Vol.2019-MUS-122 No.18
Vol.2019-EC-51 No.18
2019/2/23

End-to-End 学習を利用したスペクトログラム生成による楽器音抽出手法の提案

久野 文菜[†]　大場 隆史[‡]　中園 歩[‡]　谷口 航平[‡]　畑中衛[‡]　林 広幸[‡]　濱川礼[†]

[†]中京大学 工学部 情報工学科　　[‡]中京大学大学院 工学研究科 情報工学専攻

1　背景・目的

近年ビッグバンドの人口が増えてきており，ビッグバンドの甲子園ともいわれる「YAMANO BIG BAND JAZZ CONTEST」では，参加ビッグバンド数が 1970 年代の 13 バンドから 2018 年には 45 バンドに増えている [1]．ビッグバンドとは多楽器によるアンサンブル形態でジャズを演奏するバンドである．ジャズの楽譜には演奏すべき音符がほとんど書かれていないため，演奏練習のためにはアーティストの演奏を耳で聞き取り真似して学習する場合が多い．しかし多重奏から特定の音を聞き取るのが困難な場合がある．そこで多重奏から特定の楽器の音のみを抽出することを目的とした手法を提案する．

2　提案手法

多重奏のスペクトログラムから，抽出したい楽器音のスペクトログラムを生成する．その際 End-to-End 学習を利用し，入力スペクトログラムから出力スペクトログラムを生成する処理全てを学習により行う．関係性モデルの生成と画像生成には pix2pix[2] を用いる．

3　関連研究

小林らは楽譜からの情報と NMF を用いることで二重層の分解手法を提案している [3]．Daniel らは Wave-U-Net を使用し音源からボーカルを抽出する手法を提案している [4]．Hang らは動画から音を生成する画像領域を見つけ出し，特定の楽器の音を分離する教師なし学習アーキテクチャを提案している [5]．

ジャズの楽譜には弾くべき音符がほぼ書かれていない．またビッグバンドは楽器の数が多く，音源に収録する際に楽器ごとに位相情報を付与していない．提案手法では関連研究との差異として楽譜と位相情報を使用せず，また学習データは動画ではなく収集しやすい各楽器の音源とする．

4　システム概要

多重奏のスペクトログラムから，抽出したい楽器の音で演奏されるスペクトログラムを生成する（図 1）．学習方法は End-to-End 学習とし，画像生成には pix2pix を用いた．

学習データとして多重奏を構成する各楽器の単楽器による音源と これらの音源を合成し多重奏にした音源を用意し，それぞれ結果データ・元データとする．その後各データを 1 秒ごとに分割し，スペクトログラムに変換する．これらの元データと結果データのスペクトログラムを pix2pix に入力し，多重奏と各楽器のスペクトログラムの関係を学習し，関係性モデルを生成する．その関係性モデルを使用し多重奏に含まれている各楽器のスペクトログラムを生成する．

5　実験・考察

提案手法の有効性を評価するためトランペットとピアノによる二重奏からピアノ音源の抽出を行う．トランペットとピアノは音の特徴が大幅に異なるため今回使用した．

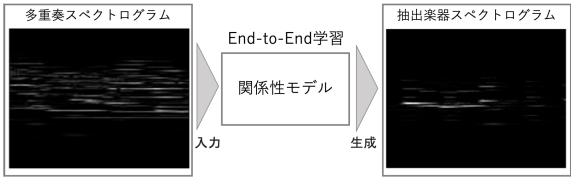

図 1: 提案手法の流れ

5.1　人間の演奏音源データによる学習

5.1.1　学習

学習データとして YouTube から取得した人間の演奏の音源を使用する．

トランペットで演奏される音源 3 曲[i]とピアノで演奏される音源 3 曲[ii]，またこの音源を組み合わせ，トランペット[iii]とピアノ[iv]による二重奏の音源を 3 曲用意した．その後各音源を 1 秒ごとに分割しスペクトログラムに変換することで，それぞれ結果データ・元データのスペクトログラムを 349 個用意した．学習は batch size 5 で 1000epoch を行った．

5.1.2　テスト

トランペットとピアノの演奏を合成した音源を 1 秒ごとに分割し，スペクトログラムに変換した．この用意した 20 個のデータをテストデータとする．テストとして，テストデータの合成音源から合成前のピアノのスペクトログラムを生成した．この合成前のピアノのスペクトログラムを正解データとする．生成されたスペクトログラムと正解データのスペクトログラムの平均二乗誤差を loss とする．

5.1.3　結果

loss の推移を図 2 に示す．400epoch，1000epoch の loss の最小値はそれぞれ 2.822963，1.797749 であり学習回数が増加するにつれ小さくなっている．また図 3 のように学習回数が増えるにつれて生成される画像が正解データに近づいていくことが分かった．しかしスペクトログラムを音源に戻し聴いたところ，元の二重奏と比べピアノの音は聴き取り易くなったもののトランペットの音がノイズへと変化しており，完全な分離はできていなかった．

5.1.4　考察

スペクトログラムは図 4 のように一見類似していても，少しの明度の差で多くの音のノイズが発生してしまう．よって loss の値を限りなく 0 に近づける必要がある．そのため各ペアの学習データを増やし batch size を最小である 1 まで小さくすることでより細かなピクセルまで特徴を捉える必要があ

[i] http://youtu.be/y5uBYc_t1NQ,http://youtu.be/pSZ5OnjBxUk,http://youtu.be/yTe4rfd062o
[ii] http://youtu.be/r5zI3m-xmIk,http://youtu.be/uKBUl07jBo0,http://youtu.be/BpQqUDgi3iY
[iii] http://youtu.be/Ry4fKCV1o3s
[iv] http://youtu.be/l3oSh4MLWMY

