公立はこだて未来大学 2018 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2018 System Information Science Practice
Group Report

プロジェクト名
AI するディープラーニングプロジェクト

Project Name
AI Love Deep Learning

プロジェクト番号/Project No
06

プロジェクトリーダ/Project Leader
1016163 濱口 和希/Kazuki Hamaguchi

グループリーダ/Group Leader
1016163 濱口 和希/Kazuki Hamaguchi

グループメンバ/Group Member
1016163 濱口 和希/Kazuki Hamaguchi
1015205 齋藤 匠/Takumi Saitou
1016065 白鳥 孝幸/Takayuki Shiratori
1016132 山田 大貴/Daiki Yamada

指導教員/Advisor
竹之内 高志/Takashi Takenouchi
香取 勇一/Yuuichi Katori
寺沢 憲吾/Kengo Terasawa
片桐 恭弘/Yasuhiro Katagiri
冨永 敦子/Atsuko Tominaga

提出日/Date of Submission
2019 年 1 月 16 日/January 16，2019

概要

声質変換 (Voice Conversion: VC) とはボイスチェンジャーの一種であり，ある人物の入力音声をまるで特定の人物が話したかのように変換して，出力する手法である．近年では、個人やバーチャルユーチューバーの動画配信において，声質変換が用いられるなど，声質変換に対する需要が高まっている．現在用いられている手法の一つとして，入力された音声を一度テキストに変換し，変換させたテキストから合成音声を出力するものがある．しかしこの手法は、入出力までの時間が長い，テキストに変換する際に誤認識が発生する，などの問題点がある．そこで私たちは，前述した問題点を解決することのできる新しい手法を提案する．具体的には，特定話者の声を，目標話者である琴葉茜の音声に変換することを目指した．特定話者から目標話者へ声質変換を行う際，最適なモデルを構築するためにディープラーニングを用いた。

適用した手法は，2 段階のモデルで構成されている．1 段階目は，元話者の声を低品質な琴葉茜の声に変換する，低品質な声質変換モデルである．このモデルでは，入力話者の基本周波数とメルケプストラムを，目標話者の基本周波数とメルケプストラムに近づけるよう変換を行った．このモデルの学習には，大量のパラレルデータが必要となる．しかし，パラレルデータの作成は非常に難しく，時間を要する．そのため，パラレルデータのかさ増しを行った．具体的には，Noise，Stretch，Shift の 3 種類で加工を行い，新しい音声データとして作成した．これにより，効率的にパラレルデータを作成することができた．2 段階目は，低品質なスペクトル包絡を，高品質なスペクトル包絡に変換する，高品質化モデルである．このモデルの学習には，目標話者の音声データが大量に必要となるため，数名で協力しながら音声データの作成を行った．

最後に，変換された基本周波数，高品質なスペクトル包絡，非周期成分から，目標話者に近付けた合成音声に変換することができた．

(文責：白鳥)

Abstract

Voice conversion is a kind of voice changer and it is a method of converting and outputting the input voice as if a specific person spoke. Recently, the demand of voice conversion is increasing in live broadcasting of individuals and virtual YouTuber. One of the method they currently use is convert the input voice to text once and output synthesized speech from the converted text. But this method has problems which it takes long time, the misrecognition may be happened, and so on. Therefore, we propose a new method that can solve the mentioned problems. Specifically, we aimed to convert the voice of a particular speaker into voice of Akane Kotonoha, the target speaker. We used Deep Neural Network in order to construct models when converting particular voice to target voice.

The method we applied consists of two models. In the first model, it converts the particular voice into the voice of Akane Kotonoha with a low quality. We converted the fundamental frequency and mel cepstrum of the input speaker closer to the fundamental frequency and mel cepstrum of the target speaker in this model. The learning of this model needs a lot of parallel data. But making parallel data is very difficult and take time. So we bulk out these parallel data. Specifically, we process the voice data with the way of Noise, Stretch, Shift and we creates new voice data. We made parallel data efficiency by doing so. The second model is a high quality model that converts low quality spectral envelopes into high quality spectral envelopes. The learning of this model needs a lot of target voices, so we cooperate to make data.

Finally, from the converted fundamental frequency, high quality spectral envelope, aperiodic component, we could convert it to synthesized speech approaching the target speaker.

(文責：白鳥)

目次

第 1 章　はじめに

この章では，プロジェクトの目的と背景について述べる．

1.1　プロジェクトの目的

世界が情報化社会に移り変わる中，大量のデータを処理するための手法として機械学習は日々発展を続けてきた．その中で，企業が持つビッグデータの存在とその価値が周知されるようになると，機械学習は一気に注目を集め，様々な企業で導入されるようになった．ビッグデータを扱う上で，特に適しているとされたのがディープラーニングである．ディープラーニングは機械学習技術の一つで，ニューラルネットワークという人間の神経構造を模した機械学習モデルのうち，隠れ層という層をいくつか重ねたものを指す．層を重ねることで，データを分析するために必要な要素とその関係を, より多くかつ複雑に蓄積することができるようになっている．ディープラーニングの精度は用意する学習データ量に左右されるため，限られたデータ量でより精度を上げるための研究開発が進められている．本プロジェクトでは，このディープラーニングを用いて新たな問題解決に挑むことを全体の目標とする．

(文責：坂下)

1.2　背景

ボイスチェンジャーと呼ばれているソフトウェアやハードウェアが世の中に多く存在している．ボイスチェンジャーは，自分の声に対して声の高さや声質を変換し，新しい声を作り出すことができるソフトウェアやハードウェアのことである．以前のボイスチェンジャーは，個人の特性や匿名性を上げるために用いられる機会が多かった．しかし，現在は自分の声を，自分の望む特定の声に変換するために用いられる機会が増えている．例えば，バーチャルユーチューバーの動画配信でボイスチェンジャーが用いられる．バーチャルユーチューバーとは，仮想空間で 3D モデルを動かしながら，主にユーチューブなどで動画配信を行う架空のキャラクターである．個人で動画配信を行っているバーチャルユーチューバーにおいて，多くの場合，モデルの作成から動画配信まですべて一人で行っている．そのため，作成されるキャラクターと本人の性別が一致していない場合がある．したがって，ボイスチェンジャーで男女間の声質を変換していることが多い．その中で声質変換という手法が用いられている．声質変換 (Voice Conversion: VC) とはボイスチェンジャーの一種であり，ある人物の入力音声をまるで特定の人物が話したかのように変換して出力する手法である．

(文責：濱口)

1.3　現状における問題点

一般に利用されている声質変換の 1 つに，入力音声を一度テキストに変換させ，変換させたテキストから合成音声を出力する手法がある．ここで，入力音声を一度テキストに変換する段階を音声認識とする．また，テキストから合成音声を出力する段階をテキスト音声合成 (Text-to-speech: TTS) とする．音声認識とテキスト音声合成を組み合わせた手法をテキストベース変換手法 (図 1.1) とする．

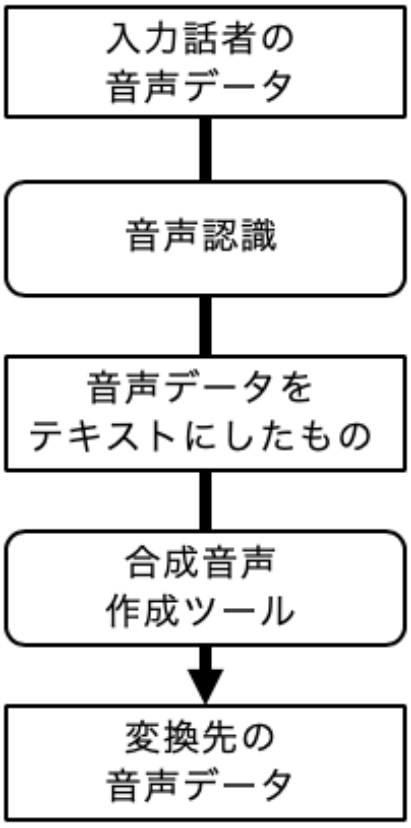

図 1.1　テキストベース手法の概要図

テキストベース変換手法にはいくつかの問題がある．まず，テキストベース変換手法では，音声認識とテキスト音声合成の 2 つの段階を経る必要がある．そのため，音声の入力から出力までが遅くなるという問題がある．実際に，テキストベース変換手法を用いた「ゆかりねっと」というツールでは，音声の入力から出力までに約 5 秒から 8 秒の時間を必要とする．また，音声認識では，音の途切れで，文章と次文章の判断をしている．したがって，一定時間以上連続して話していると，音声認識精度が落ちてしまう．その結果，誤認識が発生しやすくなる．さらに，変換させたテキストから合成音声を出力する際，テキスト音声合成を用いる．合成音声が必要となるため特定の人物の声に変換する際は，その人の合成音声を作成する必要がある．そのため，変換先が，既に合成音声として作成されている，SofTalk や VOICEROID に限られる．これらが，我々が改善すべき問題である．

(文責：濱口)

1.4 目的

本グループの目的は，テキストベース変換手法よりも高性能な声質変換手法を開発することである．その後，テキストベース変換手法と開発した変換手法をグループ内で検証して，性能評価を行う．

(文責：濱口)

第 2 章　課題設定までのプロセス

2.1　グループ目標の設定

2.1.1　前期のグループ目標

我々は，入力音声を直接変換し，合成音声を出力する手法で声質変換に取り組む．入力音声を直接変換し，合成音声を出力する手法を，直接変換手法 (図 2.1) とする．テキストベース変換手法で

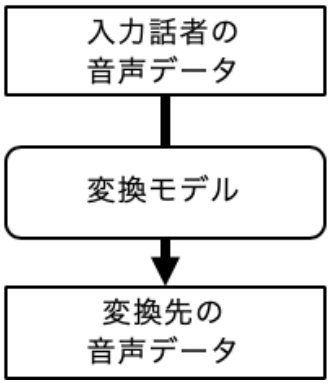

図 2.1　直接変換手法の概要図

はなく，直接変換手法に取り組むことにより，現状における問題点の改善が図れると考えられる．

まず，直接変換手法では，入力から出力にかかる時間が短縮される．テキストベース変換手法を用いた「ゆかりねっと」というツールでは，音声の入力から出力までに約 5 秒から 8 秒の時間を必要とする．しかし，直接変換手法の先行事例であるディープラーニングを用いた声質変換システム [1][2] では，入力から出力までの遅延が，約 3 秒から 4 秒である．また，テキストベース変換手法では音声認識の段階で，誤認識が発生していた．しかし，直接変換手法では音声をテキストに変換しない．そのため，誤認識は発生しない．さらに，テキストベース変換手法では音声の変換先が，SofTalk や VOICEROID に限られていた．そのため，SofTalk や VOICEROID に収録されていない特定の個人の声，アニメキャラクターなどの声に変換することが難しい．

