ニューラルネットワークによる
音楽の自動変換

教養学部後期課程学際科学科総合情報学コース

陶山大輝

学籍番号：08-192021

指導教官：金子知適准教授

目次

第 1 章

初めに

本研究ではニューラルネットワークを用いて音楽を変換することを目標とする。変換は音楽における remix と呼ばれる手法と同様に行う。remix とは既存の曲の再構成及び加工を行ってその曲の新しいバージョンを作成する方法のことである。また、本研究では、remix を音楽の構造を変化させない音色の変換による加工に限定する。

第 2 章

音楽用語の定義

本章では音楽用語の定義及びその説明を行う。また、本章の音は楽譜作成ソフトの MuseScore[*1]により作成し、音波の波形の画像は音声編集ソフトの Audacity[*2]により作成した。

2.1 音

音とは、弾性体 (空気) 中を伝播する弾性波により起こされる音波が聴覚により感じられるもののことである。また、音波に周期性があり明確な音程を持つ音として聞こえる場合は楽音と呼ばれる。

2.2 楽音の三要素

楽音は高さ, 大きさ, 音色の三つの要素 (音の三要素) から成り立ち、人間はこれらを知覚することができる。本論文では楽音のことを音と呼ぶ。

2.2.1 音の高さ

音の高さは音波の周波数により決まる。人間には周波数が高い音は高く、周波数が低い音は低く知覚さる。また、複数の周波数の音波が音に含まれる場合は最も低い周波数成分の音波 (基音) を音の高さとして知覚する。国際の音名表記では、C,D,E,F,G,A,B を西洋音楽の七音音階におけるオクターブとして定め、それぞれのオクターブに番号を振り、440Hz の音を C4 と定めることで、任意の半音の絶対的な表記を可能にしている。国際の音名表記では半音よりさらに細かい音 (微分音) を表すことはできないが、本論文では扱わない。

ここで、図 2.1 は図 2.2 で示されるある音波にフーリエ変換を行うことで得られる周波数スペクトルである。周波数スペクトルはそれぞれの周波数成分がどれだけその音波に含まれているかを示すので、この音波の基音は 264Hz とわかる。

