2024年度 秋学期

卒　業　論　文

深層学習による触覚データとオノマトペのマルチモーダル変換

指導教員: 島田 伸敬

立命館大学 情報理工学部

卒業研究3 (DC)

コース: 実世界情報コース

学生証番号: 2600190428-0

氏名: KIM Deockjung

概 要

触覚は、人間が世界を理解し、相互作用するための重要な手段の一つである。この研究は、触覚を他の感覚に表現することで、コンピュータがより人間に近い知覚と理解を獲得することを目指している。触覚情報をオノマトペ音素などの他の形式のデータに変換する研究を行うことで、コンピュータが複数の感覚情報を統合し、より総合的な理解を可能にすることを目指す。本研究では、触覚データとオノマトペ音素をそれぞれ Sequence-to-Sequence モデルを用いて特徴ベクトルに圧縮し、両方結びつけることで、双方向のデータ変換を試みた。このアプローチにより、触覚の質感をオノマトペで表現する可能性を検討し、コンピュータによる触覚の総合的な理解の基盤を築くことを目的とする。

目 次

第 1 章　はじめに

1.1　研究背景

近年は深層学習の進展に伴い、視覚・音声・テキスト・触覚など複数のモダリティを組み合わせるマルチモーダル学習の研究が多めに行われている。Baltrusaitis らの研究 [1] では、複数のモダリティ情報を統合することで単一モダリティのみでは得られない豊富な特徴量を学習できることが示されており、画像認識や自然言語処理をはじめとする多様な分野で有望視されている。

マルチモーダルとは、テキスト・イメージ・音声・触覚など、互いに異なるモダリティ(modality) のデータを結合し活用する概念を指す。人間があらゆる感覚を通して情報を習得し、総合する過程と似た方法で、各モダリティで得た情報を相互活用することで、シングルモダリティでは獲得しにくい豊富かつ正確な結果を導けるという長所がある。例えば、映画字幕と映像を同時に分析すると、字幕だけでは理解しにくい登場人物の感情や状況を把握できるようになる。このようなマルチモーダルデータを結合する技術は近年急速に発展しており、イメージ・テキスト生成モデルや自律走行などの分野で適用事例が増加している。大規模な学習データが必要になるうえ、各モダリティ間の表現や同期化を最適化する方法については研究が活発に続けられている。最終的には、人間と類似した多面的な思考や理解を可能にするアプローチとして期待され、人工知能の技術発展に大きく寄与すると考えられている。

Takahashi らの研究 [2] では、このマルチモーダル学習の一例として、視覚情報と触覚情報を組み合わせる「Deep Visuo-Tactile Learning」に着目し、画像から触覚特性を推定する手法が提案されている。このような触覚データを対象とした研究は、ロボットが物体を操作する際の接触判断や素材認識など、実世界での応用可能性を持つ点で注目されるが、他の感覚情報と比べると研究事例はまだ十分とは言えない。

また、Ngiam らの研究 [3] では、音声・映像・テキストなど複数のモダリティを統合的に扱うマルチモーダル深層学習手法を提案しており、モダリティ間の特徴を効率的に学習することで認識精度が向上することを示している。しかし、視覚や音声と比較して触覚を取り込んだ研究は少なく、特に触覚データをテキストと結びつける試みは初期段階にあるといえる。

1.2　研究目的

本研究では、触覚データとテキスト情報、特にオノマトペを結びつけるマルチモーダル学習を通じて、機械が人間に近い感覚的・総合的な理解を獲得するための基盤を築くことを目指す。具体的には、触覚データとオノマトペ音素をそれぞれ Sequence-to-Sequence（以下 seq2seq と表記する）手法によって特徴ベクトルへ圧縮・再構成し、両者を近似させる学習手法を実験することで、触覚データからオノマトペ音素を生成する、または、オノマトペ音素から触覚データを生成する双方向のマルチモーダル変換が可能かどうかを実験する。最終的には、触覚データクラスとオノマトペ音素を対応づける手法を提示し、ロボットの操作や人間との相互作用など実世界での応用における「質感」の定量的・言語的表現に役立つことを目標とする。

1.3　本論文の構成

本論文は本章を含め 5 章で構成される。2 章で seq2seq におけるオノマトペ音素のモデリングを述べ、3 章にて seq2seq における触覚データクラスのモデリングを、4 章にて深層学習による触覚データと小野間とっぺのマルチモーダル変換について述べる。

第 2 章　seq2seq におけるオノマトペ音素のモデリング

本章では深層学習モデルである seq2seq を用いてオノマトペ音素を学習させオノマトペ音素を復元する実験を行う。それに伴う設計と学習過程を説明する。

2.1　seq2seq モデルについて

本章に入る前に、まず seq2seq 深層学習モデルについて簡単に説明する。seq2seq とは、入力シーケンスを別の出力シーケンスへ変換する深層学習モデルであり、自然言語処理や時系列データなどの幅広い分野で活用されている。seq2seq モデルは主に Encoder と Decoder の 2 つの構成要素から成る。Encoder は RNN, GRU, LSTM などのアーキテクチャを用いて入力シーケンスを特徴ベクトルに圧縮し、Decoder は Encoder が生成した特徴ベクトルをもとに出力シーケンスを生成する役割を担う。Encoder と Decoder は、ともに RNN, GRU, LSTM などを用いて実装される場合が多く、生成過程で前のシーケンス情報が次のシーケンス生成に用いられるという特徴がある。

理解を深めるため、seq2seq の簡易的なモデルを図 2.1 に示す。学習の進行に伴い、Encoder の入力と Deocder の出力との誤差を表す損失関数を最小化することでパラメータが更新される。損失関数としては、Cross Entropy Loss（交差エントロピー誤差）や平均二乗誤差（以降 MSE Loss と表記する）などが一般的に用いられる。学習に際しては、入力データと正解データの両方が必要であり、データ量が学習の成果に大きく影響する。これらの特性から、seq2seq は機械翻訳、テキスト要約、音声認識、および時系列データ予測など、多岐にわたる応用分野で利用されている。

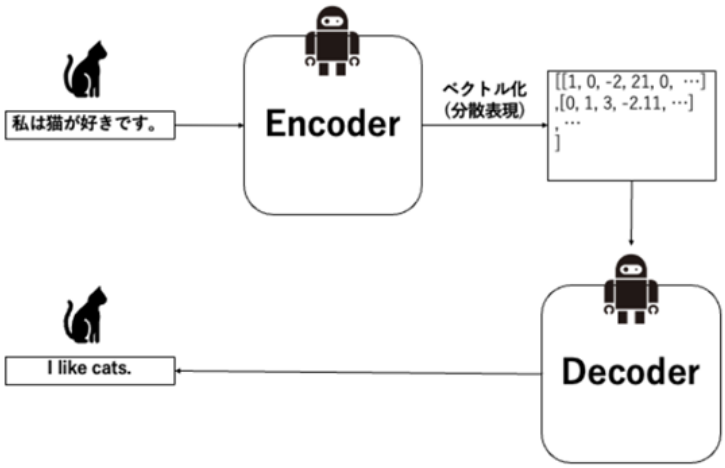


図 2.1: 図式化した seq2seq の構造 ([4] より引用)

2.2　オノマトペ音素の概要

本研究における実験の訓練に用いるオノマトペ音素のデータは、先行研究 [5][6] を基盤として作成されたものである。オノマトペ音素のパターンを表 2.1 に示す。本実験では、表 2.1 に示すような構造をもつオノマトペ音素 264 種類を対象としている。まず、使用する音素の種類として、母音（a, i, u, e, o）とその長音（a:, i:, u:, e:, o:）の計 10 種類、子音（p, b, t, d, k, g, s, z, h, n, m, y, w, r, f, j）の 16 種類、特別音素（Q, ts, sh, ch, gy, ky, ny, by, hy, py）の 10 種類、合計 36 種類を想定している。ここで、Q は日本語の促音「っ」を、N は撥音「ん」を意味する。

表 2.1: オノマトペ音素のパターン ([6] より引用)

オノマトペパタン	具体例
$CVCV$	gaba
$CVCVQ$	bataQ
$CVCVVQ$	baraaQ
$CVCVri$	batari
$CVQCVri$	baQsari
$CVCVN$	bataN
$CVQCVN$	baQtan
$CVCVVN$	bataaN
$CVQCV$	doQka
$CVNCV$	muNzu
$CVNCVri$	boNyari
$CVCVCVCVQ$	basbasaQ
$CVCVCVCVVQ$	basbasaaQ
$CVCVCVCVN$	pakapakaN
$CVCVCVCVVN$	pakapakaaN
$p_1(CVCV)p_2(CVCV)$	dotabata
$p_1(CVCVri)p_2(CVCVri)$	norarikurari
$p_1(CVCVN)p_2(CVCVN)$	gatoNgotoN
上記パタンの繰り返し	gabagaba, batabatabata, etc.

