2020
年 度

ピッキングと鍵開けにみる類似行動を機械学習により識別する研究

白 石 将 貴

宇田
研究室

[博士学位論文]

ピッキングと鍵開けにみる類似行動を機械学習により識別する研究

（指 導 教 員）　宇 田 隆 哉

コンピュータサイエンス専攻 宇田研究室

学籍番号 D2120004

白 石 将 貴

［ 2020 年度］

東京工科大学大学院バイオ・情報メディア研究科

博士学位論文

論文題目

ピッキングと鍵開けにみる類似行動を
機械学習により識別する研究

指導教員

宇田 隆哉

提出日　2021 年　3 月　31 日

提出者

専　攻	コンピュータサイエンス専攻
学籍番号	Ｄ２１２０００４
氏　名	白石 将貴

博士学位論文概要

論 文 題 目	ピッキングと鍵開けにみる類似行動を 機械学習により識別する研究
執 筆 者	白 石 将 貴
指 導 教 員	宇 田 隆 哉 講師

本論文は，家人による解錠操作と，ピッキングによる解錠操作を区別することを目的とするものである．住宅のセキュリティを考える際，出入り口以外は一般的に人が出入りしないため，簡易なセンサによって不正な侵入を検知することが可能であるが，出入り口は家人も出入りするため，単純なセンサによる検知はできない．監視カメラの設置は一般的であるが，侵入があったことを後日知ることはできても，侵入時にその侵入を検知するには有人監視を常時行うしかない．そこで，本論文では，Kinect を用いて不正な侵入を自動的に検知するシステムの提案を行う．なお，Kinect により骨格座標を取得している．Kinect を用いた動作識別の研究は既に行われており，不正な侵入のなかでも，通常の出入りとは異なる，ドアを破壊するような行為の検知は既存技術でも対応可能である．そこで，本論文では家人による解錠操作と，ピッキングによる不正な解錠操作を区別する点に重点を置いた．本研究では機械学習を用いている．大量にデータを集めて学習しさえすれば，従来の単純な機械学習でも家人による解錠操作とピッキングを区別できる可能性はあるが，学習にとても時間が掛かりコストが増加する．そこで，提案手法では，少人数の被験者のデータから，体格による分布が均一になるようにサンプルを増加させ，少ないデータで効果的に学習が行えるように工夫した．Kinect を使用することにより，撮影された画像を保存する必要がないため，被撮影者のプライバシも保護される．実験の結果，ピッキングを平均 76%，被験者によっては 94%の精度で区別ができた．なお，右腕の指先から肩までの骨格座標を取得した場合には，平均 73%，被験者によっては 97%の精度で区別ができ，ピッキングおよび鍵開け動作を行っている右腕に特徴があらわれていた．

Abstract

Title	Research for Classification of Similar Actions by Machine Learning with Examples of Picking and Key-Unlocking
Author	M a s a k i S h i r a i s h i
Supervisor	Senior Assistant Professor R y u y a U d a

The objective of this paper is to distinguish opening a door by the key from opening it by picking. In terms of security of a house, windows can be monitored by small sensors since no one goes through windows in usual. On the other hand, doors cannot be monitored by the sensors since not only thieves but also residents go through the doors. CCTV (closed-circuit television) camera is one of the solutions. However, invasion can be detected not during but after the invasion, or the movie by the camera must be always watched by a person. Therefore, in this paper, I propose a system which automatically detects invasion by Kinect. As well, action coordinates are acquired by Kinect. Of course, there are some researches in which Kinect is used to distinguish human actions, and big actions such as breaking doors can be detected with methods in the researches. In contrast, we put the stress on to distinguish a small difference such as difference between opening a door by the key and opening it by picking. We used machine learning in this paper. If big data of opening a door can be collected, the small difference would be able to be distinguished by usual way of machine learning. However, it takes too much time for learning and the cost increases. Therefore, we made an idea which can learn effectively from small samples of few examinees by increasing samples mathematically. Using Kinect also protects the privacy of people in the movie since only coordinates of joints of people are stored. The average of F-measure of detecting picking was 76% and 94% by an examinee in the results of our experiments. By the way, the average of F-measure of right arm, of detecting picking was 73% and 97% by an examinee in the results of our experiments. And In addition, the right arm performing the distinguish opening a door by the key and opening it by picking was characterized.

博士学位論文要旨

論 文 題 目	ピッキングと鍵開けにみる類似行動を 機械学習により識別する研究
執 筆 者	白 石 将 貴
指 導 教 員	宇 田 隆 哉 講師

本論文は，家人による解錠操作と，ピッキングによる解錠操作を区別することを目的とするものである．

住宅のセキュリティを考える際，出入り口以外は一般的に人が出入りしないため，簡易なセンサによって不正な侵入を検知することが可能であるが，出入り口は家人も出入りするため，単純なセンサによる検知はできない．また，その他にカードキーを使用する方法もあるが，バッテリーが切れた場合や通電しなくなった場合には，電子ロックが解除されるように設計されており信頼性が低い．したがって，ドアを開錠する最善の方法は，一般的な物理キーを使用することである．監視カメラの設置は一般的であるが，侵入があったことを後日知ることはできても，侵入時にその侵入を検知するには有人監視を常時行うしかない．

そこで，本論文では，Kinect を用いて不正な侵入を自動的に検知するシステムの提案を行う．Kinect を用いた動作識別の研究は既に行われており，不正な侵入のなかでも，通常の出入りとは異なる，ドアを破壊するような行為の検知は既存技術でも対応可能である．そこで，本論文では家人による解錠操作と，ピッキングによる不正な解錠操作という，動作の違いが少ないものを区別する点に重点を置いた．

本研究により，類似行動を区別することで，有人監視による警備員の負荷が軽減されると考えられる．具体的には，一戸建て住宅における年間のピッキング侵入件数は 11 件，年間移動回数より日本国内における鍵開け回数は 1.25×10^{11} 回となる．よって，ピッキングと鍵開けの比率は，$11 : 1.3 \times 10^{11}$ となり，正しく鍵開けを検出できるとすると，警備員の負荷が大幅に軽減されることがいえる．

本研究における被験者の骨格座標を取得する際，Kinect の高さは 185cm とし，胸の高さに近づくようにした．Kinect と被験者の距離は 300cm とし，被験者の全身が中央に収まるようにした．また，Kinect の設置位置は被験者の真横と右斜め後ろで扉に対して 30 度とした．これは，正面もしくは真後ろからが望ましいが，正面にはドアがあり，真後ろだと手元がみえなくなるためである．次に，被験者の立ち方によりばらつきが出ないように被験者 10 名の立ち位置を測定し，その平均とした．かかとの開き幅を 14cm，ドアからかかとの距離を 56cm とした．鍵開けとピッキングの動作もばらつきが出ないように動作を統一した．ピッキングの場合には手で棒状のものをもち鍵穴付近を上下に動かし，鍵開けの場合には右手で鍵穴に鍵を入れて開け閉めを行う．いずれの動作の場合にも左手はドアノブに手を添えておくこととした．この動作をサンプル取得中連続して行うこととした．なお，連続してデータを 25 サンプル取得する．1 サンプルは 40 フレームから構成されており，2 秒程で取得している．また，1 フレームは各骨格座標の X 座標，Y 座標，Z 座標の値から成っており，立体的に行動を捉えることが可能となっている．

本研究では CNN と SVM を用いている．CNN のネットワークは 11 層とし，畳込みを 2 回行っている．そして二値分類を行っている．一方，SVM は 3 層とし，分類器には線形 SVC を用いて二値分類を行っている．本論文での精度評価は，Accuracy の値ではなく，F 値としている．なお，CNN による学習の際に必要なバッチサイズやエポック数を事前実験として決定する．その際に，バッチサイズは 32 から倍々に増加させ，エポック数は 50 から 50 ずつ増加させ，F 値が 90%超かつ標準偏差が 5%未満のものを使用するものとする．これにより，10 交差検証により精度評価を行い，5 回繰り返し平均精度をとることとする．事前実験での精度評価では，全被験者データが訓練用サンプルとテスト用サンプルの両方に含まれている．一方，本実験の精度評価ではテスト用サンプルに 1 名の被験者データを入れ，残りの被験者データを訓練用サンプルとし，被験者を入れ替えながら 10 回ずつ行っている．

大量にデータを集めて学習しさえすれば，従来の単純な機械学習でも家人による解錠操作とピッキングを区別できる可能性はあるが，学習にとても時間が掛かりコストも増加する．そこで，提案手法では，少人数の被験者のデータから，体格による分布が均一になるようにサンプルを増加させ，少ないデータで効果的に学習が行えるように工夫した．具体的には，Kinect から取得した骨格座標サンプルを用いて被験者の身長や腕の長さなどの個人特徴量をなくすため，同じ身長となるようにデータ整形を行った．その後，仮想的に体格の異なる被験者のサンプル生成や，線形補間により異なる行動周期のサンプル生成を行った．また，動いている骨格座標が分類に意味があるのか，動いている骨格座標が隠れた場合に分類可能であるのかを分析するために骨格座標の一部を 0 に置き換え，行動識別を行った．

