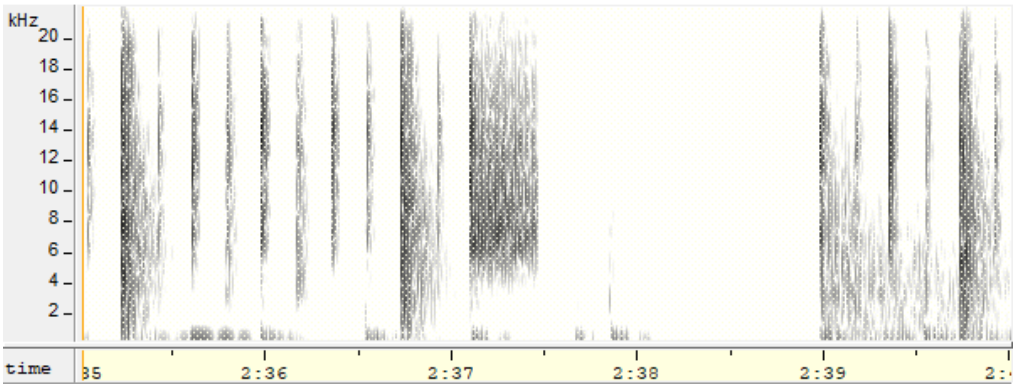


(d) proposed

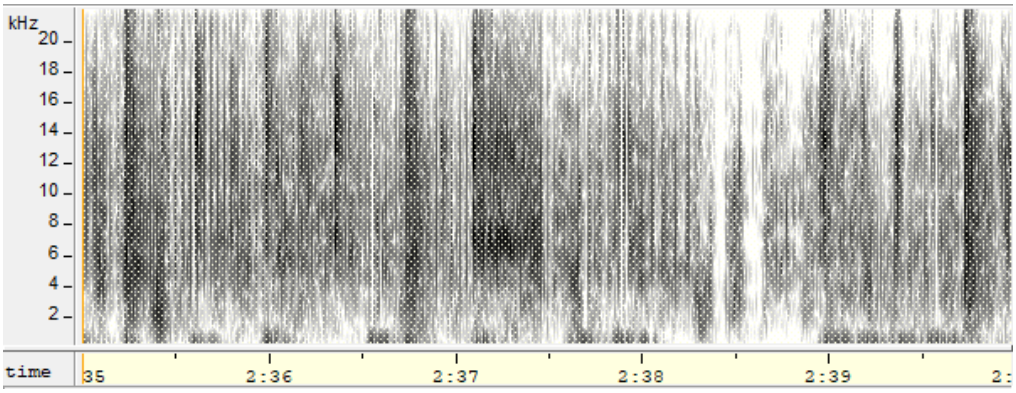
図 4.14　ジャンル：Hip-hop 楽器音：Bass



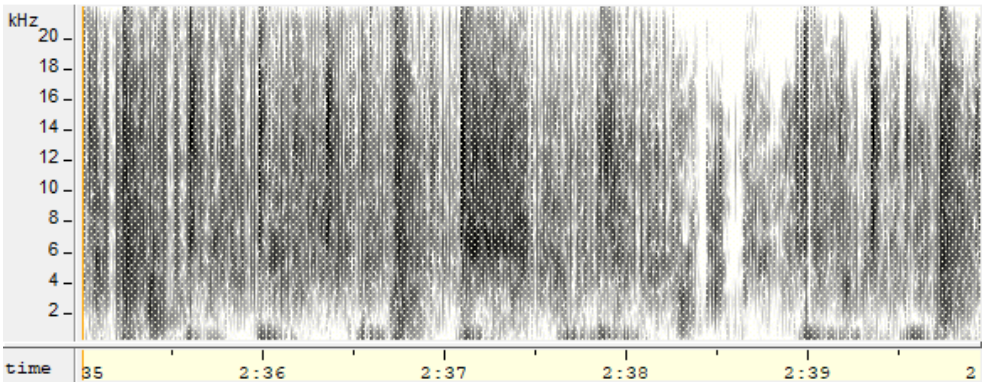
(a) mixture



(b) source

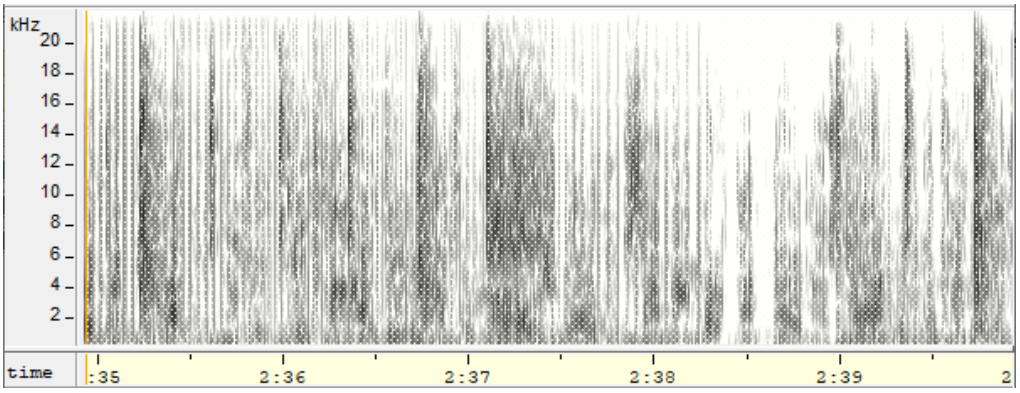


(c) baseline

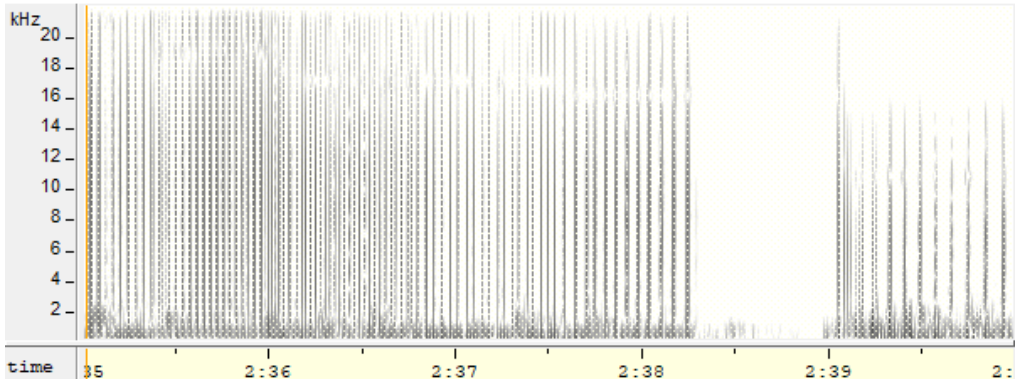


(d) proposed

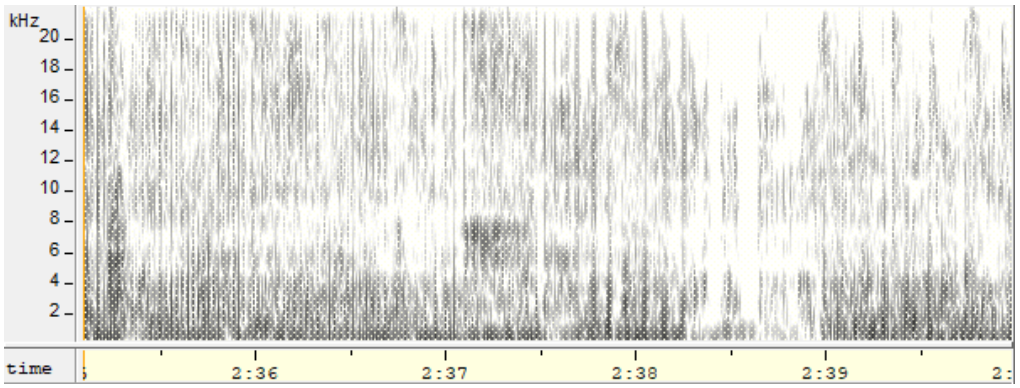
図 4.15　ジャンル：Hip-hop 楽器音：Drums



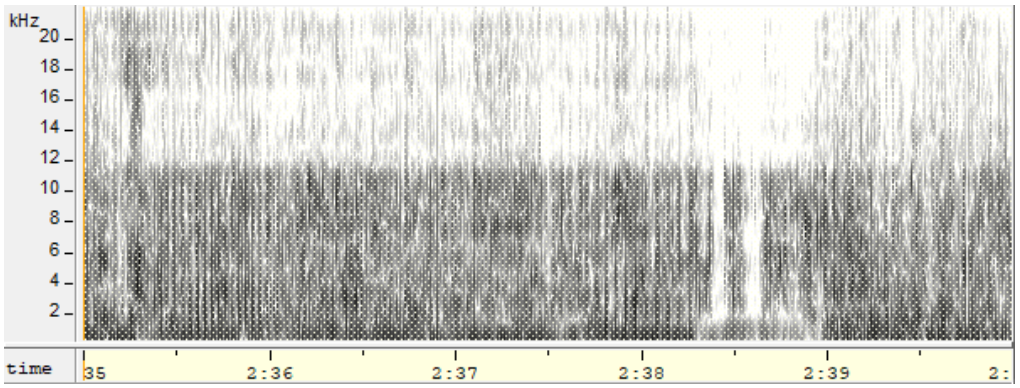
(a) mixture



(b) source

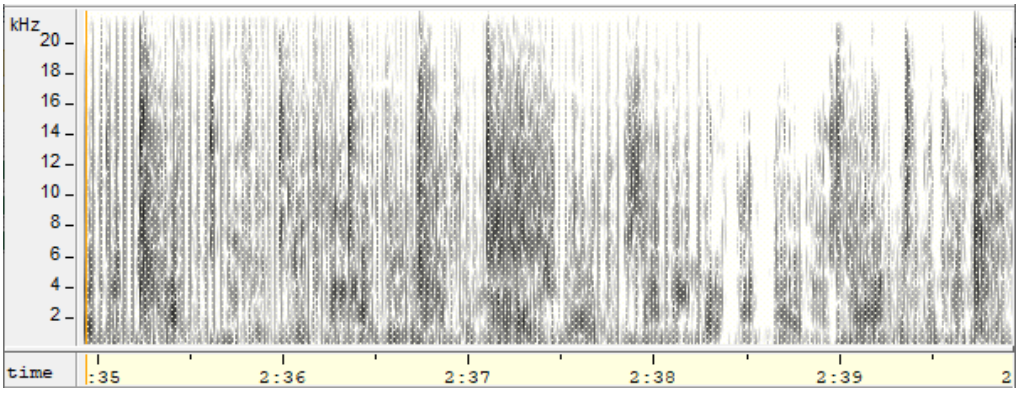