音素を切り分ける際には、分かち書きを用いて明確に分類を行う。たとえば、「ちくちく」というオノマトペの場合、パターンとして「chikuchiku」と表記できるが、「c」と「h」がそれぞれ独立した音素なのか、それとも「ch」が 1 つの音素なのかを判別しにくい。そこで、「ch i k u ch i k u」のように分かち書きを行うことで、各音素をシーケンスとして扱いやすくしている。　この実験の訓練で用いられるオノマトペ音素のデータは先行研究 [5][6] を基盤に作成されている。オノマトペ音素のパターンを表 2.1 に示す。表 2.1 のような仕組みを持っているオノマトペ音素 264 種類を用いて実験を行う。音素を確実に区切るために分かち書きを用いて音素を分類する。ちくちくと

いうオノマトペを例に挙げると、音素はパターンにより「chikuchiku」のような形になる。しかし、「chikuchiku」のような形は「c」と「h」がそれぞれ一つの音素なのか「ch」が一つの音素なのかが分かりにくい。そこで、「ch i k u ch i k u」のように分かち書きを使い各音素を区切っている。ここで、オノマトペの各音素がシーケンスである。

また、各オノマトペ音素はモデルへの入力段階で固有のトークンに変換される。音素を直接ベクトル表現に圧縮することは困難であるため、音素を数字のトークンへ変換したうえで、そのトークンをベクトルに変換するという手法を採用している。たとえば、「s a r a s a r a」という音素列を数字のトークンに置き換えると、「5 3 8 3 5 3 8 3」のようになる。これにより、モデルは入力された音素列を扱いやすい数値表現として処理可能となる。

2.3 モデルの構造

オノマトペ音素を復元するための seq2seq 構造を図 2.2 に示す。本研究で用いる seq2seq モデルは、先行研究 [5] が提案するオノマトペ音素復元モデルを基盤として構築したものである。前述の通り、seq2seq モデルの学習には入力データと正解データの双方が必要となる。本実験では、入力と正解の双方に同一のオノマトペ音素を用いることで、入力時のオノマトペ音素がどの程度正確に復元されるかを学習成果の指標とした。

本モデルを用いた学習は 250,000 イテレーション（試行回数）で行い、損失関数には Cross Entropy Loss を採用した。また、特徴ベクトルの次元数は 256 次元に設定している。これらの設定により、seq2seq モデルによるオノマトペ音素の復元精度を評価する。

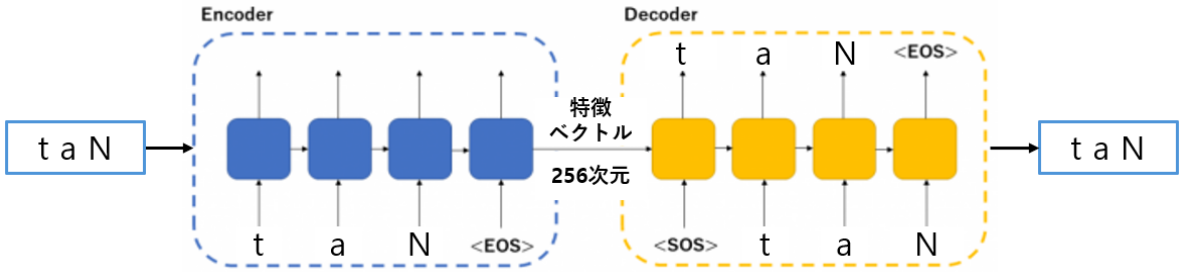


図 2.2: オノマトペ音素の seq2seq 復元モデル ([7] より引用)

2.3.1 全体の流れ

オノマトペ音素を Encoder へ入力し、特徴ベクトルに圧縮した後、Decoder を通じてオノマトペ音素を復元するという流れが、本研究における seq2seq モデルの基本的な処理手順である。ここで、Cross Entropy Loss を用いることにより、正解シーケンスと予測シーケンスの間にどの程度の差異が生じているかを定量的に評価できる。

オノマトペ音素が正確に復元されているかを確認する最も直接的な方法は、入力と出力のオノマトペ音素を視覚的に比較することである。しかし、学習の成否をより効率的に把握するためには、各イテレーションごとに損失値（Loss）を出力し、その推移を観察する方法が有効である。損失値の減少傾向を追うことで、モデルの学習が適切に進んでいるかどうかを評価可能となる。

Encoder では、先行する音素情報を考慮しながら、次の音素の情報を順次特徴ベクトルに反映させ、オノマトペ音素全体を圧縮していく。たとえば、「ｓａｒａｓａｒａ」というオノマトペ音素列は 8 つの音素（シーケンス）で構成されており、最初に「s」の音素を基に初期特徴ベクトルを生成し、次にこの特徴ベクトルと次の音素「a」を総合して、新たな特徴ベクトルを更新していく。

Decoder では、Encoder が最終的に出力した特徴ベクトルを入力として受け取り、まず「s」の音素を復元する。その後、直前に復元した音素と特徴ベクトルを合わせて参照し、次の音素を予測するという操作を順次繰り返す。これらの処理が完了すると、入力音素列と復元された音素列の誤差が計算され、1 イテレーションが終了する。本研究では、このプロセスを 250,000 イテレーションにわたって繰り返すことで、復元精度を高めることを目指している。

ただし、上記の仕組みだけでは、オノマトペ音素列の長さに関する情報をモデルに与えられないため、たとえば「ｓａｒａｓａｒａ」を入力した場合、出力が「ｓａｒａｓａｒａｓａｒａ…」といった無限長に近い形で生成されてしまう問題が起こり得る。そこで、本研究では＜ SOS ＞トークンと＜ EOS ＞トークンを導入している。＜ SOS ＞トークンは復元の開始を示すもので数値 ID 0 を、＜ EOS ＞トークンは復元の終了を示すもので数値 ID 1 をそれぞれ担っている。これにより、Decoder は復元を始めるタイミングと終了するタイミングを明示的に把握できるため、オノマトペ音素が無限に生成される事態を防ぐことが可能となる。

2.3.2 損失関数

Cross Entropy Loss は確率分布 p とモデルが予測した確率分布 q 間の類似度を測定する尺度である。p は実際の正解分布であり、q はモデルが予測した分布である。分類問題で使用される損失関数であり、特に言語モデリングまたはシーケンスモデル基盤の単語予測モデルで多めに使用されている。

一般的な Cross Entropy Loss の定義を式 2.1 に示す。ここで、*p(i)* は正解のレイブル i に対する確率であり、*q(i)* はモデルが復元のレイブル i に対する確率、または、推定値を意味する。分類問題で正解のレイブルはワンホットベクトル (one-hot vector) で表現され、復元のレイブルはモデルの出力分布 (ソフトマックスの結果) で示される。正解のレイブルのベクトル p で正解クラス j だけが 1 で他は 0 とすると式 2.2 のようになるので、実際の計算時にはー *logq(j)* へ簡単化される。

$$H(p,q) = -\sum_{i} p(i) \log q(i) \tag{2.1}$$

$$p(i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

つまり、正解クラスを y、モデルが復元した確率分布を $p_{\hat{y}}$ とすると式 2.3 のような式になる。

$$\mathcal{L}_{CE}(y, p_{\hat{y}}) = -\log\Big(p_{\hat{y}}[y]\Big), \tag{2.3}$$

この実験で Cross Entropy Loss は内部的にログソフトマックス (Log Softmax) と NLL(Negative Log-Likelihood) Loss が融合された形である。Decoder の結果生成された結果は各音素のログ確率分布 $p_{\hat{y}_t}$ でもある。また、この実験で正解のレイブル y_t は元のオノマトペ音素であるので、式 2.4 のように表現できる。つまり、この実験で Cross Entropy Loss はオノマトペ音素をシーケンスとして予測する際に使われるクラス分類損失であるのだ。

$$\mathcal{L}_{\text{ono}} = -\sum_{t=1}^{T} \log\left(p_{\hat{y}_t}[y_t]\right) \tag{2.4}$$

2.4　オノマトペ音素復元の結果

オノマトペ音素を対象とした seq2seq モデルの学習結果を図 2.3 に示す。本実験では、500 イテレーションを 1 エポックとし、これを 500 エポック反復している。学習の結果、損失値は約 2.42 から約 0.001 まで減少しており、モデルがオノマトペ音素をより正確に復元できるようになったことが示唆される。