まず，全骨格座標取得における被験者 10 名での評価を行った．CNN では，被験者全体の平均 F 値は 56%と分類は行えていないが，40%台の被験者から 80%近くの被験者までいる．しかし，ほとんどの被験者の分類精度は低い．SVM では，被験者の平均 F 値は 54%と低く，20%台の被験者から 70%台の被験者までいる．この原因として，訓練用サンプルの被験者数が少なく，テスト用サンプルの被験者と一致するサンプルが存在しなかったことがいえる．

そこで，次に，被験者 10 名のデータを用いて，仮想的に体格の異なる被験者を作り出した．この方法により仮想サンプルを増加させ，精度評価を行った．CNN では，被験者全体の平均 F 値は 59%と分類は行えていない．一番低い F 値は 41%となっているが，被験者によっては 82%，91%と分類できている．SVM では，被験者全体の平均 F 値は 54%と分類は行えていない．一番低い F 値は 22%となっているが，被験者によっては 80%，81%と分類できている．これは，55 名程度の被験者を集めて実験を行えば，一部の高精度を示した被験者に関して精度を向上させることができるが，工夫することで 10 名の被験者でも精度を向上させることがいえる．そこで，鍵開けとピッキングの行動周期に着目し，類似周期のサンプルを訓練させることで，上手く分類できるのではないかと考えた．

そこで，線形補間により鍵開けとピッキングの行動周期が異なるサンプルを作成し，精度評価を行った．CNN では，被験者全体の平均 F 値は 57%と分類は行えていない．一番低い F 値は 35%となっているが，被験者によっては 83%，94%と分類できている．SVM では，被験者全体の平均 F 値は 55%と分類は行えていない．一番低い F 値は 23%となっているが，被験者によっては 81%，85%と分類できている．

次に，被験者数を 10 名から 20 名に増加させ，精度評価を行った．CNN では，被験者全体の平均 F 値は 52%と分類は行えていない．SVM では，被験者全体の平均 F 値は 76%となっている．また 70%台が 4 名，80%台が 2 名，90%が 2 名となっており，被験者数を増加させれば精度が高くなることがわかる．ここで，CNN の精度評価がうまくいかなかったのは，被験者が行動している位置がずれていると考えた．

そこで，被験者の骨格座標の値が近くなるように座標をスライドさせ，精度評価を行った．CNN では，被験者全体の平均 F 値は 51%と分類は行えていない．SVM では，被験者全体の平均 F 値は 73%となっている．また 80%台が 2 名，90%が 3 名となっている．よって，骨格座標をスライドしても CNN には影響がみられなかった．

次に，右腕の指先から肩までの骨格座標を取得した場合に右肩に特徴があらわれているか評価を行った．CNN では，被験者全体の平均 F 値は 48%と分類は行えていない．SVM では，被験者全体の平均 F 値は 73%となっている．また 70%台が 1 名，80%台が 4 名，90%が 1 名となっている．よって，右腕の指先から肩までの骨格座標に動作の特徴があらわれていることがわかった．

次に，右腕の指先から肩まで以外の骨格座標を取得した場合の評価を行った．CNN では，被験者全体の平均 F 値は 48%と分類は行えていない．SVM では，被験者全体の平均 F 値は 45%と分類は行えていない．

次に，右腕の指先から肘まで以外の骨格座標を取得した場合の評価を行った．CNN では，被験者全体の平均 F 値は 48%と分類は行えていない．SVM では，被験者全体の平均 F 値は 50%と分類は行えていない．

次に，右腕の指先から手首まで以外の骨格座標を取得した場合の評価を行った．CNN では，被験者全体の平均 F 値は 50%と分類は行えていない．SVM では，被験者全体の平均 F 値は 46%と分類は行えていないが，被験者によっては 71%，81%となっており，分類できている．よって，右腕の肘に特徴があらわれていることがわかった．

最後に，Wilcoxon の符合付順位和検定を用いて，取得する骨格座標を変更した際に有意かどうかを確認した．まず，元データ以外の 7 つの代表値が，元データの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．その結果，元データを 0.7 倍や 1.3 倍にした線形補間データについて帰無仮説が採択され，元データの代表値との間に有意差は認められなかった．元データをスライドさせたデータ，指先から肩までのデータ，指先から肩まで以外のデータ，指先から肘まで以外のデータ，指先から手首まで以外のデータについては帰無仮説が棄却され，元データの代表値に対して有意差があった．

次に，指先から肩までのデータ以外の 3 つの代表値が，指先から肩までのデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．その結果，指先から肩まで以外のデータ，指先から肘まで以外のデータ，指先から手首まで以外のデータについて帰無仮説が棄却され，指先から肩までのデータの代表値に対して有意差があった．

次に，指先から肩まで以外のデータ以外の 2 つの代表値が，指先から肩まで以外のデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両立仮説検定を行った．その結果，指先から肘まで以外のデータ，指先から手首まで以外のデータについて帰無仮説が棄却され，指先から肩まで以外のデータの代表値に対して有意差があった．

指先から肘まで以外のデータ以外の 1 つの代表値が，指先から肘まで以外のデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．その結果，指先から手首まで以外のデータについて帰無仮説が採択せれ，指先から肘まで以外のデータの代表値との間に有意差は認められなかった．

本論文における実験の結果，CNN より SVM のほうがうまく二値分類できていることがわかった．なお，右腕の指先から肩までの骨格座標を取得した場合には，平均 73%，被験者によっては 97%の精度で区別ができ，ピッキングおよび鍵開け動作を行っている右腕に特徴があらわれていた．本システムでは，画像や映像をセンタにデータ転送する必要がなく，家庭内で行動の判別が可能である．そのため，本提案手法は，家庭の防犯対策に有用である．

目 次

図 目 次

表 目 次

プログラム目次

第1章

はじめに

監視カメラは犯罪の抑止や犯人の特定に役立っている．一方で，犯行現場をリアルタイムで捕らえて犯罪を防止するには，監視カメラを常時有人監視していなければならず，非常にコストがかかる．カメラの映像を有志で監視するという方法も考えられるが，被撮影者のプライバシの問題があるため実施は困難である．

家の周囲に警備用のセンサを設置するという方法もある．通常，家人が窓から出入りをしたり，故意に窓を割ったりすることはないため，センサを窓に設置することは有効であるが，家の扉の場合，家人が常に出入りするため，窓に設置したセンサを扉に使用すると，出入りするごとに人を検知してしまう．そのため，単純に安価なセンサを導入することはできない．

近年では，監視カメラを人工知能により監視するという技術も出現してきている．最近の研究として，三菱電機による「深層学習を使い，商業施設にいる不審者や社会的弱者を監視カメラで捉えるシステム」 [1] が挙げられる．深層学習とは，機械学習の一種である「ニューラルネットワーク」の階層を深めたアルゴリズムであり，生物の脳の神経細胞をモデル化したアルゴリズムである．この研究では，杖や乳母車などのものを属性として定義することで，深層学習を利用して映像を分析し，不審な行動をとっている人や手助けが必要な人を識別している．

監視カメラと深層学習を組み合わせれば，監視カメラを常時有人監視せずとも扉からの不正な侵入を検知でき，窓を簡易センサで守れば全てが解決するようにも思えるが，問題は単純ではない．深層学習を用いて行動を学習させたり，行動の学習モデルを更新したりする際には，各監視カメラからセンタに大量のデータを送信し，収集する必要がある．行動の学習に映像を用いる場合，インターネットを介して各監視カメラから大量のデータをセンタに集約することは現実的ではない．そして大量の映像を用いて深層学習を行う場合，学習に掛かる時間から映像の解像度を落としデータ量を削減する必要があり，それにより行動を識別するための精度も落ちてしまう．また，使用する映像には被撮影者のプライバシが含まれるため，センタにて取り扱いに注意する必要がある．

そこで，本論文では，Kinect を用いて扉から不正に侵入する行為を検知するシステムを提案する．本研究では，Kinect により撮影された画像ではなく，骨格座標のみを用いて不審な行為の判定を行うため，被撮影者のプライバシも保護される．また，4.1 節にて後述するが，映像と比較してデータサイズを大幅に圧縮できるため，インターネットを通して

センタにデータを送信したり，大量のデータを使用して学習することも画像より困難ではない．

Kinect や他のセンサデバイスを用いて行動判別を行う研究は他にも存在するが，これらの研究は，パンチやキック，ウォーキングなどの大きく異なる行動を対象にした識別手法となっているため，扉を破壊して侵入するなどの行動は検出可能と思われるが，ピッキングのように通常の鍵開けと動作の差が小さいものを高精度で区別できるのかということは定かではない．また，これらの研究では，Kinect やセンサデバイスが最善の位置に固定されて実験が行われており，例えば扉の前に人間が立つ場合，設置が推奨されている，その人間の 1〜4m 先に Kinect を設置することは物理的に不可能である．

