

平成 29 年度

筑波大学情報学群情報科学類

卒業研究論文

題目

ユビキタスインタフェースのための
無線型加速度センサと機械学習を用いた
触対象分類手法

主専攻 ソフトウェアサイエンス主専攻

著者 我妻 正太郎

指導教員 高橋伸 志築文太郎

要 旨

本研究では触覚情報収集と触覚情報分類の新たな応用として、人が触った物体や触り方をシステムに理解させることを利用したインタフェースを提案する。本論文ではこのインタフェースの開発の前段階として、人と環境の間で発生する触覚情報の収集と分類を実現するシステムを実装した。収集する触覚情報は人が物体を触った際の加速度情報に限定し、収集には既製品の無線型加速度センサを用いた。収集した情報の分類には、主に畳み込みニューラルネットワークによる機械学習を用いた。本研究において実装したシステムを用いて、収集した情報をどの程度の精度で分類できるか調査するための実験を行い、30種の触覚情報を約90%の精度で分類することに成功した。また、あらかじめ収集した情報を用いて機械学習のモデルを作成し、新たに触った物体の情報を分類する実験を行った結果、精度は約46%であった。これらの実験の結果から、より精度の高い分類を実現するため、触覚情報の収集の方法や機械学習の手法を検討した。

目次

第1章 序論	1
1.1 研究背景	1
1.2 研究目的とアプローチ	2
1.3 本論文の構成	2
第2章 関連研究または関連事例	3
2.1 触覚情報の収集に関する研究	3
2.2 触覚情報の分類に関する研究	3
2.3 物体との接触を入力としたデバイスの研究	4
第3章 触行動を検出するインタフェース	6
3.1 概要	6
3.2 利用例	6
第4章 加速度センサを用いた触覚情報の収集と分類	9
4.1 加速度センサを用いた触覚情報の収集	9
4.1.1 加速度センサ付き ZigBee マイコン	9
4.1.2 収集したデータの送信とデータの補間	13
4.2 触覚情報の分類	15
4.2.1 Multi-Layer Perceptron (MLP) による分類	15
4.2.2 Convolutional Neural Network (CNN) による分類	20
第5章 分類実験	22
5.1 MLP による分類	22
5.1.1 概要	22
5.1.2 実験の条件	22
5.1.3 結果と考察	22
5.2 CNN による分類	25
5.2.1 概要	25
5.2.2 18種類の触覚情報の分類	26
実験の条件	26
結果と考察	27
5.2.3 触察速度が異なる触覚情報の分類	29

	実験の条件	29
	結果と考察	30
5.3	学習結果に基づく触対象の識別実験	34
5.3.1	概要	34
5.3.2	触覚情報収集の条件	35
5.3.3	学習に用いるデータによる違い	35
	長時間データによる学習モデル	35
	短時間データによる学習モデル	35
5.3.4	結果と考察	36
第6章	議論	40
6.1	触覚情報の収集について	40
6.2	触覚情報の分類について	41
第7章	結論	42
	謝辞	43
	参考文献	44

目次

3.1	提案するインタフェースのイメージ図	8
4.1	本研究での触覚情報収集と分類のイメージ図	10
4.2	加速度センサ付き ZigBee マイコン, TWE-Lite-2525A	11
4.3	無線子機, MONOSTICK	12
4.4	MONOSTICK の使用例	12
4.5	触覚情報収集の様子	13
4.6	補間処理のイメージ図	14
4.7	本研究で用いる MLP の概略図	16
4.8	シグモイド関数のグラフ	17
4.9	ReLU 関数のグラフ	18
4.10	本研究で用いる CNN の概略図	21
5.1	MLP での分類実験に用いた物体	23
5.2	MLP での学習の進み具合のグラフ	24
5.3	18 種類の触覚情報分類に用いる物体	26
5.4	CNN での 18 種の触覚情報分類の学習の進み具合のグラフ	27
5.5	30 種類の触覚情報分類に用いる物体	30
5.6	分類結果の最頻値抽出のイメージ図	38

表目次

5.1	MLP の精度の一覧	25
5.2	CNN による 18 種の触覚情報分類により得られた混同行列	28
5.3	100 mm/s でなぞった場合の分類結果の混同行列	31
5.4	200 mm/s でなぞった場合の分類結果の混同行列	32
5.5	400 mm/s でなぞった場合の分類結果の混同行列	33
5.6	CNN による 30 種類の触覚情報分類の精度の一覧	33
5.7	触対象識別の精度の一覧	37
5.8	6 日間に分けて収集したデータを分類した結果から得られた混同行列	38

第1章 序論

1.1 研究背景

触覚やロボティクス、心理学を中心とした研究分野において、接触したものの弁別や、人が知覚する質感としての触覚情報処理の解明や、触覚ディスプレイのための入力信号として用いるなど、様々な目的のため、触覚情報を収集する研究が多く行われている。本論文では触覚情報とは、人が様々な物体を触った際に物体と指の間で発生する機械的振動や物体と指との温度変化等の情報を指すものとする。また、触覚ディスプレイとは触覚情報に基づく触覚刺激を人に提示する機器全般を指すものとする。

このような目的のため触覚情報を収集する手法は、様々なものが先行研究により提案されてきた [1-6]。これらの先行研究では、それぞれ収集する触覚情報は異なっており、収集する手法も研究ごとに異なっている。多くの先行研究では、収集したい情報に合わせ、様々なセンサを組み合わせた専用の機器を開発して収集を行っている。

多くの先行研究では触覚ディスプレイの入力信号として利用するため、触覚情報の収集に加えて触覚情報を分類する手法も研究されてきた [1, 3-5, 7-11]。多くの研究が行われている理由としては、収集した触覚情報をあらかじめ分類しておくことで、分類結果に基づき、触覚ディスプレイを用いて適切なコンテンツを提示することが可能になるからである。これらの先行研究において、触覚情報の分類は主に機械学習によって行われている。機械学習の手法をより効果的なものへと変更することで触覚情報の分類の精度を向上させるべく、盛んに研究が行われている。

このように、触覚情報を収集する手法、または収集した触覚情報を分類する手法は触覚ディスプレイへの応用を目的としたものが多いが、本研究ではこの手法を新たな方向に応用することを提案する。すなわち、収集した触覚情報を複数統合することで、人の触行動の検出を実現する。ここでの触行動とは、無意識的か意識的に関わらず、人が起こす何らかの行動において人が物体に対して行うインタラクション全般のことを指すものとする。人が起こす行動に起因するさまざまな日常的な触行動によって発生する触覚情報を収集し、その情報から人の触察時の動きや触察対象を分類できれば、人の触行動をシステムに理解させることが可能になる。なお、本研究において触察とは、無意識的か意識的に関わらず、人が物体に触れたときに行う、物体の表面をなぞるような動作を指すものとする。人の触行動をシステムに理解させることで、触覚情報の収集と分類を利用した新たなユビキタスインタフェースを開発することができる。

1.2 研究目的とアプローチ

本研究では触覚情報収集と触覚情報分類の新たな応用として、人の触行動をシステムに理解させることを利用したインタフェースを開発することを目的とする。本論文では、その前段階として触察時の情報を収集し、触対象の分類を行った。

本研究では、物体との接触によって起こる触行動を検出し、それに対応する操作を行うユビキタスインタフェースを提案する。具体的には、人の指や腕といった動作主体や、机やドアなどといった作用対象など、様々な場所に無線型のセンサを取り付けて触行動を検知し、触行動に対応した操作を行う。例えば指にセンサを取り付け、キーボードを叩くなど、デスクワーク作業動作を検出した場合、作業を効率化するため空調設備や照明条件を制御することで部屋の作業環境を適切に保つ。提案手法は人と環境の間で発生する様々な接触動作を入力情報として扱うことができるため、様々な用途に利用できる。

提案するインタフェース開発の第一歩として、人と環境の間で発生する触覚情報の収集と分類を実現するシステムを実装した。その後、実際に実験者が触覚情報を収集し、収集した情報を分類し、得られた結果に対する考察を行った。

本研究における触覚情報の収集には、嵯峨ら [12] が提案した手法を参考に、触察時の加速度情報に限定して収集を行った。収集には市販の無線型加速度センサ付きマイコンを用いた。無線型であるため、収集した情報を受信する機器から離れた場所でも情報収集が可能である。また、複数のマイコンから同時に情報を受信することが可能であり、様々な場所に設置しておくことで、複数のマイコンを組み合わせることで複雑な触行動を検出できる可能性がある。本研究において、触覚情報の分類は、主に畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) [13] を用いた機械学習によって行う。CNN はニューラルネットワークを用いた機械学習の手法の 1 つであり、入力データに畳み込み演算を行うことで、データの形を逃さず学習することができる特徴がある。これにより、加速度データの前後間のパターンを逃さず学習することが可能である。

1.3 本論文の構成

次章以降の本論文の構成について以下に示す。本章では、本研究の背景および目的とそれに対するアプローチを述べた。続く第 2 章では、本研究に関連する先行研究や事例を述べる。第 3 章では、提案するインタフェースを述べる。第 4 章では、触覚情報の収集と分類の手法と実装を詳説する。第 5 章では、実際に提案手法を用いて、触覚情報の収集、分類を行う実験を行った結果を考察する。第 6 章では本研究の触覚情報収集と触覚情報分類の手法の議論を行う。最後に第 7 章では結論と今後の課題を述べる。

第2章 関連研究または関連事例

この章では、関連研究または関連する事例を示す。触覚情報の収集と分類に関連する研究について示す。また、提案手法に関連する指などと物体の接触を入力に利用する研究についても示す。本研究のように触覚情報の収集と分類を両方行っている先行研究も多く存在しているが、それらについては収集と分類の2つに分け、それぞれの手法について言及する。

2.1 触覚情報の収集に関する研究

本節では、触覚情報の収集に関する研究について述べる。Abdulaliら[3]は、圧力センサと加速度センサを複合した専用の機器をPHANToM¹に取り付けて動作させることで9種類の素材の表面の情報を収集し、異方性素材のモデリング手法を提案した。PHANToM¹は仮想物体に触れて操作することを目的とした触覚ディスプレイの1つである。異方性素材とは、素材の縦方向、横方向、高さ方向のそれぞれで性質が異なる素材のことである。Streseら[1]は加速度センサ、圧力センサ、マイク、金属探知機、ウェブカメラ、赤外線センサの6つのセンサを複合した装置を開発し、108種の物体表面の触覚情報を収集し、それらの分類手法を提案し、実際に分類した。Burkaら[2]は自律型ロボットへの応用を目的とし、物体表面の視覚情報と触覚情報を収集するデバイスを提案した。Jamaliら[4]は、人工指に圧電フィルムを内蔵して、7種類の物体表面の情報を収集し、それらを分類する手法を提案した。

これらの先行研究の多くは様々なセンサを複合した専用の機器を開発し触覚情報の収集を行っている。これに対し、嵯峨ら[12]の研究では、市販の小型の無線型加速度センサ付きマイコンを指などに取り付け、センサを取り付けた指やペンで物体をなぞった際の加速度情報を触覚情報として収集する手法を提案した。嵯峨ら[12]の手法は、他の先行研究と比較し、機器が小型であることや無線型であることから、受信するデバイスから離れた場所での情報収集が可能である他、様々な場所に取り付けて情報の収集が可能という利点がある。これらの利点を考え、本研究において触覚情報の収集に関しては嵯峨ら[12]が提案した手法を参考に実装を行う。

2.2 触覚情報の分類に関する研究

先行研究では触覚情報の分類は機械学習を用いて行われているものが多い。しかし、用いる機械学習の手法は研究によって異なっている。Gaoら[8]は画像情報と触覚情報を複合して学習可能

¹Phantom premium (<https://ja.3dsystems.com/haptics-devices/3d-systems-phantom-premium> (最終閲覧日 2018年1月25日))

な CNN ベースの機械学習の手法を提案した。Tivive ら [10] は物体の表面のテクスチャの画像の分類に特化した CNN の構成を提案, その構成について議論した。Sukhoy [14] は 3 軸加速度センサを取り付けた人工爪をもったロボットに物体表面をなぞらせた際の 20 種の情報をサポートベクトルマシン (Support Vector Machine, SVM) [15] によって分類した。Gómez ら [16] は物体の触察時の振動および熱の情報を複合し, 再帰的ベイズ推定を用いることで, 物体の識別を行った。Regoli ら [7] はロボットが物体を把持する際の情報を収集し, ガウス混合モデル (Gauss Mixture Model, GMM) を用いた機械学習によって, 把持した物体を識別した。花光ら [17] は様々な素材をペン型デバイスによってなぞった触感を基に CNN モデルを作成し, それを利用して素材の触感の検索システムを提案した。

本章の冒頭で述べた通り, 触覚情報の収集と分類の両方を行った研究は多く存在しており, 前節で述べた研究においても, 分類を行っているものが存在する。Abdulali ら [3] は収集した触覚情報をニューラルネットワーク代表的構成の 1 つである Radial Basis Function Network (RBFN) [18] を用いた機械学習により分類した。Strese ら [1] はユークリッド距離, マハラノビス距離, GMM の 3 種の機械学習アルゴリズムをそれぞれを使用し分類を行い, 精度を比較した。Jamali ら [4] は単純ベイズ分類器を用いて分類を行った。

これらの先行研究に対し, 本研究では, 触覚情報の分類に多層パーセプトロン (Multi-Layer Perceptron, MLP) または CNN を用いる。MLP の 1 種である RBFN を用いた分類は Abdulali ら [3] が, CNN を用いた分類は前述の Gao ら [8] が実践し, 成果を残しているため, 高精度な分類が期待できる。先行研究が 2 種類以上の情報を複合して学習しているのに対し, 本研究の手法では収集した加速度情報のみで分類を行う。

本研究での触覚情報収集手法の参考となっている嵯峨ら [12] の研究では, 触覚情報を収集する手法を提案し触覚情報を収集したものの, 収集した触覚情報の分類については SVM で 8 種類の触覚情報の分類を行うのみとなっており, 課題が残っていた。本研究では, 嵯峨ら [12] と比べ, より多い種類の触覚情報を分類する。さらに触察速度を変化させて収集した触覚情報の分類も行う。

2.3 物体との接触を入力としたデバイスの研究

ヒューマンコンピュータインタラクションの分野において, 指などにセンサを取り付け, 物体との接触時の情報を入力とする研究はいくつか存在している。Han ら [19] は環境に関わらず常に使用可能な入力手法として, 指と手首に 3 軸の角度と加速度を計測できる Inertial Measurement Units (IMU) を装着し, 物体との接触を入力とする手法を提案した。櫛などの異なる凹凸を持つ 6 種類の物体を IMU を装着した指でなぞることで情報を収集し, なぞった物体を検出した。Yang ら [20] はオプティカルフローセンサと小型 RGB カメラ等を複合した機器を開発した。ここでのオプティカルフローとはデジタル画像内の物体の動きをベクトルとして表したものである。Yang ら [20] はこの機器を指に取り付け, 物体との接触時の物体表面の情報を収集した。実験では 32 種類の物体表面の情報を分類することに成功した。また, この機器を用いたアプリケーションについて示した。