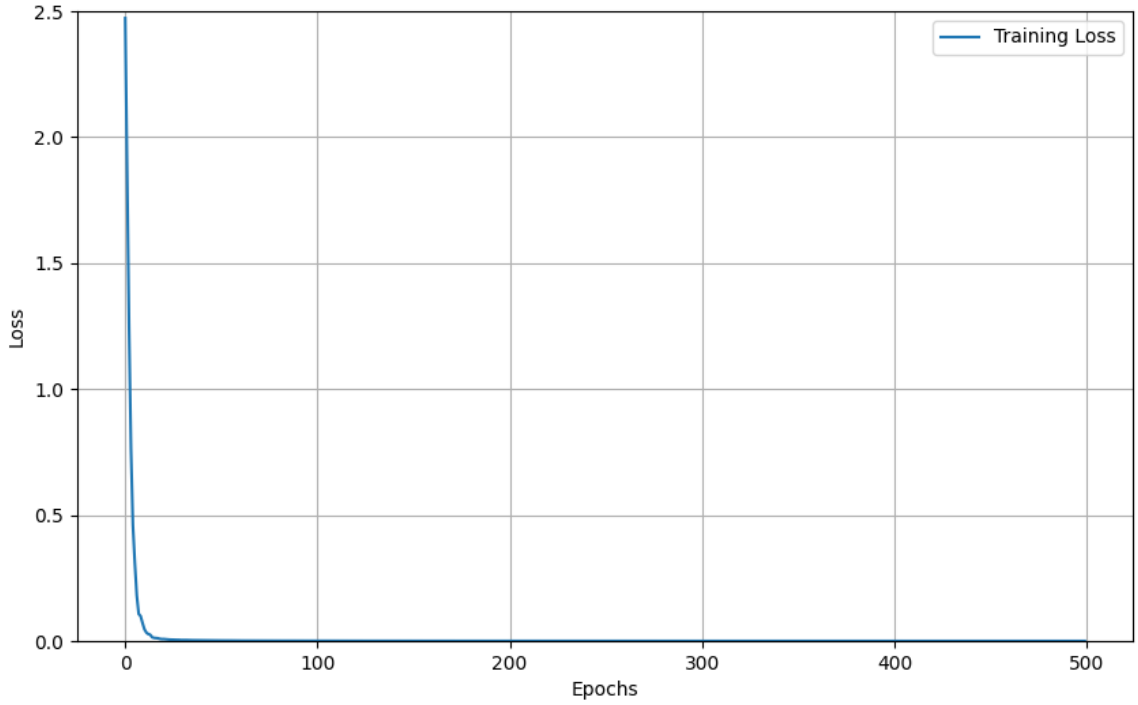


図 2.3: オノマトペ音素の復元学習のエポックによる誤差の変化

入力音素列に対する復元結果の例を、30 件抽出して表 2.2 に示す。表 2.2 からは、「b a k i b a k i」, 「k i N k i N」, 「g o ts u g o ts u」のように、入力と復元結果が概ね一致しているケースが多数確認される。一方で、すべての復元結果が完全に一致するわけではなく、たとえば「y o b o y o b o」を入力した場合には「b o y b o r o」のように一部のみ誤った復元結果が得られる例もみられた。

本研究では、オノマトペ音素がどの程度正確に復元されたかを示す指標として、式 (2.5) に示す正解率を用いる。誤差が約 0.001 に収束したモデルを用いて、訓練に用いた 264 種類の入力音素列と復元結果を比較したところ、259 件が正しく復元され、正解率は約 98.1%に達した。

$$\text{正解率 (Accuracy)} = \frac{\text{正しく復元されたオノマトペ音素の数}}{\text{オノマトペ音素の総数}} \tag{2.5}$$

表 2.2: オノマトペ音素の復元結果

入力音素	復元結果	入力音素	復元結果
b a k i b a k i	b a k i b a k i	y o r o y o r o	hy o r o y p o r
s a r a s a r a	s a r a s a r a	g u ch a g u ch a	g u ch a g u ch a
k i N k i N	k i N k i N	b o k o b o k o	b o k o b o k o
z o k u z o k u	z o k u z o k u	k e r o k e r o	k e r o k e r o
k u s u k u s u	k u s u k u s u	ch o k i ch o k i	ch o k i ch o k i
g o ts u g o ts u	g o ts u g o ts u	k o r o k o r o	k o r o k o r o
f u r a f u r a	f u r a f u r a	ts u: N	ts u: N
y o b o y o b o	b o y b o r o	b i sh o b i sh o	b i sh o b i sh o
g u b i g u b i	g u b i g u b i	i g a i g a	i g a i g a
y o r e y o r e	s o r e y p e r	p u N p u N	u N u p a r u
g o k u g o k u	g o k u g o k u	k a r i k a r i	k a r i k a r i
g i z a g i z a	g i z a g i z a	w a k u w a k u	w a k u w a k u
b u k u b u k u	b u k u b u k u	k a N k a N	k u N a k i
g i s u g i s u	g i s u g i s u	g i r i g i r i	g i r i g i r i
s u r a s u r a	s u r a s u r a	b a k u b a k u	b a k u b a k u

第 3 章　seq2seq における触覚データクラスのモデリング

本章では seq2seq 深層学習モデルを用いて触覚データの復元を学習するためにそのモデルの設計と学習過程を説明する。

3.1　触覚データ

本章でモデルの詳細を述べるに先立ち、本研究で学習に使用する触覚データクラスについて紹介する。本実験に用いた触覚データクラスは野間研より提供されたもので、Fx（外力）, Fy, Fz, Tx（回転力）, Ty, Tz の 6 次元で構成される数値データである。また、2,000 フレームが約 1 秒に相当する。図 3.1 に触覚データクラスの実例を示した。図 3.1 の赤色の四角で示した部分が実際に触覚データが測定された範囲であり、その拡大図を図 3.2 に示している。本研究では、このようなデータを用いてコルク、デニム、ガーゼ、金網、光沢（ガラス）の 5 つのデータクラスを対象とした実験を行った。