2.2.2 音の大きさ

音の大きさは音波の振幅により決まる。人間には振幅が大きい音は大きく、振幅が小さい音は小さく知覚される。

[*1] https://musescore.org/

[*2] https://www.audacityteam.org/

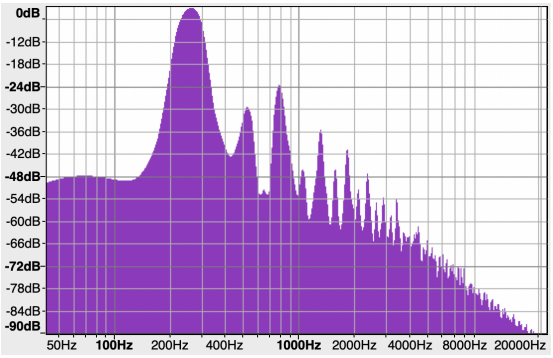

図 2.1　音波の周波数スペクトル

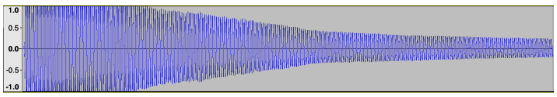

図 2.2　C4 の音波の波形

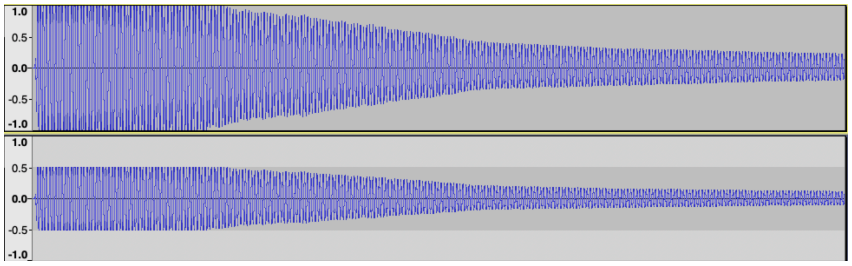

図 2.3　音の大きさの異なる音波

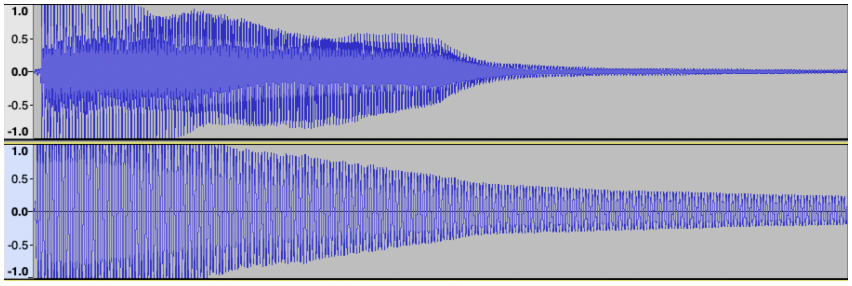

図 2.4　ギターとハープの音色

図 2.3 では、同じ楽器から出る同じ高さの音の音波を示しており、振幅の大きい後者の方が大きい音として人間には知覚される。

2.2.3　音の音色

音の高さと大きさが同じであっても異なった音として人間には知覚されることがある。この違いを音色と呼ぶ。

図 2.4 では、上側はギターの音波, 下側はハープの音波で同じ高さかつ同じ大きさである。この音波の波形の違いが音色の違いを作り出す。

第 3 章

背景

本章では、Multilayer perceptron 及びその応用例の Generative Adversarial Networks の説明を行った後、Generative Adversarial Networks を画像の変換に応用した Pix2pix を紹介する。

3.1 Multilayer perceptron

Multilayer perceptron (MLP) は複数のアフィン変換と非線形関数からなる関数により定義され、式 3.1 で定式化される。

$$F_{MLP}(\boldsymbol{x}) = f_n(W_n(f_{n-1}(W_{n-1}\cdots(f_1(W_1\boldsymbol{x}+\boldsymbol{b_1}))\cdots+\boldsymbol{b_{n-1}}))+\boldsymbol{b_n}) \tag{3.1}$$

ここで、$\boldsymbol{x}$ は実数ベクトル,n は 1 以上の整数,f_i は i 番目の非線形関数,W_i は i 番目の行列,b_i は i 番目の定数ベクトルである。また、n を層数と呼ぶ。

MLP (F_{MLP}) を用いると、あるデータ集合 $D = \{(\boldsymbol{x_j}, \boldsymbol{t_j}); 1 \leqq j \leqq N\}$ について、それぞれの j で $\boldsymbol{t_j}$ の予測値として $\boldsymbol{y_j} = F_{MLP}(\boldsymbol{x_j})$ を出力することができる。しかし、$\boldsymbol{t_j}$ に近い $\boldsymbol{y_j}$ を出力するためにはそれぞれの i で W_i を適切に決める必要がある。損失関数 $L(\boldsymbol{y_j}, \boldsymbol{t_j})$ を定義し、$W_i \leftarrow W_i - \eta\frac{dL}{dW_i}$ として更新すると、$\boldsymbol{y_j}$ を $\boldsymbol{t_j}$ に近づける方向に W_i を更新することができる (勾配降下法) 。ここで、η は任意の正の実数であり、学習率とも呼ばれる。損失関数としては、平均二乗誤差 $L_{MSE} = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{y_j} - \boldsymbol{t_j})^2$ や平均絶対誤差 $L_{MAE} = \frac{1}{n}\sum_{j=1}^{n}|\boldsymbol{y_j} - \boldsymbol{t_j}|$ などが用いられる。

一般には、ニューラルネットワークは MLP より広い概念であるが、本論文ではこれ以降 MLP のことをニューラルネットワークと呼ぶ。

3.2 Generative Adversarial Networks

Generative Adversarial Networks (GAN) [1] はニューラルネットワークの応用例であり、学習データの特徴を学習して擬似的なデータを生成することを目指す手法である。この手法は自然な手書きの文字を出力する際などに用いられる。

GAN は二つのニューラルネットワークで構成され、それぞれ識別モデルと生成モデルと呼ばれる。この二つのモデルはランダムに初期化された後に競合的に学習を進める。まず、識別モデルはデータが生成モデルの出力と学習データのどちらであるかを識別できるように学習を進める。そして、生成モデルは識別モデルが学