これらの先行研究では, 専用の機器において収集した 2 種類以上の情報を分類に使用しているが, 本研究では加速度のみを用いて触対象の分類を行う。これらの研究では物体の情報の分類に SVM を用いているが, 本研究では主に CNN を用いた分類を行う。Han ら [19] は 6 種類の物体を

分類したが、本研究は最大 30 種類の物体の情報を分類する。また、Yang ら [20] の研究と本研究の手法を比べると、本研究の手法は加速度を収集するため、収集した情報を触覚ディスプレイの出力に応用が可能であるという利点がある。

第3章 触行動を検出するインタフェース

本章では、触覚情報の収集と分類の手法の新たな応用として、人の触行動を検出することを利用したインタフェースを提案し、その概要と利用例について述べる。

触覚情報を収集する手法と分類する手法は前述の通り、主に触覚ディスプレイへの応用を目的としているが、本研究ではこれらの手法を人と物体との接触によって起こる触行動の検出に利用する。触行動の検出が可能になれば、触行動を入力として利用できる。

3.1 概要

人と物体との接触によって起こる触行動を検出し、それに対応する操作を行うユビキタスインタフェースを提案する。触覚情報を収集し、それを分類することにより、人と何がどう接触したかを検出できる可能性がある。もし検出することができれば、人と物体の接触をインタフェースとして利用できる。

具体的には、指などにセンサを取り付け、様々な触覚情報を収集し、その情報を分類する。分類した触覚情報ごとに何らかの操作を割り当てておき、再度触覚情報を収集した際、対応する操作を実行する。触察して収集できる触覚情報は物体表面の形状によって異なり、表面形状が異なる物体間であれば、収集した情報を分類し、区別することは可能であると考えられる。例えば指にセンサを取り付けて、机をなぞった場合とキーボードをなぞった場合の触覚情報から、キーボードをなぞったのか、机をなぞったのかを分類し、どちらをなぞったのか検知できれば、なぞった物体に応じて操作を割り当て実行することが可能である。このことは指で物体をなぞることを入力として利用できることを意味する。

また、人が触った物体だけでなく、触り方も分類することができた場合、さらに複雑な操作を行うことができる可能性がある。例えば、机を指でなぞる場合を考える。指を机に押し付ける力を変化させると、収集できる触覚情報は変化すると考えられ、この際の触覚情報を分類できれば、指を机に押し付ける力ごとに操作を割り当てることが可能である。

このように、触覚情報の収集と分類を応用することで、人が他の物体に対して行う触行動をインタフェースとして利用できる可能性がある。

3.2 利用例

本節では、提案手法の利用例を示す。主な用途としては、触行動をトリガとし、インタフェース使用者の周辺環境を適切にコントロールすることが考えられる。例としては以下のような利用

が考えられる。

1. センサを指に取り付け、キーボードをなぞると、なぞった際の情報をシステムが認識し、あらかじめ設定したアプリケーションが起動する。
2. センサを指に取り付け、ある部屋の特定の壁をなぞると、その部屋の照明が点灯する。
3. センサをペンに取り付け、そのペンを持った人が筆記すると、作業を効率化できるよう、空調設備が起動し、部屋の温度を適切に保つ。

上記1, 2では指にセンサを取り付けて触覚情報を収集し、分類を行う。これにより、指で何を触ったのかを検出し、対応する操作を実行する。上記2では、特定の壁をなぞった場合のみ、照明が点灯することを想定している。特定の壁に対する特定の触れ方を弁別し、システムによる入力として利用することで、特定の照明を点灯させる。

また、人にセンサを取り付けるだけでなく、物体に取り付けて動作させても有効であると考えられる。上記3ではペンにセンサを取り付け、そのペンを用いて人が筆記を行った場合の情報を収集、分類し、それに対応した操作を実行している。このように、物体にセンサを取り付け、センサを取り付けた物体に対する人の操作から情報を収集し、それに操作を割り当て実行することも可能であると考えられる。

また、複数同じセンサを用意し、センサを個別に識別できれば、上記のような利用例を同時に複数行うことができると考えられる。センサを指やペンなど、様々な場所にあらかじめ設置しておくことで、様々な操作を同時に行うことができる。

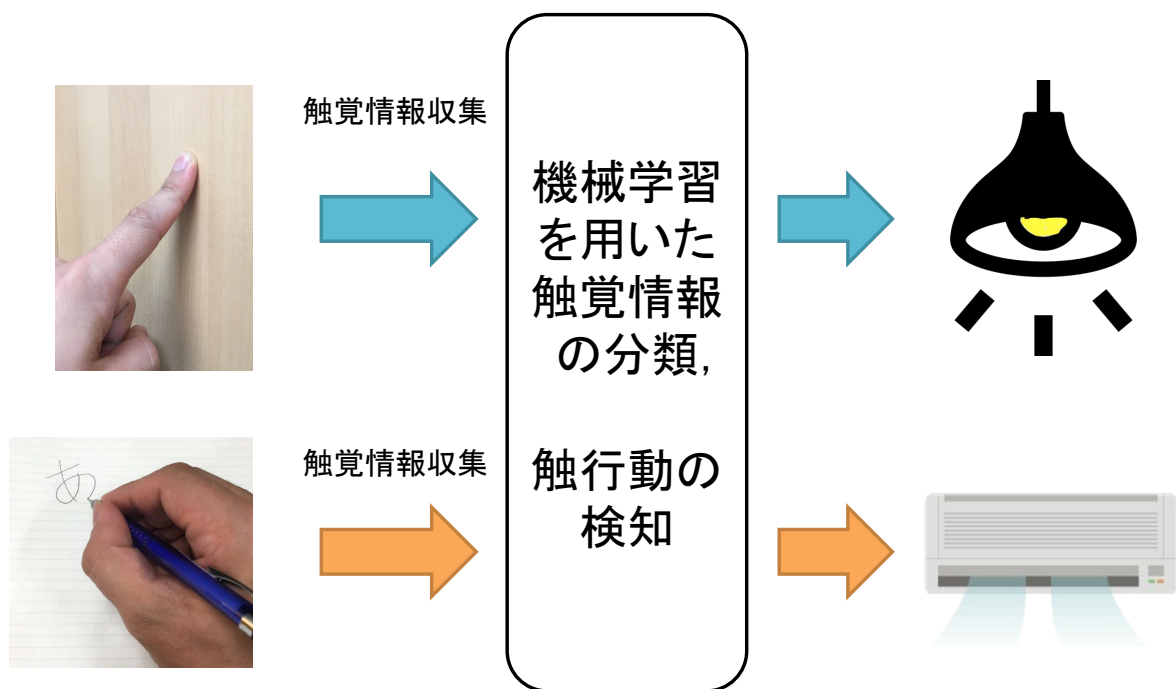


図 3.1: 提案するインタフェースのイメージ図. 触行動時の触覚情報を収集し, それを分類することで触行動を検知する. その結果を基に対応する操作を実行する. この図では, 指で壁をなぞるなどの画像が使用されているが, 実際には指やペンにセンサを装着することを想定している.

第4章 加速度センサを用いた触覚情報の収集と分類

本章では、提案するインタフェースのための触覚情報収集手法と分類手法、およびその実装について述べる。提案するインタフェースに用いる触覚情報を収集するセンサは、指やペンなどに取り付けることを考えるとできる限り小型であることが望ましい。また、無線型のセンサを用いることができれば、収集した情報を受信するデバイスから離れた場所でも情報を収集でき、受信するデバイスから離れた場所でもインタフェースを使用することができる。これらの観点から、本研究では嵯峨ら [12] の手法を参考に、加速度センサ付き ZigBee マイコンを指などに取り付けて動作させ、その際の加速度情報を触覚情報として収集する。収集した情報は PC へ送信し、後述する補間処理を行う。その後は収集した情報を MLP または CNN を用いた機械学習によって分類を行う。全体のイメージ図を図 4.1 に示す。

4.1 加速度センサを用いた触覚情報の収集

本節では、触覚情報の収集について述べる。触覚情報の収集には前述の通り、嵯峨ら [12] の手法を参考に加速度センサ付き ZigBee マイコンを用いた実装を行う。本研究では触覚情報として、センサを取り付けた指などが何かに触れた際の加速度情報を収集する。

4.1.1 加速度センサ付き ZigBee マイコン

本研究では、TWE-Lite-2525A² (MonoWireless Inc.) という既製品のデバイスを使用した。このデバイスを図 4.2 に示す。このデバイスには 3 軸加速度センサと ZigBee 無線モジュールが内蔵されており、電源にはボタン電池 (CR2032) が用いられている。大きさは 25 mm×25 mm×10 mm (縦×横×高さ) であり、ボタン電池を含め質量は 6.5 g である。デバイスに搭載されている加速度センサは ADXL345 である。ADXL345 は通常 0.1–800 Hz での加速度情報の収集が可能である。

さらに ZigBee を用いた無線通信が可能であることから、様々な物体に取り付けて動作させることが可能である。ZigBee とは無線の規格の 1 つで、センサネットワークを主な目的とした近距離無線通信規格である。データ転送速度は 250 Kbps である。ZigBee は無線機をスリープさせた場合、スリープから復帰するのが早いという特徴がある。普段はスリープ状態で、触覚情報を収集する際に起動させれば、省電力化が期待できる。また、複数機器を同時接続することが可能であ

²TWE-Lite-2525A (<https://mono-wireless.com/jp/products/TWE-Lite-2525A> (最終閲覧日 2018 年 1 月 25 日))

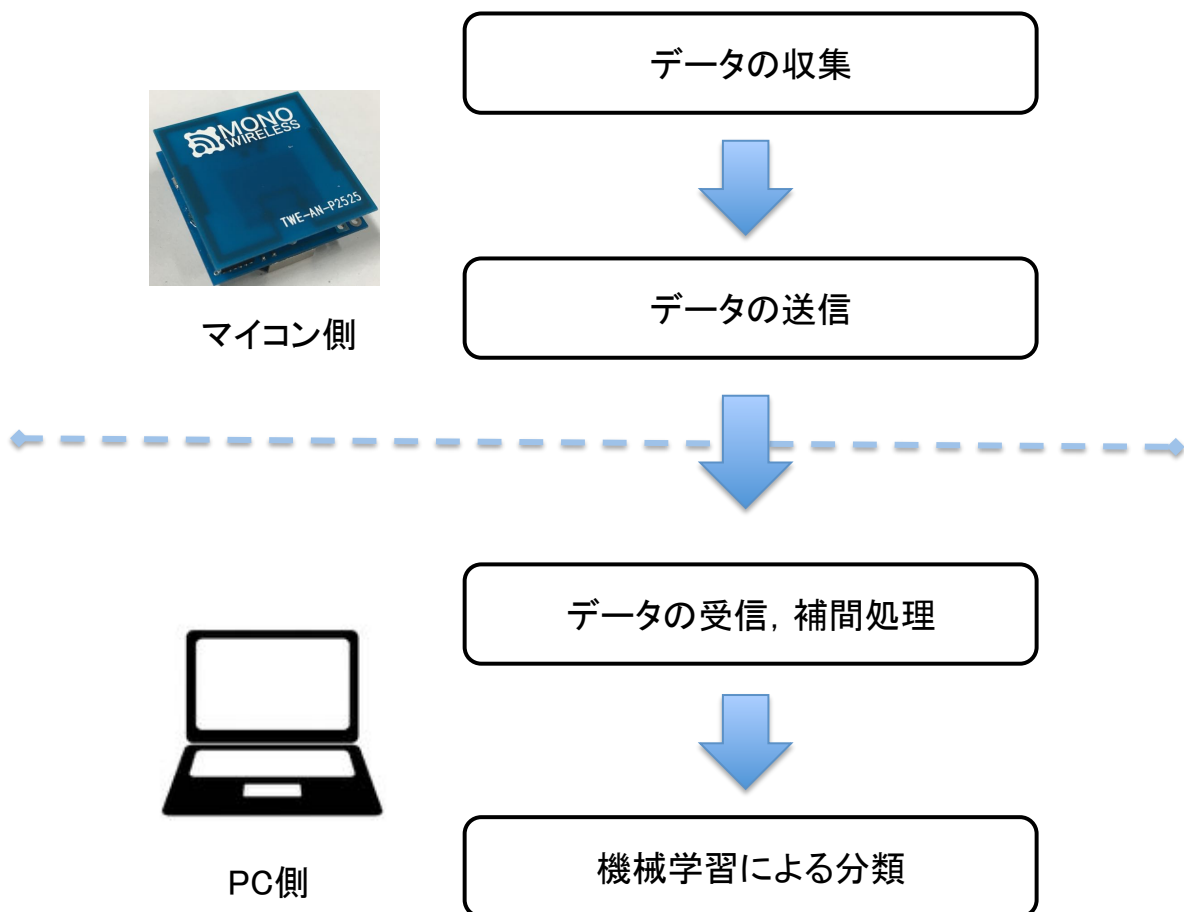


図 4.1: 本研究での触覚情報収集と分類のイメージ図. 加速度センサ付き ZigBee マイコンから触察時の加速度情報を PC へ送信, その後, PC にて機械学習を用いた分類を行う.

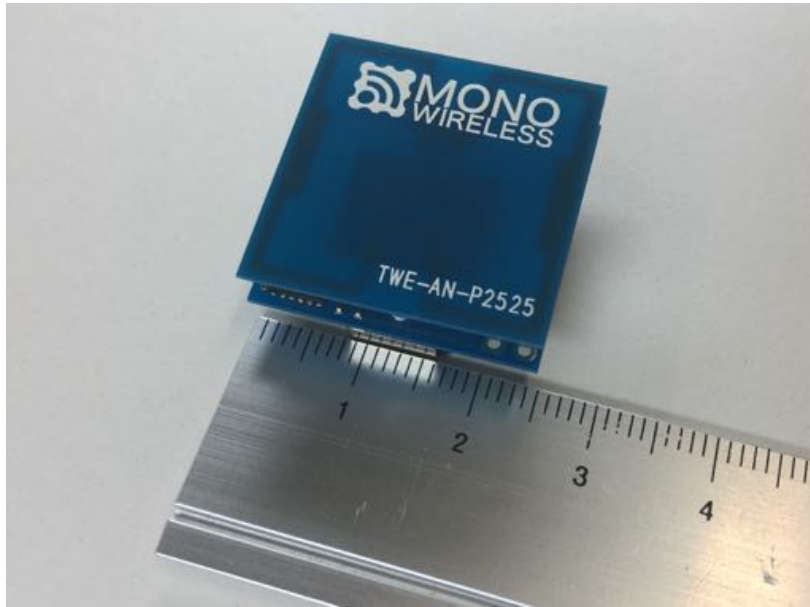


図 4.2: 加速度センサ付き ZigBee マイコン, TWE-Lite-2525A² (MonoWireless Inc.). 3 軸加速度センサ ADXL345 が内蔵されている. 電源にはボタン電池 (CR2032) を用いる.