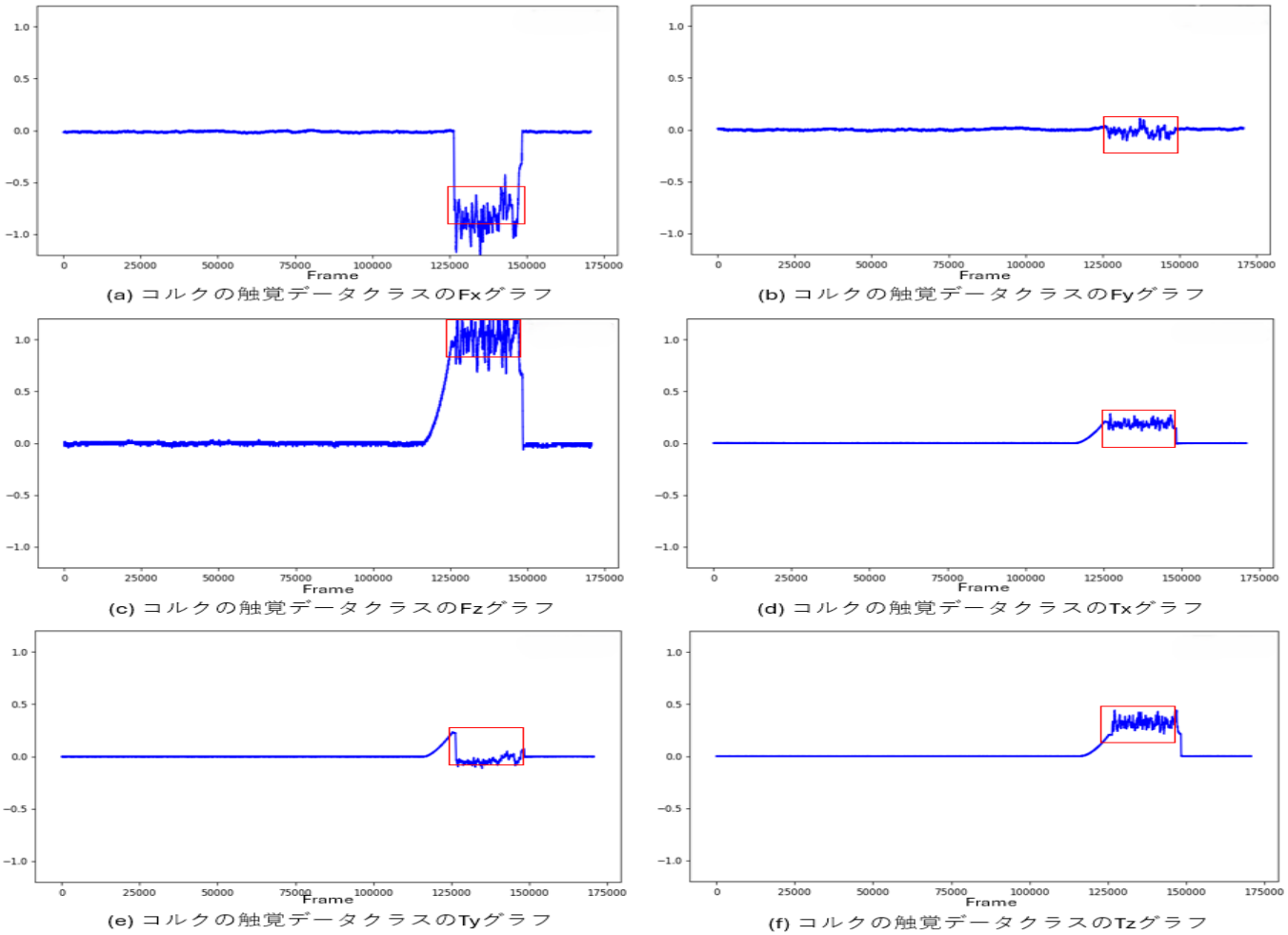


図 3.1: コルクの触覚データクラス

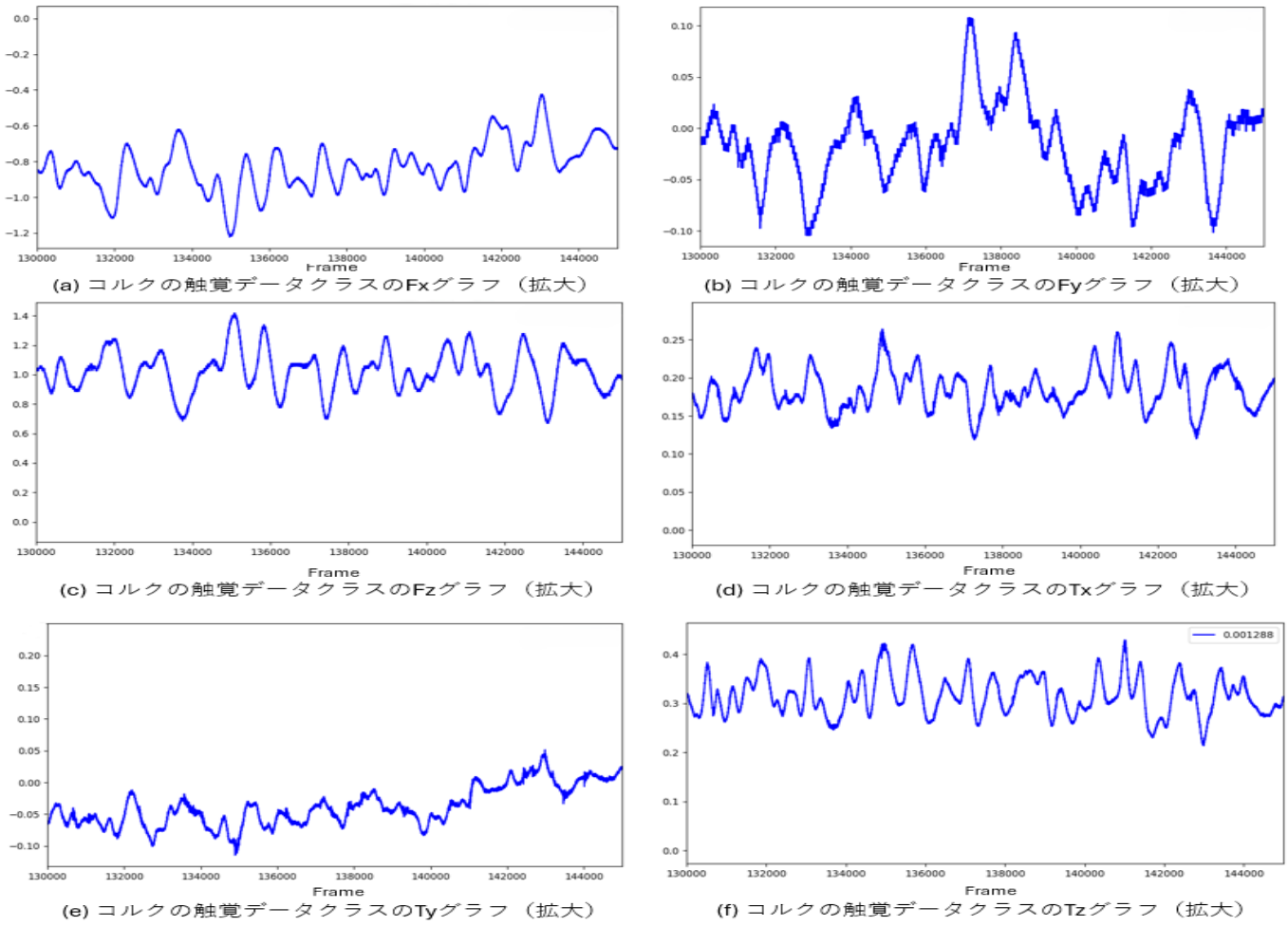


図 3.2: コルクの触覚データクラス (拡大)

3.1.1 前処理

本研究では、触覚データクラスを入力データとして利用するにあたり、適切な前処理を施す必要がある。具体的には、2,000 フレームが約 1 秒に相当するという特性を踏まえ、その半分である 1,000 フレームを 1 サンプルとして扱うようにデータを変換する。元のデータは 6 次元の情報が約 30,000 フレーム連続しているが、この前処理により 6 次元× 1,000 フレーム（計 6,000 次元）から成るサンプル列を生成する。

このような前処理を行う理由は大きく 2 点に分けられる。第一に、人間の触覚は触れた瞬間に「ふわふわ」や「ざらざら」などの感覚を認識すると考えられるため、比較的短いフレームのデータでも十分にどのような触感かを判別できる必要がある。第二に、たとえば 100 フレーム（0.05 秒相当）を 1 サンプルとした場合、1 サンプルあたり 6 次元× 100 フレーム＝ 600 次元となり、後述する実験における seq2seq モデルの特徴ベクトルを 256 次元に設定していることを踏まえると、モデルの性能を十分に評価しづらくなるおそれがある。そのため、本研究では 1,000 フレーム（約 0.5 秒）を 1 サンプルとし、6,000 次元の触覚データクラスを用いて実験を実施することとした。

3.2 モデルの構造

図 3.3 に、触覚データクラスを復元するための seq2seq モデルの構造を示す。本研究では、このモデルを用いて触覚データクラスを Encoder と Decoder に入力し、元のデータクラスへ復元する実験を行う。具体的には、入力データクラスと復元されたデータクラスとの誤差を算出し、この誤差を減少させる過程によってモデルのパラメータを学習させる。

本実験では、1,000 エポック（試行回数）にわたって学習を行い、損失関数としては MSE（Mean Squared Error）を採用した。さらに、特徴ベクトルの次元数は 256 に設定した。

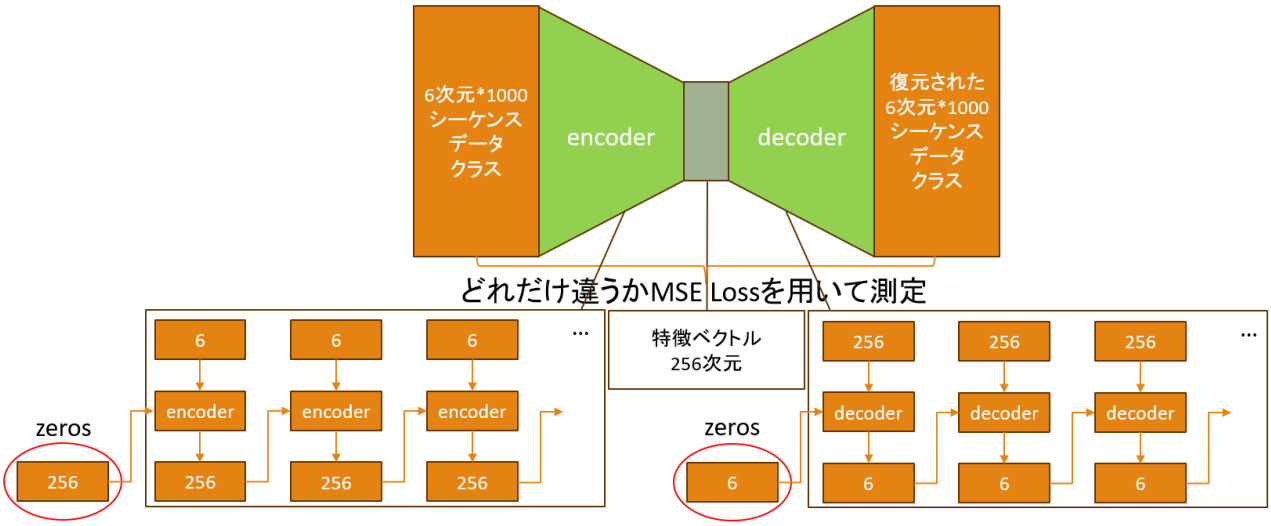


図 3.3: 触覚データクラスの seq2seq 復元モデル

3.2.1　全体の流れ

図 3.3.1 に示すように、触覚データ復元モデルの Encoder では、前処理済みの触覚データクラスを入力として受け取る。Encoder はまず、触覚データクラスの 1 フレイムに相当する 6 次元のデータを処理し、それを 256 次元の特徴ベクトルへと圧縮する。この際に得られる特徴ベクトルは、第 1 フレイムの 6 次元データに関する情報を含んでいる。続いて、次フレイム（第 2 フレイム）の 6 次元データと先の特徴ベクトルを統合して新たな特徴ベクトルを生成し、これを繰り返すことで、最終的に 1,000 フレイム分（6,000 次元）の触覚データクラスの情報が含まれた特徴ベクトルが得られる。

Encoder では毎フレイムに対して 6 次元データクラスと前の特徴ベクトルを入力とするが、第 1 フレイム目では特徴ベクトルが存在しない。そこで、zeros メソッドを用いて仮想的な特徴ベクトルを生成し、初期入力として与える。

一方、図 3.3.2 に示す Decoder は、Encoder が最終的に出力した特徴ベクトルを用いて触覚データクラスを復元する。Decoder も同様に、256 次元の特徴ベクトルと 6 次元データクラスを入力とするが、Encoder とは逆に、第 1 フレイム目の 6 次元データクラスは存在しない。そのため、zeros メソッドにより仮想的な 6 次元データクラスを生成し、Encoder から出力された最終特徴ベクトルとともに Decoder へ入力する。これにより、第 1 フレイム目の 6 次元データクラスが復元されると、今度はその第 1 フレイム目のデータクラスと最終特徴ベクトルを入力として第 2 フレイム目のデータクラスを生成する、という手続きを 1,000 フレイム分繰り返す。

Encoder と Decoder の処理が終了すると、得られた復元済みの触覚データクラスと元の触覚データクラスとの誤差を算出し、1 エポックが完了する。これを繰り返すことで、seq2seq モデルは触覚データクラスの復元性能を徐々に向上させることが期待される。