これらの研究の影響もあり，機械学習はさまざまな場面で使用されているが，類似の行動を区別できるのか興味を持った．類似の行動にはさまざまなものがあるが，情報セキュリティ分野での貢献を考え，家人による開錠操作と，ピッキングによる開錠操作を区別することを考えた．そのうえで，類似の行動を見分ける際に，どの骨格情報が機械学習に影響を与えるかについて調査を行った．さらに，少ない被験者のサンプルから，不特定の人物の行動を分類できる工夫を行った．行った実験は開錠操作とピッキングであるが，これは一例であり，汎用的な類似行動の識別に利用できると考えられる．以上をふまえ，本論文の題名を，機械学習を用いて類似行動を区別する研究の 1 つとして決定した．研究の説明にはいる前に，今後，本論文で使用する用語について補足しておく．本研究における不審行動とは，家庭のドアからピッキングにより不法侵入を試みようとすることとしている．ピッキングとは，針金などの工具を用いてドアを開錠することであり，本研究の実験においては，棒状のものを鍵穴に沿って上下する動作としている．一方，不審行動でない正常な行動とは，正しい鍵を鍵穴に差し込み鍵をまわす動作としている．なお，本研究では，類似の行動としてピッキングと鍵開けの動作を選択したが，これは監視カメラの映像から人間が瞬時に見分けることが難しいと考えたからである．ここで，Oxford Language において類似とは，よく似ていてまぎらわしいことと記されており，本研究における鍵開けとピッキングの行動比較は類似の行動と呼ぶことは適切であると考えられる [2]．また，戸建において，鍵開け動作と同じ姿勢でピッキングができるプロであれば，怪しまれないと思われた．類似動作の研究として，剣道における類似している技の違いの識別や，類似しているドラムパターンの識別，手話における意味が異なる類似動作の識別がある [3] [4] [5]．しかし，これらは全ての骨格座標を取得できることが前提の環境であり，セキュリティにおいてはそれらの一部が秘匿される可能性がある．そこで，本研究では，骨格座標の一部が隠れている場合に，どこの部位が分類精度にどの程度影響を与えるのか新規性とし調査した．

不審者とは，不審行動をするもので，家人ではない者であるが，本研究においては行動の区別のみを扱い人物の区別は行わない．対象の監視エリアはドアの前のみとする．また，本論文で精度という用語が登場するが，それはどれくらいの確率で分類が行えるかという一般的な用語の使用に留まっており，深層学習における Accuracy の値を指してはいない．なお，バイオメトリクス分野において，研究協力の対象者を被験者と呼ぶべきか，実験協力者と呼ぶべきかを電子情報通信学会（IEICE）より調査した [6]．バイオメトリクスおよび被験者で検索した結果，42 件の論文が検索に引っかかり，全論文 42 件において被験者という用語が使用されていた [7] [8] [9] [10] [11]．一方，バイオメトリクスおよび実験協

力者で検索した結果，19 件の論文が検索に引っかかったが，実際に使用されている論文は 3 件であった [12] [13] [14]．よって，本研究では，被験者という多く使われている用語を使用することとした．

第2章

関連技術

本研究の目的は，家のドアの前で人物の行動を分類することであるため，類似の分類を行っているものを関連研究として調査した．本章の関連研究における行動識別について，一覧にまとめたものを**表 2.1** に示す．また，行動の識別とは関係ないが，機械学習を用いた評価を行う場合に，テスト用サンプルと訓練用サンプルの比率をどのように扱っているか，関連研究を調査した．

2.1 Kinect を用いて機械学習を用いない行動分類手法

Pang らの研究では，Kinect を用いてパンチやキックの行為を検知するシステムの提案をしている [15] ．この手法では，関節座標の位置と速度から、通常行動か異常行動かを判別している．実験の結果，パンチは 95.83%，キックは 100%の精度で検知することが出来ている．Pang らの研究では，機械学習を用いずに取得した関節座標の数値を細かく分類して行動を定義づけているため，新たな行動を判別するためには新たに行動を定義づける必要がある．そして，新たな行動を追加する際に，全ての行動の定義づけを見直す必要がある．当然，類似した行動の定義づけは困難になる．Pang らの研究では，パンチやキックといった骨格座標が大きく異なる行動のみの区別を行っており，ピッキングと鍵開けといった類似の行動を高い精度で区別可能かどうかは定かではない．

2.2 機械学習を用いた人体の行動予測

Horiuchi らの研究では，機械学習を用いて 0.5 秒後の人体の動きをリアルタイムで推定し出力するシステムの提案をしている [16] ．実験の結果，ジャンプしたときの体の重心は実際の動作の 0.5 秒前後に 7.0cm の誤差で推定できる．しかし，これは被験者が途中で意図的に行動を変えることができない，ジャンプという動作に限定して予測を行うものであり，ドアの前に立った人間が次にピッキングをするか鍵開けをするかというような，意図的に行動を選択できるものの予測は行えない．

表 2.1: 関連研究における行動識別の比較一覧

	識別可能な動作	鍵開けとピッキングの区別
Pang ら	パンチとキックの検知が可能	座標が類似しているので不可能
Horiuchi ら	ジャンプ中の 0.5 秒後の動きを推定可能	意図的に変更可能な動作のため不可能
中原ら	歩行しているかどうかを高精度で区別	言及されておらず不明
中島ら	遮蔽された骨格座標を首の位置から想定	首関節だけでは骨格座標の細かい位置予測は不可能
渡邊ら	上方から撮影した距離画像による姿勢	大まかな姿勢推定でも 60%の精度であるため不可能
森ら	歩行における人物の区別	歩行以外は識別対象にならないため不可能
Dehbandi ら	軽度・中度障害者と健常者の上肢の動き	不審者の行動を学習できないので不可能

2.3 移動型センサと Kinect を用いた家庭内の行動ロギング

中原らの研究では，お掃除ロボットなど将来的に一般家庭に普及が進むロボットに 3D センシング機器を付加し，家庭内や施設内で高齢者を認識・追跡して日常行動を把握し，その各行動における運動量蓄積を行うための手法を提案している [17] ．この手法は，定点センサを利用せず、家庭内に普及する簡易ロボットに Kinect センサを搭載することで，深度画像を用いて対象の行動認識を行うものである．実験の結果，歩行のような特徴ある行動に関しては高い精度での認識が可能である．中原らの研究では，歩行しているかどうかといった行動を Kinect の深度センサで識別できるため，ドアの前に Kinect を設置すれば，本研究の目的のうち，ドアを破壊して侵入するような行動を高い精度で識別することはおそらく可能である．よって，不審者のこのような明らかな行動に関しては，既存研究を利用することで解決可能であると思われるが，ピッキングと鍵開けといった類似の行動を高い精度で区別可能かどうかは定かではない．

2.4 遮蔽された骨格座標の追跡

中島らの研究では，人物画像から検出した骨格画像を用いて，人体に遮蔽が起こる状況でも人体骨格を検出し追跡する手法を提案している [18] ．この手法は，OpenPose と呼ばれる RGB 画像から人体骨格を検出することで，遮蔽がある環境下でも関節ごとに関節を追跡するものである．実験の結果，安定して検出される首関節を追跡することで，人物追

跡が可能である．よって，この手法を取り入れることにより，部分的に関節の情報が取れなくても，残りの関節を追跡できる．しかし，中島らの研究では，首関節の位置からそれ以外の骨格座標の位置を大まかに予測しているに過ぎず，首から下がどのように動いているか細かく判別することはできない．ドアの前に人が立っているかどうかであれば首関節の位置から予測は可能であるが，当然，ピッキングと鍵開けといった類似の行動における骨格座標の位置の違いを予測することは不可能である．

2.5　上方視点距離画像を用いた人物姿勢推定手法

渡邊らの研究では，人物を上方から撮影した距離画像から人物姿勢を推定するための手法を提案している [19] ．彼らは，点群をボクセルリスト化し人体の部位の長さで探索する逐次探索型のルールベース手法とランダムフォレストによる機械学習手法を用いることで，推定処理やトラッキング処理の精度向上や，上方視点から特徴量を抽出することで処理高速化を目指している．その結果，逐次探索手法では TOF による正解率は 75%でリアルタイムでの姿勢推定が可能である．また，機械学習手法では正解率が 60%でリアルタイムでの姿勢推定が可能である．しかし，大まかな姿勢推定でも 60%の精度であるため，ピッキングと鍵開けといった類似の行動における骨格座標の位置の違いを予測するには不十分である．