るため, 複数台のデバイスで複数の触覚情報を同時に収集することが可能である. このデバイスを指やペンに取り付けて動作させることで, 触覚情報の収集を行う.

また, 無線通信が可能であることにより, 従来の有線の触覚情報収集機器に比べて, 情報収集できる距離が長くなるという利点がある. これは, 収集した情報を受信する端末から, 有線では届かない離れた場所でも情報収集ができる可能性があることを意味している. また, このデバイス取り付けられた物体にコードが絡まるなどといった問題が発生しないため, 取り付けられた指や物体の行動を邪魔になりにくい. これらの性質により, 先行研究における, 実験環境内でしか触覚情報を収集することができないという問題に対処し, 日常生活における触覚情報を収集できる可能性がある.

このデバイスの無線通信を受信するためには, ZigBee 通信が可能な同型の TWE-Lite マイコンが必要になる. 同型のマイコンを用いるため, ZigBee の通信は, それぞれのデバイスも受信し中継が可能であり, さらに通信距離を延長することも可能である. これらの通信を PC などで受信し処理するためには, これらのデバイスと同一の ZigBee マイコンを搭載し, USB 通信も可能な専用のデバイス, MONOSTICK³ (MonoWireless inc.) が必要になる. MONOSTICK の外観図を図 4.3 に示す. MONOSTICK の使用例を図 4.4 に示す. MONOSTICK は USB をインタフェースとして備えており, スマートフォン, PC, タブレット端末等様々な機器に接続することが可能である. この性質により, 様々な端末で収集した触覚情報を受信できる他, 受信した端末をさらにイーサネットを介して中継することで, 別の端末に触覚情報を送信することが可能となる.

このデバイスは前述の通り既製品であり, 3000–4000 円にて販売されており, 入手が容易であ

³MONOSTICK (<https://mono-wireless.com/jp/products/MoNoStick/index.html>(最終閲覧日 2018 年 1 月 25 日))



図 4.3: 無線中継器, MONOSTICK³ (MonoWireless Inc.). USB 端子が搭載されており, スマートフォン, タブレット等, 様々な機器に接続することができる.



図 4.4: MONOSTICK の使用例.

る。また、開発環境も整備されている。モノワイヤレス社が自社の ZigBee マイコンのプログラムを自由に書き換えることができる GUI アプリケーション、TWELITE プログラマを配布している他、ソフトウェア開発環境、TWELITE NET SDK を配布している。これらのアプリケーション開発環境を利用することで、容易にマイコンのプログラムを開発し、更新することができる。マイコンのプログラムには C 言語が用いられている。

ZigBee 通信の特徴を考慮し、省電力化のため、デバイスにスリープ機能を追加した。測定した加速度が設定した閾値を超えるまで、通常の測定より低周波で測定し、閾値を超えて初めて通常の測定を開始する。

実際にこのデバイスを用いて触覚情報を収集する際には、3D プリントにて作成した専用のケースに入れた上で、指やペンに装着する。その状態で指やペンで様々な物体をなぞることによって触覚情報を収集する。ケースに入れ、触覚情報を収集している様子を図 4.5 に示す。



図 4.5: 触覚情報収集の様子。TWE-Lite-2525A² を 3D プリントで作成したケースに入れ、指やペンに装着する。実際の収集の際には TWE-Lite-2525A² を覆い隠すようにケースに蓋をし、触察時にデバイスがケースの外に出ないようにする。

4.1.2 収集したデータの送信とデータの補間

機械学習による分類を行うため、収集した加速度データを ZigBee を用いた無線通信によって PC に送信する。前述の通り、本研究で使用するデバイスに搭載されている加速度センサ ADXL345 では、0.1–800 Hz での加速度計測が可能である。しかし、無線モジュールと PC 間の ZigBee 通信を考慮しなければならず、実際の値は 800 Hz に比べて著しく下がる。本研究で使用する TWE-Lite-2525A² の加速度計測レートは購入時に書き込まれているプログラムでは 33 Hz となっており、そのままの実装では十分に触覚情報が収集できない可能性がある。これには、人の感知する振動の周波数帯域が関係している。人の皮膚は 0–1 kHz 程度までの振動を感知する。その中でも、0–200 Hz 程度の情報から、物体と接触した際の速度や皮膚の曲がり方などの細かい情報を得ている¹と

¹ヒトの触覚のメカニズム, 触覚の生理・心理学の初歩 (<http://www.rm.mce.ucc.ac.jp/sice/2006HapticsCommitteePPT.pdf>(最終閲覧日 2018 年 1 月 25 日))

される。このことから、33 Hz での計測では不十分であると考えられる。そこで、連続する 10 点の加速度計測値をまとめて 1 つのパケットとして送信する方式を採用し、マイコンのプログラムを書き換えた。これにより 3 軸加速度計測値を約 330 Hz で計測することが可能となった。

連続する 10 点の加速度計測値をまとめて送信する処理のため、前後のパケット間の計測時間間隔に若干不規則なずれが生じる場合がある。そこで送信されたデータに補間処理を施すことにより、計測時間間隔を揃え、データのずれを解決する。この補間処理のイメージ図を図 4.6 に示す。

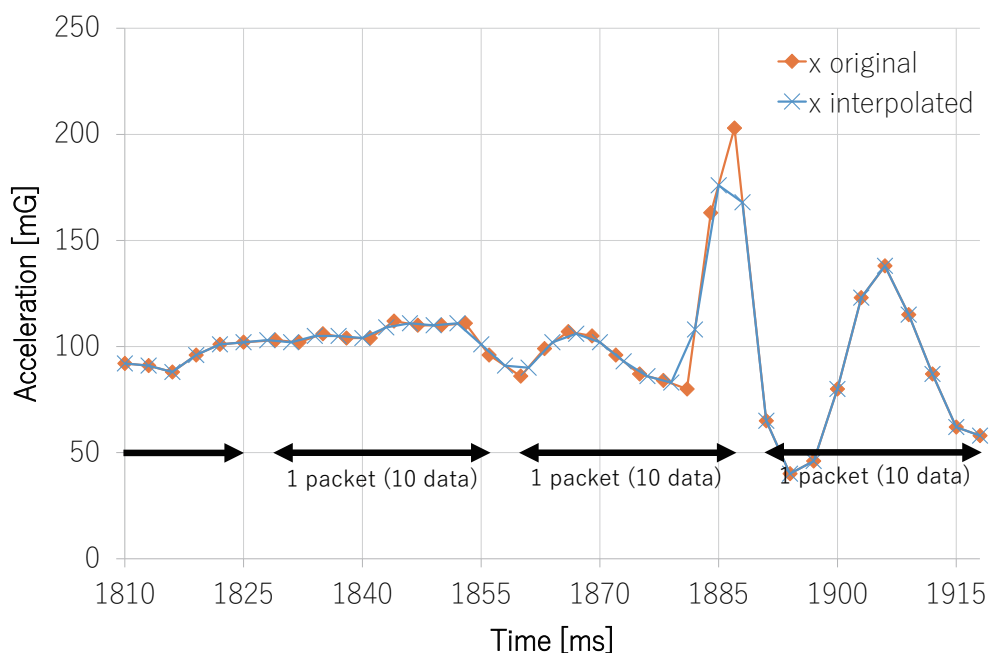


図 4.6: 補間処理のイメージ図。前後 2 点の計測データを用いて 2 次方程式を求め、1 ms 間隔の値を算出することで線形に補間する。

補間処理の詳細について説明する。前後のパケット間の計測時間間隔のずれを解消するため、線形補間により、一定間隔のデータに整える。ここでは、一定間隔を 3 ms とする。これは本研究で使用するマイコンの 1 時刻が 3 ms 間隔で計測されているからである。線形補間の計算式について説明する。時系列の前後 2 点を用いて、2 次方程式を解く。x をタイムスタンプ、y を加速度の値とすると、1 軸ごとの 2 値 $(y_1, x_1), (y_2, x_2)$ から式 3.1 の方程式を作成し、解く。

$$\begin{aligned} y_1 &= ax_1 + b \\ y_2 &= ax_2 + b \end{aligned} \tag{4.1}$$

解いて得られた線形方程式 $(y = ax + b)$ から 2 値間 $(x_2 - x_1)$ の間の 1 ms 間隔の値を求めた。これをすべての 2 値間にて行い、最後に 3 ms 間隔に値を分割する。

4.2 触覚情報の分類

この節では、触覚情報を分類について述べる。触覚情報の分類は、MLP または CNN を利用した機械学習によって行う。どちらもニューラルネットワークを利用した機械学習の一手法である。先行研究においてもニューラルネットワークを用いて触覚情報を分類し、成果を残した研究が存在し、高精度な分類が期待できる。ニューラルネットワークを利用した機械学習は、データ量を用意できれば、高い精度で分類することができる利点があるが、人があらかじめ定義しておかなければならないハイパーパラメータが多くなってしまふのが欠点である。ハイパーパラメータは解決すべき問題に応じて、調整が必要になる。この節で述べる機械学習の構成は、本研究におけるの基本となる構成であり、後述の実験の際に調整を行っている場合がある。本研究での機械学習は、プログラム言語 Python と Google 社が提供する機械学習ライブラリ TensorFlow⁴ を使用し実装した。さらに、GPU と TensorFlow⁴ を連携させることで、計算の高速化を図った。

この節での、ニューラルネットワークへの入力には収集した 3 軸加速度データの中から連続 200 点分の 600 次元データ、出力は、入力したデータが何であるかの確率とする。また、入力するデータにはあらかじめ教師データとしてのラベル付けを行う。ラベルは、要素の値が 0 か 1 で構成されている配列で、正解のインデックスの要素のみ 1 が格納されている。

4.2.1 Multi-Layer Perceptron (MLP) による分類

MLP を用いた触覚情報の分類について述べる。MLP は順伝搬型のニューラルネットワークの 1 つである。MLP はデータを入力する入力層、データを推測する複数の中間層、推測結果を出力する出力層によって構成される。推測結果から、各層の重みパラメータを更新することで学習する。

本研究での MLP は入力層 1 層、中間層 3 層、出力層 1 層の計 5 層のニューラルネットワークを用いた。ニューラルネットワークの概略図を図 4.7 に示す。

ニューラルネットワークの入力は前述の通り、連続 200 点の 3 軸加速度データであるが、ニューラルネットワークへ入力する際、1 次元の配列に変換する必要がある。よって、 3×200 点のデータ (x, y, z の 3 軸で 200 点) を一列に並べ、 1×600 のデータとし、それをニューラルネットワークの入力とした。

細かな設定について述べる。全ての中間層において、ニューラルネットワークのノード数は 600 に固定した。次の層へ値を伝搬する際に必要な活性化関数は、入力層から中間層第 2 層までをシグモイド関数、中間層第 3 層から出力層の間は ReLU (Rectified Linear Unit) 関数 [21] を用いている。

シグモイド関数は式 (3.2) で表される。グラフにすると図 4.8 のようになる。シグモイド関数は、比較的単純な非線形関数であることから、微分の計算が容易であり、重みパラメータを更新する際の計算負荷を減らすことができるため、多くのニューラルネットワークの活性化関数として使用されている。

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (4.2)$$

⁴TensorFlow (<https://www.tensorflow.org/> (最終閲覧日 2018 年 1 月 25 日))

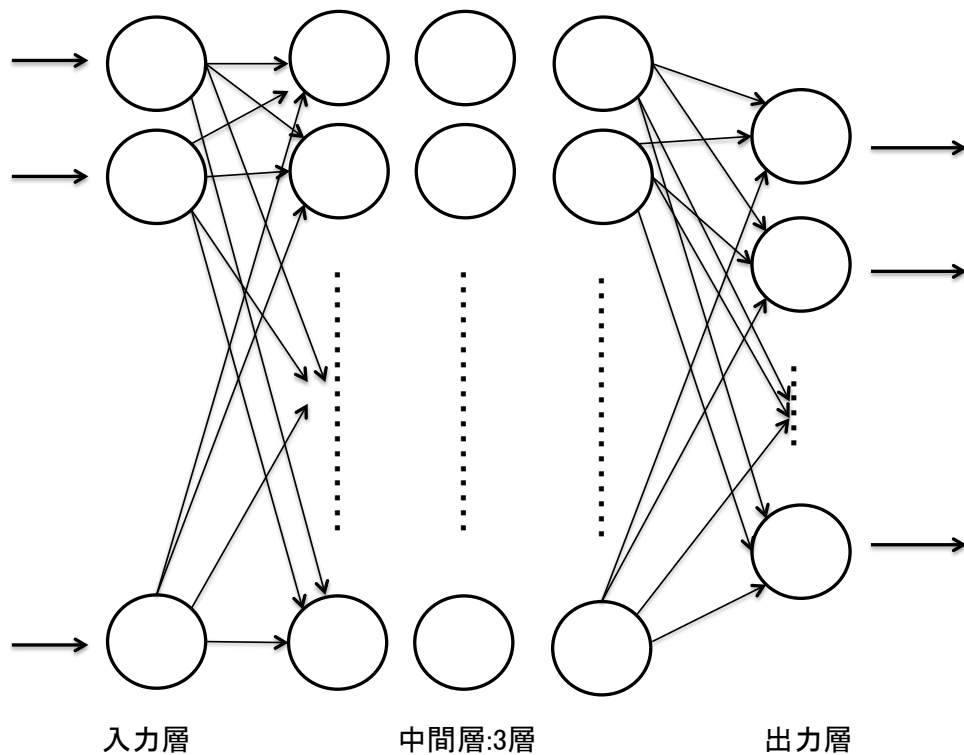


図 4.7: MLP の構成の概略図. 計 5 層のニューラルネットワークである. 本研究では, 入力は 1×600 の加速度データ, 出力は入力がどのデータであるかの確率である. 中間層のノード数はいずれも 600 としている.