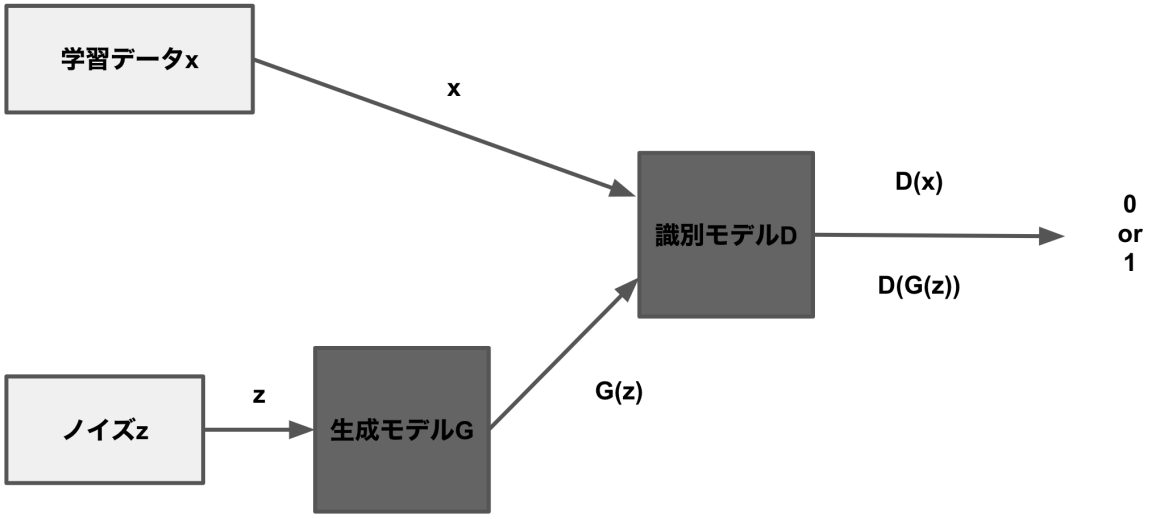

図 3.1　GAN のネットワーク

習データであると誤って識別するよう学習データに近いデータを出力する。この二つの学習を交互に繰り返すことで、漸進的に生成モデルが学習データにより近いデータを生成できるようになると期待される。

生成モデルの目的関数は式 3.2, 識別モデルの目的関数は式 3.3 として定式化される。

$$\arg\min_{\theta_G} \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z};\theta_G);\theta_D))] \tag{3.2}$$

$$\arg\max_{\theta_D} \mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x};\theta_D)] + \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z};\theta_G);\theta_D))] \tag{3.3}$$

ここで、$\boldsymbol{x}$ は学習データ,$\boldsymbol{z}$ は生成モデルへの入力のノイズ,$G(\boldsymbol{z};\theta_G)$ はノイズ $\boldsymbol{z}$ を入力とする生成モデル,$D(\cdot;\theta_D)$ は識別モデル,θ_G は生成モデル G のパラメータ,θ_D は識別モデル D のパラメータである。

3.3　Pix2pix

Pix2pix [2] はある条件下で画像間の変換を行う GAN である。例えば、図 3.3 のようにピクセルの対応関係を変えずにスタイル変換を行うことができる。Pix2pix は GAN に変換先の学習データを条件として与えることでスタイル変換を行う。生成モデルの目的関数は式 3.4, 識別モデルの目的関数は式 3.5 でとして定式化される。

$$\arg\min_{\theta_G} \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}}[\log(1 - D(\boldsymbol{y}, G(\boldsymbol{y},\boldsymbol{z};\theta_G);\theta_D))] + \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[\|\boldsymbol{x} - G(\boldsymbol{y},\boldsymbol{z};\theta_G)\|_1] \tag{3.4}$$

$$\arg\max_{\theta_D} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y};\theta_D)] + \mathbb{E}_{\boldsymbol{y},\boldsymbol{z}}[\log(1 - D(\boldsymbol{y}, G(\boldsymbol{y},\boldsymbol{z};\theta_G);\theta_D))] \tag{3.5}$$

ここで、$\boldsymbol{x}$ は変換先の学習データ,$\boldsymbol{y}$ は変換元の学習データ,$\boldsymbol{z}$ は生成モデルへの入力のノイズ,$G(\boldsymbol{y},\boldsymbol{z};\theta_G)$ は

