部 門

# 競技部門

No.1 登録番号

1) 予定開発期間:5ヶ月 2) 予定開発人数: 3人

No.2

|           | 4月 |  | 5月 |   |  | 6月 |  |  | 7月 |          |  | 8月 |  |  | 9月 |  | 10月 |             |  |  |
|-----------|----|--|----|---|--|----|--|--|----|----------|--|----|--|--|----|--|-----|-------------|--|--|
| 問題分析      |    |  |    | 1 |  |    |  |  |    |          |  | ٧  |  |  |    |  |     |             |  |  |
| 設計        |    |  |    |   |  | •  |  |  |    |          |  | ٧  |  |  |    |  |     |             |  |  |
| 実装        |    |  |    |   |  |    |  |  |    | <b>+</b> |  |    |  |  | ¥  |  |     |             |  |  |
| 試用・トレーニング |    |  |    |   |  |    |  |  |    | <b>+</b> |  |    |  |  |    |  |     | <b>&gt;</b> |  |  |

## 実現方法

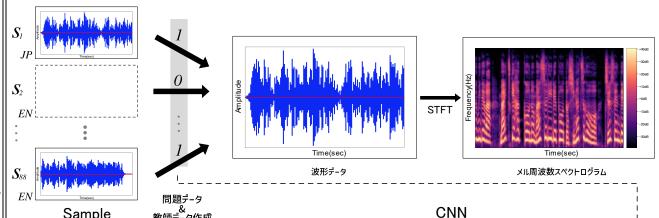
# 1) 音声の解析アルゴリズム

#### 全体の方針

機械学習を行い、合成された音声を分離する。

# ・音声データの前処理

合成された音声データ(波形データ)をSTFT(短時間フーリエ変換)を用いてメル周波数スペクトログラムに変換する。 本来必要であった時系列ごとの処理や、入力データである 48,000 \* t(s) 個のデータを、画像化することによって、 最大 縦\*横(ピクセル)のデータ数まで削減することができ、時系列ごと(リカレントニューラルネットワーク)の 処理が不要になり、CNN(畳み込みニューラルネットワーク)での処理が可能になる。



No.3

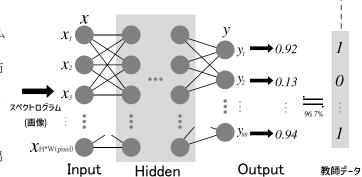
# ・機械学習の方針

Sample

問題データは単体のサンプルデータ(88種)をランダム (3~20種)で合成して作成し、分割やずらしなども表 現する。その際にどの音声が合成されたかを表す教師 データも作成する。

& 教師データ作成

スペクトログラム化された問題データを入力データと スペクトログラム して渡し、CNN で画像の機械学習を行い、どの音声が 使われたのかを表す0~1の88個の配列を出力する。 問題データを作成した際に生成された教師データと出 力データとの差で誤差をとり、学習を行う。



# ・本番(本戦)での出力

CNN の出力は、使われた音声が 1, 使われていない音声が 0 に近い値になっている 88 種類の配列である。 これを 1, 0 に近い ほうに変換し、取り札が44種類のため、日本語・英語のどちらかが1だと、その音声に対応した札を1として出力する。

#### 2) その他(独創的なところ)

・予め、どこまでダウンサンプリングを行うとスペクトログラムの品質が落ちるのかを調べておく。

これを行うことで、CNNの処理を高速化でき、かつあまり劣化していないスペクトログラムを使うことができる。

・スペクトログラムにメル周波数スペクトログラムを用いることで、人間の聞き取れる周波数をスペクトログラム化する。 これにより、音声の特徴などを見つけやすくなるのではないかと考えた。

No.4

開発言語:Python

開発環境:Anaconda, Visual Studio Code, TensorFlow, Neural Network Console, Git Hub

# 第33回 全国高等専門学校 プログラミングコンテスト: 群馬大会