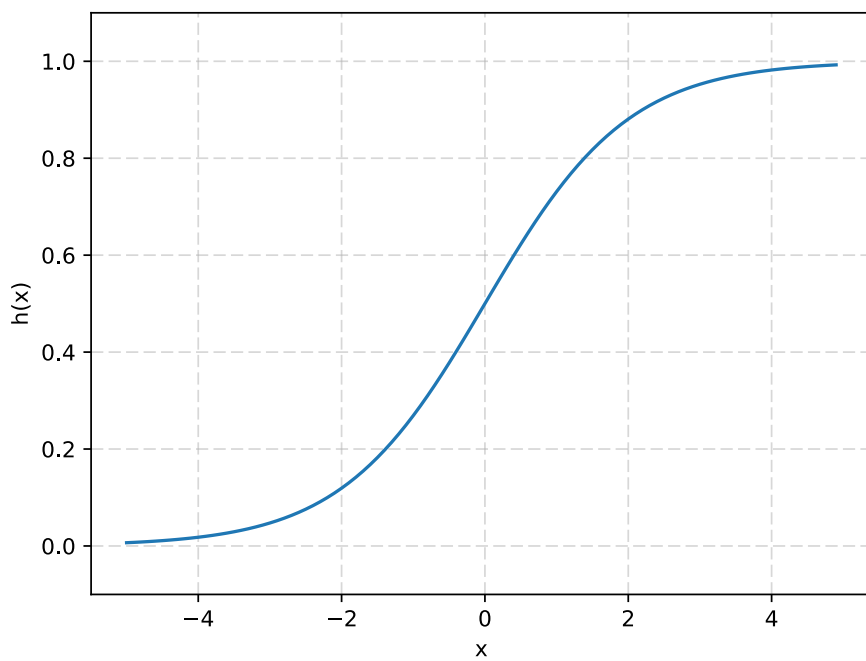


図 4.8: シグモイド関数のグラフ

ReLU 関数は式 (3.3) で表される。0 か入力値のどちらか大きい方を出力する単純な関数であるが、Glorot ら [21] が活性化関数として ReLU 関数を使用することで、他の活性化関数より精度を改善できる可能性があることを示した。関数が非常に単純であることから、シグモイド関数よりさらに計算負荷を減らすことが可能である。また、 $x > 0$ において微分値が常に 1 であるため、重みに関する微分の値が小さくなりすぎることによって学習が止まる危険性を減らすことが可能である。

$$h(x) = \max(0, x) \quad (4.3)$$

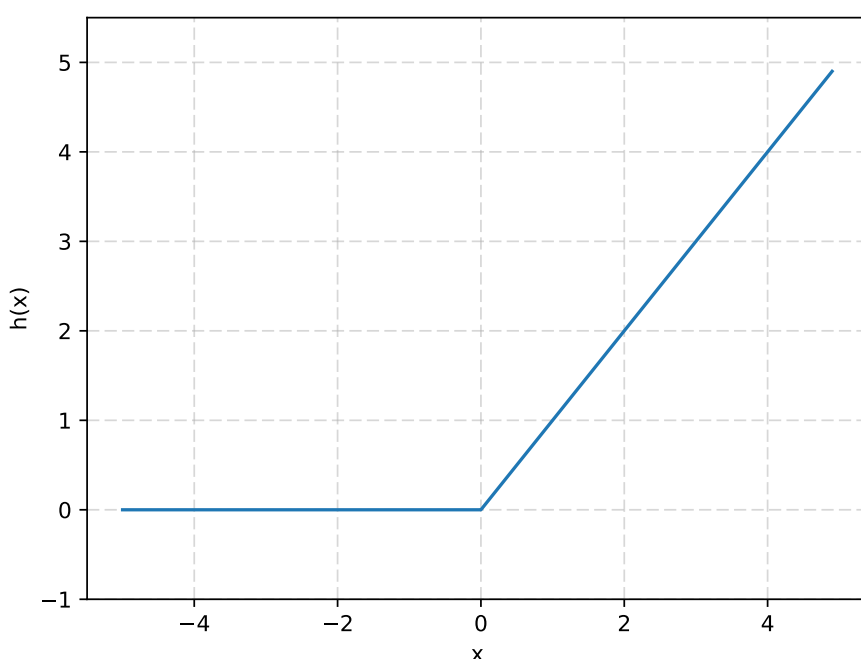


図 4.9: ReLU 関数のグラフ

LeCun ら [22] によれば、通常、活性化関数に ReLU 関数を用いるのが最善とされているが、後述の実験の際、中間層の活性化関数を一部シグモイド関数に変更すると精度の向上が見られたため、ReLU 関数だけでなく、シグモイド関数も使用した構成とした。

ニューラルネットワークの出力層においては、伝搬してきたデータをソフトマックス関数を用いて、0-1 までの値のデータに変換する。このデータを確率として扱うことで、結果の統計的な分析が可能になる。 k 番目の出力データを y_k とし、ソフトマックス関数への入力を a_k 、出力データ全体の数を n とすると、ソフトマックス関数は式 (3.4) で表される。出力層での値を指数関数を用いて正の値へ変換し、全ての出力の指数関数の和で割ることで、0-1 までの値に変換する。指数関数を用いているため、要素間の大小関係を保ったまま、0-1 の値に変換することができる。ニュー

ラルネットワークを用いた機械学習による分類では、ニューラルネットワークの出力の値が最も大きいものに対応する要素を分類結果として採用する機会が多いことから、出力の要素の大小関係を保ちつつ、統計的な分析を可能にするソフトマックス関数は広く用いられている。

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)} \quad (4.4)$$

ニューラルネットワークの学習は、出力データを基に損失関数を用いてニューラルネットワークの性能の悪さを評価し、損失関数の値をより減らすことができるように重みを更新することで行われる。損失関数、重みの更新手法はそれぞれ様々なものが存在する。本研究のMLPでは、損失関数には正解の確率分布と分類器の予測分布の交差エントロピーを用いる。 t_k を正解ラベル、 y_k を出力データとすると、本研究で用いる交差エントロピー E の式は(3.5)で表される。本研究の実装においては前述の通り、正解ラベルとなる t_k について、正解のインデックスだけが1で、その他が0となるベクトルとして表現する。これにより、式(3.5)の計算は正解のインデックスの場合の対数の計算 ($E = -\log y_k$) のみに削減できる。また、実際の計算の際には、 y_k に微少な数値 (1×10^{-7}) を足し、 y_k の値が0であっても計算を可能にする。

$$E = - \sum_k t_k \log y_k \quad (4.5)$$

また、過学習を抑えるため、損失関数により得られた値に各層の重みの2乗ノルムを加算する[23]。損失関数によって得られた値を L 、重みを W 、加算後の値を L' とすると、式(3.6)で表される。ここでの λ はハイパーパラメータで、本研究では0.1とした。

$$L' = L + \frac{1}{2} \lambda W^2 \quad (4.6)$$

本研究のMLPにおける重みの更新手法に関しては、Kingmaら[24]が提案したアルゴリズム、Adamを用いる。 W を更新する重みパラメータ、損失関数の値を L 、 $\frac{\delta L}{\delta W}$ を W に関する損失関数の勾配とすると、式は(3.7)のように表される。ここでの $\alpha, \beta_1, \beta_2, \epsilon$ はハイパーパラメータで、本研究ではKingmaら[24]の論文を参考に、 $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ とした。

$$\begin{aligned}
m_0 &= 0 \\
v_0 &= 0 \\
m_{t+1} &= \beta_1 m_t + (1 - \beta_1) \frac{\delta L}{\delta W} \\
v_{t+1} &= \beta_2 v_t + (1 - \beta_2) \frac{\delta L}{\delta W} \odot \frac{\delta L}{\delta W} \\
\hat{m} &= \frac{m_t + 1}{1 - \beta_1^t} \\
\hat{v} &= \frac{v_t + 1}{1 - \beta_2^t} \\
W^{t+1} &= W^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v} + \epsilon}}
\end{aligned} \tag{4.7}$$

4.2.2 Convolutional Neural Network (CNN) による分類

CNN を用いた触覚情報の分類について述べる。CNN はニューラルネットワークを用いた機械学習手法の 1 つである。通常の順伝搬型ニューラルネットワークと比べて、畳み込み層とプーリング層が追加されている。畳み込み層では、あらかじめ設定した畳み込みフィルタを基に入力データに畳み込み演算をすることによって、データの特徴を抽出する。この際、通常のニューラルネットワークと違い、データの空間的形狀を維持する。これにより、データの形状やパターンが重要となるデータの分類を高精度で行うことが可能である。プーリング層では、データを圧縮し、扱いやすい形に変形する。これにより、データの小さな位置の変化に頑健となる他、計算コストを抑えることができる。

本研究では、計 13 層の CNN を用いた。その概略図を図 4.10 に示す。CNN の入力と出力に関しては、前述の通り、3 軸加速度データ連続 200 点分 (3×200) を入力し、入力したデータの分類結果の確率を出力する。

本研究の CNN の構成は代表的な CNN の構成の 1 つである VGG [25] の構成を参考にしている。VGG [25] は一回の畳み込み層において畳み込みする量を減らし、畳み込み層自体を増やすことで精度の向上を図った構成である。本研究の CNN においても、畳み込みフィルタのサイズを小さくする代わりに、層を増やすことによって、精度の向上を図っている。本研究の CNN の畳み込みフィルタのサイズは全畳み込み層において 1×5 とした。フィルタの数は、VGG [25] の設定を参考に、畳み込み第 1 層、第 2 層では 64 枚、第 3 層、第 4 層では 128 枚、第 5 層、第 6 層では 256 枚とした。プーリングには全プーリング層において 1×2 の Max プーリングを用いた。 1×2 のデータの領域から最大値を取り出し、新たなデータとする。これを繰り返すことで、データの量を半分に圧縮する。

次の層に値を伝搬する際に用いる活性化関数は全ての層において、前述の ReLU 関数を用いた。出力層では MLP の場合と同じく、ソフトマックス関数を用いて出力データを確率に変換した。損失関数は交差エントロピーを用いた。重みの最適化アルゴリズムには Adam [24] を用いた。Adam を扱う際必要になるハイパーパラメータは MLP の場合と同様である。

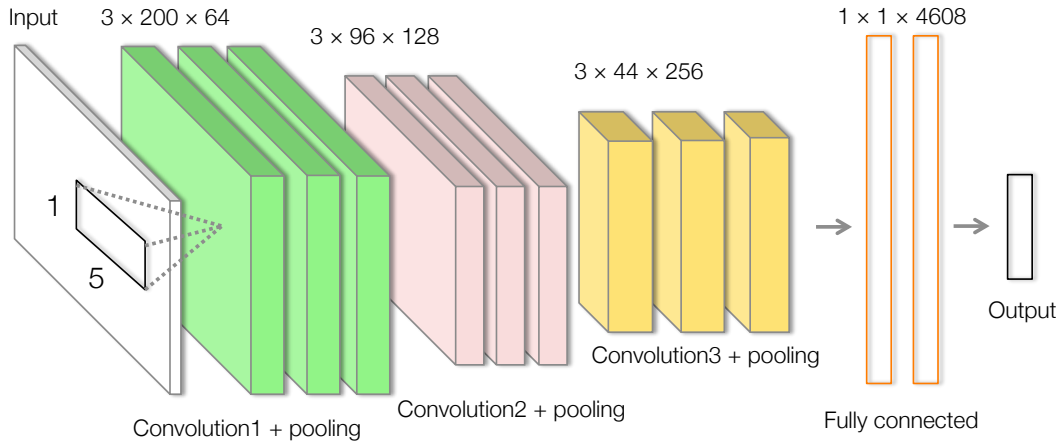


図 4.10: 本研究で用いる CNN の概略図. 各層に Batch Normalization, 全結合層と出力層の間のみ Dropout を用い, 過学習を抑制している. CNN への入力, 3×200 の連続 200 点の加速度データとする. 出力は, 入力データの確率とする.

また, 過学習を抑制するため, Batch Normalization [26] と Dropout [27] を用いる. Batch Normalization [26] は学習を行う際の入力データの集合に対し, 平均と分散を求めて, それらを基に入力データの分布について平均が 0, 分散が 1 になるように正規化を行う. m 個の入力データ $\{x_1 \dots x_i \dots x_m\}$ とそれに対する平均 μ_B , 分散 σ_B^2 とし, m 個の出力データ $\{y_1 \dots y_i \dots y_m\}$ とすると, このアルゴリズムは式 (3.8) で表される. γ と β は学習によって最適化されるパラメータである. 初期値は $\gamma = 1$, $\beta = 0$ である. ϵ は微小量 (1×10^{-5}) である. 各層において活性化関数を使用した後にこのアルゴリズムを用いることで, 過学習を抑制する.

$$\begin{aligned}
 \mu_B &= \frac{1}{m} \sum_{i=1}^m x_i \\
 \sigma_B^2 &= \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\
 \hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \\
 y_i &= \gamma \hat{x}_i + \beta
 \end{aligned} \tag{4.8}$$

Dropout [27] は学習を行う際, ニューラルネットワークのノードをランダムに消去しながら学習を進める手法である. Dropout [27] を行うことで, 擬似的に複数のモデルを個別に学習させ, 結果の平均を出力する操作と同じ効果を実現し, 過学習が抑制される. 本研究の実装では, 全結合第 2 層から出力層の間に Dropout [27] を使用し, ランダムに半分のノードを消去している.