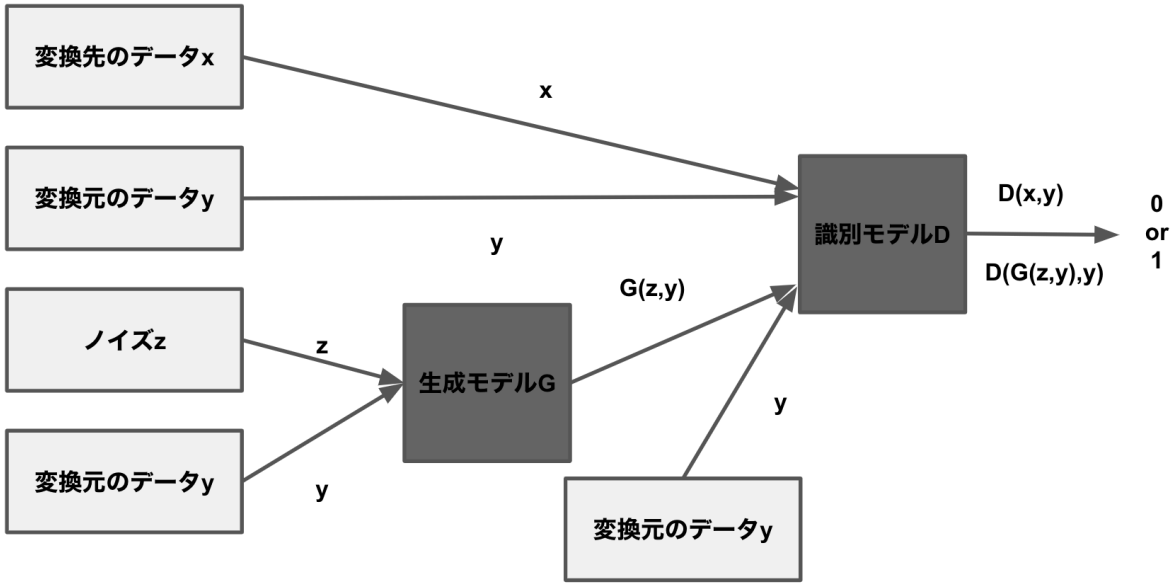

図 3.2　pix2pix のネットワーク

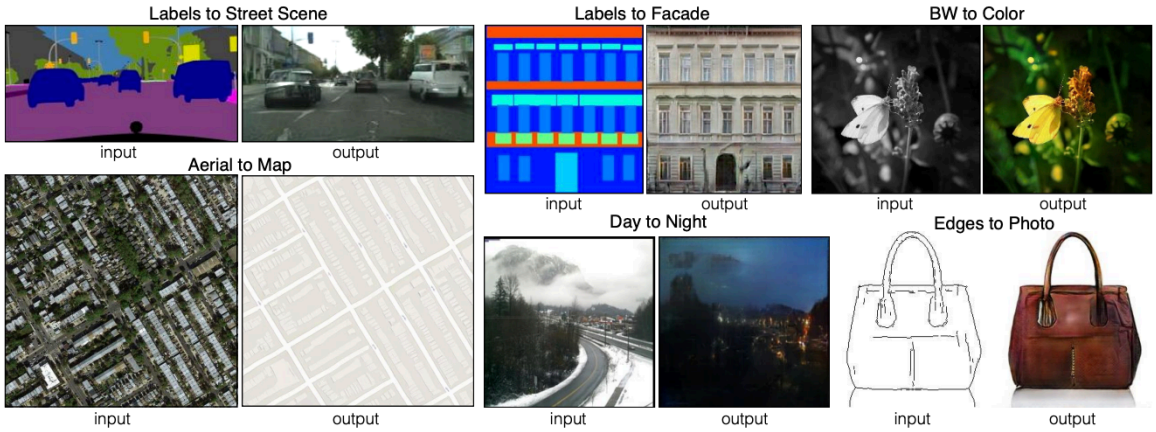

図 3.3　pix2pix のスタイル変換の例

$\boldsymbol{y}$ を条件としノイズ $\boldsymbol{z}$ を入力とする生成モデル,$D(\boldsymbol{y}, \cdot; \theta_D)$ は $\boldsymbol{y}$ を条件とする識別モデル,θ_G は生成モデル G のパラメータ,θ_D は識別モデル D のパラメータである。

3.3.1　生成モデルの構造

変換を行うための生成モデルには Encoder-decoder のネットワークが用いられる。Pix2pix の生成モデルでは、画像データの基礎的な構造を保持するために、図 3.4 のように U-net[3] というスキップコネクションを持ったネットワークが用いられる。

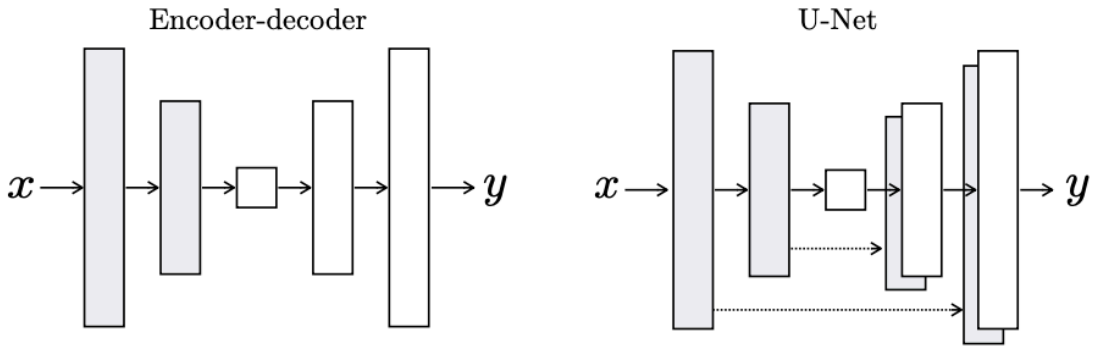

図 3.4　Encoder-decoder のネットワークと U-net のネットワーク。図は文献 [3] の図の○○から引用。

3.3.2　識別モデルの構造

Pix2pix の識別モデルでは、PatchGAN という手法が用いられる。PatchGAN は画像全体の誤差を求めるのではなくパッチと呼ばれる小領域ごとで誤差を求めて平均を取っている。これにより、局所的な部分の識別の精度が高まることが期待される。