ここで，直接変換手法内の変換方法に，自分の音声をまるで特定の人物が話したかのように変換し， かつ変換先の対象を自由に変更する方法 [3] がある．したがって，この手法を用いることで自由な対象への声質変換ができると考えられる．

現状，直接変換手法には，様々な種類の変換方法がある．例えば，隠れマルコフモデル (Hidden Markov Model : HMM) を用いる方法，混合ガウスモデル (Gaussian Mixture Model： GMM) を用いる方法，ディープニューラルネットワーク (Deep Neural Network) を用いる方法などがある．その中でも，DNN は HMM に比べ，より自然な合成音声を出力できる [4]．また，DNN は GMM と比べ，より高精度な変換が可能である [5]．以上より我々は，DNN を用いた手法での開発を前提とする．

(文責： 齋藤)

2.1.2 後期のグループ目標

前期の終了時点では，提案手法と最も関係が深い先行事例 [7] の代替策として，先行事例 [1][2] の追実験を実施していた．そのため，後期では先行事例 [7] の追実験を優先的に実施するため，先行事例 [1][2] の追実験は実施しないことに決定していた．

夏季休暇中に，パラレルデータを必要としない Phonetic PosteriorGrams (PPGs) を用いた多対 1 の声質変換手法 [5] を利用した先行事例 [7] の追実験に取り組んだ．しかし，先行事例 [7] に対する詳細な説明を見つけることができず，環境の構築や，データの割り当てが上手く行かなかった．そのため，自分たちが想定していた以上に，追実験を進めることができなかった．したがって，最終発表までにデモを行うと考えた場合，このまま先行事例 [7] に取り組むことは，スケジュール面から難しいと考えた．

一方で，前期に追実験を試みていた，畳込みニューラルネットワークを用いた音響特徴量変換と，スペクトログラム高精細化による声質変換手法 [6] の先行事例 [1][2] がある．先行事例 [1][2] を用いて追実験をしていた．前期の追実験の際に，学習に使用する音声データの作成が上手く行えていなかった．しかし，夏季休暇中にさらに追実験を試みたところ，前期に問題であった箇所が解消できた．

そこで，プロジェクトのグループ目標を「ノンパラレルデータを用いたリアルタイム声質変換」から，「パラレルデータを用いた声質変換」に変更し，先行事例 [1][2] を利用していくことに決定した．

(文責：齋藤)

2.2 期間ごとの課題設定

2.2.1 前期の課題設定

はじめに，グループ内で前期の目標を話し合った．話し合いの中で下記の課題が上がった．

・音声処理，音声認識についての知識を習得
・声質変換についての知識を習得
・実際に収集し選択した手法の追実験の実施
・入力音声を直接変換して，合成音声を出力する手法を用いて　簡単なモデルの構築
・音声データの作成または，購入

上記で設定した課題から具体的な活動方針を決定した．まず，音声処理，音声認識についての知識を習得する点と，声質変換についての知識を習得する点から，声質変換手法の収集と選択という課題を設定した．また，実際に声質変換手法の収集と選択をしていくなかで，実際に収集し選択した手法の追実験を行うと，さらに声質変換に関する理解が深まると考えた．そこで，先行事例の追実験を行うという課題を設定した．そして，先行事例の追実験を行うため，音声データが必要であると判明した．したがって，音声の作成・収集という課題を設定した．

(文責：齋藤)

2.2.2　後期の課題設定

後期の課題設定において，グループ内で話し合った．話し合いの中で下記の課題が上がった．

・参考にするソースコードの選択

・音声データの作成及び加工

・音声データの組み合わせを検証

設定した課題から具体的な活動方針を決定した．実装の際，参考にするソースコードを動かすのに環境を設定する必要があったため，環境構築という課題を設定した．また，前期から活動方針が大きく変わったため，前期に作成していた音声データを使用することができなくなった．そのため，音声データの作成・加工という課題を設定した．加工した音声データのかさ増しを行い，様々な組み合わせで試しに変換してみたところ，精度が高い音声データの組み合わせと，精度の低い音声データの組み合わせが存在した．したがって，音声データの組み合わせの検証という課題を設定した．

(文責：濱口)

2.3　期間ごとの担当割り当て

2.3.1　前期の担当割り当て

まず設定した課題より，グループ全体で声質変換について学習した．そして，音声データの作成を行う音声収集班と，ネットワーク構築を行うネットワーク班に別れた．音声の作成には，濱口の提案で VOICEROID を用いて行うことにした．さらに，濱口が VOICEROID を 2 つ所有していたため，所有者である濱口と山田を音声収集班にした．また，ニューラルネットワークや音声処理に興味がある，齋藤と白鳥をネットワーク班にした．

(文責：齋藤)

2.3.2　後期の担当割り当て

後期で設定した課題である環境構築，音声データの作成・加工，音声データの組み合わせを解決するため，グループ内で柔軟に担当を割り振り解決した．

(文責：濱口)

第 3 章　活動内容

3.1　前期の活動内容

3.1.1　声質変換手法の収集と選択

まず，個人の Web サイト，書籍や論文で声質変換に関する資料を各メンバーが個別に収集した．そして，収集した資料の中で，特に我々が求める変換手法と関係が深い，個人の Web サイト，書籍や論文をグループ全体で読み合わせた．

読み合わせた文献の中に，パラレルデータを必要としない Phonetic PosteriorGrams (PPGs) を用いた多対 1 の声質変換 [5] がある．声質変換手法の概要を図 3.1 に示す．

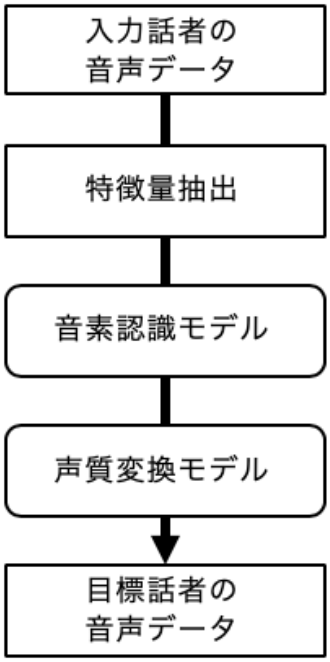

図 3.1　先行事例 [5] の概要図

図 3.1 より，多数のノンパラレルデータで学習可能な音素認識モデルと，少数の対象音声で学習可能な声質変換モデルを構築する．これにより，自分の音声をまるで特定の人物が話したかのように声質変換し，かつ変換先の対象を自由に変更することが可能になる．また，畳込みニューラルネットワークを用いた低品質な声質変換モデルと高品質化モデルによる声質変換 [6] がある．声質変換手法の概要を図 3.2 に示す．

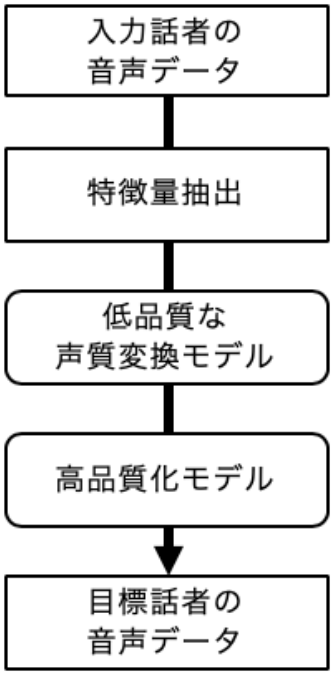

図 3.2　先行事例 [6] の概要図

図 3.2 より，少数のパラレルデータで学習可能な声質変換モデルと，多数のノンパラレルデータで学習可能な高精細化モデルを構築する．これにより， 必要なパラレルデータ数を抑えつつ，高品質な声質変換を行うことが可能である．パラレルデータとは，入力話者と目標話者が全く同じ内容の文章を話し，対になっている音声データのことである．

(文責：齋藤)

3.1.2　先行事例の追実験

ニューラルネットワークに関する理解を深めるため，実際にコードが公開されている先行事例の追実験を行うことに決定した．

はじめに，我々の目標に近い，自分の音声をまるで特定の人物が話したかのように声質変換し，かつ変換先の対象を自由に変更することが可能である先行事例 [7] の追実験を実施しようと試みた．この先行事例は，パラレルデータを必要としない Phonetic PosteriorGrams (PPGs) を用いた多対 1 の声質変換 [5] で提案された手法を用いている．そのため，音素認識モデルと声質変換モデルを構築する必要がある．また，音素認識モデルでは学習を行うため，多数のノンパラレルデータが必要である．声質変換モデルの学習を行うには，学習済みの音素認識モデルが必要ある．しかし，音声資源コンソーシアムから購入した音声データが届くのに時間がかかり，音素認識モデルの学習を行うことができなかった．したがって，先行事例 [7] を実際に動作させることはできなかった．そのため，コードを読み，システムの構成どのようになっているか，どのような関数により音素認識や声質変換を行っているのかを把握するにとどまった．

以上より，先行事例 [7] が前期中には行えないと判明した．そこで，必要なパラレルデータ数を抑えつつ，高品質な声質変換が可能である先行事例 [1][2] の追実験を実施しようと試みた．この先行事例は，畳込みニューラルネットワークを用いた低品質な声質変換モデルと，高品質化モデルによる声質変換 [6] で提案された手法を用いている．そのため，声質変換モデルと高精細化モデル

を構築する必要がある．また，声質変換モデルでは学習を行うために，最低 50 文ほどのパラレルデータが必要である．そのため，パラレルデータを作成した．その後，追実験を開始した．しかし，用意した音声データから学習に用いるデータに変換する段階で， 学習に用いるデータを作成するプログラムが動作しないという問題が発生した．したがって，その後の実験を行うことができず，追実験を完遂することができなかった．本来，提案手法と最も関係が深い先行事例 [7] の代替策として，先行事例 [1][2] の追実験を実施した．そのため，後期では先行事例 [7] の追実験を優先的に実施するため，先行事例 [1][2] の追実験は実施しないことに決定した．

(文責： 齋藤)

3.1.3 音声の作成・収集

音素認識モデルを構築するために，様々な人の音声データが必要であった．そのため，音声資源コンソーシアムから「日本音響学会 新聞記事読み上げ音声データ」を購入した．この音声データは，16679 文，時間にして約 70 時間である．また，声質変換モデルを構築するために変換先の音声データも必要であった．そのため，濱口が所持している VOICEROID を用いて約 1000 文，時間にしておよそ 1 時間分の音声データと，対応するテキストデータを作成した．

(文責： 齋藤)

3.2 前期の個人活動

ここでは，前期末時点における個人の活動及び，後期の活動に対する展望を述べる．

3.2.1 濱口 (グループリーダー、音声収集班)