第5章 分類実験

本章では、実際に触覚情報を収集し、分類する実験を行い、どの程度の精度で収集した触覚情報が分類可能か調査する。MLP と CNN それぞれで分類実験を行った。

5.1 MLPによる分類

5.1.1 概要

MLP を用いた触覚情報の分類を行う。MLP を用いて 5 種類の触覚情報を分類した結果から、本研究の手法により収集した情報を MLP を用いてどの程度の精度で分類可能であるかを調査する。

5.1.2 実験の条件

TWE-Lite-2525A を専用のケースを用いて指に装着し、5 種類の物体をなぞることによって著者自ら触覚情報を収集した。MLP による分類実験で用いた物体を図 4.1 に示す。carpet1, carpet2, carpet3 が素材の異なる絨毯の一部、sponge-g, sponge-y が家庭用スポンジの表と裏である。これらの物体の大きさは縦横共に 8–12 cm 程度である。それぞれの物体の表面において、指を左右に往復させて 3 分間なぞることで、その際の加速度情報を触覚情報として収集した。なぞる際の指の角度は物体表面から 45 度程度になるようにした。この操作を 3 回行うことで、物体 1 つにつき、9 分間の加速度データを収集した。物体をなぞる速さは、物体を 1 秒間に左右に 2 往復 (約 400 mm/s) できる速さで、折り返し以外の場所ではおおまかに一定となるようにした。

収集したデータは、前章にて述べた MLP にて分類を行う。収集した加速度データの中からランダムで連続 200 点を取り出し、MLP への入力とする。出力は入力したデータがどの物体のデータであるかの確率とする。この際、収集した全データをランダムに 8:2 に分割し、前者を訓練データ、後者をテストデータとした。訓練データによって学習させ、テストデータを分類することによってどの程度の精度で分類が可能なのか調査する。また、学習の進行を確認するため、訓練データにおいても分類を行う。学習は全データから抜き出した連続 200 点のデータを 100 個学習させるミニバッチ学習を 1 エポックとし、10000 エポック行った。

5.1.3 結果と考察

MLP による分類実験の結果を述べる。5 種類の触覚情報を分類した結果、精度は約 30%であった。分類を行った際の学習の進み具合のグラフを図に示す。

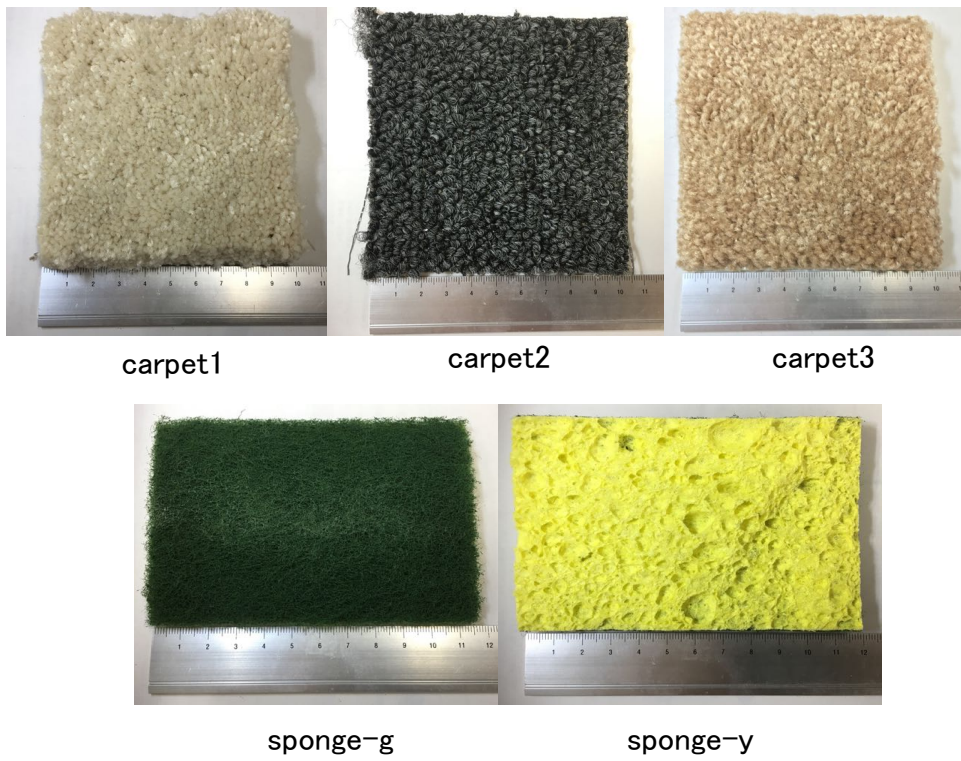


図 5.1: MLP での分類実験に用いた物体

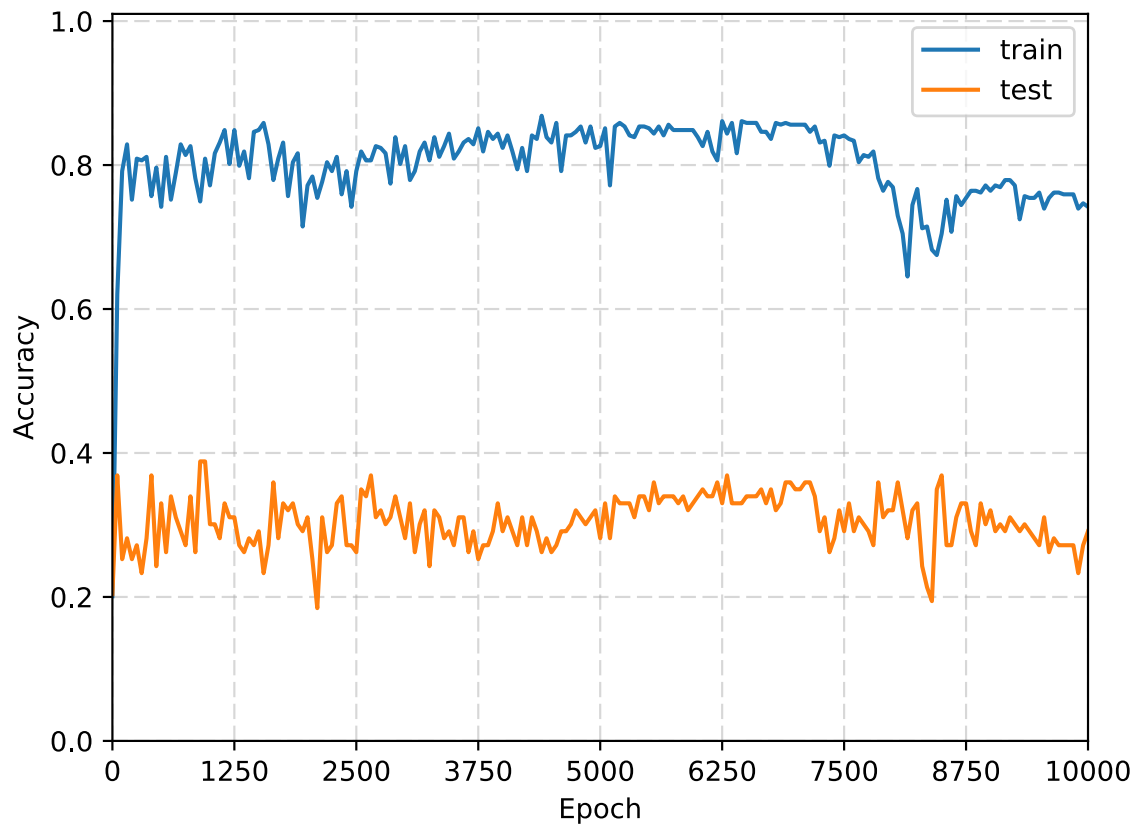


図 5.2: 中間層 3 層の MLP での学習の進み具合のグラフ。横軸が学習回数，縦軸が精度である。学習 50 回ごとに記録している。

グラフを見ると、訓練データとテストデータの分類結果の差が非常に激しく、過学習が起こっていると考えられる。訓練データの分類結果は約 80%であった。訓練データの分類は、学習に使ったデータを分類しているため、通常 100%に近い値となるはずであり、この精度 80%という結果は良いものとは言えない。訓練データが 100%に近い値とならない場合、原因としては単純にモデルの表現力が足りないことが考えられる。そこで、モデルの表現力を増加させるため、MLP の中間層の数を変化させて分類実験を行った。中間層の数を 3 層から 5 層、7 層に増加させて分類を行った。増やす中間層については、ノード数 600、活性化関数はシグモイド関数とする。結果を表に示す。

表 5.1: MLP の精度の一覧。

中間層数	3	5	7
精度 (%)	29	31	29

結果は 5 層や 7 層の場合も精度約 30%ほどであり、3 層の場合とほぼ変化がなかった。また、シグモイド関数を全て ReLU 関数に変更して実験をした結果、中間層の数に関わらず、学習が全く進まず、精度約 20%ほどで学習が止まってしまった。これらの結果から、本研究の手法によって収集した情報を MLP を用いて分類するのは難しいと考えられる。MLP での学習が難しい原因としては、MLP ではデータの形状やパターンを維持しつつ学習するのが困難であるからだと考えられる。MLP のような全結合層のみのニューラルネットワークでは、入力データの要素を全て別々のノードとして学習するため、入力データの形状を活かした学習を行うことができない。本研究における加速度データにおいても、物体ごとの加速度データのパターンを学習することができず、結果低い精度になったと考えられる。

5.2 CNN による分類

5.2.1 概要

MLP では良い結果を得ることができなかつたため、CNN を用いた触覚情報の分類を行う。MLP を用いた分類で良い結果が得られなかつた理由としては前節の考察から、MLP ではデータの形状を活かした学習を行うことができないからであると考えられる。CNN では、入力データに畳み込み演算を行うことで、データの形状を維持したまま学習が可能であり、データの形状やパターンが重要な問題の解決に適している。この CNN の特徴から、CNN を用いることで、物体ごとの加速度データのパターンを効果的に学習することが可能であり、MLP よりも良い結果を得ることができると考えられる。

CNN を用いた分類実験は大きく分けて以下の 2 通りの実験を行った。

- 18 種類の触覚情報の分類
- 触察速度の異なる触覚情報の分類

18種類の触覚情報の分類実験では、本研究の手法により収集した触覚情報をCNNを用いてどの程度の精度で分類可能であるかを調査する。触察速度の異なる触覚情報の分類実験では、触察速度を変化させた情報を分類し、触察速度が分類の精度に与える影響を調査する。また、18種類よりも多くの情報を分類可能であるかを調査する。

5.2.2 18種類の触覚情報の分類

実験の条件

MLPの場合と同様に、TWE-Lite-2525Aを指またはペンに装着し、指で10種類、ペンで8種類の物体を計9分間なぞり、著者自ら18種類の加速度データを収集した。18種類の分類に用いた物体を図4.3に示す。



図 5.3: 18種類の触覚情報分類に用いる物体。

分類には、MLPで用いた5種類に加え、新たに5種類追加した計10種類の物体を用いる。sandpaper40, sandpaper80がそれぞれ紙やすりの40番と80番である。stonatile, whitetile, woodtileがそれぞれ素材の違うタイルの一部である。これらの物体を用いて加速度データを収集するが、ペンではsandpaper40, sandpaper80を用いた加速度データの収集は行わない。これは紙やすりをペンでなぞった際、ペン先が削れて正確なデータが収集できない恐れがあるからである。物体をなぞる速さはMLPの場合と同様である。収集したデータは、前章にて述べたCNNによって分類を行う。入力は一全加速度データの中からランダムで取り出した連続200点、出力は入力したデータがどの物体のデータであるかの確率とする。分類の際、このCNNによって作成されるモデルの汎化性能を確認するため、全データを10分割し、10分割交差検証を実施する。交差検証を行い得られた結果の平均を精度とする。学習は全データから抜き出した加速度データを500個学習させるミニバッチ学習を1エポックとし、5000エポック行った。

結果と考察

CNNによる18種類の触覚情報の分類の結果について述べる。収集したデータをもとに10分割交差検証を実施し、得られた混同行列を表4.1に示す。混同行列において、ペンで収集したデータにはsponge-g-penのように”物体名”+”-pen”で表現している。また、CNNでの18種類の分類を行った際の学習の進み具合のグラフを図4.4に示す。

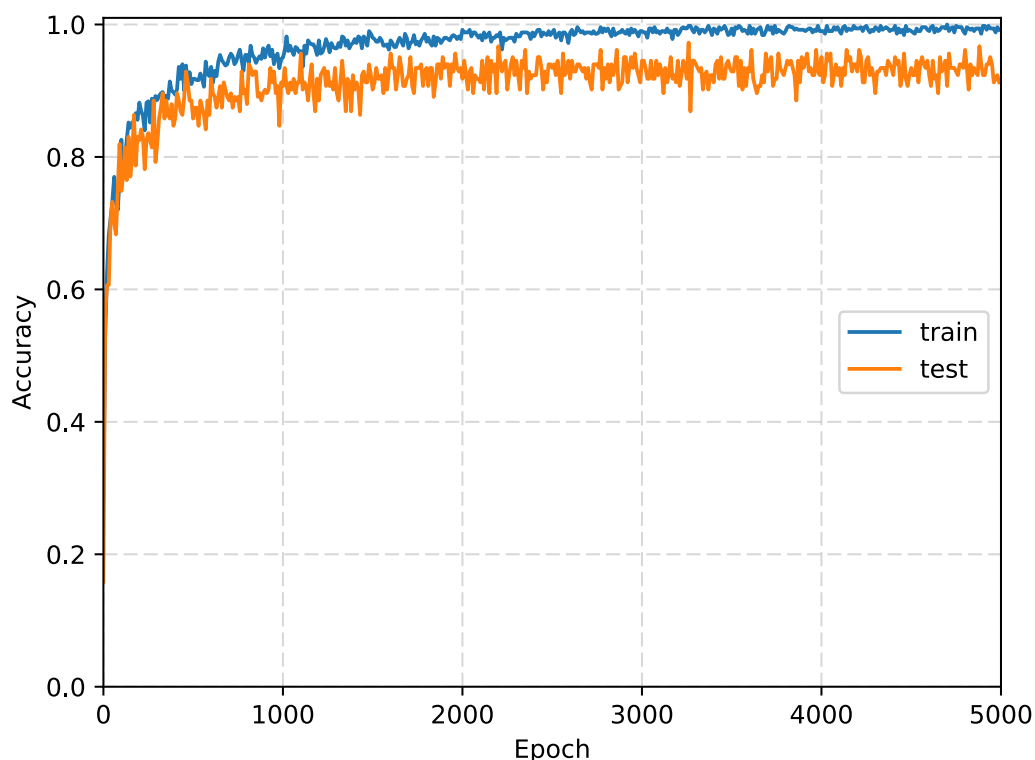


図 5.4: CNN での 18 種の触覚情報分類の学習の進み具合のグラフ。横軸が学習回数、縦軸が精度である。学習 50 回ごとに記録している。

結果から、平均して約 93%での触覚情報の分類に成功したことがわかった。MLP での分類よりも、多い種類のデータを高い精度で分類することに成功した。

表 4.1 の混同行列を見ると、素材の異なる絨毯同士など、ある程度触感が似ている物体同士においても分類に成功していることがわかる。このことから、本研究の触覚情報収集と分類により、細かい触覚情報を分類できる可能性があることがわかった。しかし、sandpaper40 の分類の精度が他の物体の分類結果と比べて低い。荒さの違う紙やすり sandpaper80 や sponge-g-pen, stonetile と主に誤認識している。このことについては、2 通りの原因が考えられる。1 つ目の原因としては物体の質感が似ており、単純に収集したデータが似ている可能性である。2 つ目の原因としては、