2.6　深度センサによる歩容特徴量を用いた個人識別・追跡方式

森らの研究では，歩容情報を用いることで個人識別を行うための手法を提案している [20] ．この手法では，Kinect v2 を用いて取得したデータから足首間の距離を用いて 1 周期分の歩行動作を抽出している．そして，静的距離，動的距離，関節角度に関しての特徴量を平均値，中央値，最大値から求めている．実験の結果，体の大きさと動きの大きさとの影響を受けて個人差が大きくなるので，動的距離が識別に最も有効であることが分かったとしている．よって，本研究においては，ドアに人物が歩いて近づく際に，家人か不審者かを区別することには使用可能と思われる．ただし，家人と体格が類似している不審者は区別できず，一度ドアの前に立たれてしまうと，ピッキングやドアの破壊などの行動は一切識別できない．

2.7　Kinect を用いた上肢障害推定

Dehbandi らの研究では，Kinect を用いてデータを取得して機械学習することにより，手や腕の障害や上肢障害のレベルを部類するための手法を提案している [21] ．この手法では，健常者が上肢機能臨床的検証するためのテストである Wolf Motor Function Test を行っている．なお、健康な被験者が、健康な個体、軽度障害、中度障害をエミュレートとして検証を行い、分類している．実験の結果，健康な個体の識別精度は 100%，軽度障害は 83.3%，中度障害は 91.7%，平均では 91.7%となり，高精度で分類することが可能である．しかし，Dehbandi らの研究では，区別している動きが障害者と健常者の動きの違い

だけであり，どちらも同じ距離に同じ角度で Kinect が設置されている状態での比較となっている．さらに，学習とテストの被験者が同一人物である．我々の提案手法では，家人による鍵開けか不審者によるピッキングかを区別するが，これから犯罪を行う予定のものが学習に協力することはあり得ないため，不審者による鍵開けの学習を行うことは不可能である．また，6 章にて詳述するが，学習とテストの被験者が異なる場合には単純に学習は行えない．

2.8 評価に用いるテスト用サンプルと訓練用サンプルの比率

本節では，行動識別をしている関連研究ではなく，機械学習を用いた評価を行う場合に，テスト用サンプルと訓練用サンプルの比率をどのように扱っているか，関連研究を調査した．

山口らの研究では，エッジコンピューティングのような分散計算環境においても，シームレスに知的データ処理を実現するための手法を提案している [22] ．この研究では評価に用いるデータセットのうち 9 割を訓練用サンプルとし，1 割をテスト用サンプルとしている．

川村らの研究では，ユーザが商品検索した際に，その検索が購入目的なのか情報収集目的なのかの目的抽出を行い，分類するためのエージェントを提案している [23] ．この研究では評価に用いるデータセットとして，9 割を訓練用サンプル，残りの 1 割をテスト用サンプルとしている．

川村らの研究では，Twitter 上の投稿やマスメディアのニュース記事などから属性情報をもつ事象情報を抽出し，それらを Linked Data 化という方法でインターネットを介してデータ共有する手法を提案している [24] ．この研究では事象抽出精度の評価を行うデータセットとして，9 割を訓練用サンプル，1 割をテスト用サンプルとしている．

以上より，これらの研究に倣い，我々の研究においても，9 割を訓練用サンプル，1 割をテスト用サンプルとして評価を行った．なお，テスト用サンプルは全サンプルからランダムに抽出し，テスト用サンプルの抽出を複数回繰り返して平均と標準偏差を算出している．

第3章

要素技術

3.1 Kinect

Kinect とは，マイクロソフト社が Xbox360 用のゲームコントローラとして 2010 年末に販売された機器である．なお，Kinect を使用することで，身振り手振りや声をカメラやマイク，センサにより捉えることが可能である．本来，ゲームコントローラであるが，USB インタフェースとなっている．そのため，Kinect を Windows や Linux から利用することが可能である．

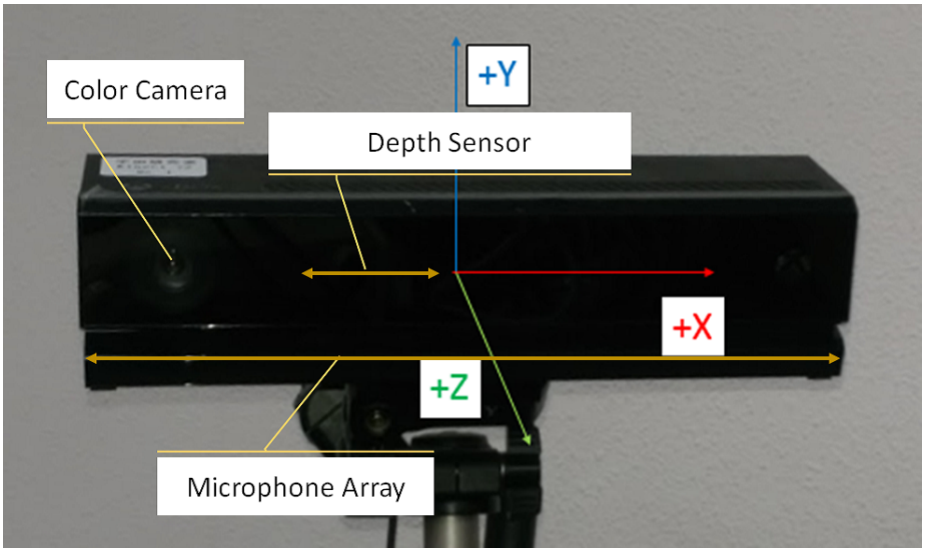

図 3.1: Kinect 機器

この発売により，多くの人の手に届く価格帯でモーション・キャプチャを入手することが可能となった．また， **図 3.1** に示すように，RGB カメラ，深度センサ，マイク等から構成されている．Kinect v1 の Depth センサは，投光した赤外線パターンを読み込み，パターンのゆがみから Depth 情報を取得する「Light Coding」方式を採用している．Kinect v2 の Depth センサは，投光した赤外線が反射して戻ってくる時間から Depth 情報を得る「Time of Flight」方式を採用している．Kinect v1 は，2012 年に発売され Depth（深度）

や Skeleton（人物姿勢）などの情報を手軽に取得できるため，世界中の開発者や研究者から注目された．Kinect v1，v2 ともに取得できる人物領域は 6 人となっている．しかし，Kinect v1 の color カメラは解像度が 640 × 480 と低かったが，Kinect v2 では 1980 × 1080 と解像度が向上しており，背景と人物を分離できるようになっている．また，Kinect v2 では Depth データの精度とその取得範囲も向上している．2014 年に発売された Kinect v2 は，ハードウェア，ソフトウェアともに大きく進化し，Joint（関節）が取得できる Skeleton は 2 人から 6 人まで取得できるようになった．また，取得できる Joint は，20Joint から 25Joint 取得できるようになった．

3.2 深層学習

深層学習とは，機械学習の 1 種である「ニューラルネットワーク」の階層を深めたアルゴリズムであり，生物の脳の神経細胞（ニューロン）をモデルとしたアルゴリズムである [25]．深層学習モデルは，焦点とすべき特徴を自己学習することが可能であるので, 感情分析やコンピュータ・ビジョンなどの複雑なタクスで大きな効果を上げてきた．ただし，深層学習アルゴリズムでは，学習プロセスに時間がかかる．以下深層学習についてまとめた [26].

3.3 順伝播型ネットワーク

3.3.1 ユニットの出力

順伝播型（ニューラル）ネットワーク（feedforward neural network）は，最も基本かつ最もよく使われているユーラルネットワークで，層状に並べたユニットが隣接層間でのみ結合した構造をもち，情報が入力側から出力側に一方向にのみ伝播するニューラルネットワークである．また，多層パーセプトロン（multi-layer perceptron）ともいう．