前期では，テーマ出しにおいて声質変換への熱意を大きく伝え，グループを率いて目標を立てた．また，音声収集班として VOICEROID を用いて音声データを作成した．作成した音声データは約 1000 文で時間は 1 時間ほどである．音声データ作成後はネットワーク班と協力して，追実験を実施しようと試みた．しかし，環境構築の段階においてエラーが多発し，実施することができなかった．私はこの原因が Python の理解不足とミドルウェア設定の違いだと考え，参考書 [8][9] を用いて学習を行っている．また，前期はグループ開発がスムーズにできていなかった．理由は，データや情報の管理において，グループ内でのルールが，明確に定められていなかったためであると考えられる．そのため，グループ内でのデータや情報の管理を行う人を決める必要がある．

後期では，まず，グループ開発の環境を整えるため，Git/GitHub を用いたオンラインで管理を行う．Git とはバージョン管理システムである．GitHub はオンライン上でファイルの管理が行えるツールである．次に，追実験を成功させるため，環境構築および Python の理解を深める．環境構築においては全員の環境を同じにするため環境設定を記入したファイルを作成する．Python の理解を深めるため，サンプルコードを実行させ，内部を確認しながら，理解不足の部分を明確にさせる．

(文責： 濱口)

3.2.2 山田 (音声収集班)

前期では，音声収集班としてグループで購入を考えていた音声データの候補を，インターネットでできるだけ多く探した．その後，グループリーダーと購入する音声データを決定した．その後，VOICEROID を用いて音声データを 250 文作成した．また，参考書 [10] を用いて簡単な音声処理プログラムを自分の PC で実行した．その後，ネットワーク班に合流した．ネットワーク班が追実験を試みていて，Python の知識や声質変換の知識がほかのメンバーよりなかったため，手伝うことができなかった．

後期では，グループ内でのデータ管理を Git/GitHub を用いて行うことに決まったので，Git/GitHub に関する知識を Web サイトや書籍を用いてつけていきたい．また，声質変換についての知識が，ほかのグループメンバーよりも少ないので，後期に向けて自分で論文や書籍を読んで知識を蓄えたいと考えている．

(文責：山田)

3.2.3 白鳥 (ネットワーク班)

前期では，書籍または Web サイトを用いて声質変換に関連する文献を調査・検討した．そして，多様な手法を主にネットワーク班で比較し, その中で PPGs を用いて声質変換を行うことを決定した．また，声質変換に対する理解を深めるため，先行事例の追実験 [1][2] を試みたが, 失敗した．その理由のうちの一つとして，先行事例の元コードを読むことができるほどの知識が私たちになかったことが挙げられる．そのため，夏季休暇中に Python，機械学習，音声信号処理など，全般について幅広い知識を学習する必要があると考える．私は，特に機械学習に関する学習を重点的に行うことを考えている．具体的には，東京大学の松尾研究室の公開しているコンテンツ [11] である，GCI データサイエンティスト育成講座，Deep Learning 基礎講座を修了させることを目標とする．また，後期では GitHub を使った開発が予想されるため，夏季休暇中に書籍を用いて学習を行う．

後期では，夏季休暇中に学び得た知識を活用し，GitHub を用いて実際に開発を行う．また，Slack 等のコミュニケーションツールを用いて，メンバーと積極的にコミュニケーションを取る．メンバー同士で理解の至らない部分を補いながら，グループの中心として開発を行いたい．

(文責：白鳥)

3.2.4 齋藤 (ネットワーク班)

前期では，Web サイトや書籍，論文から声質変換に関連する技術についての資料を収集した．そこから，声質変換に対する理解を深めるためには自分の音声をまるで特定の人物が話したかのように声質変換し，かつ変換先の対象を自由に変更することが可能である先行事例 [7] の追実験を行うべきだとグループ内で主張した．しかし，この先行事例では，モデルの構築に音声データが必要である音声データの到着が予想以上に遅くなってしまったため，前期中に追実験を始めることができなかった．そこで，別の手法で声質変換している追実験 [1][2] を実施しようと試みた．こちらは，音声データを変換する段階でエラーが発生し，モデルの学習を始めることができなかった．したがって，コードを読み，システムの構成どのようになっているか，どのような関数により音素認

識や声質変換を行っているのかを把握するにとどまった．

後期では，前期で学習した知識を用いて，実際に先行事例 [7] の追実験を実施し，声質変換を行うシステムのプロトタイプの作成を行いたいと考えている．先行事例 [1][2] の追実験は我々の目標とは少々異なっているのでこれ以上は行わない．プロトタイプの作成をする段階からはそれぞれ具体的な実装が始まると考えているので，その際に理論や方法などの問題に答えて行けるように，変換対象の限られない声質変換に関連する技術についての学習を続けていく．

(文責：齋藤)

3.3 後期の活動内容

3.3.1 環境構築

後期の活動において使用する手法の変更があったため，環境を一から作り直した．環境構築の手順を図 3.3 に示す．また，環境構築において使用したツールとバージョンを表 3.1 に示す．

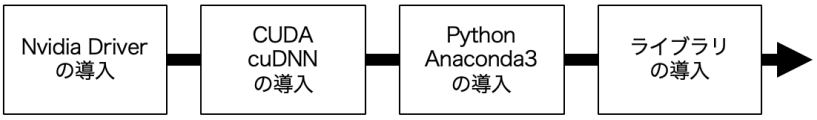

図 3.3 環境構築の手順

表 3.1 環境構築に使用したツール及びバージョン

ツール	バージョン
Ubuntu	17.10
Python	3.6.3
Anaconda3	5.0.1
Nvidia Driver	410.48
CUDA	9.0.176-1
cuDNN	7.7.1 for CUDA 9.0

構築の際に注意した点が 4 つある．

1 つ目は，Nvidia Driver をインストールする際に，セキュアブートを無効にすることである．セキュアブートを無効化せずに Nvidia Driver をインストールし再起動すると，Ubuntu 側が GPU を認識しなくなる．そのため，セキュアブートを無効化したのち，Nvidia Driver をインストールする必要がある．

2 つ目は，cuDNN をインストールするバージョンである．cuDNN をインストールする際に CUDA と Ubuntu のバージョンを確認し，指定されたバージョンに揃える必要がある．しかし，Ubuntu17.10 に合った cuDNN のバージョンは存在しなかった．調べた結果，Ubuntu17.04 の cuDNN をインストールする [12] と，Ubuntu17.10 でも使用可能だったため，Ubuntu17.04 及び CUDA9.0 の cuDNN7.1.1 をインストールした．

3 つ目は，Python と Anaconda3 におけるバージョンを一致させることである．基本 Python と Anaconda3 のバージョンは一致していない．例えば，Anaconda3 のバージョン 5.2.0 にインス

トールされている Python のバージョンは 3.6.5 であった．さらに，Anaconda3 における Python のバージョンは，インストールしてからでないと Python のバージョンを確認できない状態だった．したがって，Python のバージョンを 3.6.3 にするため，Anaconda3 を一つ一つインストールし，Python を起動させ確認する必要があった．その結果，Python のバージョンが 3.6.3 であった，Anaconda3 のバージョン 5.0.1 をインストールした．

4 つ目は，ライブラリのインストールである．Anaconda3 にはすでに基本的な機械学習を行うためのライブラリがインストールされているが，声質変換を行うために，追加した主なライブラリを表 3.2 に示す．ライブラリ中で音声処理を行っているのが，librosa，pysptk，world4py である．

表 3.2　導入したライブラリ及びバージョン

ライブラリ	バージョン
librosa	0.6.2
pysptk	0.1.11
world4py	0.1
chainer	4.5.0
chainerui	0.3.0
cupy	4.5.0
tensorflow-gpu	1.11.0

機械学習を行うライブラリが chainer，chainerui であり，GPU で並列処理を行うためのライブラリが cupy，tensorflow-gpu である．これらのライブラリをインストールするときに注意しなければいけないのが，cupy，world4py，chainerui のインストールである．

まず，cupy において注意しなければいけないのが，バージョンである．pip install cupy を用いてインストールを行うと，コードを動かすときにエラーが出てしまう．このエラーは CUDA のバージョンと cupy のバージョンが違うときに出てしまう．今回は CUDA のバージョンが 9.0 のため，cupy のインストールには pip install cupy-cuda90 を用いてインストールを行う必要がある．

次に，pyworld のインストール時において pip install world4py を用いてインストールするのではなく world4py の github[13] から直接ダウンロードを行い，インストールする必要がある．なぜなら，pip install world4py を用いてインストールを行うと，setup.py というインストールが完全に行えたかを確認するコードでエラーが出てしまうからである．そのため，world4py をインストールする際，github のサイトから直接ダウンロードを行い，インストールをする必要がある．

最後に，chainerui のインストール時において，pip install chainerui==0.3.0 のようにバージョンを 0.3.0 以内に指定する必要がある．これはバージョンが 0.4.0 以上だと JSON データの扱いが変わってしまいコードを動かすことができないからである．そのため，chainerui のインストール時にはバージョンを 0.3.0 以内に指定してからインストールする必要がある．

以上 4 点が，環境構築を行った際に注意すべき点である．

(文責：濱口)

3.3.2 音声データの作成・加工

低音質な声質変換モデルと高音質化モデルの構築するために，それぞれのモデルの学習に必要な音声データを作成した．低品質な声質変換モデルでは，入力話者と目標話者のパラレルデータを作成した．パラレルデータとは入力話者と目標話者が全く同じ内容の文章を話し，対になっている音声データのことである． パラレルデータは入力話者が読み上げの時，話す速度や話すタイミング，話し方を真似る必要があり，作成に時間がかかる．入力話者の音声データは，Audacity[14] を用いてマイク入力した声を録音することによって作成した. 録音した入力話者の音声データは，wav 形式，サンプリングレートが 22050Hz，量子化ビット数を 16bit である． 目標話者の音声データは，VOICEROID + EX 琴葉茜 [15] を起動させ，読ませたい文章を入力することによって作成した．低品質な声質変換モデルと高品質化モデルに使う目標話者の音声データは，どちらとも wav 形式，サンプリングレートが 22050Hz，量子化ビット数を 16bit である．また，琴葉茜の音声データを作成する際には，文章を入力し，音声を再生し，音声データに名前を付け保存するという操作が必要で時間がかかる．そこで，作成する時間を短縮するために VOICEROID-YMM 連携マクロ [16] を使用した．VOICEROID-YMM 連携マクロを使うことにより，VOICEROID + EX 琴葉茜で音声データを作成する際，音声の再生や，音声の保存など，決められた操作をショートカットキーを押すことによって自動化してくれる．また，このようにソフトの決まりきった操作を自動化したプログラムをマクロという．この VOICEROID-YMM 連携マクロを使うために UWSC[17] が必要なため，インストールした．UWSC と VOICEROID-YMM 連携マクロを起動させ，キーボードの Ctrl キーと P キーを同時に押すと音声の再生，Ctrl キーと R キーを同時に押すと音声の保存を行うことができる．次に，低品質な声質変換モデルと高品質化モデルに使用した入力話者と目標話者の文章内容について説明する．低品質な声質変換モデルで使用したパラレルデータの文章内容は， ATR 音素バランス 503 文 [18] である．音声データを収集した後，パラレルデータとして扱える音声データを抽出した．高品質化モデルに使用した目標話者の音声データの文章内容は，Yahoo!ニュースや web 小説など作成者同士で作成する音声データの文章が同じにならないようにした．また，音声データの長さは 2.3 秒以上 10 秒未満の長さになるように作成した．