表 5.2: 18 種の分類により得られた混同行列. 0-1 の範囲で 1 に近づくほど黒くなる, グレースケールで表現している.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
A: carpet1	0.95	0	0	0	0.02	0	0	0	0.01	0	0	0	0.01	0	0	0	0.01	0
B: carpet1-pen	0	0.95	0	0.03	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0
C: carpet2	0	0	0.89	0	0.04	0	0.01	0.02	0	0.02	0.01	0	0	0	0.01	0	0	0
D: carpet2-pen	0	0.01	0	0.95	0	0.01	0	0	0	0	0	0.01	0	0.01	0	0.01	0	0
E: carpet3	0.03	0	0.04	0	0.91	0	0	0	0	0	0.02	0	0	0	0	0	0	0
F: carpet3-pen	0	0	0.01	0.02	0	0.95	0	0	0	0	0	0	0.01	0	0	0.01	0	0
G: sandpaper40	0	0	0	0	0	0	0.83	0.06	0	0.05	0	0	0.05	0	0.01	0	0	0
H: sandpaper80	0	0	0.02	0	0	0	0.08	0.89	0	0	0	0.01	0	0	0	0	0	0
I: sponge-g	0	0	0	0.01	0	0	0	0	0.92	0.01	0	0.05	0	0	0	0.01	0	0
J: sponge-g-pen	0.02	0	0	0	0	0.01	0.03	0	0	0.92	0.02	0	0	0	0	0	0	0
K: sponge-y	0.01	0	0.03	0	0	0	0.01	0.01	0	0	0.94	0	0	0	0	0	0	0
L: sponge-y-pen	0	0.01	0	0	0	0	0	0	0	0	0	0.97	0.01	0.01	0	0	0	0.01
M: stonatile	0	0	0	0	0	0	0.01	0	0	0	0.02	0.02	0.93	0	0.03	0	0	0
N: stonatile-pen	0	0	0	0	0	0	0	0	0	0	0	0.01	0.01	0.97	0	0.01	0	0
O: whitetile	0.01	0	0	0	0	0	0.01	0.01	0	0	0	0	0.01	0	0.92	0	0.04	0
P: whitetile-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.95	0	0.04
Q: woodtile	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01	0	0.01	0	0.97	0
R: woodtile-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.02	0	0.97

CNN が過学習を起こしている可能性である。sandpaper40 の分類の精度のみ他と比べて低く、逆に woodtile など精度 97% で分類できている物体も存在しており、CNN によって作成したモデルが、woodtile などの物体に過剰に適合している可能性がある。これら 2 つの原因の解決策は、どちらも CNN を調整することで対応が可能であると考えられる。1 つ目のデータが似ていることに関しては、CNN の層を増やすなど、モデルの表現力を上げることによって精度が改善する可能性がある。2 つ目の原因に関しては、重みの初期値を過学習が発生しにくいもの [28,29] に変更することでさらに過学習を抑制できる可能性がある。

学習の進み具合のグラフを見ると、訓練データとテストデータの精度の差が少なく、過学習が抑えられていることがわかる。しかし、差が全く無いわけではなく、訓練データとテストデータの精度には約 7-8% ほどの開きがあり、過学習を起こしている可能性がある。また、訓練データの分類精度は 3000 エポックほどでほぼ 100% に収束しており、それ以降のエポックでは訓練データの分類精度、テストデータの分類精度ともに大きな変化が見られなかった。このことから、学習は十分に進んでおり、これ以上エポックを増やすことで学習回数を増やしても、精度の向上は見込めないことがわかった。

この実験においては 18 種類の触覚情報の分類を行ったが、検証すべき事項は多く存在している。例えば触察時の指の角度や押し付ける力、なぞる速さ等を変化させて触覚情報を収集した場合、結果が異なる可能性がある。分類するデータの種類はこの実験では 18 種類分としているが、さらに分類するデータの種類を増加させた場合、分類の精度は変化すると考えられる。また、CNN の条件を変更した場合、さらに精度を向上させられる可能性がある。

5.2.3 触察速度が異なる触覚情報の分類

実験の条件

18 種類の触覚情報の分類の際と同様に、TWE-Lite-2525A を指やペンに装着し、1 種類の物体につき、計 9 分間のデータを著者自ら収集した。本節の実験では、紙やすりを除いた前節の物体 8 種類に新たに 7 種類を加えた計 15 種類の物体を指またはペンのそれぞれでなぞることにより、30 種類の触覚情報を収集した。この実験において用いた物体を図 4.5 に示す。

用いる物体については、発泡スチロールのような素材の sponge-g、質感がやや異なる素材のタイルの一部の whitetile2 と whitetile3、石のような質感のタイルの一部である stonetile2 と stonetile3、木製のタイルの一部である woodtile2 と woodtile3、これらの 7 種類の物体が新たに追加されている。前節で用いた紙やすりについては、ペンでの触覚情報の収集はペン先が削れる恐れがある他、指での触覚情報の収集の際、怪我をする恐れがあるため、この実験では用いなかった。この実験では、データ収集時の触察の速度に着目し、物体をなぞる速度を変化させてデータを収集した。前述の 30 種類のデータについて、物体を 1 秒間に左端から右端までなぞる速度 (約 100 mm/s)、1 秒間に左右に 1 往復してなぞる速度 (約 200 mm/s)、1 秒間に左右に 2 往復してなぞる速度 (約 400 mm/s) の 3 種類の速度でなぞることで収集を行った。

収集したデータを用いて前章にて述べた CNN によって分類を行う。CNN の入力はいままでと同様に、 3×200 の加速度データ、出力はどの物体であるかの確率である。この実験においては、



図 5.5: 30 種類の触覚情報分類に用いる物体.

まず 30 種類の触覚情報を分類できるか調査するため、約 100 mm/s, 約 200 mm/s, 約 400 mm/s で収集した 30 種類それぞれに対し、分類を行った。次に、触察の速度が分類に与える影響を調査するため、図 4.5 の carpet1, carpet2, carpet3, sponge-g, sponge-y, sponge-b, stonetile1, stonetile2, stonetile3, whitetile1 の 10 種類の物体を約 100 mm/s, 約 200 mm/s, 約 400 mm/s でなぞった場合の計 30 種類 (10 種類の物体 × 3 種類の速度) での分類を行った。この実験においても、18 種類の触覚情報の分類の場合と同様に 10 分割交差検証を行う。

結果と考察

この実験における触覚情報の分類の結果について述べる。収集したデータを基に 10 分割交差検証を行い、得られた混同行列を表 4.3, 表 4.4, 表 4.5 に示す。また、精度の平均を表 4.6 に示す。なお、この実験においても、前節の実験同様、訓練データの分類結果は 100% 付近に収束しており、十分に学習を進めることに成功している。

100 mm/s, 200 mm/s, 400 mm/s の速度で収集した 30 種類のデータをそれぞれ分類した結果、精度は 100 mm/s と 200 mm/s が平均約 88–89%, 400 mm/s が平均 93% という結果となった。さらに、10 種類の物体について 3 種類の速度で収集した 30 種類のデータを分類した結果、精度は平均 90% ほどという結果になった。この結果から、18 種類から 30 種類へと分類する触覚情報を増加させても高い精度で分類が可能であることがわかった。

表 5.3: 100 mm/s でなぞった場合の分類結果の混同行列. 0-1 の範囲で, 1 に近づくほど黒くなる, グレースケールで表現している.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD		
A: carpet1-pen	0.9	0	0.01	0	0.05	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0.01	0	0	0	
B: carpet1	0	0.87	0	0.01	0	0	0.01	0.02	0	0.02	0	0.02	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0.02	0	0	0	0.03	
C: carpet2-pen	0	0	0.94	0	0.03	0	0	0	0.02	0	0	0	0	0	0	0	0.01	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	
D: carpet2	0	0	0	0.93	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0.03	0	0	0	0	0	0	0	0	0	
E: carpet3-pen	0.03	0	0.01	0	0.93	0	0	0	0	0	0	0.02	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
F: carpet3	0	0	0	0.02	0	0.93	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0.01	0	0	0	0	0	
G: sponge-b-pen	0.01	0	0	0	0.01	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	
H: sponge-b	0	0	0	0	0	0.02	0	0.94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0	0.02	
I: sponge-g-pen	0	0	0.01	0	0.01	0	0	0.01	0.82	0	0.06	0	0.01	0.01	0	0	0.04	0	0	0.01	0	0	0	0	0	0	0	0.01	0	0.01	0	
J: sponge-g	0	0	0	0	0	0	0	0	0.89	0	0.05	0	0.02	0	0.01	0	0	0.01	0	0	0	0	0.02	0	0	0	0	0	0	0	0	
K: sponge-y-pen	0.02	0.01	0.03	0	0.04	0	0	0	0.06	0	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L: sponge-y	0	0.01	0	0	0	0	0	0	0	0.01	0.01	0.93	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	
M: stonetile1-pen	0.01	0	0	0	0	0	0	0	0	0	0.02	0	0.88	0	0.03	0	0.01	0	0	0	0	0	0	0	0	0.03	0	0	0	0.02	0	
N: stonetile1	0.01	0.02	0.01	0	0.01	0	0	0	0	0	0.02	0	0.88	0	0.04	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
O: stonetile2-pen	0.01	0	0	0	0	0	0	0	0.02	0	0	0.04	0	0.85	0	0.02	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0.03	0	0
P: stonetile2	0	0.01	0	0	0	0	0	0	0	0.01	0	0.01	0	0.03	0	0.89	0	0.02	0.01	0	0.01	0	0.01	0	0	0	0	0	0	0	0	0
Q: stonetile3-pen	0	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0.01	0	0.92	0	0	0	0.01	0	0	0	0	0	0.01	0	0.02	0	0	0
R: stonetile3	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0.01	0	0.02	0	0.92	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
S: whitetile1-pen	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.96	0	0	0	0.01	0	0.02	0	0	0	0	0	0	
T: whitetile1	0	0	0	0	0	0	0	0	0.02	0	0	0.02	0.01	0	0	0	0	0	0	0.9	0	0	0.02	0.01	0.01	0	0	0	0	0	0	0
U: whitetile2-pen	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.04	0.88	0	0.01	0	0.04	0	0.01	0	0	0.01	0	
V: whitetile2	0	0	0	0.01	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0.94	0	0.01	0	0.01	0	0	0	0	0	0
W: whitetile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0.02	0.02	0	0.91	0	0.01	0	0.01	0	0	0	0	
X: whitetile3	0	0	0	0	0	0	0	0	0.02	0	0	0	0.01	0	0	0	0	0	0.01	0	0	0.03	0	0.9	0	0	0	0	0	0	0.02	
Y: woodtile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.02	0	0.07	0	0.83	0	0.05	0	0	0	0	0	
Z: woodtile1	0	0.03	0	0	0	0.01	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0	0.02	0	0.89	0	0.01	0	0.01	0	
AA: woodtile2-pen	0.01	0	0	0	0	0	0	0	0	0	0	0.04	0	0	0	0	0.01	0	0	0	0	0	0	0	0.05	0	0.84	0	0.05	0	0	
AB: woodtile2	0	0	0	0.01	0	0.05	0	0.05	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.01	0	0	0	0.01	0	0.87	0	0	0
AC: woodtile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.03	0	0	0	0.01	0	0.05	0	0.91	0	0
AD: woodtile3	0.01	0.01	0	0.01	0	0.01	0	0.05	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0.86

表 5.4: 200 mm/s でなぞった場合の分類結果の混同行列. 0-1 の範囲で, 1 に近づくほど黒くなる, グレースケールで表現している.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	
A: carpet1-pen	0.91	0	0.02	0.01	0.03	0	0	0	0.02	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B: carpet1	0.01	0.94	0	0.01	0	0.01	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C: carpet2-pen	0.01	0	0.94	0	0	0	0	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	
D: carpet2	0	0.02	0	0.76	0	0.13	0	0	0	0.02	0	0.04	0	0	0	0	0	0.01	0	0	0	0	0	0.01	0	0	0	0	0	0.02	
E: carpet3-pen	0.05	0	0.03	0	0.83	0.01	0.02	0	0.01	0	0.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
F: carpet3	0	0	0.08	0	0.83	0	0	0	0.01	0	0.06	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0.01	0	0	
G: sponge-b-pen	0	0	0	0	0.01	0.01	0.95	0	0.01	0	0	0	0	0.01	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	
H: sponge-b	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
I: sponge-g-pen	0	0	0.02	0	0	0.02	0	0.92	0	0.02	0.01	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	
J: sponge-g	0	0	0	0.01	0	0.03	0	0	0	0.87	0	0.02	0	0	0	0	0	0.01	0	0	0	0	0.04	0	0	0	0	0.01	0	0.02	
K: sponge-y-pen	0	0	0.01	0	0.06	0	0	0.03	0	0.89	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L: sponge-y	0	0.05	0	0.01	0	0.01	0	0	0.01	0.01	0.86	0	0	0	0	0	0	0	0	0	0	0	0.03	0	0	0	0	0	0	0.01	
M: stonatile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.82	0	0.11	0	0.05	0	0	0	0	0	0	0	0	0	0.01	0	0.01	0	
N: stonatile1	0	0	0	0	0.01	0.02	0	0	0	0.01	0	0.03	0	0.87	0	0.01	0	0.02	0	0	0	0.01	0	0	0	0	0	0	0	0.02	
O: stonatile2-pen	0	0	0	0	0	0	0.01	0	0	0	0	0	0.06	0	0.88	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0
P: stonatile2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.97	0	0.02	0	0	0	0	0	0	0	0	0	0.01	0	0.01	
Q: stonatile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.03	0	0.89	0	0.03	0	0	0	0	0	0	0	0.01	0	0.02	0	
R: stonatile3	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.02	0	0	0	0.85	0	0.06	0	0.01	0	0	0	0.01	0	0.02	0	0.02	
S: whitetile1-pen	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0	0.02	0	0.88	0	0.01	0	0	0.02	0	0.04	0	0	0	0	0	
T: whitetile1	0	0	0	0	0.01	0.01	0	0	0	0	0	0	0	0	0	0	0.01	0	0.87	0	0	0.01	0	0.02	0	0.03	0	0.03	0	0.05	
U: whitetile2-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.96	0	0	0.01	0	0.02	0	0	0	0	0	
V: whitetile2	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0.02	0	0.01	0	0.89	0	0	0.01	0	0	0	0.04	0	
W: whitetile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0.01	0	0.92	0	0.03	0	0	0	0	0	
X: whitetile3	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0.91	0	0.01	0	0.01	0	0.04	
Y: woodtile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0.01	0	0	0	0.94	0	0	0	0	0	
Z: woodtile1	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0.91	0	0.04	0	0.02	
AA: woodtile2-pen	0.01	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0.93	0	0.04	0	
AB: woodtile2	0	0	0	0	0	0.01	0.01	0	0	0.02	0	0	0	0.02	0	0.01	0	0.01	0	0.05	0	0.03	0	0	0	0.04	0	0.7	0	0.09	
AC: woodtile3-pen	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.07	0	0.92	0		
AD: woodtile3	0.01	0.01	0	0.01	0	0	0	0	0	0.04	0	0	0	0	0	0	0.01	0.01	0	0.04	0	0	0	0	0.01	0	0.05	0	0.05	0	0.76