第 4 章

提案手法

本章では、提案手法である単音の音色の変換について説明する。

一般的な音楽で特定の音色へと変換を行うことは難しい。そこで、著者は三つの要素に分解することに着目した。一つは楽器の重ね合わせ、もう一つは音の重ね合わせ、そして最後は時間方向の音の繋ぎ方である。具体的には以下で説明をする。

楽器の重ね合わせ

音楽はそれぞれの楽器により奏でられた音の重ね合わせになっている。楽器ごとに音色は異なるので楽器ごとの音波に分解して音色変換を行う (音源分離) が必要であると考えられる。なお、楽曲の作成時に楽器ごとに分離したデータ (パラデータ) で保存しておけば、音源分離を行わずに直接楽器ごとの音波を利用できる。

音の重ね合わせ

ある単位時間の音に注目した時、楽器ごとの音波に分解してもその単位時間で異なった高さや大きさの音の重ね合わせになっていることがある。この場合は、今回の提案手法で用意したデータセット以外に和音のデータセットも加えて学習させることで表現可能であると考えられる。

時間方向の音の繋ぎ方

楽器ごとの音波に分解し単位時間の音が表現できた時、時間方向に音を繋ぐ必要がある。時間方向については先程定めた単位時間で区切って順に変換していくことで可能であると考えられる。

以上の三つの要素に分解すれば、単音での特定の音色への変換手法を確立することで一般的な音楽に適用可能であると考えられる。そこで、本研究では単音での特定の音色への変換手法を提案する。

4.1 データセット

本節では、本研究のデータセットの作成方法及びその形式についての説明を行う。

4.1.1 データセットの作成方法

楽譜作成ソフトの MuseScore[*1]により、A0 から C8 までの半音を 88 音生成した。この 88 音は最も一般的な音域として今回の実験では選んだ。また、ギターからハープの音へと音色の変換を行うので、そのどちらも

[*1] https://musescore.org/

88 音を生成した。

4.1.2 データ形式

音のファイル形式としては非圧縮形式の WAV 形式を用いる。WAV 形式は音波の波形データを直接保持しているために扱いやすいので本論文では採用した。また、WAV 形式は音波の波形データ以外にメタデータも持つ。以下では、本論文で関係のあるメタデータについて説明する。

サンプリング周波数

サンプリング周波数とは、デジタル信号の 1 秒あたりの標本化の回数のことである。本論文では 44100Hz に固定して実験を行う。

サンプリング数

サンプリング数とは、デジタル信号の標本化の合計の回数のことである。本論文では 44100 回に固定して実験を行う。

量子化ビット数

量子化ビット数とは、デジタル信号の細かさを表現するビット数のことである。本論文では 16bit に固定して実験を行う。

チャンネル数

チャンネル数とは、モノラルな音声の出力の総数のことである。本論文では 1 に固定して実験を行う。

4.2 手法

Pix2pix を用いて十分に音色が異なると考えられるギターからハープへの音の変換を行った。また、学習と評価の際のパラメータは付録 A の A.1 節に示す。

4.2.1 ニューラルネットワークのモデル

本実験では、Pix2pix を元に Pix2pix においては以下のようなニューラルネットワークのモデルを用いた。

4.2.2 生成モデルの表現力の評価

生成モデルの表現力を測るために学習データと評価データに同じ 88 音のデータセットを用いて実験を行う。また、88 音と小さいサイズのデータセットへの過学習を防ぐためにそれぞれのデータの振幅を乱数で変化させる工夫をした。

具体的には、乱数のシードをあらかじめ固定した後、各エポックで学習データとして与えるデータごとに振幅を c 倍した。c の範囲は $[0.3, 1]$ であり、一様乱数により c を生成している。

4.2.3 生成モデルの汎化能力の評価

一般に、単音の音色変換において任意の高さと大きさの音を学習データとして用意するのは不可能である。従って、未知のデータも適切に処理できるニューラルネットワークを構築する必要がある。本手法のニューラルネットワークが未知のデータも適切に処理できるか調べるため、88 音のうち $\frac{3}{4}$ を学習データ,$\frac{1}{4}$ を評価デー