次に，モデル別に作成したデータの個数について説明する．始めに，低品質な声質変換モデルでは，447 個のパラレルデータを作成し，学習に使用した．高品質化モデルでは，2362 個の目標話者の音声データを作成し，学習に使用した．

また，先ほども述べたように，低品質な声質変換モデルにおいてパラレルデータが必要となるが，パラレルデータの作成には時間がかかる．そこで，パラレルデータに対して加工を行うことによって，加工前の音声データとは別の音声データとして学習に使用できるようにした．音声データをかさ増しすることによって、パラレルデータ作成の手間を省いた．パラレルデータに対して，Noise，Stretch，Shift の 3 種類の加工を行った．1 つ目の Noise は，元の音声データに対して，ある一定の大きさの雑音を加える加工である．まず，元の音声の wav データの配列と同じ大きさの配列を用意する．この配列は，すべての要素において，平均値が 0，標準偏差が 1 で正規分布するようにランダムで作成される．そして，ランダムで作成された配列を 0.005 倍する．この配列を元の音声の wav データの配列に加算することによってノイズを追加している． 2 つ目の Stretch は，音声の再生速度を変える加工である．今回は，パラレルデータに対して再生速度を 1.1 倍に伸ばした．これは librosa の time-stretch と呼ばれる関数を使用して加工を行った． 3 つ目の Shift は，読み上げの開始地点を変える加工である．例えば，「ワインと日本酒等を問わず，原産地，成分表示を

急ぐべきではないか」という音声データに Shift を行うと「日本酒等を問わず，原産地，成分表示を急ぐべきではないかワインと」のように元のデータと比べ，読み上げの始まる位置を変える加工である．この加工は，numpy の roll を用いて行った．roll は，配列の要素を回転させる関数であり今回は元の音声データに対してこの関数を用いて加工を行った．また，「ワインと日本酒等を問わず，原産地，成分表示を急ぐべきではないか」と琴葉茜が話している加工前の音声データの波形を図 3.4 に示す．先ほど説明した 3 つの加工を, 加工前の音声データに行ったときの波形を Noise, Stretch, Shift の順番で図 3.5, 図 3.6, 図 3.7 に示す.

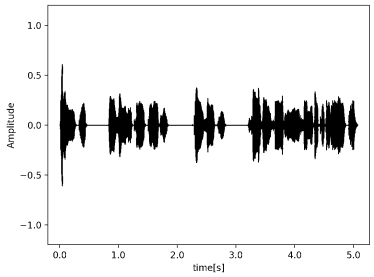

図 3.4　加工前の琴葉茜の音声データ

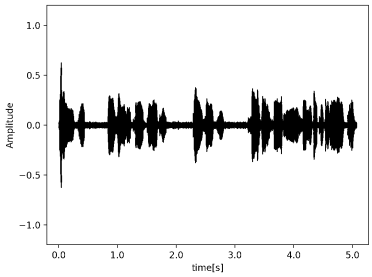

図 3.5　Noise 加工後の琴葉茜の音声データ

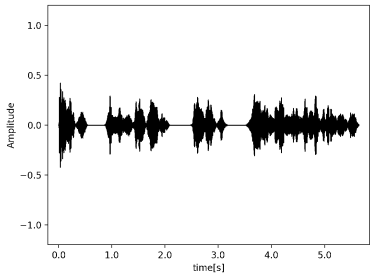

図 3.6　Stretch 加工後の琴葉茜の音声データ

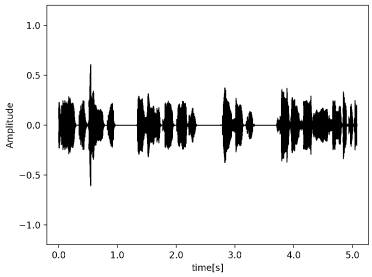

図 3.7　Shift 加工後の琴葉茜の音声データ

(文責：山田)

3.3.3 学習データの組み合わせ

3.3.2 の音声データ作成・加工より元の音声データから Noise，Stretch，Shift のデータを作成した．その後，低品質な声質変換モデルの学習時に 4 種類のデータを組み合わせ学習させた．学習後，実際に変換を行った音声データを聞き主観的に聞き取れるかの評価をした．評価した結果を表 3.3 に示す．

表 3.3　元の音声のみで学習したモデルとの比較

学習に利用した データの組み合わせ	主観的評価	日本語として 聞き取れるか
元の音声，noise	さ行がかすれて聞こえた	○
元の音声，shift	言葉が崩れ聞き取れなかった	×
元の音声，stretch	元の音声とほぼ変わらなかった	○
元の音声，noise，shift	言葉が崩れ聞き取れなかった	×
元の音声，noise，stretch	母音が連続する場所が聞き取れなかった	○
元の音声，shift，stretch	言葉が崩れ聞き取れなかった	×
元の音声，noise，shift，stretch	言葉が崩れ聞き取れなかった	×

表 3.3 より評価基準は主観的評価と日本語として聞き取れるかの 2 つで評価を行った．評価基準を 2 つ設けたのは，主観的評価で琴葉茜の音声に変換できているかどうかの確認及び，変換後の音声が日本語として聞こえるかを評価するためである．評価の結果，Shift を組み合わせた学習データでは，変換後のノイズが酷く日本語としては聞き取ることができない部分が多く存在した．これは，音声データの読み上げ位置を変更したため不安定な発話部分が学習されたためだと考えられる．Noise を組み合わせた学習データでは，変換後も日本語として聞き取ることができた．しかし，母音が連続する単語，例えば，「あおいあおい（青い青い)」などの部分では精度が著しく下がり非常に聞き取りにくかった．また，さ行が含まれる単語，例えば，「さげた（下げた)」などの部分でも声がかすれるように聞こえ発音が正常にできていなかった．Stretch を組み合わせた学習データでは，変換後も元の音声とほぼ変わりなく聞き取ることができた．以上より元の音声と Stretch の組み合わせ音声データを，低品質な声質変換モデルに使用した．

(文責：濱口)

3.4　後期の個人活動

ここでは，後期末時点における個人の活動を述べる．

3.4.1 濱口 (グループリーダー)

後期では，チーム全体の技術力が不足していたため，個人で実装できることを探して少しずつスキルアップが行えるようにした．その後，プロジェクトのゴール地点をノンパラレルを用いたリアルタイム声質変換から，パラレルデータを用いた声質変換に変更しようと話し合い決めた．自分が担当した役割が OSS[1] を動かすための環境構築，1 段階目における自身と琴葉茜のパラレルデー

タの作成，2 段階目における琴葉茜の音声データの作成，齋藤によってデータかさ増しされたパラレルデータの最適な組み合わせの考案，OSS の使用方法が解説されている web サイト [2] を参考にし OSS を実際に動かし，必要なフォルダ構成及びファイルの確認，スライド・ポスターの作成，プロジェクトに必要な備品の申請と多岐に渡る部分を担当した．

まず，個人で行ったスキルアップにおいて，ピアノの単音を識別する識別器を作成した．データセットとして，音階を 7 段階，音の高さを 6 段階，音の強さを 4 段階に分け，全部で 168 個のピアノ単音音声を作成した．データセットの分け方として，学習用データ，パラメータチューニング用データ，テスト用データを 17:5:2 の割合で分割した．学習には scikit-learn[19] の SVM を識別器として使用した．実際に学習をさせ，チューニングを行い，テストで確認すると SVM の *gamma* 値によって識別器の性能が大きく変わった． *gamma* 値による正答率の違いは表 3.4 で示す．

表 3.4　各 *gamma* 値と正答率

gamma 値	正答率
1.0×10^{-1}	42.9%
1.0×10^{-2}	50.0%
1.0×10^{-3}	100.0%
1.0×10^{-4}	92.8%
1.0×10^{-5}	64.2%
1.0×10^{-6}	21.4%
1.0×10^{-7}	0.0%

表 3.4 より，*gamma* 値は 1.0×10^{-3} が最も高い精度であった．これは SVM を用いて音声を線形分類する際に，$gamma = 1.0 \times 10^{-3}$ が最も適切なマージンを取ることができたと考えられる．

OSS を動かすための環境は 3.3.1 の環境構築で述べたことを行った．

1 段階目における自身と琴葉茜のパラレルデータ作成及び，2 段階目における琴葉茜の音声データ作成，パラレルデータのデータかさ増しにおいては 3.3.2 音声データの収集・加工で述べたことを行った．自身と琴葉茜のパラレルデータ作成において，工夫したところは琴葉茜の話速がやや早かったため，読み上げた文章を 2 回ほど繰り返し 3 回目で録音を行うようにした．さらに，琴葉茜の音声データは有声音か無声音の 2 種類しかないため自身の音声も有声音以外のノイズが入っている場所を無声音にする加工を行ったところである．作成には 500 個で 15 時間ほどかかったため，時間短縮につながる方法を考える必要があると考えた．

OSS の使用方法が解説されている web サイトを参考に実際の動作，必要なフォルダやファイル構成の確認においては，4 章で述べたことを行った．ここでの注意点は OSS 自体の解説があまりなかったため，同じサイトを見て確認を行った．そのため，そのサイトでは述べられていないエラーに遭遇したときに，解決するのに非常に時間がかかった．しかし，ここでのエラーに対処したことによって，Linux における apt のパッケージ管理，shell における Path の通し方など OS に関する知識を学ぶことができた．また，学習モデルの動作チェックを行うため，Python でテストコードやチェックコードを書くことが多かった．これにより，音声を操作すること，音声波形図の作成，行列の操作など Python を用いて何かを行うことができるようになった．

スライド・ポスターの作成においては，自分たちが作業してきた内容のアウトプットを相手に伝えることができた．

課題は，チーム作りに時間をほとんど割くことができなかったことである．

まず，連絡ツールとして Slack を導入したが，チーム内での連絡が活発に行うことができなかった．原因として，Slack を定期的に確認するということをチーム全体で義務付けなかったためであると考えられる．また，連絡ツールを Slack だけにしてしまったため，言葉のやりとりでは伝わりにくい部分ができてしまった．解決策として，ビデオ通話などが行える Discode や Skype を用いることによって，改善が行えたと考えられる．以上より連絡ツールにおいては，チャット式とビデオ通話機能があるツールを導入すべきだと感じた．また，連絡がつかなかったときのために各自の携帯電話番号を控えるのも有効的だと感じた．