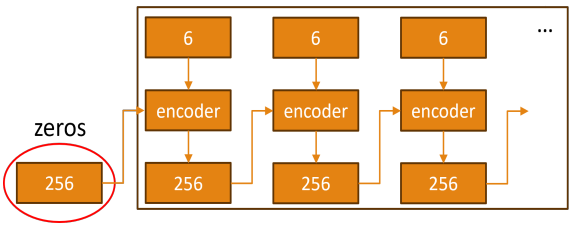


図 3.3.1: 触覚データ復元モデルの Encoder の構造

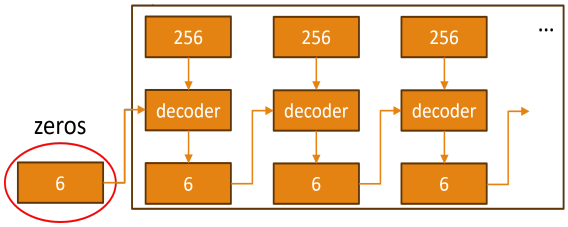


図 3.3.2: 触覚データ復元モデルの Decoder の構造

3.2.2 損失関数

触覚データクラス復元モデルでは MSE Loss を用いてその差を求めている。MSE Loss は予測値と原本データクラス間の平方誤差の平均を計算する代表的な回帰損失関数である。MSE Loss を式 3.1 に示す。予測結果が入力に近いほど誤差は減少し、遠いほど誤差は増加する。N はミニバッチ数、n は次元数を意味する。$Y_{i,j}$ は入力シーケンスに該当する値、$\hat{Y}_{i,j}$ はモデルの予測値を意味する。

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N \times n} \sum_{j=1}^{N} \sum_{i=1}^{n} \left| \hat{Y}_{i,j} - Y_{i,j} \right|^2 \tag{3.1}$$

MSE 損失関数の特徴は連続的な値を予測する回帰問題で使用される。また、誤差を平方することによりマイナスとプラス誤差を同様に扱い、誤差が大きいほど損失が急激に増加する。この特徴で学習過程の大きい誤差を速めに減少することが可能になる。誤差を足した後、データの個数で平均を取るので、データ数に変化があっても損失のスケイルを一定に維持することができる。

3.3 触覚データ復元の結果

学習のエポック (Epoch) による誤差の変化を図 3.4 に示す。図 3.4 を見るとエポックが増加することにより誤差が減少していくことが分かる。このような誤差は約 78 から 0.05 まで減少した。

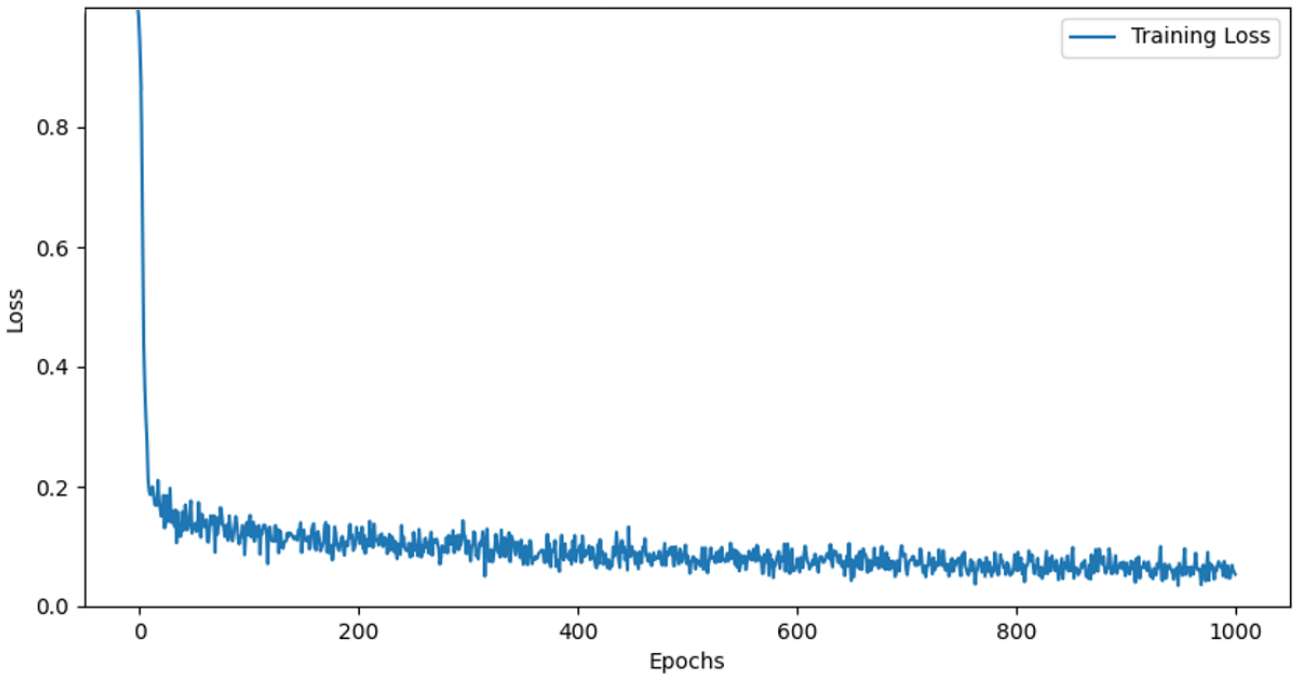


図 3.4: 触覚データクラスの復元学習のエポックによるの誤差の変化

図 3.5 に、入力された触覚データクラスと seq2seq モデルによる復元結果を比較したグラフを示す。さらに、図 3.5 のグラフを拡大したものを図 3.6 に示した。青色の線は入力された触覚データクラスを、黄色の線はモデルが復元した触覚データクラスをそれぞれ示している。

図 3.6 の上段 2 つのグラフでは、入力と復元結果がほぼ一致しているように見受けられる。一方、下段のグラフは復元が不十分に見えるものの、直流成分の振幅を比較すると、1 番目が 0.15、2 番目が 0.46、3 番目が 0.06 となっている。上段のグラフにおいても、さらに拡大して観察すると同程度の誤差が含まれていることがわかるため、下段のグラフにおける復元も一定の精度が確保されていると考えられる。

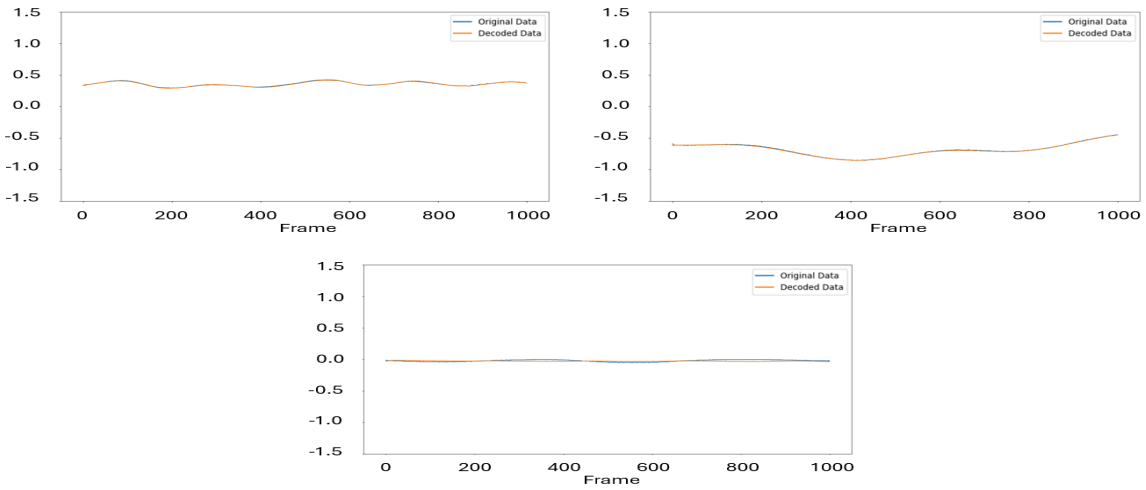


図 3.5: 触覚データクラスの原本と復元値の比較

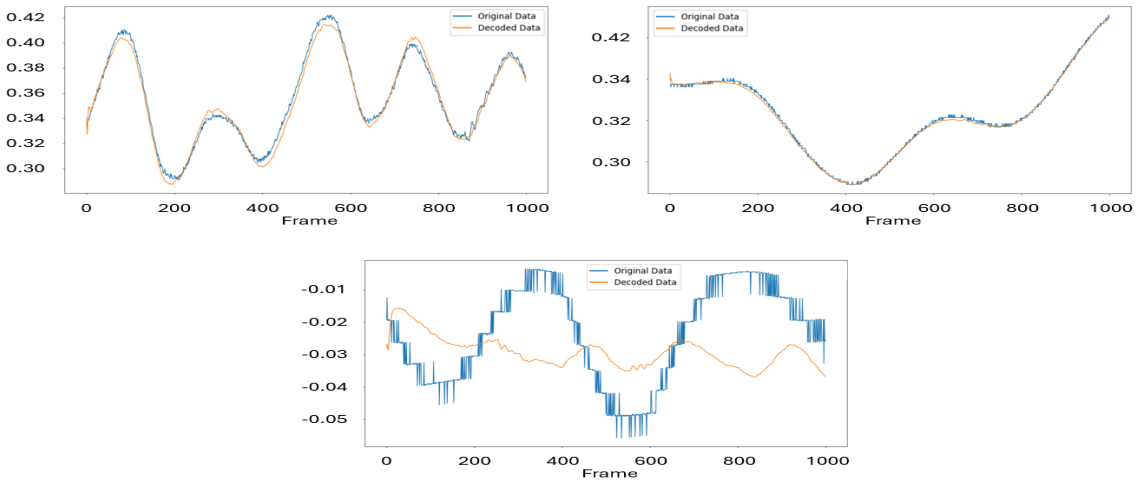


図 3.6: 触覚データクラスの原本と復元値の比較 (拡大)