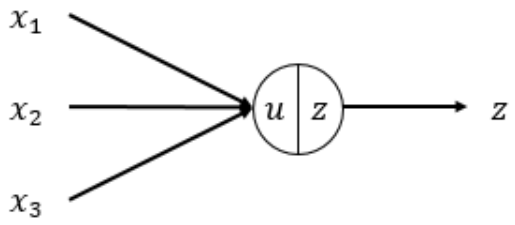

図 3.2: ユニット 1 つの入出力

図 3.2 のように，ネットワークを構成する各ユニットは，複数の入力を受け取り，1 つの出力を計算する．同図の場合，3 つの入力 x_1，x_2，x_3 を受け取ると，このユニットが受け取る総入力は

$$u = w_1x_1 + w_2x_2 + w_3x_3 + b \tag{3.1}$$

となり，各入力にそれぞれ異なる重み（weight） w_1，w_2，w_3 を掛けたものに，バイアス（bias） b を足したものになる．このユニットの出力 z は，総出力 u に対する活性化関数（activation function）と呼ばれる関数 f の出力である．

$$f = f(u) \tag{3.2}$$

順伝播型ネットワークでは，**図 3.3** のようにユニットが層状に並べられ，層間でのみそれらは結合をもつ．この結合をつうじ信号は，左の層から右の層へと一方向に伝わる．同図のネットワークでは，右の層の 3 つのユニットはそれぞれ，左の層の 4 つのユニットからの出力を入力として受け取り，1 つ 1 つの結合に異なる重みが与えられる．式であらわすと，

$$u_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1 \tag{3.3}$$

$$u_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b_2 \tag{3.4}$$

のように計算され，これらに活性化関数を適用したものが出力

$$z_j = f(u_j) \quad (j = 1, 2) \tag{3.5}$$

となる．

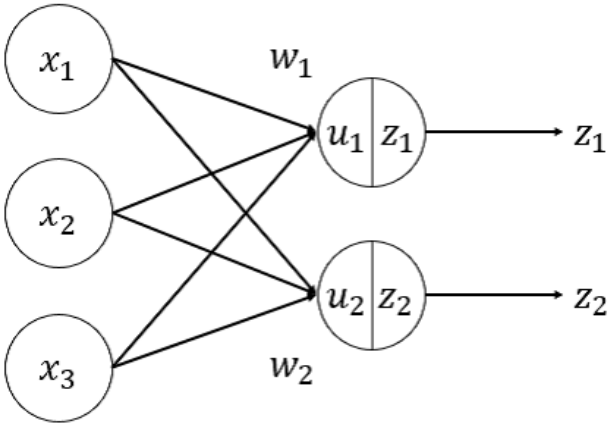

図 3.3: 2 層に並べられたユニットをもつネットワーク

第 1 層のユニットを $i = 1, \ldots, I$，第 2 層のユニットを $j = 1, \ldots, J$ であらわすと，第 1 層のユニットの出力から第 2 層のユニットの出力が決まるまでの計算は，

$$u_j = \sum_{i=1}^{I} w_{ji}x_i + b_j \tag{3.6}$$

$$z_j = f(u_j) \tag{3.7}$$

のように一般化される．

3.3.2 活性化関数

ユニットがもつ活性化関数には通常, 単調増加する非線形関数が用いられる．古くから最も用いられているのが, ロジスティックシグモイド関数（logistic sigmoid function）あるいはロジスティック関数（logistic function）といい，

$$f(u) = \frac{1}{1 + e^{-u}} \tag{3.8}$$

である．この関数は実数全体 $(-\infty, \infty)$ を定義域にもち，$(0, 1)$ を値域とする．ロジスティック関数の代わりに類似の双曲線正接関数

$$f(u) = tanh(u) \tag{3.9}$$

を使うことがある．また，$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$ ともあらわせるように，ロジスティック関数とよく似た性質をもつ．これらの関数を一般にシグモイド関数（sigmoid function）と総称される．これらは，生物の神経細胞がもつ性質をモデル化したものである．

近年，これらの活性化関数に代わり，正規化線形関数（rectified linear function）

$$f(u) = max(u, 0) \tag{3.10}$$

がよく使われている．

3.3.3 多層ネットワーク

図 3.4 に示すような 2 層構造のネットワークを考える．情報は左から右へと一方向に伝わり，この順に各層を $l = 1, 2, 3$ であらわす．なお，$l = 1$ の層を入力層， $l = 2$ を中間層， $l = 3$ を出力層という．

3.3.4 学習の枠組み

順伝播型ネットワークが表現する関数 $y(x; w)$ は，ネットワークのパラメータを変えると変化する．よい w を選ぶことで，このネットワークが望みの関数を与えるようにすることを考える．

目標とする関数は，1 つの入力 x に対する望ましい出力を d と書くと，そのような出力ペアが複数，

$$(x_1, d_1), (x_2, d_2), ..., (x_N, d_N) \tag{3.11}$$

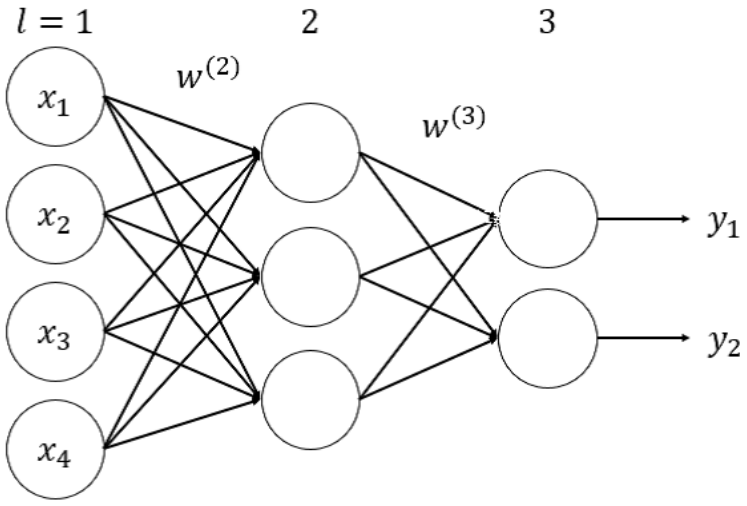

図 3.4: 3 つのユニットを中間層にもつ 2 層ネットワーク

の様に与えられてるとすると，これらのペア (x, d) 1 つ 1 つを訓練サンプル（training samples）と呼び，その集合を訓練データ（training data）と呼ぶ．このとき，どの $n(= 1, \ldots, N)$ のペア (x_n, d_n) に対しても，入力 x_n を与えたときのネットワークの出力 $y(x_n; w)$ がなるべく d_n に近くなるように w を調整する．

このとき，ネットワークがあらわす関数と訓練データとの近さ $(y(x_n; w) \approx d_n)$ をどのように測るか，つまりそれらの近さの尺度が大事となる．この尺度のことを，誤差関数（error function）という．

3.3.5 回帰

回帰（regression）とは，主に出力に連続値を取る関数を対象に，訓練データをよく再現するような関数を定めることをいう．出力層の活性化関数を選んだうえで，ネットワークの出力 $y(x_i)$ が，訓練データの目標出力 d_i に可能な限り近くなるようにすることを考える．その際，二乗誤差

$$\| d - y(x; w) \|^2 \tag{3.12}$$

を使って「近さ」の尺度を決めるのが一般的である．

3.3.6 二値分類

入力 x を内容に応じて 2 種類に区別する問題を考える．例えば，人の顔の写真が与えられ，そこからあるきまった方法で特徴ベクトル x を取り出すとき，x をもとにその人が笑

顔かそうでないかを区別する場合である．$d = 0$ なら笑顔，$d = 1$ なら笑顔でないという具合に種類を 2 値の変数 $d \in \{0, 1\}$ で表現することにする．こうすると問題は，入力 x から d の値を推測することになる．

この問題を定式化する方法は，いくつかあるが，ここでは x を指定したとき，$d = 1$ となる事後確率 $p(d = 1|x)$ をモデル化とする方法を考える．与えられた x に対する d の推定は，このモデルを使って事後確率を計算し，その値が 0.5 を超えれば $d = 1$，下回れば $d = 0$ と判断することとする．

この事後確率をモデル化するのにニューラルネットを使い，出力層にユニットを 1 つだけもち，その活性化関数はロジスティック関数 $y = \frac{1}{(1+exp(-u))}$ とする．また，このネットワーク全体の入出力関係 $y(x; w)$ を事後確率モデル

$$p(d = 1|x) \approx y(x; w) \tag{3.13}$$

とする．ネットワークのパラメータ w を変えることで，さまざまな事後確率を表現することができる．

パラメータ w は，訓練データを用いて，モデルが与える事後分布 $p(d|x; w)$ が，データが与える分布と最もよく整合するように決める．具体的には，最尤推定（maximum likelihood estimation）を行うことを考える．

3.3.7　多クラス分類

クラス分類とは，入力 x を内容に応じて有限個のクラスに分類する問題である．その一例として，**図 3.5** に手書き数字の認識を示す．数字が 1 つ書かれた画像 1 枚が与えられたとき，その数字が 0〜9 のどれかであるかを答えることが目標である．画像の画素値を全画素分，成分にもつベクトルを入力 x とする（画像のサイズが 32 × 32 画素なら x の成分数は 1024）．そのような x が与えられたとき，それを 0〜9 までの 10 クラスのいずれかに分類するのが目的である．