次に，ソースコード，議事録，課題，スケジュールの管理においては前期で GitHub を導入すると検討し，実際に導入したが，有効的に使用することができなかった．原因としては，GitHub をほぼ個人で使用する形になってしまったため，チーム内で GitHub を使用する人と使用しない人ができてしまった．原因として，チーム内で GitHub を使用する有用性が感じられない，学習するコストが大きくてする必要がないなどの意見が出てしまったためである．解決策として，提案した本人が GitHub の基盤構築と有用性の証明をする必要があった．基盤構築においては，ソースコードと議事録のディレクトリを分割作成し，Issues を用いて課題，スケジュールの管理するという使用方法を GitHub の wiki に記入する必要があった．GitHub 管理における有用性の証明は 3 つある．1 つ目は，開発の際に情報を一つの場所にまとめると，その場所を確認するだけで，ソースコード，議事録，課題，スケジュールの現状が把握できるということである．2 つ目は，GitHub は Slack と連携ができるため，GitHub の情報を Slack を通しても確認することが可能である．3 つ目は，ローカルのみでファイルを管理していると誰かの変更によって作業が止まってしまうため，クラウド上にデータを保存しておくことで，作業が止まってしまうことを避けることが可能となる．

以上の GitHub における基盤構築と有用性の証明を準備した上で，チーム内でハンズオンを開き，使い方を理解させた後，有用性を証明し，日常的に使用するツールとして定着させることが必要だったと考えられる．

最後に，目標の達成は行えたが作業量の偏りがチーム内で起こったことである．原因はプロジェクト時間中にチーム全体で，達成目標の解決方法がわからなくなったとき，どのように対応するのかを決めていなかったためである．グループ内で，タスクを大量に提案し一つずつ作業に入って行く方法をとったが，作業途中で目標の解決方法がわからなくなり手が止まってしまうことが多くあった．また，他のタスクを行っている人に原因を聞いたとしてもすぐに解決できないため，時間を多く消費してしまうことがあった．さらに，長期に渡って原因の解決方法がわからないと個人のモチベーションが保てないため，作業に集中できない人が出るようになった．その結果，作業を多くこなす人にタスクが集中してしまった．解決策としては，何がわからないのかを調べることを徹底する必要があった．そのためには，自分が行っているタスクに関連する言葉を調べ，その結果得られた情報を整理し，全体でその問題を共有する必要があった．情報の共有をプロジェクト前半開始 15 分，及びプロジェクト後半開始 15 分に報告する時間をとるべきだと考えられた．

以上より後期の課題はチーム作りに時間を割くことがなかったことである．この経験を生かして，チーム作りはプロジェクト開始時からなるべく早く行う必要があると感じた．

(文責：濱口)

3.4.2 山田

後期では，始めにチーム全体での技術力の不足のため，各自で実装できることを探してスキルアップを行うことになった．私は，Python を用いて簡単な識別器を実装した．その後，2 段階目の高品質化モデルの学習に使う琴葉茜の音声データの作成，最終成果発表会の発表原稿の作成，改良を行い，本番での発表を行った．

まず，個人で行ったスキルアップでは，犬か猫の鳴き声の wav データを入力し，その wav データはどちらの鳴き声のデータかを識別する識別器を Python を使用して作成した．また，機会学習ライブラリとして，scikit-learn と Keras の 2 つを使って識別器を作成したかったため，この 2 つを使用して犬か猫の鳴き声かを識別する同じ識別器を 2 つ作成した．識別器の作成に当たり，データセットとして，学習用に wav 形式の犬の鳴き声データ 36 個，猫の鳴き声データ 36 個，精度を調べるためのテスト用に，wav 形式の犬の鳴き声データ 4 個，猫の鳴き声データ 4 個の計 80 個を用意して使用した．始めに scikit-learn で作成した 1 つ目の識別器について詳しく説明する．

1 つ目の識別器では，scikit-learn のサポートベクトルマシンを使用して学習させた．精度のいいものを見つけるために，gamma 値を 0.05，0.1，0.01，0.001，0.0001，0.00001 の 5 種類に設定して，それぞれの gamma 値ごとに識別器を作成した．精度のいいものは，テスト用に使った犬と猫の鳴き声データ 4 個ずつ計 8 個のデータを用いて正答率を調べるためにテストを行い探した．学習させた識別器の精度としては，テストを行って一番良いものは gamma = 0.0001 の時に 87.5 % の正答率だった．精度が低かった理由としては，学習に使う犬と猫の鳴き声のデータが少ないことが原因だと考えられる．

2 つ目の識別器では，Keras を用いて作成した．まず，Python と Keras によるディープラーニング [20] を読んで，この本に沿って実際に Keras を用いて二値分類，多クラス分類，回帰についての簡単なモデルを実装しながら勉強した．その後，本を読んで作成した二値分類のモデル作成を参考にして，scikit-learn で実装したのと同じように犬か猫かを判別する識別器を作成した．

今回は，スキルアップということもあり犬と猫の少ない音声データで学習を行ったが，今後，このように識別器を作成する機会があれば大量のデータを用意し，学習やテストに利用できるようにしたい．機会があれば，今回は犬か猫の二値分類だけに挑戦したので，分類する動物を増やして様々な動物の鳴き声を判別する多クラス分類に挑戦したり，分類問題だけでなく，回帰問題にも取り組んでみたい．

2 段階目の高品質化モデルの学習に使う琴葉茜の音声データの作成は，自分のパソコンに VOICEROID + EX 琴葉茜をインストールし，プロジェクト学習の活動中や自宅で行った．また，琴葉茜の音声データ作成を短縮するため，3.3.3 で説明したように，VOICEROID-YMM 連携マクロを使うため，自分のパソコンに必要なものをインストールし使用した．高品質化モデルの学習に使用する琴葉茜の音声データの話している内容は，全く同じでなければどんな内容でもよく自由である．そこで，私は，Yahoo!ニュースニュースやブログなどの web サイトの記事の文章をもとに作成した．音声データの長さは，5 秒から 10 秒くらいの長さになるように最適なところで文章を分け作成した．また，音声データの個数としては約 1500 個程作成した．

最終成果発表会の原稿作成と改良では，濱口と齋藤が作った発表スライドをもとに，初めて声質変換やディープラーニングについて話を聞く人も理解できるよう，注意して原稿を作成した．いったん原稿を作成した後，グループメンバーに評価してもらい同じことを言っているところを削ったり，同じ言葉だが，ところどころで言い回しが変わっている言葉を統一したり，A グループの設定

されている発表時間よりも早く読み終えてしまったため，文量の追加を行った．原稿の作成後，濱口に最終確認を行ってもらい，その後，学校や自宅でテレビのモニターを使い，本番を想定しながら発表練習を行った．聴講者にわかりやすく伝えるために手で図を指し示したり，聴講者の顔を見ながら話せるように練習を個人で行った．加えて，本番は同じフロアでほかのプロジェクトも同時に発表しているため，後ろの聴講者が話を聞き取りにくいことが想定された．そのため，大きな声で発表することを意識した．

最後に，最終成果発表会の発表では，後半の 3 回の発表を担当した．発表内容は，本プロジェクト全体の概要と，A グループの活動内容を担当した．1 回目の発表は緊張もあり，スライドや原稿を見てしまい，聴講者のほうを向きながら話すことができなかった．しかし，2 回目以降からはスライドや原稿を見る時間を減らすことができ，スライドを指しながらであったり，聴講者の顔を見ながら大きな声で発表することができた．また，時間通りに発表をすることができた．3 回発表を行った結果として，伝えたいことを聴講者に伝えることができたと感じた．しかし，スライドや原稿を見ず，常に聴講者のほうを向いて発表することができなかったので，これから先，発表する機会があれば，このことを気を付けていきたい．

プロジェクト活動全体を通して，目的の達成ができたが，私は声質変換の技術的な面で活躍することができなかった．わからないことをはやめにグループメンバーや担当教員に報告し、相談するべきだったと考えている．これから先も，会社や大学でこのプロジェクト学習のような作業をすることがあると思うので，今回の経験を生かして，グループメンバーとのコミュニケーションを活発にとるようにしたい．

(文責：山田)

3.4.3 白鳥

後期のはじめは，前期に遂行することのできなかった声質変換に関する先行研究 [1][2] の追実験を引き続き行った．その先行研究は，前期の目標であったノンパラレルデータではなくパラレルデータを用いるものであったが，完遂することで音声変換に対するより深い知見を得られると考えたため, 実験に関する問題の解決を行なった．まず初めに，追実験のコードを動かすためのパラレルデータを約 30 組作成した．追実験の問題点として，上の階層のディレクトリに存在するモジュールの読み込みに失敗することが挙げられていた．前期では, bashrc に読み込みを行いたいディレクトリの相対 PATH を記述したが，実行することができなかった．問題を調査した結果，通常の PATH ではなく, PYTHONPATH を通さなければいけないことがわかった．そこで，bashrc に読み込みを行うディレクトリの相対パスを PYTHONPATH として記述したところ，抱えていた問題を解消することができ，無事に追実験を遂行することができた．その後，プロジェクトの残り日数や，自分たちの技術力を鑑みたところ，追実験で動かしたコードを参考にプロジェクトの成果物を作っていくことを決定した．決定後は，声質変換モデルの精度を上げるため，パラレルデータの作成数を 30 組から 447 組に増やした．また，高品質化モデルの精度を上げるため，琴葉茜の音声データを 1000 組作成した．また，先行研究には存在しなかった音声のリアルタイム入出力の処理を試みた．

(文責：白鳥)

3.4.4 齋藤

後期では，前期で学習した知識を用いて，実際に先行事例 [7] の追実験を実施し，声質変換を行うシステムのプロトタイプの作成を行いたいと考えている．先行事例 [1][2] の追実験は我々の目標とは少々異なっているのでこれ以上は行わない．プロトタイプの作成をする段階からはそれぞれ具体的な実装が始まると考えているので，その際に理論や方法などの問題に答えて行けるように，変換対象の限られない声質変換に関連する技術についての学習を続けていく．

後期に自分が担当した役割は，音声信号処理についての学習，参考にした OSS[1] のコードを読み，全体の動作を考察，1 段階目・2 段階目に使用する音声データをかさ増しするためのプログラムの作成，スライド・ポスターの作成である．

技術力向上の課題では，ディープラーニングを行うモデル構築したいと考え，環境音・自然音を Convolutional Neural Network で分類する問題に取り組んだ．ここで環境音・自然音は ESC-50[20] を利用し，コードは「ディープラーニングで音声分類」[21] を参考にした．ESC-50 は環境音を 50 クラス，2,000 ファイル集めたデータセットである．ESC-50 の音声を判別する分類器を作成した．分類器を作成する前に，データの前処理を行った．ESC-50 は 1 クラスあたりに収録されている音声が最大で 40 個ほどであり，学習を行うには，少なかった．そのため，データかさ増しを行った．かさ増し方法は 3.3.2 に記述している方法と同様である．モデルを Python で作成し，実際に学習を始める段階で，GPU 上で動かすことができないという問題が発生した．自分の MAC では学習が始まるが，GPU を利用して学習を始めようとするとエラーが発生してしまう．解決を試みたが，うまくいかず残された時間も少なくなり，モデルを構築したいという目標からは少し遠ざかっていると感じたため，やむなく中断した．しかし，音声信号処理とモデル構築に関する技術の習得は達成できた．その後は，本グループで利用していた OSS[1] のコードを読み，関数の動きやデータの受け渡し方法，モデル構造の把握に務めた．スライド・ポスターの作成においては，ディープラーニングや声質変換という言葉を聞いたことのない人々にも伝わりやすいものを作成できた．