表 5.5: 400 mm/s でなぞった場合の分類結果の混同行列. 0-1 の範囲で, 1 に近づくほど黒くなる, グレースケールで表現している.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD		
A: carpet1-pen	0.94	0	0	0	0.02	0	0.03	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B: carpet1	0	0.92	0	0	0	0.03	0	0	0	0	0	0.01	0	0	0	0.01	0	0	0	0	0	0.01	0	0	0.01	0	0	0	0	0	0	
C: carpet2-pen	0	0	0.92	0	0.05	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	
D: carpet2	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0.01	
E: carpet3-pen	0.02	0	0.02	0	0.95	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
F: carpet3	0	0.04	0	0.03	0	0.9	0	0.01	0	0.01	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
G: sponge-b-pen	0	0	0	0	0	0	0.96	0	0.01	0	0.01	0	0.01	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
H: sponge-b	0	0	0	0	0	0	0	0.92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0.02	0	0.01	
I: sponge-g-pen	0	0	0	0	0	0	0	0	0.98	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
J: sponge-g	0	0	0	0	0	0	0	0	0	0.94	0	0.03	0	0	0	0.01	0	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	
K: sponge-y-pen	0	0	0.01	0	0	0	0	0	0.03	0	0.95	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
L: sponge-y	0	0.04	0	0.03	0	0	0	0	0	0	0	0.91	0	0.01	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	
M: stonatile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0.98	0.01	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	
N: stonatile1	0	0	0	0	0	0.01	0	0	0	0.01	0	0.01	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01	
O: stonatile2-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.86	0.01	0.02	0	0	0.02	0	0.01	0	0	0	0	0	0	0	0.08	0	
P: stonatile2	0	0	0	0.01	0	0.01	0	0	0	0	0	0	0	0	0	0.92	0	0.03	0	0	0	0	0	0	0.01	0	0	0.02	0	0	0	
Q: stonatile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.9	0	0.01	0.01	0	0	0	0	0	0	0	0	0.07	0	
R: stonatile3	0	0	0	0	0	0	0	0	0	0.02	0	0.01	0	0	0	0.01	0	0.94	0	0	0	0	0	0	0	0	0.01	0.02	0	0	0	
S: whitetile1-pen	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0.02	0	0.87	0.01	0	0.02	0	0	0	0.03	0.04	0	0	0.01	0	
T: whitetile1	0	0	0	0	0	0	0.01	0	0	0	0	0	0.01	0	0.03	0	0.01	0	0	0.85	0	0.01	0	0	0	0.03	0	0	0.06	0		
U: whitetile2-pen	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0	0.97	0	0	0	0	0	0	0	0	0	0	
V: whitetile2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0.95	0	0	0	0.01	0	0	0.01	0	0	
W: whitetile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0.01	
X: whitetile3	0	0	0	0	0	0	0	0.01	0	0.02	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0	0.89	0.01	0.02	0.01	0	0	0.01	
Y: woodtile1-pen	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0.01	0	0	0	0	0	0.97	0	0	0	0.01	0	
Z: woodtile1	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0.01	0.01	0	0	0.01	0	0	0	0	0	0	0.94	0	0	0.01	0	
AA: woodtile2-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0.01	0	0	0	0	0	0	0	0	0	0.94	0	0	0	
AB: woodtile2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0.99	0	0	
AC: woodtile3-pen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.08	0.01	0.02	0	0	0.03	0	0.01	0	0	0	0	0	0	0	0.86	0
AD: woodtile3	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.03	0	0	0	0	0	0.94

表 5.6: 分類精度の一覧表. "3 種の速度"は 10 種類の物体について 3 種類の速度で情報を収集し, 30 種類として分類した結果の精度である.

触察速度 (mm/s)	100	200	400	(3 種の速度)
精度 (%)	89	88	93	90

混同行列を見ると、sponge-g-pen (精度約 98%) のように他と比べて高いものや、whitetil1 (精度約 85%) のように低いものが存在しており、やや過学習を起こしている可能性がある。精度が低いものに関して、誤認識する傾向としては物体ごとに異なっているのに加えて、数種類の物体と少しずつ誤認識している場合が多く、誤認識の傾向を捉えることは難しいという結果となった。

精度に関して、以前の 18 種類の分類の場合は平均約 93% であったが、今回の 30 種類の分類では、100 mm/s と 200 mm/s で情報収集した場合が以前の精度を下回っている。これについては、分類する物体の種類を増やしたため、分類の精度が低下したものと考えられる。しかし、今回用いた物体には 3 種類の絨毯など、物体同士において触感がある程度似ている物体が含まれているため、触感が全く違う物体を多数用意し、実験を行った場合、精度が低下しない可能性がある。逆に、触感が似ている物体を多数用意した場合、精度が低下する場合も考えられ、今後検証が必要である。

この実験において、400 mm/s で情報収集した場合の精度は 18 種類の分類の場合と同程度の約 93% であり、100 mm/s や 200 mm/s の場合と異なり、分類する情報の種類が増加しても精度を保ち分類することに成功した。この要因として、400 mm/s の場合、他の 2 つの速度の場合より、情報収集の際の加速度の変化が大きいため、CNN の入力として用いた連続 200 点中の加速度の変化が他の場合よりも激しく、その分様々なパターンを学習できたからと考えられる。この実験では連続 200 点の加速度データを使用して学習を行っているが、使用する加速度データの長さを変化させることで結果が変化する可能性は高く、今後検証を行う必要がある。また、この実験で用いた CNN の設定が 100 mm/s、200 mm/s のデータより 400 mm/s のデータの分類に適していた可能性もある。

表から、10 種類の物体について 3 週類の速度で収集した 30 種類のデータを分類した結果、約 90% であった。また、表の結果とは別に、今回収集した 90 種類のデータを全てまとめて分類した結果、約 88% だった。さらに、同じ物体をなぞった 3 種類の速度が異なるデータに同じラベルをつけて 30 種類分類を行った結果、約 88% であった。これらのことから、同じ物体について異なる速度で収集したデータは別のデータとして分類されていることがわかった。これの要因としてはデータ収集の手法に原因があると考えられる。この実験では、物体を左右に往復してなぞることで情報収集を行っているが、左右に往復する際の方向転換時の加速度がデータに含まれている。なぞる速度によって左右に方向転換する頻度は変化するため、その違いから分類できている可能性がある。また、なぞる速度によってなぞる際の指の振動などが変化的ることにより収集するデータに変化が生じ、その変化を逃さず学習できたから分類できたとも考えられる。これらの考察から、データ収集の手法について検討する必要があることがわかった。

5.3 学習結果に基づく触対象の識別実験

5.3.1 概要

ここでは、あらかじめ作成した機械学習モデルを用いて、収集した情報をその場で分類することで触対象の識別を行い、どの程度の精度で触対象を識別できるか調査する。機械学習モデルの作成には、前節の実験同様、CNN を用いる。

5.3.2 触覚情報収集の条件

これまでの実験同様、TWE-Lite-2525A を用いて物体をなぞった際の加速度情報を著者自ら収集する。収集したデータをあらかじめ作成したモデルによって分類することで、触対象の識別を行う。この実験では図 5.5 に示された 15 種類の物体を用いる。指で 5 秒間物体をなぞり、そのデータをあらかじめ作成した機械学習モデルにその場で入力することで識別を行う。なぞる際の速度は約 400 mm/s 程度とした。

5.3.3 学習に用いるデータによる違い

長時間データによる学習モデル

まず、前節での学習に用いた長時間連続で取得したデータにより、再度学習モデル作成を行い、触対象の識別テストを実施した。しかし、400 mm/s のデータでモデルを作成した場合、収集したデータを分類した精度は平均約 10% であった。各物体それぞれの結果において、2 種から 3 種ほどの別の物体のデータと誤認識している場合が多く、正答率は低い結果となった。この結果の原因はモデル作成時の訓練データにあると考えられる。400 mm/s のデータでモデルを作成すると、400 mm/s で計 9 分間なぞる場合のデータにモデルが特化してしまっている可能性がある。この触対象識別に実験では収集の条件は 5 秒間 400 mm/s でなぞるというものであるため、モデル作成時の訓練データの条件と異なり、分類ができなかったと考えられる。また、データを収集する際、速度は約 400 mm/s、指の角度は約 45 度となるようになぞって収集を行っているが、人が自然に物体をなぞって情報を収集する以上、収集の際に速度や指の角度が微小に変化している可能性がある。400 mm/s のデータは 3 分間同じ指の角度と速度でなぞることを 3 回繰り返したデータであるため、400 mm/s のデータで作成したモデルは、このような収集ごとの微小な変化に弱い可能性がある。

短時間データによる学習モデル

前項の結果をうけ、ある収集の条件にモデルが特化するのを防ぐため、日を分けて収集したデータを用いてモデルを作成することにした。収集する日を分けることで、日ごとに条件の微小な変化が発生し、その変化をモデルに反映できると考えられる。図 5.5 の 15 種類の物体に対し 5 秒間なぞることで 1 個のデータを収集する。1 日 15 個のデータを収集し、それを 6 日間続けて行うことで、1 つの物体につき 90 個のデータを収集した。これらのデータを用いて学習を行うが、これらのデータをどの程度で分類できるか調べるため、モデル作成の際にはこれまで同様全データを 8:2 に分割し、前者を訓練データ、後者をテストデータとし、モデルのテストを行った。CNN の入力、出力はこれまでと同様である。CNN の構成もこれまでと同様のものを用いた。学習は約 400 mm/s で収集した全データから抜き出した加速度データを 250 個学習させるのを 1 エポックとし、5000 エポック行った。

分類を行う際、収集した 5 秒間のデータから連続 200 点を 10 個抜き出すことで、1 回の収集ごとに 10 回の分類を行い、実験の効率化を図る。1 回データを収集し、分類することを 1 セットとし、1 物体あたり 20 セット行う。

5.3.4 結果と考察

収集した情報を用いて機械学習モデルを作成し、触対象識別を行った結果を述べる。ここでは、収集した情報をモデルに入力した際の分類結果を識別結果として扱う。例えばある情報をモデルに入力し、情報が図 5.5 の carpet1 と分類された場合、識別結果は carpet1 として扱う。収集した 5 秒間のデータから 10 個のデータを抜き出し、分類した精度は平均で表 5.7 のようになった。また、全データを分割し、モデルの性能をテストした結果得られた混同行列を表 5.8 に示す。なお、この実験においても、これまでの実験同様、訓練データの分類結果は 100% 付近に収束しており、十分に学習を進めることに成功している。

触対象識別実験を行った結果、精度は全体平均で約 46% という結果となった。この実験では、収集した 5 秒間のデータから 10 個のデータを抽出しているが、10 個の中で最も分類された頻度が高かったものを全体の分類結果として扱った場合、精度は全体平均で約 71% だった。この最頻のものを抽出する操作のイメージ図を図 5.6 に示す。なお、分類にかかった時間は 5 秒間のデータを 1 つ分類するのに約 0.3 秒であった。また、これまでの実験と同じく、全データを訓練データとテストデータに分割し、分類のテストを行った結果、平均約 77% という結果となった。これらの結果から、現段階では本研究の手法による触対象の識別を実用化することは難しいことがわかった。

表 5.7 に示した分類の精度を見ると、分類の精度は最大のものでも約 68% であり、最低のものは約 25% であった。全体平均で考えても精度は約 46% であり、改善が必要であることがわかった。これに対し、得られた混同行列の精度の平均は約 77% であり、触対象識別を行った際と精度との差がある。これについては、前述の通り、収集ごとに物体をなぞる条件が微小に変化しているからだと考えられる。触対象識別の実験での触覚情報収集の際、の指のなぞる速度や押し付け力などが以前より微小に変化したことにより、結果として収集したデータは作成したモデルでは対応できないものになってしまった可能性がある。微小な変化をモデルに取り込めるよう、日を分けて収集したデータを学習に用いたが、まだ十分にモデルに取り込めていないことがわかった。

低い精度をカバーする方法として、分類結果を用いて何らかの操作を行うことが考えられる。この実験での分類結果は、1 回の収集で得られる 5 秒間のデータから 10 個のデータを抽出してそれぞれ分類を行い、その精度の平均としている。この分類結果に行う操作として、例えば 10 個データの分類結果の中から最も分類された数が多かったものを抽出し、全体の 5 秒間のデータの分類結果として扱うようにすることが考えられる (図 5.6)。このようにすることで 10 個のデータ 1 つ 1 つの分類精度が低くても、触った物体を認識できる可能性がある。単純に計算すると、10 個のデータの分類結果の中で 6 回正解していれば、正しい物体を認識できることになる。

この操作を行った場合の分類精度は表 5.7 において“最頻抽出精度平均”として示されており、全体の平均では約 71% となった。分類結果の平均をそのまま使用するよりは良い結果が得られた。しかし、10 個のデータの分類精度の平均が 50% 以下の場合、大きな精度の改善は見られなかった。これは精度平均 50% 以下の物体の分類結果では、10 個のデータの分類結果の中で半分以上不正解であるため、正しい分類結果が最頻の結果にならない可能性が高いからであると考えられる。実際の分類結果を見ると、最も多く分類されたものが複数存在したり、意図する物体とは別の物体が最も多く分類されている状態であった。このような状態のため、最頻のものを抽出しても効果が無かったと考えられる。一方で分類精度平均が 50% を上回っているものについては大きな改善

表 5.7: 触対象識別の精度の一覧. “精度平均”は 20 セット分類を行った精度の平均である. “最頻抽出精度平均”は 5 秒間のデータの中から抜き出した 10 個の分類結果の中で最も分類された数が多かったものを抽出し, それを 5 秒間のデータの分類結果とした場合の精度の平均である.

	精度平均	最頻抽出精度平均
carpet1	0.625	0.95
carpet2	0.685	1
carpet3	0.55	0.8
sponge-b	0.435	0.65
sponge-g	0.41	0.7
sponge-y	0.48	0.8
stonetile1	0.315	0.5
stonetile2	0.53	0.8
stonetile3	0.255	0.3
whitetile1	0.5	0.85
whitetile2	0.565	0.85
whitetile3	0.615	0.95
woodtile1	0.325	0.35
woodtile2	0.3	0.4
woodtile3	0.435	0.75
全体平均	0.468	0.71

表 5.8: 6 日間に分けて収集したデータを分類した結果から得られた混同行列. 0-1 の範囲で, 1 に近づくほど黒くなる, グレースケールで表現している.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A: carpet1	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0
B: carpet2	0.07	0.73	0	0	0.13	0	0	0	0.07	0	0	0	0	0	0
C: carpet3	0.04	0.08	0.79	0	0	0	0	0	0	0	0	0.04	0.04	0	0
D: sponge-b	0	0	0	0.86	0	0	0.07	0	0.07	0	0	0	0	0	0
E: sponge-g	0	0	0	0	0.94	0	0.06	0	0	0	0	0	0	0	0
F: sponge-y	0	0	0	0	0.03	0.86	0	0.03	0	0	0	0.03	0.03	0	0
G: stonetile1	0.06	0	0	0	0	0	0.82	0.06	0.06	0	0	0	0	0	0
H: stonetile2	0	0	0	0	0	0	0	0.73	0.09	0	0	0	0	0.18	0
I: stonetile3	0	0	0	0	0	0.11	0	0.16	0.47	0	0.11	0	0	0.11	0.05
J: whitetile1	0	0	0.06	0	0	0	0	0	0.06	0.75	0	0	0	0	0.12
K: whitetile2	0	0	0	0	0.06	0	0	0.06	0.06	0.06	0.71	0	0.06	0	0
L: whitetile3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
M: woodtile1	0	0	0	0	0	0	0	0	0	0.12	0	0.06	0.75	0.06	0
N: woodtile2	0	0	0	0	0	0	0.04	0.04	0	0.04	0.04	0	0.08	0.62	0.12
O: woodtile3	0	0	0	0	0	0	0.08	0	0.08	0.08	0	0.17	0	0	0.58

[carpet1, carpet2, carpet1, carpet1, carpet2]
 [carpet3, carpet1, carpet1, carpet1, carpet1]



carpet1

図 5.6: 分類結果の最頻値抽出のイメージ図. 例えば 5 秒間収集したデータの中から抽出した 10 個のデータの分類結果が図の上段のようになった場合, 最も出現頻度の高い carpet1 を全体の分類結果として用いる.