(c) baseline

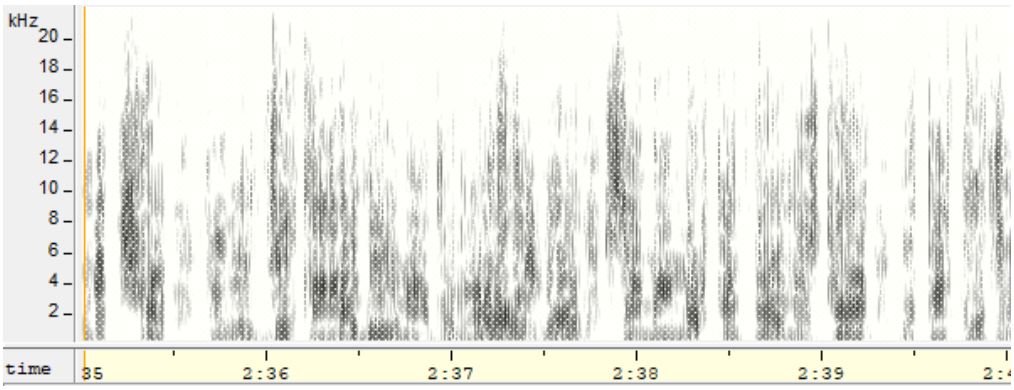


(d) proposed

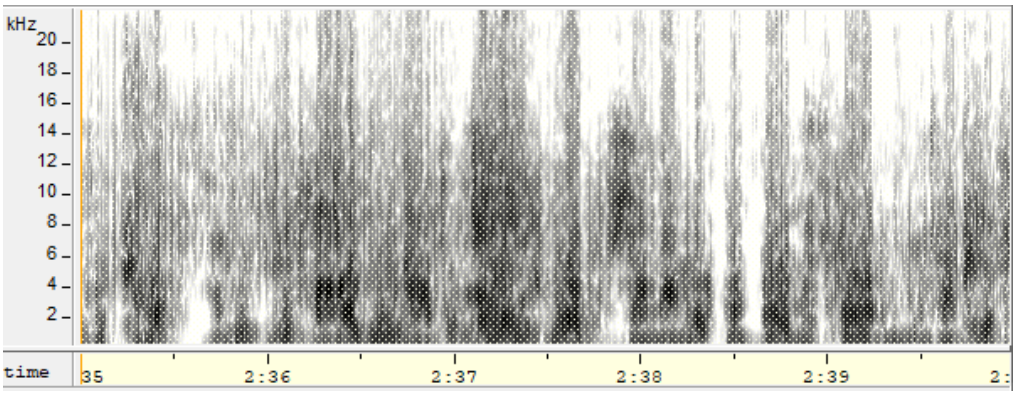
図 4.16　ジャンル：Hip-hop 楽器音：Other



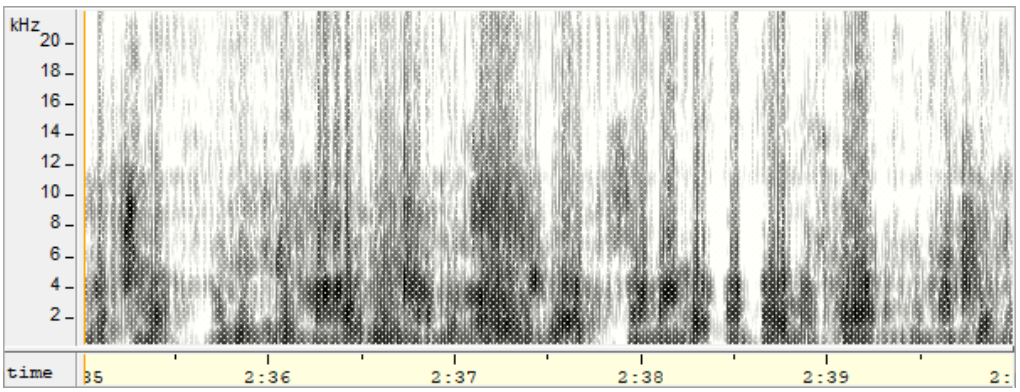
(a) mixture



(b) source



(c) baseline



(d) proposed

図 4.17　ジャンル：Hip-hop 楽器音：Vocals

第 5 章

おわりに

5.1 まとめ

本論文では，楽器音分離モデルの性能向上のためにジャンル情報などの楽曲から得られる情報を活用することを目的として，楽曲情報を DNN ベースの分離モデルに挿入する手法を提案した．第 1 章では最近の楽器音分離の研究動向及び現在の問題点について述べ，本研究の目的を示した．