(文責：齋藤)

第 4 章　開発した手法

4.1　開発プロセスの概要

先行事例 [1][2] で声質変換に利用していた，スペクトル包絡，メルケプストラムを理解するにあたって，いくつか必要な前程知識があった．前程知識の学習には「人工知能に関する断創録」[23]，「メルケプストラムについてのまとめ」[24]，「音楽と機械学習 前処理編 MFCC　メル周波数ケプストラム係数」[25] を参考にした．

前提として，人間の音声は，音源を声道フィルタに通した線形システムとしてモデル化できる．はじめに，音声データを扱いやすくするために，音声データをスペクトル領域に変換する，スペクトルとは単位時間あたりに，どの周波数の音が，どれほどの大きさで含まれているかを示したものである．スペクトル領域では，出力信号のスペクトルを $Y(\omega)$，入力信号のスペクトルを $S(\omega)$，声道フィルタの特性を $H(\omega)$ として，$Y(\omega) = S(\omega)H(\omega)$ と表すことができる．ここで，ω は周波数を表している．式から分かるように，スペクトル領域では入力信号のスペクトルと声道フィルタの特性が掛け合わさった数値で表されている．

そこで，声道フィルタの情報を得るために，スペクトル領域から変換して対数スペクトル領域で考える．対数をとることにより，$\log Y(\omega) = \log S(\omega) + \log H(\omega)$ と表すことができる．ここで，$\log S(\omega)$ はスペクトル微細構造であり，$\log H(\omega)$ はスペクトル包絡である．対数スペクトル領域でデータを可視化したものが図 4.1 である．滑らかな線はスペクトル包絡表しており，ギザギザの線はスペクトル微細構造を表している．

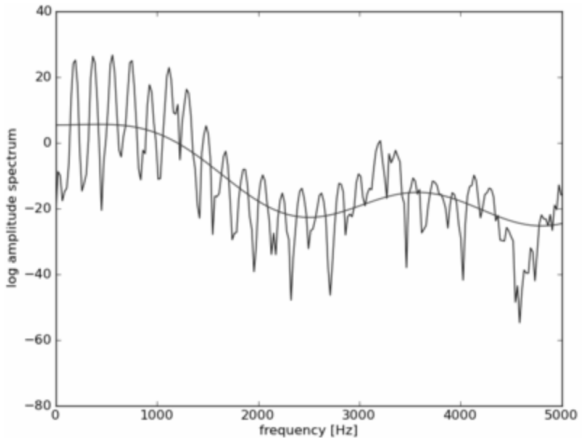

図 4.1　スペクトル包絡及びスペクトル微細構造のグラフ

声質変換に重要である音韻性や声質は，スペクトル包絡に含まれているので，スペクトル包絡とスペクトル微細構造を分離したい．そこで，対数スペクトルをさらに変換することにより，分離を行う．変換した対数スペクトルはケプストラムと呼ばれる．ケプストラムを図 4.2 で示す．

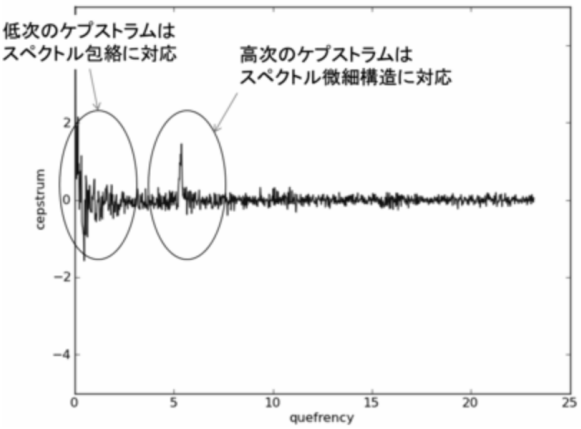

図 4.2　スペクトル包絡とスペクトル微細構造の分離

ケプストラムから，高次元部分にあるスペクトル微細構造を除去したものが，スペクトル包絡である．ケプストラムを低次元からいくつ取り出すかにより，スペクトル包絡の粗さが変わってくる．取り出す次元数が多いほど，スペクトル包絡は細かいものになる．実際にケプストラム次数を 20 と 100 にしてそれぞれ対数スペクトル領域に戻したものを図 4.3，4.4 で示す．

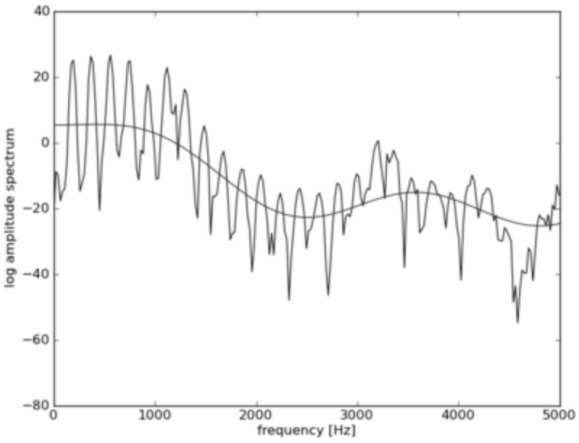

図 4.3　ケプストラム次数 20 のスペクトル

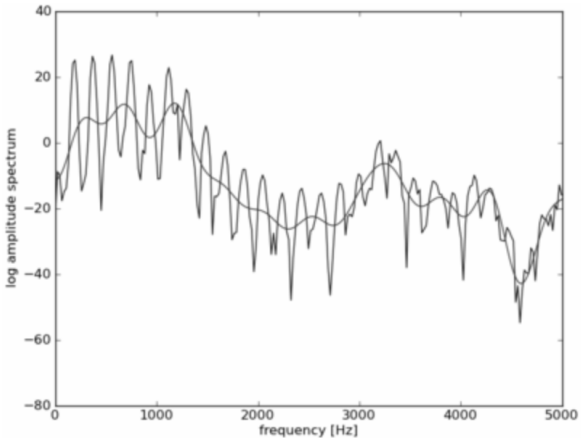

図 4.4　ケプストラム次数 100 のスペクトル

ケプストラムを等間隔でサンプリングすると，人間における低周波数の区別がつきやすく，高周波数は区別しにくいという聴覚特性が考慮されない．そのため，メルケプストラムを利用する．メルケプストラムとは，スペクトルをサンプリングする際，低次元では狭く，高次元では広くサンプリングを行って得られたケプストラムである．ケプストラムとメルケプストラムのサンプリング位置の違いを図 4.5 に示す．メルケプストラムを利用する利点としては，通常のケプストラムよりも次数が少なく済むことが挙げられる．

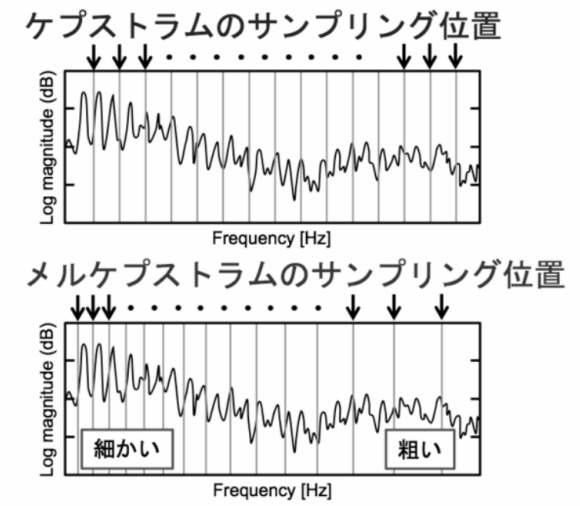

図 4.5　ケプストラムとメルケプストラムの違い

声質変換を行うシステムの開発には，参考文献 [6] で提案された手法を用いている先行事例 [1][2] を利用した．実際に，入力話者の音声データを，目標話者の音声データに変換するまでを図 4.6 に示す．

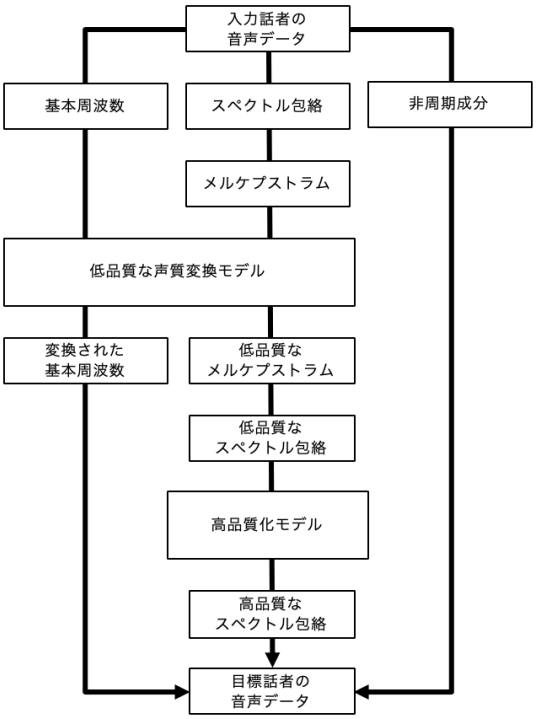

図 4.6　声質変換モデル

まず，入力話者の音声データから声質変換に必要な特徴量を抽出する．抽出する特徴量は，基本周波数，スペクトル包絡，非周期成分の 3 つである．ここで，基本周波数は，声の高さを表しており，スペクトル包絡は，音韻性や声質を表している．また，非周期成分は，声のかすれやブレスを表している．

次に，変換を行う．先行事例 [1][2] では 2 つのモデルを利用していた．1 つ目のモデルは，低品質な声質変換モデルである．このモデルでは，入力話者の基本周波数とメルケプストラムを，目標話者の基本周波数とメルケプストラムに近づけるように変換を行う．

1 つ目のモデルで変換が終わった段階では，精度の低い変換が行われている．しかし，高品質化モデルの入力はスペクトル包絡である．そのため，低品質なメルケプストラムを，低品質なスペクトル包絡に変換する必要がある．

2 つ目のモデルは，高品質化モデルである．このモデルでは，低品質なスペクトル包絡を高品質なスペクトル包絡に変換する．

最後に，変換された基本周波数，高品質なスペクトル包絡，非周期成分から，目標話者の音声データ (wav 形式) を生成する．

(文責：齋藤)