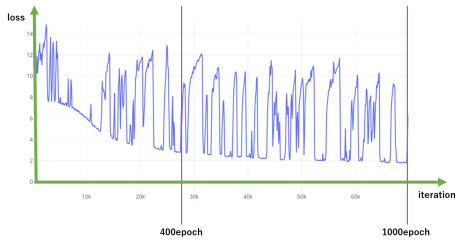

図 2: 1000epoch における loss の推移

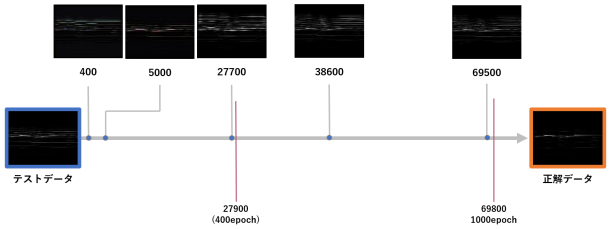

図 3: 各 epoch における生成画像

ると考られる．打ち込み音源データによる実験では学習データを増やし，また batch size 5 から batch size 1 に小さくし試行した．

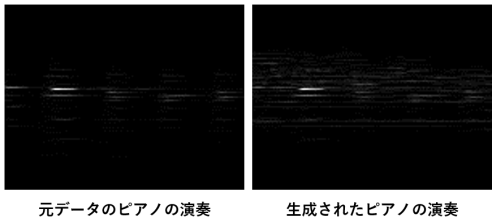

図 4: 1000epoch における生成画像との比較

5.2 打ち込み音源データによる学習

5.2.1 学習

打ち込み音源のデータセットでさらに大量のデータを用意し，トランペットとピアノの演奏可能音域にあわせてランダムに打ち込み音を生成し，それぞれ学習データの元データ・結果データを 961 個用意した．学習は 5.1 の考察より，batch size を小さくすることで画像生成の精度が上がると考えられたので batch size 5 と batch size 1 で行い比較した．

5.2.2 テスト

トランペットとピアノによる打ち込み音源を作成した．これを打ち込みテストデータとする．また打ち込み音源による関係性モデルでも人間の演奏音源から楽器音を抽出できるのかを試行するため，5.1 と同じトランペットとピアノで演奏された音源を合成した音源を用意した．これを人間テストデータとする．それらを 5.1 と同様に 1 秒ごとに分割し，スペクトログラムに変換した．テストとして，打ち込みテストデータから打ち込まれているピアノのスペクトログラム，また人間テストデータの合成音源から合成前のピアノのスペクトログラムを生成した．このピアノのスペクトログラムをそれぞれ打ち込み正解データ・人間正解データとする．

5.2.3 結果

loss の推移は図 5 に示す．batch size 5，batch size 1 の loss の最小値はそれぞれ 5.775213，5.360880 である．今回 batch size を変えたが epoch 数は変えていないため，batch size 5 のグラフは途中で切れている．また図 6 は今回の学習で loss が最小値だった batch size 1 の学習モデルを使って生成されたスペクトログラムと，打ち込み正解データであるスペクトログラムの比較である．

5.2.4 考察

batch size 1 にしたところ，iteration 39100 から loss が右肩上がりになってしまった．これは batch size を極端に減らしたことによる過学習だと考えられる．

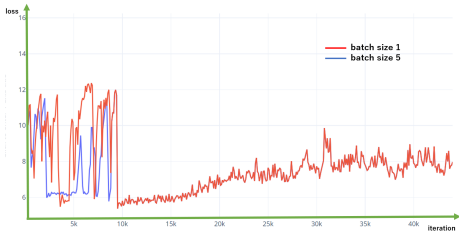

図 5: batch size 1 と 5 における loss の推移

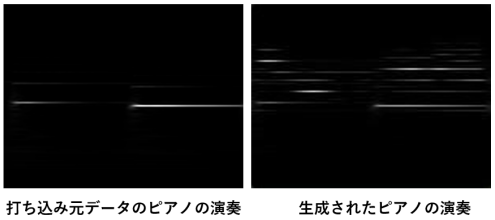

図 6: batch size 1 における生成画像との比較

6 まとめ・展望

End-to-End 学習を使用しスペクトログラムのペアから関係性モデルを生成した．そのモデルを使用し二重奏のスペクトログラムから抽出したい楽器の音で演奏されるスペクトログラムを生成したが，結果は抽出したい音の他にもノイズが含まれてしまった．

今後は batch size を徐々に小さくしていくことで画像生成の精度向上を図る．また各データのスペクトログラムを見たときに必ず黒色の部分があるためその部分を切り取り，画像サイズを小さくさせて学習させる．更に打ち込み音源による学習回数の増加・人間の演奏による音源データセットの購入など，様々な手法を試行し考察していく．

参考文献

[1] History of ybbjc. `https://www.yamano-music.co.jp/docs/ybbjc/siryoukan/history.html`.
[2] Isola Phillip, Zhu Jun-Yan, Zhou Tinghui, and Efros Alexei A. Image-to-image translation with conditional adversarial networks, Nov 2018.
[3] 瑞紀小林, 宏史手塚, 真理稲葉. 楽譜を用いた楽器音分離手法の提案. EC2015 論文集, 2015.
[4] Stoller Daniel, Ewert Sebastian, and Dixon Simon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In *19th ISMIR*, Jun 2018.
[5] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, September 2018.