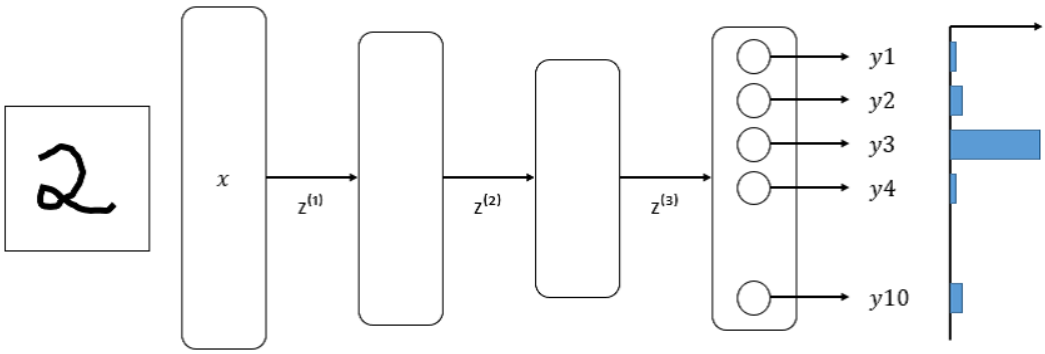

図 3.5: クラス分類の問題の例

このような多クラス分類を対象とする場合，ネットワークの出力層に分類したいクラス数 K と同数のユニットを並べ，この層の活性化関数を次のように選びます．出力層

$l = L$ の各ユニット $k(= 1, ..., K)$ の総入力は，1 つ下の層 $l = L - 1$ の出力をもとに ${u_k}^{(L)}(= W^{(L)} z^{(L-1)} + b^{(L)}$ と与えられる．これをもとに，出力層の k 番目のユニットの出力を

$$y_k = z_k^{(L)} = \frac{exp(u_k^{(L)})}{\sum_{j=1}^{K} exp(u_j^{(L)})} \tag{3.14}$$

とする．この関数をソフトマックス関数（softmax function）という．

3.4 確率的勾配降下法

3.4.1 勾配降下法

学習のゴールは，選んだ $E(w)$ に対し最小値を与える $w = argmin_w E(w)$ を求めることであるが，$E(w)$ は一般に凸関数ではなく，大域的な最小解を直接得るのは不可能である．そこで代わりに $E(w)$ の局所的な極小点 w を求めることを考える．$E(w)$ の極小点は一般に多数存在するので，たまたまみつけた極小点が $E(w)$ の大域的な最小値を与えることはほぼない．

そのような極小解は，何らかの初期値を出発点に w を繰り返し更新する反復計算によって求める．そうする方法はいくつもあるが，なかでも最も簡単な方法が勾配降下法（gradient descent method）である．勾配降下法の勾配（gradient）とは

$$\nabla E \equiv \frac{\partial E}{\partial w} = [\frac{\partial E}{\partial w_1} \cdots \frac{\partial E}{\partial w_M}]^\top \tag{3.15}$$

というベクトルである（ここで M は w の成分数である）．なお，$\top$ は，転置であることをあらわしている．勾配降下法は現在の w を，負の勾配 $(-\nabla E)$ 方向に少し動かし，これを何度も繰り返している．つまり，現在の重みを $w^{(t)}$，動かした後の重みを $w^{(t+1)}$ とすると

$$w^{(t+1)} = w^{(t)} - \epsilon \nabla E \tag{3.16}$$

の様に更新する．ここで，ϵ は w の更新量の大きさを定める定数であり，学習係数（learning rate）という．

3.4.2 確率的勾配降下法

勾配降下法では，全訓練サンプル $n = 1, \ldots, N$ に対して計算される誤差関数 $E(w)$ を最小することを考えた．回帰とクラス分類のいずれであっても，$E(w)$ は各サンプル 1 個だけについて計算される誤差 E_n の和として

$$E(w) = \sum_{n=1}^{N} E_n(w) \tag{3.17}$$

と与えられる．w の更新**式 3.17** では，この $E(w)$ の勾配を用いている．この方法をバッチ学習（batch learning）という．

これに対し，サンプルの一部，最も極端にはサンプル 1 つだけを使ってパラメータの更新を行う方法がある．この方法を，確率的勾配降下法（stochastic gradient descent）と呼び，頭文字を取って SGD と略される．この方法では，w の更新は 1 つのサンプル n について計算される誤差関数 $E_n(w)$ の勾配 ∇E_n を計算し，

$$w^{(t+1)} = w^{(t)} - \epsilon \nabla E_n \tag{3.18}$$

のように w を更新する．

バッチ学習の勾配降下法に対し，確率的勾配降下法にはいくつかの長所があることから，ディープネットの学習ではより一般的である．まず，訓練データに冗長性がある場合には，計算効率が向上し学習が早く実行できる．さらに，確率的勾配下降法を使うと，反復計算が望まない（誤差関数の値が相対的にそれほど小さくない）局所的な極小解にトラップされてしまうリスクを低減できる．バッチ学習の場合, 望まない局所極小解にいったんトラップされてしまうと，二度とそこから抜け出せない．一方で，確率的勾配下降法では，そのようなリスクが小さくなり，反復の度にランダムにサンプルを選べばその効果を最大化できると考えられる．

確率的勾配下降法の他の利点として，パラメータの更新が小刻みに行われるので，学習の途中経過をより細かく監視できることや，オンラインでの学習，すなわち訓練データの収集と最適化の計算を同時並行で行えることが挙げられる．

3.4.3 「ミニバッチ」の利用

規模が大きいニューラルネットの学習は大きな計算コストを要するが，数値計算を効率化するには，計算機がもつ並列計算資源の利用が不可欠である．そのため，重みの更新をサンプル 1 つ単位で行うのではなく，少数のサンプルの集合をひとまとめにし，その単位で重みを更新する．このようにひとまとめにしたサンプル集合をミニバッチ（minibatch）という．具体的には，まず 1 つのミニバッチを D_t とする．これは少数のサンプルの集合を指し，添字は t 回目の更新ごとにそのサンプルが変わることをあらわす．そして，D_t の含む全サンプルに対する誤差

$$E_t(w) = \frac{1}{N_t} \sum_{n \in D_t} E_n(w) \tag{3.19}$$

を計算し, その勾配の方向にパラメータを更新する．

3.5 誤差逆伝播法

3.5.1 勾配計算の難しさ

勾配下降法を実行するには，誤差関数 $E(w)$ の勾配 $\nabla E = \frac{\partial E(w)}{\partial w}$ を計算する必要がある．勾配のベクトルの各成分は，各層の結合重みと各ユニットのバイアスでの誤差関数の微分 $\frac{\partial E}{\partial w_{ji}}$ と $\frac{\partial E}{\partial b_j}$ である．しかし，これらの微分の計算は中間層，特に入力に近い深い層のパラメータ程，その計算が面倒になる．このため，プログラミングが面倒であるとともに，計算量も大きくなってしまう．誤差逆伝播法（back propagation）は，この問題を解決する．

以降, 表記を簡素化するため, $+1$ をいつも出力する特別な第 0 番ユニットを各層に導入し, バイアス b_j をそのユニットと各ユニット j との結合の重み ${w_{0j}}^{(l)} = b_j^{(l)}$ と考えることとする．つまり l 層のユニットへの入力は, $l-1$ 層の第 0 ユニットの出力が常に $z_0^{(l-1)} = 1$ となることで

$$u_j^{(l)} = \sum_{i=1}^{n} {w_{ji}}^{(l)} {z_i}^{(l-1)} + b_j = \sum_{i=0}^{n} {w_{ji}}^{(l)} {z_i}^{(l-1)} \tag{3.20}$$

と簡潔に書きあらわせる．

3.5.2 2 層ネットワークでの計算

2 層ネットワークを考える．このとき，回帰問題への適用を念頭に，出力層の活性化関数は恒等写像とする．ただし，中間層のユニットは任意の活性化関数 f をもつとする．このネットワークでは，入力 $x = [x_1 \quad x_2 \quad x_3 \quad x_4]^\top$ は出力へ向けて次のように伝播する．入力層の出力をいつも通り ${z_i}^{(1)} = x_i$ とすると，中間層の出力は

$${z_j}^{(2)} = f({u_j}^{(2)}) = f(\sum_i w_{ji}^{(2)} z_i^{(1)}) \tag{3.21}$$

となる．出力層の活性化関数は恒等写像なので，その出力は

$$u_j(x) = z_j^{(3)} = u_j^{(3)} = (\sum_i w_{ji}^{(3)} z_i^{(2)}) \tag{3.22}$$

である．このネットワークの誤差関数に二乗誤差

$$E_n = \frac{1}{2} \parallel y(x) - d \parallel^2 = \frac{1}{2} \sum_j (y_j(x) - d_j)^2 \tag{3.23}$$

を選んだとき，重みでの微分 $\frac{\partial E_n}{\partial w_{ji}^{(3)}}$ と $\frac{\partial E_n}{\partial w_{ji}^{(2)}}$ はどのように計算されるかを考える．まず，出力層の重みについての微分 $\frac{\partial E_n}{\partial w_{ji}^{(3)}}$ は比較的簡単に計算できる．これは

$$\frac{\partial E_n}{\partial w_{ji}^{(3)}} = (y(x) - d)^\top \frac{\partial y}{\partial w_{ji}^{(3)}} \tag{3.24}$$