4.2 低品質な声質変換モデル

低品質な声質変換モデルでは，入力に入力話者の基本周波数及び，低次元メルケプストラムを必要とし，出力では目標話者に近づいた基本周波数及び，低次元メルケプストラムを出力する．学習時に必要なデータは，入力話者と目標話者のパラレルデータである．

このモデルでは，特徴量と時間の依存関係に注意しつつ，特徴量の過学習を避けるために 1 次元の CNN と GAN を使用した文献 [26] を参考にしている．さらに，単純な CNN では全結合層において特徴量の消失があるため，全結合層が存在しない U-Net[27] を採用する．

学習時には，3.3.2 で述べたように ATR 音素バランス 503 文の 1 文読み上げた 2.3 秒以上 10 秒未満の wav データを 001.wav から作成した．入力話者を akane_00X.wav，目標話者を hama_00X.wav として，パラレルデータをそれぞれ 447 個ずつ，合計で 894 個使用した．

wav データ 1 個あたりのサイズは，およそ 100kB から 200kB であり，全体ではおよそ 160MB である．wav データから，f0，spectrogram，aperiodicity，mcep，voiced をキーに持った dict 型の npy 配列を作成し，学習を行った．f0 には基本周波数，spectrogram はスペクトログラム，aperiodicity は非周期成分，mcep はメルケプストラム，voiced はその位置に音があるかどうかを「true」と「false」で表している．学習時に使用したのは，f0 と mcep である．

層の数，各ネットワーク層のチャンネル数，活性化関数は文献 [27] と同じ値を用いた．学習時の概要を図 4.7 に示す．

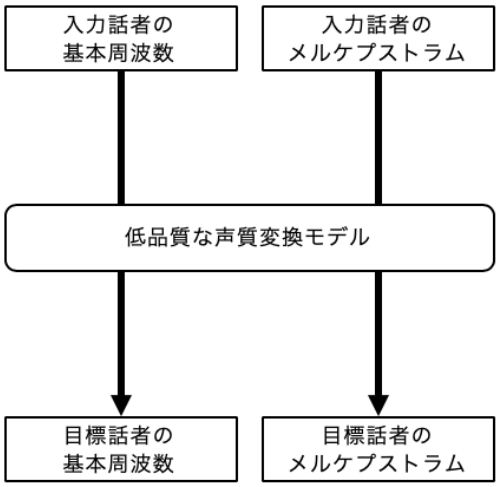

図 4.7　低品質な声質変換モデルにおける学習時の概要

(文責：齋藤)

4.3 高品質化モデル

高品質化モデルでは，入力に変換された低品質なスペクトル包絡を必要とし，出力では高品質化されたスペクトル包絡を出力する．学習時に必要なデータは，目標話者の音声データである．前処理として，目標話者の音声データから，高品質なスペクトル包絡と低品質なスペクトル包絡を作成する．高品質なスペクトル包絡とは，目標話者の音声データからそのまま取り出したスペクトル包絡のことである．また，目標話者の低品質なスペクトル包絡とは，目標話者の音声データから取り出したスペクトル包絡を，一度メルケプストラムに変換し，意図的に低品質化してスペクトル包絡に再構成したものである．

ここで，スペクトル包絡の予測誤差のみを最小化するように学習すると，推定されたスペクトル包絡が過剰に平滑化する．この現象を抑制するために，敵対的生成ネットワーク (GAN) [5] を用いている．高品質化モデルでは，GAN を利用したモデルの中でも Conditional GANs を用いており，Image-to-Image Translation with Conditional Adversarial Networks[12] を参考にして，モデルを作成し，スペクトル包絡を高品質化している．

学習時には, 3.3.2 で述べたように Web 小説などの 1 文を読み上げた 2.3 秒以上 10 秒未満の wav データを 0001.wav から作成していき，合計で，目標話者の音声データを 2632 個使用した．目標話者の wav データ 1 個から高品質なスペクトル包絡と低品質なスペクトル包絡を作成し， 学習に使用した．

wav データ 1 個あたりのサイズはおよそ 200kB から 400kB であり，全体ではおよそ 610MB である．目標話者の wav データから，low，high をキーに持った dict 型の npy 配列を作成し，学習を行った．low は低品質なスペクトル包絡，high は高品質なスペクトル包絡を表している．

層の数，各ネットワーク層のチャンネル数，活性化関数は文献 [28] と同じ値を用いた．学習時の概要を図 4.8 に示す．

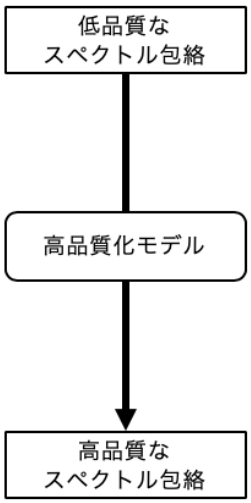

図 4.8　高品質化モデルにおける学習時の概要

(文責：齋藤)

第 5 章　発表の評価

5.1　中間発表に対する評価シートの内容と考察

2018 年 7 月 13 日, 公立はこだて未来大学でプロジェクト学習の中間発表が行われた.

中間発表では，発表後に聴講者に発表評価シートを記入してもらった．発表評価シートに記入してもらう内容は，2 つある．1 つ目は，発表技術についてである．声の大きさ，目線，ジェスチャ，が行えていたかを 1 点から 10 点の点数でつけてもらった．また，発表技術についてのコメントも記入してもらった．2 つ目が発表内容についてである．発表技術について評価してもらったのと同様に，スライドのわかり易さ，発表内容の理解を 1 点から 10 点の点数でつけてもらい，発表内容についてコメントしてもらった．中間発表で受けた評価をまとめた表を以下の表 5.1 に示す.

表 5.1　中間発表の評価

評価	発表技術	発表内容
10	7	8
9	10	7
8	18	17
7	16	15
6	4	5
5	2	3
4	3	0
3	0	2
2	0	0
1	0	0
無回答	3	5
平均点	7.7	7.6

中間発表で, 聴講者から寄せられた発表技術や発表内容についてのコメントを以下に示す.

・スライドと身体が被っていて見づらかった.
・研究の内容と目的がはっきりしていてよかったと思う.
・重要なところがわかりにくい.
・発表者の身振り手振りがより効果的になるともっといいと思う.
・ジェスチャをうまく使っていてわかりやすかった.
・プレゼンの構成内容がとてもよく考えられていてわかりやすかった.
・目標設定が明確で活動内容も理解しやすかった

表 5.1 より発表技術は，平均が 7.7 点と高めの数字であった．コメントを見てみると，スライドと身体がかぶっていたことや，ジェスチャーを効果的に使用できていないことなど発表者の動きについてのコメントが多かった．声の大きさや，聞き取りやすさについてはあまりコメントされてい

なかったのでしっかり意識して発表できていたのだと考えられる．発表内容も，平均が 7.6 点と高めの数字であった．プロジェクトの目的や活動の内容がわかりやすいというコメントもあり，発表内容は聴講者に伝わったのだと考えられる．

(文責：山田)

5.2 最終発表に対する評価シートの内容と考察

2018 年 12 月 7 日, 公立はこだて未来大学でプロジェクト学習の最終成果発表会が行われた．

最終成果発表では，中間発表と同様に発表後に聴講者に発表評価シートを記入してもらった．発表評価シートに記入してもらう内容も同様で，発表技術と発表内容についてである．発表技術と発表内容について，それぞれ 1 点から 10 点の点数とコメントをつけてもらった．最終成果発表で受けた評価をまとめた表を以下の表 5.2 に示す．

表 5.2 最終成果発表の評価

評価	発表技術	発表内容
10	6	6
9	6	3
8	13	21
7	14	9
6	3	3
5	0	0
4	1	0
3	0	0
2	0	0
1	0	0
無回答	2	3
平均点	7.9	8.0

最終成果発表で, 聴講者から寄せられた発表技術や発表内容についてのコメントを以下に示す．

・スライドがわかりやすい．目標を設定した背景をもっと知りたい．
・ディープラーニングについてもっと詳しく話してもいい．
・機械学習初心者でもわかりやすく聞けて良かった．
・図解があったため，視覚的にも目標や工夫がわかりやすかった．
・バーチャルユーチューバーなどの例があったためどんなことをしているのかわかりやすい．
・スライドで何が重要なのかがわかりにくい, フォントサイズや色での区別が欲しい．
・ニーズがあって面白いサービスだと思う．
・発表の最初にデモをやったほうがわかりやすいと思う．
・考察をしっかりしていてほしい．
・発表も聞こえやすく, スライドも非常にわかりやすかった．
・音声の変換についてもう 1 段あればさらに精度は上がる？

・棒読みなのが気になった.
・声質変換についての内容の可視化がわかりやすかった.
・下を向いていて聞き取りづらかった
・音声のデモがあってわかりやすかった.
・声が小さく誤操作があった.
・目標をしっかり設定して，達成もできていた.
・目を見ていた.
・声質変換についてはある程度いい結果が出ていたと思う.
・音声がかき消され気味で聞き取りにくい, 声を変換する過程がわかりやすい.

表 5.2 より発表技術は，平均が 7.9 点と中間発表よりも多少良い結果であった．コメントを見てみると，発表が聞こえやすいや，目を見ていたなどのコメントがあった．しかし，棒読み感が出ていたことや，下を向いて発表していると指摘されていた．これについては，もう少し本番を想定して，原稿を見ずにスライドを指しながら読むといった練習をするべきであった．また，メンバー同士で発表練習をして，確認しあっていればよりよくなったと考えられる．発表内容は，平均が 8.0 点と中間発表よりも少しではあるが良い結果であった．スライドがわかりやすいというコメントが多く見られたので，中間発表よりも要点を整理できたと考えられる．しかし、フォントサイズや文字の色を変えることによって大事なところがわかるようにしたほうが良いというコメントがあったので，スライドが見やすいかどうかを念入りに確認しておくべきだった.