第 4 章　深層学習による触覚データとオノマトペのマルチモーダル変換

本章では 2,3 章で紹介した二つの seq2seq を結合し、触覚データとオノマトペのマルチモーダル変換を説明する。また、マルチモーダル変換の結果を分析し、不足や問題点を考察する。

4.1　マルチモーダル変換

4.1.1　モデルの構造

図 4.1 に、触覚データとオノマトペを相互に変換するマルチモーダル変換モデルの構造を示す。本モデルは、第 2 章および第 3 章で述べた触覚データクラスとオノマトペ音素の単独復元手法を基盤として構築されている。具体的には、触覚データクラスとオノマトペ音素の特徴ベクトルが互いに近似するよう MSE（Mean Squared Error）Loss を用いた学習を追加することで、図 4.1 の赤色の矢印が示すように、触覚の Encoder に触覚データクラスを入力するとオノマトペの Decoder 経由でオノマトペ音素を生成し、逆に青色の矢印のようにオノマトペ音素をオノマトペの Encoder へ入力すると触覚の Decoder 経由で触覚データクラスを生成できる構造になっている。

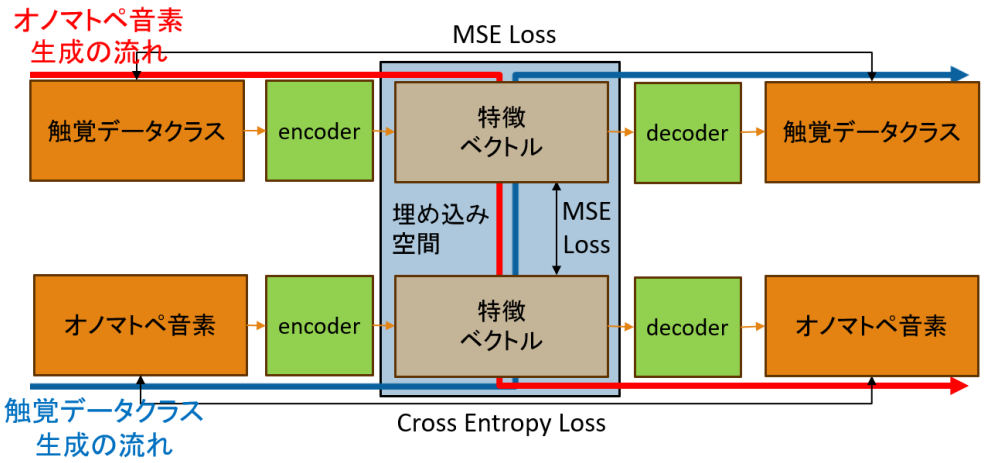


図 4.1: 触覚データとオノマトペのマルチモーダル

モデルの動作手順を以下に説明する。まず、触覚データクラスを対象とした seq2seq による復元処理を実行し、入力データクラスと復元結果との誤差を MSE Loss として保存する。次に、オノマトペ音素に対して 500 イテレーションの学習を行い、その最終的な Cross Entropy Loss を保存する。この際、オノマトペ音素の seq2seq モデルは学習を行うが、触覚データクラス側は誤差の保存のみを行う点が異なる。続いて、両データ（触覚データクラスおよびオノマトペ音素）をそれぞれ対応する Encoder に入力し、生成された特徴ベクトル同士を埋め込み空間上で MSE Loss を用いて比較し、その誤差を求める。すなわち、触覚データクラスの MSE Loss、オノマトペ音素の Cross Entropy Loss、および両特徴ベクトル間の MSE Loss という 3 つの誤差を、それぞれ重み付けしたうえで総和し、この総和の勾配を逆伝播させることで学習を進める。複数の目標（触覚データクラスの復元、オノマトペ音素の復元、両者の埋め込み空間での近似）を同時に満たすため、これら 3 つの損失を一回のエポック内で総合的に低減させる必要があるからである。

本実験では、触覚データクラスに関しては、1 エポックごとにデータセット全体を対象に seq2seq 復元を行い、その復元誤差（MSE）を算出しつつ学習を進める。これを 1,000 エポックにわたって繰り返し、触覚データクラス復元にかかわる MSE Loss を継続的に更新する。一方で、オノマトペ音素については、1 エポックあたり 500 イテレーションを行い、Cross Entropy Loss を最小化するよう学習し、同様に 1,000 エポック継続する。結果として、オノマトペ音素に対する総学習回数は、500 イテレーション× 1,000 エポック = 500,000 イテレーションに達する。触覚データクラスとオノマトペ音素を並行して扱い、それぞれのエポックまたはイテレーションごとに 3 つの損失を同時に低減することで、触覚情報とオノマトペ情報が埋め込み空間上で適切に対応付けられるよう学習を行った。

4.1.2 損失関数

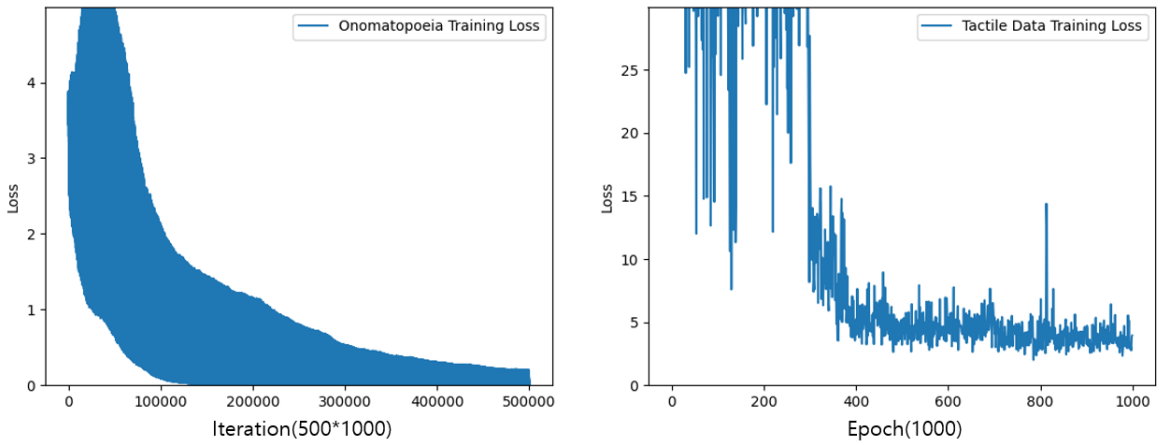
本モデルでは、触覚データクラスの再現誤差を低減するための MSE Loss、オノマトペ音素の再現誤差を低減するための Cross Entropy Loss、そして特徴ベクトル同士を近似させるための MSE Loss という 3 つの損失関数を用いて学習を行う。さらに、これら 3 つの損失関数の重み付け和の勾配を逆伝播するため、式 (4.1) に示すような Loss 関数を採用している。式 (4.1) 中の α, β, γ それぞれ触覚データクラス用 MSE Loss、オノマトペ音素用 Cross Entropy Loss、特徴ベクトル用 MSE Loss に対する重みを表す。本実験では、$\alpha = 0.01$, $\beta = 0.2$, $\gamma = 1$ と設定して学習を行った。

$$\mathcal{L} = \mathcal{L}_{\mathrm{sk}} \cdot \alpha + \mathcal{L}_{\mathrm{ono}} \cdot \beta + \mathcal{L}_{\mathrm{cv}} \cdot \gamma \tag{4.1}$$

4.1.3　モーダル変換の結果

図 4.2 に、エポックおよびイテレーション数に応じた各損失の変化を示す。ここで、図 4.2(a) はオノマトペ音素の復元 Loss（最小値約 0.2）、(b) は触覚データクラスの復元 Loss（最小値約 3）、(c) は触覚データクラスとオノマトペ音素の特徴ベクトル間の Loss（最小値 0.01）、(d) は (a), (b), (c) にそれぞれの重みをかけて合計した総合 Loss（最小値約 0.08）を表している。

まず、図 4.2(a) に示すオノマトペ音素の復元 Loss は、第 2 章で述べた結果と同様、イテレーションの進行に伴い誤差が減少する傾向が見られる。一方、図 4.2(b) に示す触覚データクラスの復元 Loss は、第 3 章の結果と比較すると最小値がおよそ 3 まで高くなっている。図 4.2(c) の特徴ベクトル間の Loss は、学習開始から約 50 エポックまでは急激に減少したものの、その後は振動を含む形でゆるやかに減少し続ける傾向が確認できる。最後に、図 4.2(d) の総合 Loss は、エポックを重ねるにつれて徐々に低下しており、学習の進行とともにモデル全体の損失が総合的に改善していることを示している。



(a) エポックによるオノマトペ音素のCross Entropy Lossの変化
(b) エポックによる触覚データクラスのMSE Lossの変化
(c) エポックによる特徴ベクトルを近付けるMSE Lossの変化
(d) 総合Loss

図 4.2: マルチモーダル変換学習のエポックやイテレーションによる誤差の変化

触覚データクラスとオノマトペ音素の特徴ベクトルを主成分分析し、プロットした結果を図 4.3 に示す。ここで、コルクは緑、デニムは黄、ガーゼは赤、金網は青、光沢 (ガラス) は黒の色で表している。丸形が触覚データクラスの特徴ベクトルを意味し、星形がオノマトペ音素の特徴ベクトルを意味する。対応付けしたオノマトペ音素の特徴ベクトルが触覚データクラスの特徴ベクトルに遮られ、矢印がそのオノマトペ音素の特徴ベクトルを表す。