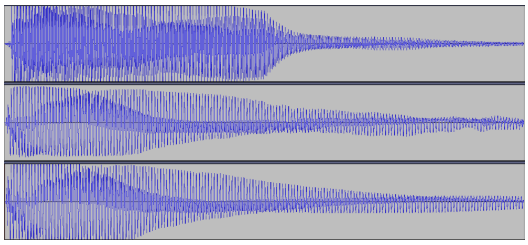

図 4.1　F3 の 0.800 秒から 1.000 秒までの音波

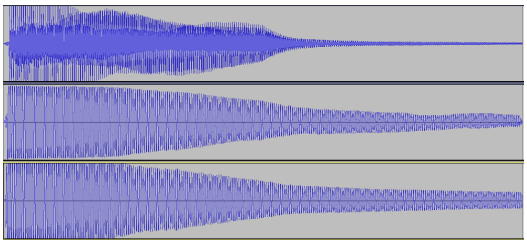

図 4.2　C4 の 0.200 秒から 0.300 秒までの音波

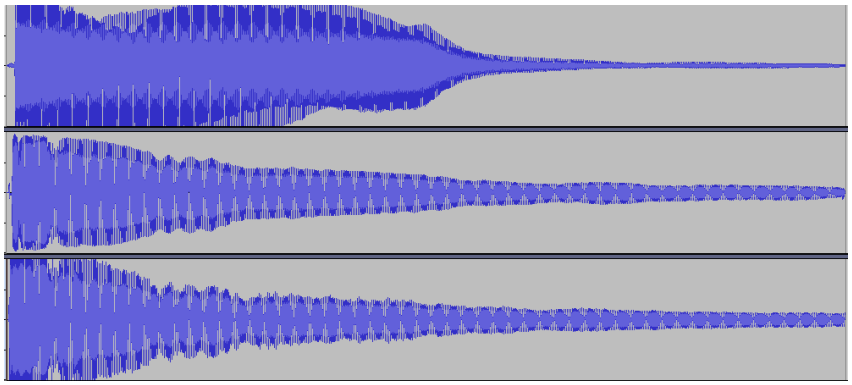

図 4.3　C5 の音波

タとする 4 分割交差検定を行う。この際のデータセットの分割方法は付録 A の A.2 節に示した。また、先程の実験と同様、各エポックで学習データとして与えるデータごとに振幅を $c \in [0.3, 1]$ 倍して学習を行った。

4.3　実験結果

本節では、実験結果及びその考察をまとめる。また、生成された音波の波形を画像で添付する際に音声編集ソフトの Audacity[*2]を用いた。実験結果に記載する画像については全て上から順に元のギターの波形, 変換後の波形, ハープの波形となっている。

4.3.1　生成モデルの表現力

学習の際の loss は図のようになり学習が進んでいることがわかる。

まず、F2 から G6♯ の音は耳で聴いても波形を見ても図 4.1 のように正確にハープの音を表現することができていた。特に C4 から D5♯ の音は図 4.2 のように上音の音が少ないために綺麗なハープの音を生成することができていた。他の音についても下記に挙げた改善点はあるものの基音の部分は表現することができていた。以下では、ハープの音を生成において困難であった点を列挙する。