(文責：山田)

第 6 章　まとめ

本グループでは，ディープラーニングを用いて，入力音声を直接変換し，合成音声を出力する変換手法を作成する．そして，音声認識とテキスト音声合成を組み合わせた手法よりも，優れた変換手法の作成を目的に活動してきた．

6.1　前期のまとめ

まず，中間目標として，中間発表までに，入力音声を直接変換して，合成音声を出力する手法を用いて簡単なモデルを構築し，プロトタイプの公開まですることを決定した．その後，グループ全体で声質変換の知識を深めるため，学習を行った．次に声質変換手法をグループ全体で収集し，どの文献を用いるかを選択した．そして，構築に必要な音声データの収集を行う音声収集班と，ネットワーク構築を行うネットワーク班に分かれた．

ネットワーク班では，グループ全体で選択した文献をもとに，より詳細な文献を探す活動を行った．その活動の中で，先行事例 [1][2] の追実験の実施を決定した．また，追実験を実施するため，プロジェクト用 PC の環境構築を行った．そして，追実験に必要な音声データを音声収集班とは別に用意した．

音声収集班は音素認識モデル構築のため，音声資源コンソーシアムから「日本音響学会 新聞記事読み上げ音声データ」を購入した．購入した音声データ量は，16679 文で約 70 時間分である．また，声質変換モデル構築のため，VOICEROID を用いて音声データの作成を行った．作成した音声データの量は，約 1000 文で約 1 時間分である．音声収集班は，データ作成および購入する音声データを選択，購入した後，ネットワーク班に合流した．

前期の活動を振り返ると，中間目標で決定した，モデルの構築とプロトタイプの公開はできなかった．理由は，技術習得において基礎学習や開発環境構築の際，出てきた問題を解決できなかったためである．したがって，後期の活動に向けて，夏季休暇中に簡易的なプロトタイプの作成を行う．

ネットワーク班では，追実験を実施する際，Python のエラーを解決することができず，先行事例 [1][2] の追実験を実施できなかった．そのため，夏季休暇中では，追実験中に学んだ知識，経験を先行事例 [7] の追実験に活用する．また，先行事例 [7] の追実験も考えていた．しかし，購入した音声データが届くのに時間がかかっており，追実験に着手できなかった．したがって，購入した音声データが届き次第，加工して追実験を実施する．以上より，ネットワーク班は，音声に関する知識を深め，別の先行事例 [1][2] の追実験を実施する環境を，整えるまでに留まった．しかし，多方面の技術や先行研究について，知見を深めることができたため，有意義な活動であったといえる．

音声収集班では，ネットワーク班が必要とする音声データの作成，購入を行った．音声データの作成，購入後は音声収集班の課題は達成されたと考え，ネットワーク班と合流し活動を共にした．

後期では，先行事例 [7] より音素認識モデルと声質変換モデルの構築を行う．また，購入した音声データが届き次第，音素認識モデルの学習で用いる音声データの加工を行う．音声データの加工が終了したのち，音素認識モデルの学習を行う．また，前期で作成した音声データを用いて，声質変換モデルの構築も行う．その後，実際に動作する入力音声を直接変換して，合成音声を出力する

手法を用いて，構築したモデルの確認をする．そして，構築したモデルの精度を向上させ，最終的にはツール化も視野に入れて活動をしていきたい．

課題として，我々の学習不足と見通しが不十分であったため，先行事例の追実験を実施することができなかった点が挙げられる．そこで，まずはグループメンバー全員の技術力を向上が必要である．その上で，夏季休暇中に先行事例 [7] の追実験を実施する．また，音素認識モデル構築のため，様々な人の音声データが必要である．そのため，購入した新聞記事読み上げ音声データが届き次第，データの加工・前処理を行う．

(文責：濱口)

6.2 後期のまとめ

夏季休暇中に, パラレルデータを必要としない Phonetic PosteriorGrams (PPGs) を用いた多対 1 の声質変換手法 [5] を利用した先行事例 [7] の追実験に取り組んでいた. しかし, チーム内の技術力不足により実装が行えないと判断したため, 活動方針の見直しを行った. 見直しの中で, 前期に追実験を試みていた, 畳込みニューラルネットワークを用いた低品質な声質変換モデルと，高品質化モデルによる声質変換手法 [6] を利用した先行事例 [1][2] があった. 先行事例 [1][2] は, 前期の追実験の際に, 用意した音声データから学習に用いるデータに変換する段階で,　音声学習に用いるデータを作成するプログラムが上手く動作しないという問題があった. しかし, 夏季休暇中にさらに追実験を試みたところ, 前期に問題があった箇所が解消できた. そこで, プロジェクトのグループ目標を「ノンパラレルデータを用いたリアルタイム声質変換」から,「パラレルデータを用いた声質変換」に変更し, 先行事例 [1][2] を利用していくことに決定した.

活動方針を決定したことにより環境を再構築する必要があったため, 一度内部の設定をすべて初期化し, 一から環境を構築し直した.

また, 前期に用意していた音声データセットも新しいモデルでは使えなかったため, 一から作り直した. その際, パラレルデータの作成に時間がかかったため, データかさ増しを行い少しでもデータ作成の負荷を減らそうと考え, 取り組んだ.

開発手法では, 先行事例 [1][2] を参考にして低品質な声質変換モデル, 高品質化モデルの実装を行った. 低品質な声質変換モデルの学習では, 入力話者と目標話者のパラレルデータからそれぞれの基本周波数・メルケプストラムを抽出し, 入力話者が目標話者に近づくように学習を行った. そして, 入力話者の基本周波数・メルケプストラムが, 目標話者の基本周波数・メルケプストラムに近づいているかどうか, 変換された基本周波数・メルケプストラムと目標話者の基本周波数・メルケプストラムで比較を行い誤差を算出した. この誤差が小さくなれば小さくなるほど, 変換の精度が向上する. 高品質化モデルの学習では, 目標話者の音声データから, 意図的に低品質にしたスペクトル包絡と高品質なスペクトル包絡を作成し, 低品質なスペクトル包絡を高品質なスペクトル包絡に変換できるように学習させた.

最終成果発表では, 実際に構築したモデルを利用し, 声質変換を行い, 変換した音声を流すことまでを行うことができた.

課題としては，動画配信分野での応用を考慮すると，声質変換にリアルタイム性を持たせることが重要であると考える．しかし，グループメンバーの技術力の不足により，最終発表までにリアルタイム機能の実装を行うことができなかった．また，学習に用いる音声データは非常に大量であり，後期の半分程度の時間を音声データの作成に費やした．そのため，後期後半での声質変換シス

テムの改善や試行錯誤に時間を割くことができなかった．プロジェクトの成功において，時間の見積もりや計画的な予定を立てることが大切であるとわかった．

通年の活動を通じ，ディープラーニングに対する知見や，音声処理に関する包括的な知識を得ることができ，非常に有意義なプロジェクト学習にすることができた．

(文責：白鳥)

参考文献

[1] Hiho, 「ディープラーニングの力で結月ゆかりの声になってみた」, 2018.
[online]https://blog.hiroshiba.jp/became-yuduki-yukari-with-deep-learning-power/ 2018 年 7 月 20 日アクセス

[2] Y. Oota,「「ディープラーニングの力で結月ゆかりの声になる」ための基礎知識とコマンド操作」, 2018.
[online]https://qiita.com/atticatticattic/items/575d71dab4ee716e4969 2018 年 7 月 20 日アクセス

[3] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, “Phonetic posteriorgrams for many-to-one voice conversion without parallel data training,”Multimedia and Expo (ICME), 2016 IEEE International Conference on, pp. 1-6, 2016.

[4] O. Watts, G. E. Henter, T. Merritt, Z. Wu and S. King, “From HMMs to DNNs: Where do the importants come from?” ,Proc. ICASSP, pp. 55055509, 2016.

[5] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad. Voice conversion using artificial neural networks. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 38933896, 2009.

[6] 廣芝 和之, 能勢 隆, 宮本 颯, 伊藤 彰則, 小田桐 優理, 「畳込みニューラルネットワークを用いた音響特徴量変換とスペクトログラム高精細化による声質変換」, 研究報告音声言語情報処理, Vol. 2018-SLP-122, No. 27, pp.1-4, 2018.

[7] さんこ, 「自分の声を水瀬いのりさんの声にする他対 1 声質変換」, 2017.
[online]http://sesenosannko.hatenablog.com/entry/inori_vc1 2018 年 7 月 20 日アクセス

[8] 池田尚史・藤倉和明・井上史彰, 「チーム開発実践入門：共同作業を円滑に行うツール・メソッド」, 技術評論社, 2014.

[9] Bill Lubanovic,「入門 Python3」斎藤康毅監訳, 長尾高弘訳, O’Reilly Japan , 2015.

[10] 伊藤克亘・花泉弘・小泉悠馬,「Python で学ぶ実践画像・音声処理入門」, コロナ社, 2018.

[11] 松尾研究室,「松尾研究室 演習コンテンツ 公開ページ」, 2018.
[online]http://weblab.t.u-tokyo.ac.jp/% 演習コンテンツ公開ページ/ 2018 年 7 月 24 日アクセス

[12] A. Kaminkure,「If you’re trying to install Tensorflow on Ubuntu 17.10」, 2018.
[online]https://medium.com/@adakaminkure/if-youre-trying-to-install-tensorflow-on-ubuntu-17-10-703b971d98d 2018 年 12 月 20 日アクセス

[13] yamachu, 「world4py」, 2017.
[online]https://github.com/yamachu/WORLD4py 2018 年 12 月 20 日アクセス

[14] Audacity
[online]https://www.audacityteam.org/ 2018 年 12 月 20 日アクセス

[15] VOICEROID+ EX 琴葉茜・葵
[online]https://www.ai-j.jp/kotonoha/ 2018 年 12 月 20 日アクセス

[16] さぼてん,「VOICEROID-YMM 連携マクロ Ver3 公開しました」, 2016.

[online]https://ch.nicovideo.jp/sabotenda/blomaga/ar1018500 2018 年 12 月 19 日アクセス

[17] UWSC
[online]https://web.archive.org/web/20180224033340/http://uwsc.info/ 2018 年 12 月 20 日アクセス

[18] 音声資源コンソーシアム,「ATR 音素バランス 503 文」, 2007.
[online]http://research.nii.ac.jp/src/ATR503.html

[19] scikit-learn
[online]https://scikit-learn.org/stable/ 2018 年 12 月 20 日アクセス

[20] F. Chollet,『Python と Keras によるディープラーニング』, 巣籠悠輔訳, マイナビ出版 , 2018.

[21] karoldvl, ESC-50, 2017.
[online]https://github.com/karoldvl/ESC-50 2018 年 1 月 9 日アクセス

[22] cvusk,「ディープラーニングで音声分類」, 2018.
[online]https://qiita.com/cvusk/items/61cdbce80785eaf28349 2018 年 1 月 9 日アクセス

[23] aidiary,「人工知能に関する断創録」, 2017.
[online]http://aidiary.hatenablog.com/entry/20120211/1328964624 2018 年 12 月 20 日アクセス

[24] tam5917,「メルケプストラムについてのまとめ」, 2017.
[online]http://tam5917.hatenablog.com/entry/2016/03/15/113555 2018 年 12 月 20 日アクセス

[25] martin-d28jp-love,「音楽と機械学習 前処理編 MFCC　メル周波数ケプストラム係数」, 2017.
[online]https://qiita.com/martin-d28jp-love/items/34161f2facb80edd999f 2018 年 12 月 20 日アクセス

[26] Kaneko, T., Kameoka, H., Hiramatsu, K. and Kashino, K.: Sequence-to-Sequence Voice Conversion with Similarity Metric Learned Using Generative Adversarial Networks, Interspeech, pp. 12831287, 2017.

[27] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A. A.: Image-to-image translation with conditional adversarial networks, CVPR, 2017.

[28] Ronneberger, O., Fischer, P. and Brox, T.: U-net: Convolutional networks for biomedical image segmentation, MICCAI, pp. 234241, 2015.