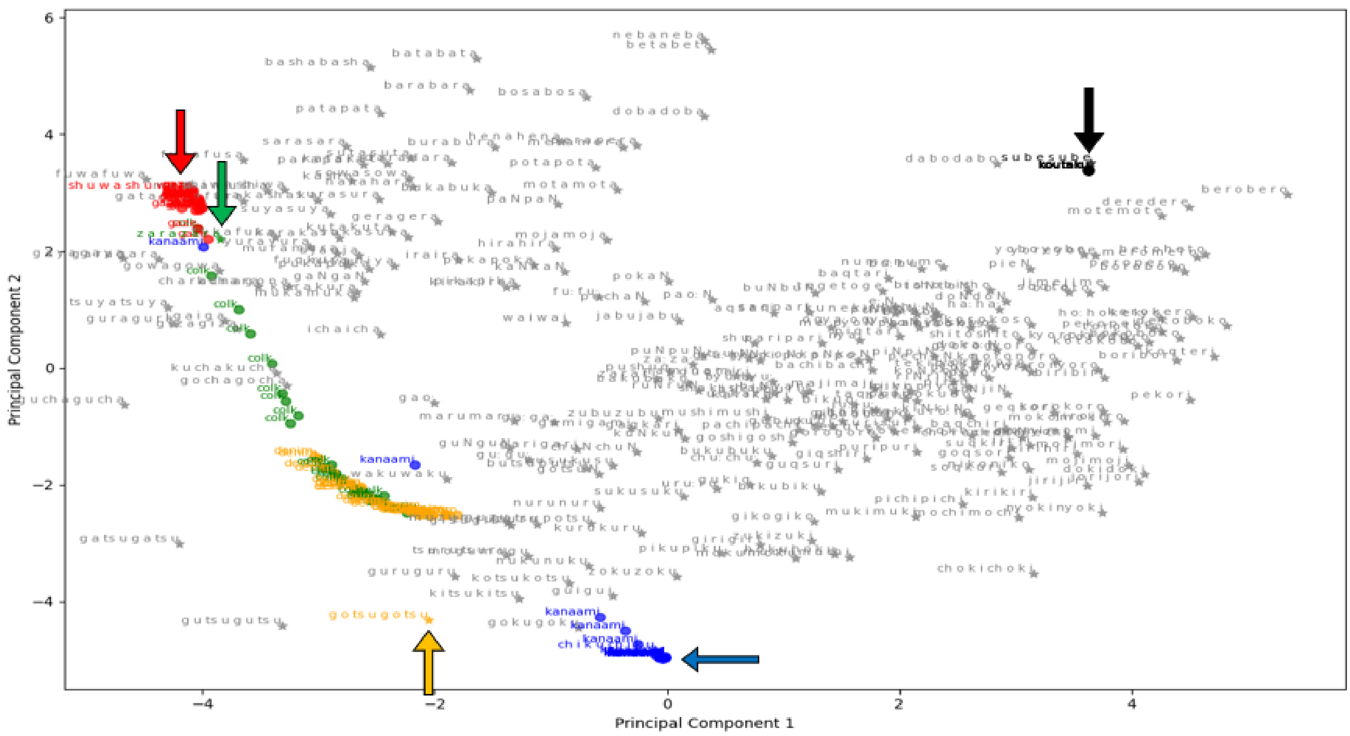


図 4.3: 触覚データクラスとオノマトペ音素の特徴ベクトルのプロット図

触覚データクラスの入力によるオノマトペ音素の生成結果、正解率 (式 4.2) とその具体例を表 4.1 に示す。図 4.3 に示したように触覚データクラスとオノマトペ音素の特徴ベクトルが一番近いオノマトペ音素を生成する。図 4.3 で黒の光沢はほぼ同じ一点に集まり光沢の触覚データクラスを入力すると光沢のオノマトペを意味する「ｓｕｂｅｓｕｂｅ」が生成されることが分かる。実際に表 4.1 を見ると光沢の触覚データクラスを入力した場合「ｓｕｂｅｓｕｂｅ」が復元される確率がサンプル数 30 個のうち 30 個が復元され 100%に当たる結果が出た。一方、緑のコルクや黄のデニムは混ぜており互いに生成結果が重なる結果を表した。表 4.1 を見るとコルクの復元結果がデニムの生成結果である「ｇｏ ts ｕ ｇ ｏ ts ｕ」が 8 回も生成されたことが分かる。また、デニムの触覚データクラスの特徴ベクトルもかなり散らばっており正解率がサンプル数 30 個のうち 18 個である 60%に達していた。表 4.1 で具体例を見ると正解と音素一つが間違って正解でない場合もあり、完全に違う復元結果を示した場合もあった。

$$\text{正解率 (Accuracy)} = \frac{\text{正しく復元されたオノマトペ音素の数}}{\text{訓練に用いられた触覚データクラスのサンプル数}} \tag{4.2}$$

表 4.1: 触覚データクラスの入力によるオノマトペ音素の生成結果

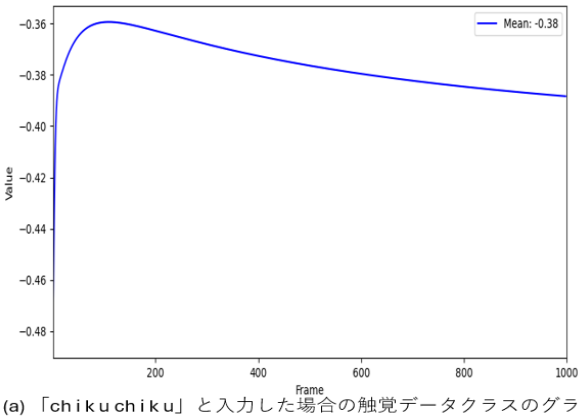
case	正解率	正解
コルクの触覚データクラスを入力した場合	3.33%	ｚａｒａｚａｒａ
デニムの触覚データクラスを入力した場合	60%	ｇｏ ts ｕ ｇ ｏ ts ｕ
ガーゼの触覚データクラスを入力した場合	93.33%	ｓｈ ｕ ｗ ａ ｓｈ ｕ ｗ ａ
金網の触覚データクラスを入力した場合	93.33%	ｃｈ ｉ ｋ ｕ ｃｈ ｉ ｋ ｕ
光沢（ガラス）の触覚データクラスを入力した場合	100%	ｓｕｂｅｓｕｂｅ

入力	出力	数	入力	出力	数
コルクの触覚データクラス	ｚａｒａｚａｒａ	1	ガーゼの触覚データクラス	ｓｈ ｕ ｗ ａ ｓｈ ｕ ｗ ａ	28
	ｇｏ ts ｕ ｇ ｏ ts ｕ	8		ｚａｒａｇｕｗａ	1
	ｇｏ ts ｕ ｇ ａ ｒ ｏ	5		ｆ ｕ ｗ ａ ｓｈ ｕ ｗ ａ	1
	ｇｏ ts ｕ ｇ ａ ｒ ｉ	5	金網の触覚データクラス	ｃｈ ｉ ｋ ｕ ｃｈ ｉ ｋ ｕ	28
	ｇａｒａｇｕｒａ	5		ｃｈ ｉ ｋ ｕ ｗ ａ ｋ ｉ	1
	ｇｏ ts ｕ ｇ ａ ｒ ｕ	3		ｓｈ ｕ ｗ ａ ｓｈ ｕ ｗ ａ	1
	ｇｕｚａｒｏｇａ	1	デニムの触覚データクラス	ｇｏ ts ｕ ｇ ｏ ts ｕ	18
	ｚａｒａｚａ	1		ｇｏ ts ｕ ｇ ａ ｒ ｉ	6
	ｇｕ gy ａ ｇｕｒａ	1		ｇｏ ts ｕ ｇ ａ ｒ ｕ	3
光沢の触覚データクラス	ｓｕｂｅｓｕｂｅ	30		ｇｏ ts ｕ ｇ ａ ｒ ｏ	3

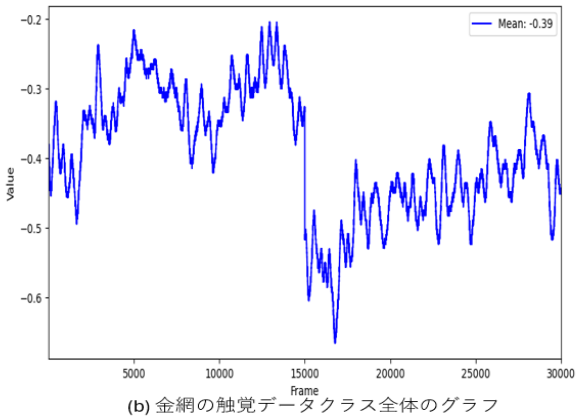
図 4.3 の結果、触覚データクラスとオノマトペ音素の特徴ベクトルが一番近いオノマトペ音素を生成すると述べているが、黄色いデニムはその周りに「ｗａｋｕｗａｋｕ」や「ｇｕｒｕｇｕｒｕ」が生成されることはない。その理由はこのモデルの特徴ベクトルは 256 次元で構成されており、図 4.3 はその中で 1,2 番目で目立つ次元を示しているので、実際に式 4.3 のユークリッド距離を用いて計算してみると「ｗａｋｕｗａｋｕ」や「ｇｕｒｕｇｕｒｕ」より「ｇｏｔｓｕｇｏｔｓｕ」が近いことが確認できた。