のように展開でき，右辺の $\frac{\partial y}{\partial w_{ji}^{(3)}}$ はベクトルですが，**式 3.24** よりその j 成分のみが $z_i^{(2)}$ で，それ以外の成分は 0，つまり

$$\frac{\partial y}{\partial w_{ji}^{(3)}} = [0 \dots 0 \quad z_i^{(2)} 0 \dots 0]^\top \tag{3.25}$$

の形になる．したがって，

$$\frac{\partial E_n}{\partial w_{ji}^{(3)}} = (y_j(x) - d_j) z_i^{(2)} \tag{3.26}$$

のように求まる．

3.5.3 多層ネットワークへの一般化

2 層のネットワークを任意の層数のネットワークに拡張する．具体的には，2 層ネットワークの中間層の重みに関する計算を一般化することを考える．

まず，式を第 l 層の重み $w_{ji}^{(l)}$ についての式とみると

$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = \frac{\partial E_n}{\partial u_j^{(l)}} \frac{\partial u_j^{(l)}}{\partial w_{ji}^{(l)}} \tag{3.27}$$

となる．式の右辺第 1 項を考える．各 $u_k^{(l+1)}$ を経由した微分連鎖により

$$\frac{\partial E_n}{\partial w_j^{(l)}} = \sum_k \frac{\partial E_n}{\partial u_k^{(l+1)}} \frac{\partial u_k^{(l+1)}}{\partial u_j^{(l)}} \tag{3.28}$$

と分解できる．そこで，この式の右辺の各校を計算する．式の両辺に，第 l 層と第 $l+1$ の層での入力に関する微分 $\frac{\partial E_n}{\partial u_j^{(\cdot)}}$ が現れていることに着目し，

$$\delta_j^{(l)} = \frac{\partial E_n}{\partial u_j^{(l)}} \tag{3.29}$$

とおく．この量をデルタ（delta）と呼ぶ．デルタは各層 l の各ユニット j に対して定義されることに注意する．

3.6　畳込みニューラルネットワーク

3.6.1　全体の構造

物体のカテゴリ認識などの画像認識でよく使われる畳込みネットの典型的な構造を以下の**図 3.6** に示す．畳込み層とプーリング層をペアとしてこの順に並べ，複数回繰り返す．ただし，畳込み層だけが繰り返された後，プーリング層が 1 層続く場合もある．また，畳込み層とプーリング層の後に，局所コントラスト正規化（local contrast normalization，LCN）層を挿入する場合もある．

畳込み層とプーリング層の後に，隣接層間ユニットが全結合した層が配置され，これを全結合層（fully-connected layer）と呼ぶ．

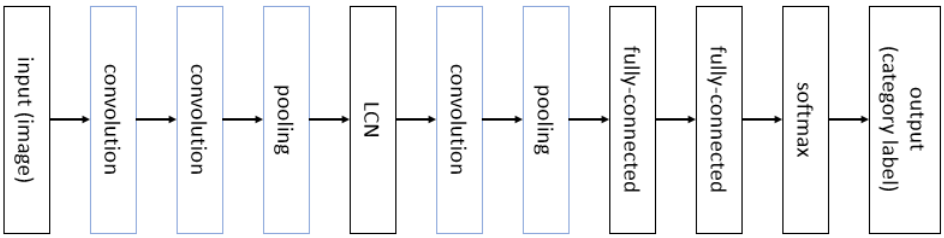

図 3.6: 典型的な畳込みネットの構造

3.6.2　畳込みの定義

濃淡値を各画素に格納したグレースケールの画像を考える．画像サイズを $W \times W$ 画素とし，画素をインデックス $(i,j)(i=0,\ldots,W-1,j=0,\ldots,W-1)$ であらわす．画素 (i,j) の画素値を x_{ij} と書き，負の値を含む実数値を取るとする．そして，フィルタと呼ぶサイズの小さい画像を考え，そのサイズを $H \times H$ 画素とし，フィルタの画素はインデックス $(p,q)(p=0,\ldots,H-1,q=0,\ldots,H-1)$ であらわし，画素値を h_{pq} と書く．h_{pq} は任意の実数である．

画像の畳込みとは，画像とフィルタ間で定義される次の積和計算である．

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq} \tag{3.30}$$

3.6.3　畳込みの働き

画像の畳込みは，フィルタの濃淡パターンと類似した濃淡パターンが入力画像上のどこにあるかを検出する働きがある．つまり，フィルタがあらわす特徴的な濃淡構造を，画像から抽出する働きである．

3.6.4　パディング

式 3.30 のように，畳込みは画像にフィルタを重ねたとき，画像とフィルタの重なり合う画像同士の積を求め，フィルタ全体の和を求めることで計算ができる．したがって，画像からフィルタがはみ出さないように画像内でフィルタを動かす必要がある．そのときの畳込みの結果の画像サイズ（u_{ij} のインデックスの範囲）は入力画像よりも小さくなる．このとき，そのサイズは

$$(W - 2\lfloor\frac{H}{2}\rfloor) \times (W - 2\lfloor\frac{H}{2}\rfloor) \tag{3.31}$$

とあらわせる．ただし，$\lfloor\cdot\rfloor$ は小数点以下を切り下げて整数化する．

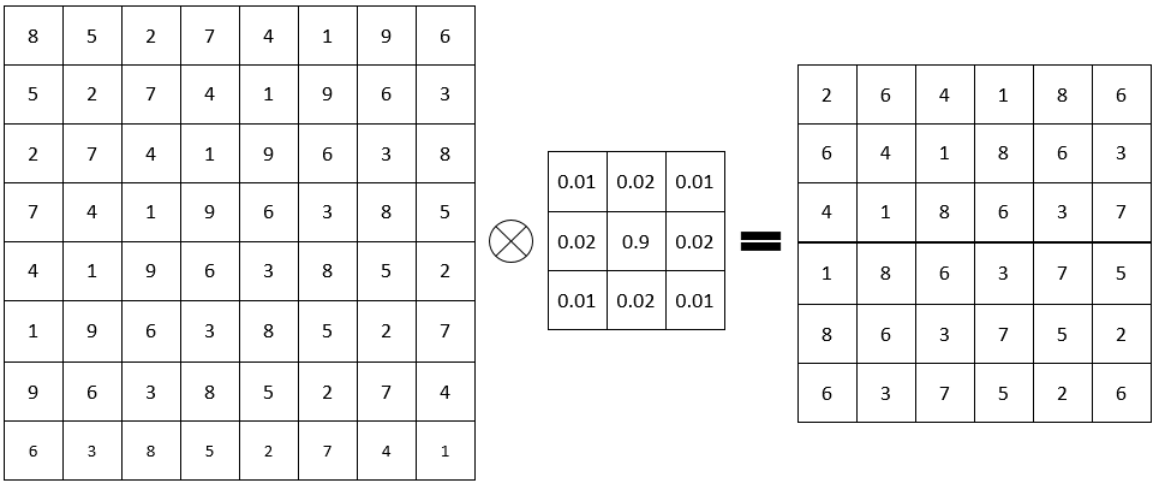

図 3.7: サイズ 8 × 8 の画像と 3 × 3 のフィルタの畳込みによって生成される画像

畳込み結果の画像が，入力画像と同サイズになると便利な場合がある．その場合には，入力画像の外側に幅 $\lfloor\frac{H}{2}\rfloor$ の「ふち」をつけ，もとの入力画像と出力画像のサイズを同じになるようにする必要がある．最も一般的な方法は，この「ふち」部分に画素値を 0 にセットする方法で，これをゼロパディング（zero-padding）と呼ぶ．しかし，この方法を用いると，画像の周辺部が自動的に暗くなってしまう．

3.6.5　ストライド

画像上を縦横方向にフィルタの適用位置を 1 画素ずつずらしていくのではなく，数画素ずつずらして計算する場合もある．このフィルタの適用位置の間隔をストライド（stride）と呼び，ストライドを s とするとき，出力画像の画素数は，

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{si+p,sj+q} h_{pq} \tag{3.32}$$

となり，出力画像サイズは約 $\frac{1}{s}$ となる．上述のパディングを行う場合の正確なサイズは，

$$(\lfloor\frac{W-1}{s}\rfloor+1)\times(\lfloor\frac{W-1}{s}\rfloor+1) \tag{3.33}$$

となる．

3.6.6　畳込み層

畳込み層は，畳込みの演算を行う単層ネットワークである．実用的な畳込みネットでは，グレースケールの画像 1 枚に対してではなく，多チャネルの画像に対し，複数このフィルタを並行して畳込む演算を行う．

3.6.7　プーリング層

プーリング層は通常，畳込み層の直後に設置される．プーリング層のユニットは，畳込み層で抽出された特徴の位置感度を若干低下させることで，対象とする特徴量の画像内での位置が若干変化した場合でも，プーリング層の出力が不変になるようにする．

3.7　2 値クラス分類

以下サポートベクトルマシンについてまとめた [27].