音波の振幅

図 4.3 の波形の前半のように、生成された波形の振幅が小さいという問題が発生した。これにより、減

[*2] https://www.audacityteam.org/

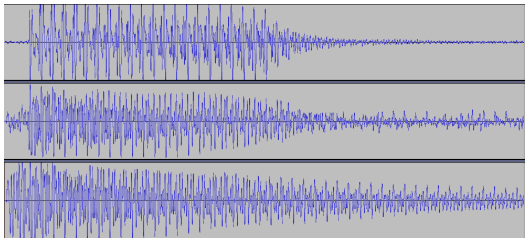

図 4.4　F1♯ の 0.000 秒から 1.000 秒までの音波

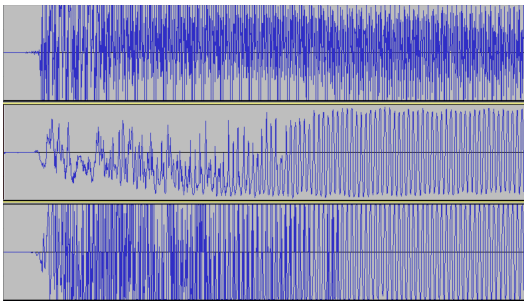

図 4.5　A7 の 0.000 秒から 0.030 秒までの音波

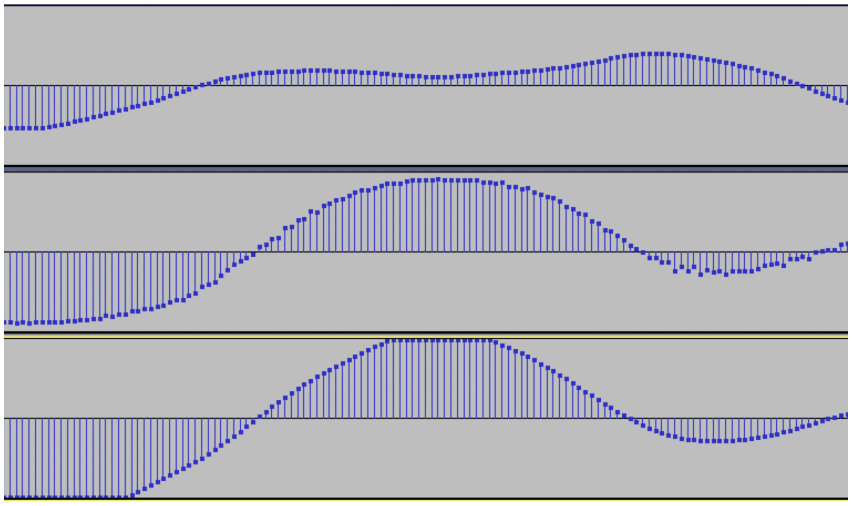

図 4.6　D2♯ の 0.100 秒から 0.103 秒までの音波

衰していく部分の表現が難しい場合があった。原因は、振幅をランダムに小さくしたためであると考えられ、学習の初段階では振幅を固定するなどの工夫する必要があると思われる。

音の鳴り出しの遅延

ギターとハープは弦楽器なので鳴り出すまでに遅延がある。特に、今回用いたデータセットでは図 4.4 のようにギターの音に遅延がある場合や図 4.5 のように周期的な音になるまでに遅延がある場合、その部分を学習することが難しかった。

音波の滑らかさ

図 4.6 の上から 2 番目の波形と 3 番目の波形を比較すると、生成された波には滑らかさがないことがわかる。人間の耳にはこの滑らかでない音波の部分はノイズまじりの音として聞こえるので、この波を滑らかにするために工夫が必要であると考えられる。

音波の振動の減衰

音波の振動の減衰が全く表現できていない音波があった。具体的には、A0,A0♯,B0,C1,C1♯,D1,D1♯,E1,F1,F1♯,G1,G1♯D2,D2♯,E2,A6,A6♯,B6,C7,C7♯,D7,D7♯,F7,F7♯,G7,G7♯,A7,A7♯,B7,C8 の音である。また、これ以外の音についても減衰が一部表現できていないものはあった。