が見られた。50%を超えているものに対して最頻のものを抽出した結果、全て80%以上の精度となった。また、60%を超えているものについては全て90%以上の精度となった。この結果については、10個のデータの分類結果の中で平均すると過半数以上が正解しているため、最頻の分類結果を抽出した場合、必然的に正解の分類結果を抽出することができたと考えられる。このことから、この実験の手法を用いた場合、様々な条件を調整し、CNNの分類結果が60%を超えることができれば、触対象を高い精度で識別することが可能であると考えられる。

表5.8に示された混同行列を見ると、stonetile3の分類結果が約47%と低く、逆にwhitetile3の分類結果が100%と不自然に高い。このことから過学習が起こっていることがわかる。混同行列から全体の精度の平均を計算すると約77%であり、これまでの実験の結果と比べると低い。これらの結果から考察すると、この実験におけるデータを分類するには現時点でのCNNの構成では力不足である可能性が考えられる。ただし、条件によっては、CNNの構成を変えずとも分類精度が改善する可能性はある。この実験では5秒間なぞったデータを1物体につき90個収集して学習に使用しているが、単純に学習に使うデータ量を増やすことによりモデルの性能を上昇させることができる可能性がある。また、この実験では前述の通り5秒間のデータを分類する操作を20セット行ったが、セット数が少なかったために結果に偏りが出ている可能性も考えられる。これらの考察をもとに今後CNNの構成や実験のやり方について見直す必要がある。

第6章 議論

本章では実験の結果を受け、本研究の触覚情報収集と触覚情報分類のシステムについて議論する。また、収集と分類のシステムそれぞれに対する今後の課題についても検討する。

6.1 触覚情報の収集について

触覚情報の収集には前述の TWE-Lite-2525A を用いている。主な利点として、無線である点、開発環境が整備されているため実装が容易である点、小型である点が挙げられる。さらに受信機である MONOSTICK には USB 接続端子が搭載されているため、様々なデバイスで収集した触覚情報を受信することができる。嵯峨ら [12] はこのことを応用し、MONOSTICK を接続したスマートフォンを中継機として、クラウドサーバに触覚情報を送信する手法について言及している。しかし、現段階での実装では 330 Hz までしか加速度情報を収集することができない。人の触覚は 0-1 KHz 程度までの振動を検知することが可能であり、330 Hz での収集では十分に情報が収集できていない可能性がある。また、標本化定理 [30] について考える必要がある。標本化定理では、得られたアナログ信号をデジタル信号に変換する際、元の周波数成分の半分の値の周波数までしか忠実に再現できないとされている。本研究の手法では 330 Hz の情報を収集しているが、標本化定理を考えると、正確に収集できているのは 165 Hz までの情報に限られている可能性がある。今後、正確な情報を収集するため、現在のものより高周波を計測できる加速度センサを用いた専用の機器を作成することを検討する必要がある。

前述の実験では触覚情報の収集は指やペンにマイコンを取り付け、物体表面を左右に往復させて情報を収集している。現在触覚情報収集の際はこの方法を用いているが、実験の考察でも述べた通り、なぞる方向を反転させた際の加速度の変化もデータとして収集してしまっており、分類へ影響を与える可能性がある。今後、指などで物体をなぞって行う触覚情報収集では、例えば物体表面を円を描くようになぞるなど、なぞる方向を変化させた際の影響が少なくなるように情報を収集する必要がある。さらに、実験において、日ごとに触覚情報収集の条件が変化している可能性があることがわかったため、厳密に調査を行う必要がある。また、現段階では指やペンで物体をなぞった際の情報収集と分類のみしか行っておらず、他の条件で触覚情報の収集と分類を行うことができるかは不明である。このことを考え、今後は、例えばペンで文字を書いた場合や指でキーボードを叩いた場合などの実生活で行う触行動について触覚情報の収集と分類の実験を行う予定である。

6.2 触覚情報の分類について

触覚情報の分類には、MLP または CNN を用いた。MLP で 5 種類のデータを分類した際、精度は約 30% であった。MLP での分類が低い精度となった原因としては、MLP ではデータの形状を維持した学習ができないことが原因と考えられる。この問題に対応し、データの形状を維持したまま学習することができる CNN を用いた分類も行った。CNN を用いた分類では、3 分間なぞったデータを分類した際、最大 30 種類のデータを約 90% で分類を行うことに成功した。しかし、6 日間に分けて収集したデータを分類した際には 3 分間なぞったデータを分類した場合より精度が落ち、その場で収集したデータを分類した際にはあまり良い結果が得られなかった。良い結果が得られなかった理由としては、実験の考察でも述べた通り、作成したモデルの表現力が足りないからであると考えられる。他には、モデル作成の際の学習データが少ないという可能性も考えられる。実験では 1 物体につき 90 個のデータを用いてモデルを作成したが、モデル作成に用いるデータ量を増加させることで精度を改善できる可能性がある。

CNN での分類にかかった時間は、実験の結果から、連続 200 点の加速度データを 10 個分類するのに約 0.3 秒ほどであった。また、モデルの作成にかかった時間は、本研究での CNN の構成では約 15 分ほどであった。モデル作成に時間がある程度かかるものの、あらかじめモデルを作成しておけば情報の分類を短い時間で行うことが可能であることがわかった。

また、実験の結果から、触察速度が違うデータは、触察する物体が同じでも違うデータとして分類されることがわかった。インタフェースへの応用を考えると、どのような速度で物体をなぞった場合でも物体を識別できるのが望ましい。このため、触察速度が違うデータでもなぞった物体が同じであれば同じ物体をなぞったものとして認識されるように分類の手法を工夫する必要がある。

学習データ量の増加や触察速度の考慮の他に、用いる機械学習手法を調整または変化させることで分類の精度を向上させることができると考えられる。現段階では主に CNN を用いて分類を行っている。CNN の構成について、畳み込み層を増加させるなどの調整を行った場合、精度が向上する可能性がある。また、時系列データを扱うことに優れているとされる再帰型ニューラルネットワーク (Recurrent Neural Network, RNN) ベースの機械学習手法に変更することで、精度を向上させることができる可能性がある。

第7章 結論

本研究では、触覚情報収集と触覚情報分類の新たな応用として、人の触行動をシステムに理解させることを利用したインタフェースを提案した。インタフェース開発の第一歩として、人と環境の間で発生する触覚情報の収集と分類を実現するシステムを実装した。触覚情報の収集には市販の無線型加速度センサを、触覚情報の分類にはMLPまたはCNNによる機械学習を使用した。実装したシステムを用いて、実際に触覚情報を収集し、機械学習を用いて分類する実験を行い、得られた結果に対する考察を行った。

分類実験の結果、本研究の実装により収集した情報をCNNを用いた機械学習によって、30種類の触覚情報を約90%で分類することに成功した。さらに、90種類の触覚情報を約88%で分類することに成功した。これらの分類結果から、触察速度が分類に与える影響を調査したところ、触察速度が異なる情報を分類した場合、触察した対象が同じでも別の情報として分類されることがわかった。また、実際に指でなぞった物体を分類結果から識別する実験を行った結果、精度は約46%と低く、触覚情報を収集する際の条件や、CNNの構成などを見直す必要があることがわかった。

今後は、実験の考察やシステムに対する議論をもとに、CNNの構成や実験の条件の調整を行い、触覚情報の分類による触対象識別の精度を改善する。機械学習の手法についても着目し、RNNを利用した機械学習を実装し、CNNを用いた場合との精度を比較する。

触対象を識別する精度の改善に成功した場合、触対象の識別だけでなく、触り方を分類する実験を行う。本論文では触察速度を変化させる実験を行ったが、今後は指にセンサを取り付けた際のなぞる方向の違いなど、本論文とは異なる条件での実験を行い、分類できるか調査する。また、提案インタフェースを用いたアプリケーションを作成する。具体的には、第3章で示した利用例を実際に実装する。さらに複数のマイコンから同時に情報収集を行うことができるように実装し、提案手法を、1本の指やペンなどの物体だけでなく、複数の物体においてインタフェースが使用できるようにする。

また、現段階では触覚情報の収集は著者1名にて行っており、結果が偏っている可能性がある。結果の偏りの可能性を減らすため、今後は多くの被験者を用いた触覚情報収集と分類の実験を行う。

謝辞

本論文の執筆に際し、指導教員である高橋伸准教授と志築文太郎准教授には多大なご助力を賜り、深く感謝致します。Simona Vasilache 助教には、国際論文投稿時の英文作成の際に多くのご指摘を頂き、感謝致します。また、熊本大学の嵯峨智准教授には、研究に対する細やかなご指摘や、論文の添削など熱心なご指導を賜りました。研究生生活においても、様々なご指導を賜り、ここに深い感謝の意を表します。先生方だけでなく、インタラクティブプログラミング研究室の皆様には研究においての様々なご支援を頂きました。特に UBIQUITOUS チームの皆様には、論文執筆の際の多くの助言や添削等、様々なご支援を頂きました。深く感謝致します。最後に、私の研究生生活を支えてくださった家族や友人の皆様、研究においてお世話になった全ての方々に深く感謝致します。

参考文献

- [1] Matti Strese, Yannik Boeck, and Eckehard Steinbach. Content-based Surface Material Retrieval. In *2017 IEEE World Haptics Conference (WHC), Fürstenfeldbruck (Munich), Germany*, pp. 352–357. IEEE, 2017.
- [2] Alex Burka, Siyao Hu, Stuart Helgeson, Shweta Krishnan, Yang Gao, Lisa Anne Hendricks, Trevor Darrell, and Katherine J. Kuchenbecker. Design and implementation of a visuo-haptic data acquisition system for robotic learning of surface properties. In *Proc. IEEE Haptics Symposium*, pp. 350–352, April 2016.
- [3] Arsen Abdulali and Seokhee Jeon. Data-Driven Modeling of Anisotropic Haptic Textures: Data Segmentation and Interpolation. In *Haptics: Perception, Devices, Control, and Applications: 10th International Conference, EuroHaptics 2016, London, UK*, pp. 228–239. Springer International Publishing, 2016.
- [4] Nawid Jamali and Claude Sammut. Material classification by tactile sensing using surface textures. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2336–2341. IEEE, 2010.
- [5] Matti Strese, Clemens Schuwerk, and Eckehard Steinbach. Surface classification using acceleration signals recorded during human freehand movement. In *World Haptics Conference (WHC), 2015 IEEE*, pp. 214–219. IEEE, 2015.
- [6] Haitian Zheng, Lu Fang, Mengqi Ji, Matti Strese, Yigitcan Özer, and Eckehard Steinbach. Deep learning for surface material classification using haptic and visual information. *IEEE Transactions on Multimedia*, Vol. 18, No. 12, pp. 2407–2416, 2016.
- [7] Massimo Regoli, Nawid Jamali, Giorgio Metta, and Lorenzo Natale. Controlled tactile exploration and haptic object recognition. In *Advanced Robotics (ICAR), 2017 18th International Conference on*, pp. 47–54. IEEE, 2017.
- [8] Yang Gao, Lisa Anne Hendricks, Katherine J. Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. *abs/1511.06065*, 2015.
- [9] Jivko Sinapov, Vladimir Sukhoy, Ritika Sahai, and Alexander Stoytchev. Vibrotactile recognition and categorization of surfaces by a humanoid robot. *IEEE Transactions on Robotics*, Vol. 27, No. 3, pp. 488–497, 2011.

- [10] Fok Hing Chi Tivive and Abdesselam Bouzerdoum. Texture classification using convolutional neural networks. In *TENCON 2006. 2006 IEEE Region 10 Conference*, pp. 1–4. IEEE, 2006.
- [11] Manik Varma and Andrew Zisserman. Texture classification: Are filter banks necessary? In *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, Vol. 2, pp. II–691. IEEE, 2003.
- [12] 嵯峨智, 中川真史, 小野智義, 潘振雷, 張嘉袁. Zigbee を利用した日常の触覚情報収集. 電気学会研究会資料 (知覚情報研究会・力触覚提示デバイス), Vol. 2017, No. 52, pp. 11–14, 2017.
- [13] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, No. 5, pp. 826–834, 1983.
- [14] Vladimir Sukhoy, Ritika Sahai, Jivko Sinapov, and Alexander Stoytchev. Vibrotactile recognition of surface textures by a humanoid robot. In *IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 145–150. IEEE, 1989.
- [15] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, Vol. 24, pp. 774–780, 1963.
- [16] A. Gómez Eguíluz, I. Rañó, Sonya A. Coleman, and T. Martin McGinnity. A multi-modal approach to continuous material identification through tactile sensing. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4912–4917. IEEE, 2016.
- [17] 花光宣尚, 南澤孝太, 館. Convolutional neural network を用いた触感検索システム. 第 19 回日本バーチャルリアリティ学会大会論文集, pp. 300–301, 2014.
- [18] David S. Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [19] Teng Han, David Ahlström, Xing-Dong Yang, Ahmad Byagowi, and Pourang Irani. Exploring design factors for transforming passive vibration signals into smartwear interactions. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction, NordiCHI’16*, pp. 35:1–35:10. ACM, 2016.
- [20] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 147–156. ACM, 2012.
- [21] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proceedings of Machine Learning Research*, pp. 315–323. PMLR, 2011.

- [22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.
- [23] Stephen José Hanson and Lorien Y Pratt. Comparing biases for minimal network construction with back-propagation. In *Advances in neural information processing systems*, pp. 177–185, 1989.
- [24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [25] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014.
- [26] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *abs/1502.03167*, 2015.
- [27] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, Vol. 15, No. 1, pp. 1929–1958, 2014.
- [28] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, 2010.
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034. IEEE, 2015.
- [30] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, Vol. 27, No. 3, pp. 379–423, 1948.