第 2 章では本論文のベース手法である MMDenseNet 及び MMDenseLSTM について述べた．第 3 章では本論文の提案手法である，楽曲のジャンル情報を one-hot vector 形式で挿入し楽器音分離の学習を行うモデル及び，楽曲の音源から抽出した特徴量を挿入し楽器音分離の学習を行うモデルについてその構造を述べた．第 4 章では DSD100 と呼ばれる楽器音分離用のデータセットを用いて，従来法と提案手法の精度を比較した．実験の結果から，ジャンル情報を挿入する提案法は Bass, Drums, Vocal の 3 つの楽器音において従来法を上回る結果を得ることが分かった．また，one-hot vector とバッチ正規化の相性の問題から，ジャンル情報はバッチ正規化を採用していない一部の層のみに挿入するべきであると判明した．楽曲の音源から抽出した特徴量を挿入した場合の分離結果は，BPM を用いた場合は従来法を下回る結果となったが，Flatness を用いた場合は一部楽器音で多少の精度向上が見られた．

5.2 今後の課題

本論文の実験では，ジャンル情報を挿入することで楽器音分離の精度を向上させた．しかし，楽曲にジャンル情報のラベリングを行うことはコストがかかるため，入力音源から抽出できる楽曲特徴量の活用を考え，本論文の実験でも行ったがジャンル情報を用いた場合ほどの精度向上とはならなかった．そのため，今後の課題として，楽器

音分離に有用な楽曲特徴量の調査及び適切な挿入方法の検討を行う必要がある．また，バッチ正規化と one-hot vector の相性の問題のために本論文では分離モデルの一部分のみにジャンル情報の挿入を行ったが，音声合成の分野では全層に楽曲情報を挿入する形式の方が一般的であり，精度向上が見込める．そこで，楽曲情報を挿入するために適切な形式を調査・実験することも今後の課題として考えられる．

謝辞

本研究を行うにあたり，多くのご指導・ご鞭撻をいただいた杉田泰則准教授に厚く感謝申し上げます。また，本論文の審査にあたり適切なご指示をいただいた本学の岩橋政宏教授ならびに圓道知博准教授に誠に感謝申し上げます。最後に，本研究に対して多くのご指摘をくださった信号処理応用研究室の皆様に深く感謝いたします.