これらの音のうち、E2 以下の低音域の場合は図 4.7 のようにハープとは全く異なる波形で減衰し、A6

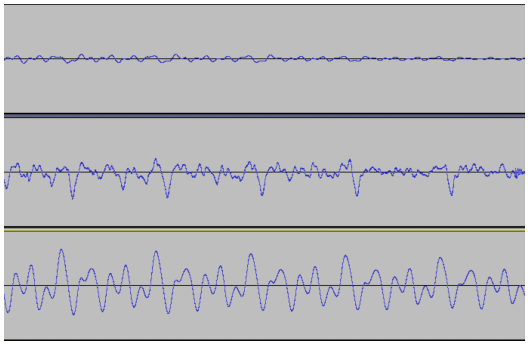

図 4.7　A0 の 0.800 秒から 1.000 秒までの音波

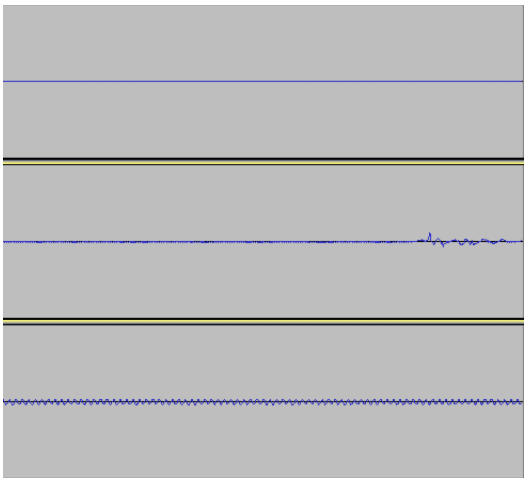

図 4.8　B7 の 0.980 秒から 1.000 秒までの音波

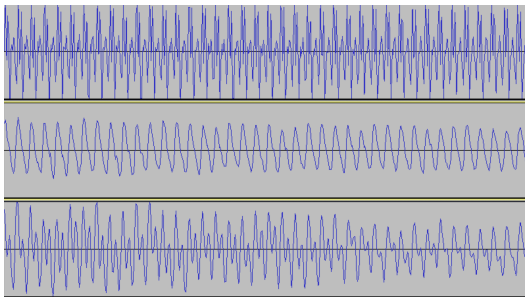

図 4.9　E7 の 0.055 秒から 0.070 秒までの音波

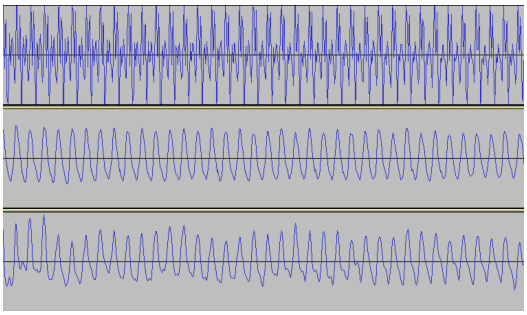

図 4.10　D7♯ の 0.055 秒から 0.070 秒までの音波

以上の高音域の場合は図 4.8 のようにほとんど振動が見られなかった。原因としては、微小な振動をニューラルネットワークが学習するのが難しいことと今回のデータセットではギターの方がハープよりも音の鳴る時間が短かったことが挙げられる。

安定したデータセットの作成

E7 の音については、図 4.9 のように上記に挙げた以外の部分で波が表現できていなかったが、ハープのデータセットの音波の振動の振幅が安定していないために、ニューラルネットワークによる学習が難しかったと考えられる。

また、E7 の半音下の音である D7♯ でも図 4.10 のようにハープのデータセットの音波の振動の振幅は安定しておらず、特に高音は安定した振幅のデータセットを生成することが難しいと考えられる。

4.3.2　生成モデルの汎化能力

第 5 章

まとめ

ネットワーク変えれるのでは？もっとランダムにできるのでは？本研究では○○を確認することができた。以後は以下をやる。

バランス悪いデータセットの可能性→バランス良くしたら

ハープからギターの方向別の楽器との比較

二音のみの組み合わせでできるのか前処理を工夫するかフーリエ変換を用いて sin 波に分解するか

本研究では波形の観察による考察を行ったが、より定量的な判定を行うには判定器の導入が必要であると考えられ、具体的には以下の二点での判定が必要である。

(1) 音程が維持されているか

(2) 変換されて音色が変換されているか

波形とスペクトログラム？

謝辞

本研究を進める際に丁寧なご指導をして頂いた指導教官の金子知適准教授に厚く感謝を申し上げます。また、研究に関して助言を頂いた金子研の構成員の方々にも感謝の意を表します。

参考文献

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

付録 A

A.1　実験時のパラメータ

実験時のパラメータを表 A.1 に示す。

表 A.1

パラメータ	値
インプットのバッチサイズ	1
学習でのエポック数	1000
学習での学習率	0.0002

A.2　データセットの分割方法

生成モデルの汎化能力を調べた際のデータセットのサブセットへの分割を表 A.2 に示す。表記は国際の階名表記に従い、音の高さの昇順に並んでいる。データセットの分割方法としては 88 音をシャッフルして配列に格納した後に 22 音ずつ順に選んでいる。また、4 分割検証を行っているため、番号 i $(0 \leqq i \leqq 4)$ の 22 音のサブセットを評価に用いた際は番号 j $(0 \leqq i \leqq 4$ かつ $i \neq j)$ のサブセットの 66 音を全て学習に用いている。

表 A.2

番号																					
0	C1	E1	G1	C2♯	F2♯	G2	A2♯	G3	D4♯	E4	F4	G4♯	A4	F5	A5♯	B5	E6	F6	B6	F7	A7♯	B7
1	C1♯	D1	D1♯	F1♯	G1♯	A1	A1♯	C2	D2♯	A2	B2	A3♯	G4	C5	D5♯	F5♯	A5	C6♯	D6	E7	G7	G7♯
2	A0♯	B0	D3	D3♯	E3	F3	A3	C4	C4♯	D4	C5♯	D5	G5	G5♯	D6♯	G6	A6	A6♯	C7♯	D7♯	F7♯	C8
3	A0	F1	B1	D2	E2	F2	G2♯	C3	C3♯	F3♯	G3♯	B3	F4♯	A4♯	B4	E5	C6	F6♯	G6♯	C7	D7	A7