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{4.3}$$

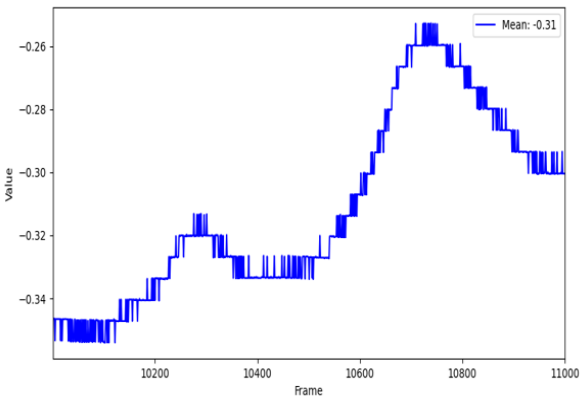
図 4.4 にオノマトペ音素の入力による触覚データクラスの生成結果と触覚データクラスのグランドトゥルースを示す。図 4.4 の (a) は「ｃｈｉｋｕｃｈｉｋｕ」というオノマトペ音素を入力した場合の触覚データクラスの生成結果である。図 4.4 の (b) は「ｃｈｉｋｕｃｈｉｋｕ」と対応している触覚データクラスの金網のをグランドトゥルースを図式化したものである。(a) と (b) の平均は似ているが、その波形は全く復元できていない結果が示された。グランドトゥルースを 1,000 フレイムずつランダムに取った波形を (c), (d) に示す。確かにその平均は近似しているがその波形の復元はできなかった。



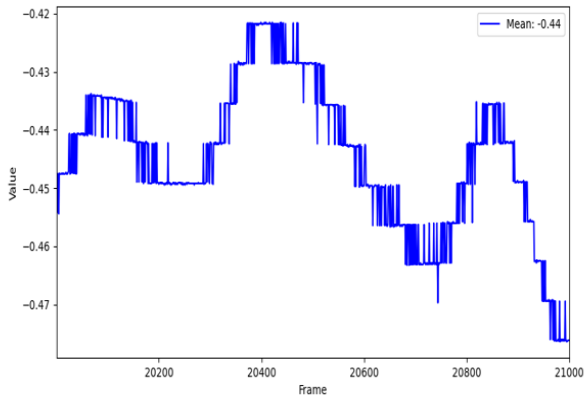
(a) 「ｃｈｉｋｕｃｈｉｋｕ」と入力した場合の触覚データクラスのグラフ



(b) 金網の触覚データクラス全体のグラフ



(c) 金網の触覚データクラス一部のグラフ(10001~11000フレイム)



(d) 金網の触覚データクラス一部のグラフ(20001~21000フレイム)

図 4.4: オノマトペ音素の入力による触覚データクラスの生成結果

4.2 考察

結果から触覚データクラスによるオノマトペ音素の復元は成功している。しかし、オノマトペ音素の入力による触覚データクラスの生成結果はその平均を復元しているが、波形は全く復元できなかった。このような結果から触覚データクラスの直流成分だけを用いて復元されているのではないかと考えた。それを証明するために表 4.2 に触覚データクラスの移動平均による正解率 (式 4.2) の変化を示す。表 4.2 を見ると移動平均を適用してもかなり復元が可能であるように見える。これは直流成分によりデータの分類が可能であることを指す。

表 4.2: 触覚データクラスの移動平均による正解率の変化

復元率 / case	コルク	デニム	ガーゼ	金網	光沢	全体
原本の触覚データクラス	3.33%	93.33%	60%	93.33%	100%	70%
100フレイム移動平均した場合	0%	0%	89.66%	96.55%	100%	57.24%
1000フレイム移動平均した場合	0%	0%	86.21%	96.55%	100%	56.55%
10000フレイム移動平均した場合	50%	100%	100%	100%	100%	90.00%

表 4.3 に触覚データクラスの直流成分の平均を 0 にした場合の正解率を示す。表 4.3 を見ると全く復元に成功していない。このような結果は波形 (フレイムによる変化) は全く分類に影響を与えなかった。これは波形は全くデータの分類に影響を与えなかったことを意味する。実際に平均を 0 にして復元を進めると「ｇｕｃｈａｋｕｒｉ」のようなあらゆるオノマトペ音素が結合された結果が示された。

表 4.3: 触覚データクラスの平均を０にすることによる正解率の変化

復元率 / case	コルク	デニム	ガーゼ	金網	光沢	全体
原本の触覚データクラス	3.33%	93.33%	60%	93.33%	100%	70%
平均を0に変えた場合	0%	0%	0%	0%	0%	0%
平均を 0 に変えて 100フレイム移動平均した場合	0%	0%	0%	0%	0%	0%
平均を 0 に変えて 1000フレイム移動平均した場合	0%	0%	0%	0%	0%	0%
平均を 0 に変えて 10000フレイム移動平均した場合	0%	0%	0%	0%	0%	0%

第5章　おわりに

5.1　まとめ

本研究では seq2seq でオノマトペ音素や触覚データクラスが復元できるかを確認し、それらを用いてマルチモーダルの学習により別データ同士の復元ができるモデルを作成した。オノマトペ音素と触覚データクラスという異なるモダリティを対象に、どちらも seq2seq によって復元できるかを検証し、モダリティ間の相互変換を行うモデルの構築を試みた。第 2 章では、オノマトペ音素を seq2seq で学習・復元させた結果、オノマトペ音素においては高い正解率を得られることを示した。第 3 章では、触覚データクラスに対して seq2seq を適用し、こちらも単体データでは概ねに復元可能であることを確認した。第 4 章にて、両方を組み合わせたマルチモーダル学習を導入し、触覚データクラスとオノマトペ音素の相互変換を試みたが、波形の細部を再現するには至らず、触覚データの直流成分を用いた分類・生成が中心となった。以上の結果から、単一モダリティでの seq2seq モデルの有効性は確認できたが、異なるモダリティ間の変換については波形情報の再現性に課題が残ることが分かった。

5.2　今後の課題

今後の課題として波形をより細かく復元するために、触覚データクラスとオノマトペ音素の特徴ベクトルを近似させる MSE Loss をほかの Loss を用いて実験を行いたい。触覚データクラスの細かいの波形を復元できない理由として、MSE Loss だけではその平均を近似されるだけの作業しかできないと思うからだ。加えて、エポック数やバッチサイズ、学習率などのハイパーパラメータを体系的に調整し、その誤差を減らすことを目指す。実際に、このようなハイパーパラメータを変えることでモデルの性能が上がったからである。また、オノマトペ音素の入力による触覚データクラスの生成結果は直流成分に依存しているが、触覚データクラスだけの seq2seq ではその波形もうまく復元されているので、マルチモーダルの学習の際にうまく学習が終了した触覚データクラスのモデルをそのまま使用し、オノマトペ音素を近似させる形式に実験を進めていきたいと思う。最後に、触覚データクラスをオノマトペ音素へ復元する過程でその正解率が厳しいと考えて編集距離を用いて実際にどれだけ近いのかを確認したいと思う。そうすると波形の復元は確かでオノマトペ音素の復元に集中できるからだ。さらに、オノマトペ音素を復元する際にこうした改良を通じて、より人間の感覚に近い質感理解や多面的なマルチモーダル変換の実現を目指していきたいと思う。

謝辞

本研究においては、研究の方針や着想のみならず、論文執筆や発表会に関する基本的な事柄に至るまで多くのご指導を頂きました。立命館大学 情報理工学部 島田伸敬教授に心より深謝致します。また、本研究の環境構築に当たり多大なご協力・ご助言を頂きました同学部の先輩である 松尾さんに深く感謝致します。加えて、貴重な触覚データクラスを提供してくださった野間研のご協力がなければ、本研究の成立は困難であったことをここに記してお礼申し上げます。そして、研究室内において種々のアドバイス・サポートをしてくださった秘書の皆様や、日々の議論を通して有益な知見を提供してくださった研究室の皆様に深くお礼を申し上げます。

参考文献

[1] Baltrusaitis, T., Ahuja, C., Morency, L. P. (2019).「Multimodal Machine Learning: A Survey and Taxonomy」, IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(2), 423–443.

[2] Takahashi, K. and Tan, J. (2018).「Deep Visuo-Tactile Learning: Estimation of Tactile Properties from Images」, arXiv preprint, arXiv:1803.03435.

[3] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A. Y. (2011).「Multimodal Deep Learning」, Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, Washington, 689–696.

[4] LLM へ至る道 Seq2Seqってやつがいるらしい . https://dev.classmethod.jp/articles/road-to-llm-advent-calendar-2023-14/(最終検索日:2025/01/16)

[5] 村尾航, 島田伸敬, オノマトペと画像のクロスモーダル分散表現の獲得, 立命館大学, 卒業論文, 2022

[6] 馬場睦也, 楠和馬, 波多野賢治, 音素単位の分散表現に基づくオノマトペ辞書構築法の提案, DEIMForum2020 G4-3

[7] Sequence To Sequence(Seq2Seq). https://blog.octopt.com/sequence-to-sequence/(最終検索日:2025/01/19)

[8] マルチモーダル AI とは?. https://www.aist.go.jp/aist_j/magazine/20231129.html(最終検索日:2025/01/20)