令和 3 年 2 月

付録 A

付録

A.1 ideal binary mask

ideal binary mask(IBM) は分離後の各楽器音が得られている前提で計算される理想的な楽器音分離手法である．具体的には，以下の計算式の通りとなる．

$$IBM(t,f) = \begin{cases} 1 & SNR(t,f) \geq 0 \\ 0 & SNR(t,f) < 0 \end{cases} \tag{1.1}$$

$$SNR(t,f) = 20\log\frac{S_{target}(t,f)}{S_{noise}(t,f)} = 20\log\frac{S_{target}(t,f)}{S(t,f) - S_{target}(t,f)} \tag{1.2}$$

$$\hat{S}_{target}(t,f) = IBM(t,f)S(t,f) \tag{1.3}$$

ここで，t は時間サンプル，f は周波数ビン，S_{target} は分離したい楽器音の教師信号，S_{noise} は目的以外の楽器音の信号，$\hat{S}_{target}$ は分離したい楽器音の推定信号，S は入力信号 (混合音源) である．

参考文献

[1] A. Liutkus, D. Fitzgerald, and R. Badeau, “Cauchy nonnegative matrix factorization,” in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA New Paltz, NY, USA, pp. 1–5, 2015.

[2] J. LeRoux, J. R. Hershey, and F. Weninger, “Deep NMF for speech separation,” in Proc. ICASSP, p. 6670, 2015.

[3] Y. Mitsufuji, S. Koyama, and H. Saruwatari, “Multichannel blind source separation based on non-negative tensor factorization in wavenumber domain,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, pp. 56–60, March 2016.

[4] N. Q. K. Duong, E. Vincent, and R. Gribonval, “Under-determined reverberant audio source separation using a full-rank spatial covariance model,” IEEE Trans. Audio Speech Language Process., vol. 18, no. 7, pp. 1830–1840, 2010.

[5] 北村大地, 角野隼斗, 高宗典玄, 高道慎之介, 猿渡洋, 小野順貴, "独立深層学習行列分析に基づく多チャネル音源分離の実験的評価", 信学技報, vol. 117, no. 515, pp. 13-20, 2018.

[6] A. A. Nugraha, A. Liutkus, and E. Vincent, “Multichannel music separation with deep neural networks,” in Proc. EUSIPCO, 2015.

[7] S. Uhlich, F. Giron, and Y. Mitsufuji, “Deep neural network based instrument extraction from music,” in Proc. ICASSP, pp. 2135–2139, Apr. 2015.

[8] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving Music Source Separation Based On Deep Networks Through Data Augmentation And Augmentation And Network Blending,” in Proc. ICASSP, pp. 261–265, 2017.

[9] N. Takahashi, N. Goswami, and Y. Mitsufuji, “MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation,” in 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC). IEEE, pp. 106–110, 2018.

[10] G.Huang, Z.Liu, L.van der Maaten, K.Q.Weinberger. ”Densely Connected Convolutional Networks,”IEEE Conference on Pattern Recognition and Computer Vision (CVPR), 2016.

[11] N. Hojo, Y. Ijima, and H. Mizuno, “DNNbased speech synthesis using speaker codes,” IEICE T. Inf. Syst., vol. 101, no. 2, pp. 462–472, 2018.

[12] N. Takahashi and Y. Mitsufuji,“Multi-scale multi-band DenseNets for audio source separation,” in Proc. WASPAA, pp. 261–265, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," In International Conference on Machine Learning, pp. 448–456, 2015.

[15] P. Grosche, M. Müller, and F. Kurth, “Cyclic tempogram—A midlevel tempo representation for music signals,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 5522–5525, Mar. 2010.

[16] S. Dubnov, “Generalization of spectral flatness measure for non-gaussian linear processes,”IEEE Signal Processing Letters, Vol. 11, no. 8, pp. 698-701, Aug. 2004.

[17] 亀岡弘和, 中村友彦, 高宗典玄, "音楽音響信号処理技術の最先端," 電子情報通信学会誌 Vol. 98, No. 6, pp. 467-474, 2015.

[18] 山田真司, 三浦雅展, "音楽情報処理で用いられる音響パラメータによる音楽理解の可能性," 日本音響学会誌 70 巻 8 号, pp.440-445, 2014.

[19] 福本颯太, 奥健太, "楽曲-景観データに基づく音響特徴量の分析," DEIM Forum, pp.2-4, 2018.

[20] 山崎瑞己, 櫻井美緒, 三浦雅典, "音楽音響信号を対象としたアニメソングに対する公開年代の自動推定," 日本音響学会誌 72 巻 4 号, pp. 182-189, 2016.

[21] D. Misra,”Mish: A Self Regularized Non-Monotonic Activation Function,”BMVC, 2020

[22] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, ”On the variance of the adaptive learning rate and beyond,” ICLR, 2020