3.7.1　2 クラス分類問題

与えられた入力データが 2 つのどちらに部類されるかを識別する問題を 2 クラス分類問題（binary classification problem）といい，この部類をクラスという．ここで，入力された画像が人かそうでないかを判別する 2 クラス分類問題を考える．**図 3.8** には，人検知システムにおける処理の流れを示す．この際，適当なサイズで画像を切り出し，人であるかを分類する．なお，このシステムでは，まず画像をどのような形式で表現するのか前処理を行い，その後，画像をもとに分類器（classifier）によるクラスタ推定処理を行う．その後，後処理により検出した場所を示している．

2 クラス分類問題の設定では，あらかじめ人かそうでないかがすでに分かっている画像を，コンピュータに学習させることで分類を行っている．分類問題は，訓練データから分類の規則を抽出するように，分類器を構築することで，入力された画像が人かそうでないかを判別することが可能となる．

訓練データは n 枚の画像と各々に人が写っているかの情報によって構成されているとすると，$(x_i, y_i)_{i\in[n]}$ とあらわすことができる．その際，x_i は画像を表現する何らかの数値ベクトルであり，特徴ベクトル（feature vector），または入力ベクトル（input vector）という．画像を数値表現するには，各画素値を並べて画素列を作る方法や，画像中の画素値の局所的変化から特徴を生成するような複雑な方法も存在する．このような方法を特徴抽

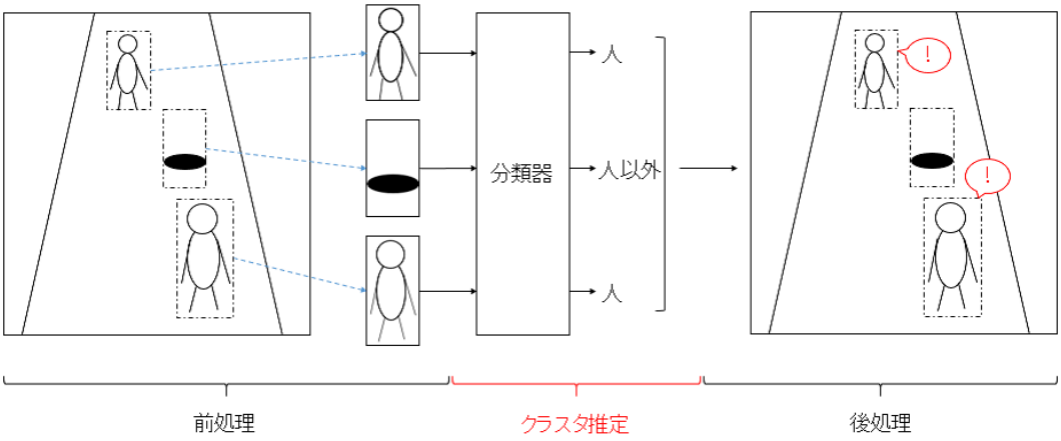

図 3.8: 人検出システムにおける処理

出（feature extraction）といい，特定のタスクに依存するため前処理とし，どのように特徴生成されたかは問題とならない．一方，y_i は x_i に人が写っている場合 1 をとり，また人が写っていない場合 -1 をとる 2 値変数とする．なお，出力 y_i をクラス表現するラベル（label）といい，1 つの x_i と y_i の組 (x_i, y_i) を事例（instance）という．なお，一般的な特徴ベクトルとラベルを考えるときは，単に x や y と書く．

なお，本研究で用いるサポートベクトルマシン（SVM: support vector machine）は，2 クラス分類問題の代表的手法であり，未知データにおいて高い分類精度をもつ分類器により構築が可能である．

3.7.2 線形 SV

n 個の事例からなる訓練集合 $\{(x_i, y_i)\}_{i\in[n]}$ が d 次元実数ベクトル $x_i \in R^d$ と，1 か -1 の値を取るラベル $y_i \in \{-1, 1\}$ から構成されていることとする．決定関数（decision function）と呼ばれる実数値関数 $f : R^d \to R$ を用いて，

$$g(x) = \begin{cases} 1 & f(x) > 0 \text{ の場合} \\ -1 & f(x) < 0 \text{ の場合} \end{cases}$$

のように分類器を定義でき，ここでは，$f(x)$ として，

$$f(x) = w^\top x + b \tag{3.34}$$

の 1 次関数を考える．ただし，d 次元実数ベクトル $w \in R^d$ とスカラー $b \in R$ は未知の実数であるので，どのように推測できるかを考える．$g(x)$ の定義では，$f(x) = 0$ が分類結果の変化する境目になっているため，x は 2 つのクラスを分ける境界を形成しており，そのような境界を分類境界（classification boundary）という．**図 3.9** には 2 次元空間の例を示

す．なお，決定関数として**式 3.34** を用いた SV 分類を線形サポートベクトル分類（linear support vector classification）ともいう．

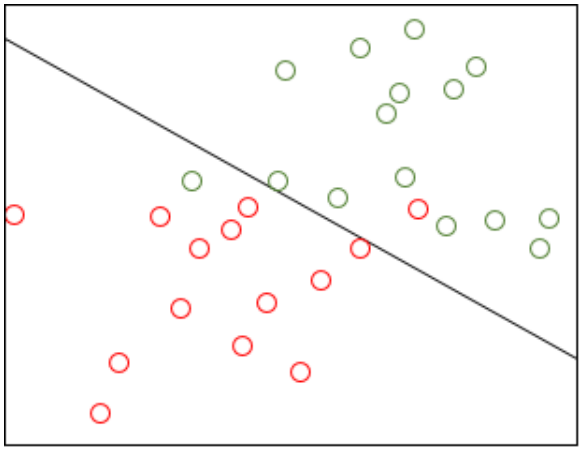

図 3.9: 2 次元における分類境界の例

3.7.3　ハードマージン

一般に，訓練集合を分離できる分類境界は複数存在する可能性がある．そのため，サポートベクトルマシンでは，クラスのデータが分類境界から離れるように分類境界を定めている．**図 3.10** に分類境界の例を示す．なお，分類境界から 2 つのクラスがどのくらい離れているかをマージン (margin) といい，マージンが大きくなるように分類境界を求める．これを，マージン最大化（margin maximization）という．

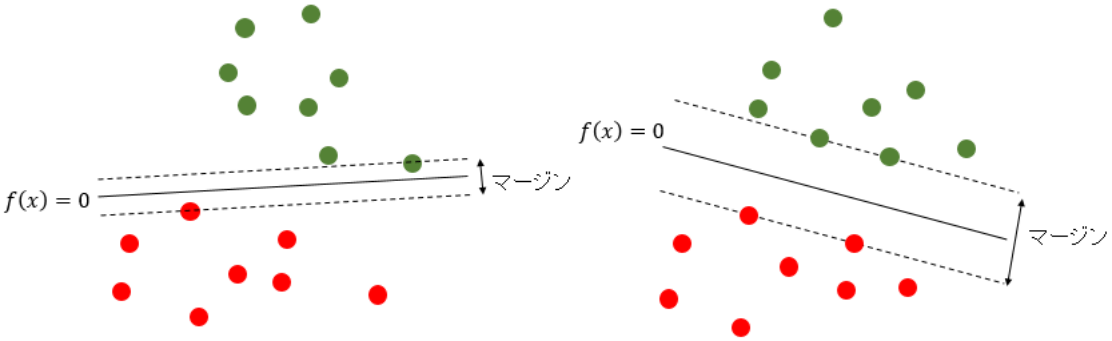

図 3.10: 分類境界とマージン

なお，x_i から分類境界までの距離は，点と平面の距離の公式から

$$\frac{|w^\top x_i + b|}{\| w \|}$$

となる．

一方，全ての点を正しく分類する必要がある．つまり，$y_i f(x_i) > 0$ となる必要がある．ある正の実数 $M > 0$ に対して $y_i f(x_i) \geq M$ が全ての $i \in [n]$ に成立しているとすると，マージン最大化は

$$\begin{aligned} &\max_{w,b,M} \frac{M}{\parallel w \parallel} \\ s.t. \quad &y_i(w^\top x_i + b) \geq M, \quad i \in [n] \end{aligned}$$

となる．$\frac{1}{\|w\|}$ の最大化が，逆数である $\parallel w \parallel$ の最小化と等価であり，$\parallel w \parallel$ の最小化はノルムを 2 乗した $\parallel w \parallel^2$ の最小化と等価であることより，最小化は

$$\begin{aligned} &\min_{w,b} \parallel w \parallel^2 \\ s.t. \quad &y_i(w^\top x_i + b) \geq 1, \quad i \in [n] \end{aligned} \tag{3.35}$$

となる．分離可能性を仮定した SV 分類をハードマージン（hard margin）という．また，図 3.10 のように破線上に存在する点，つまり分類境界を支えている点のことをサポートベクトル（support vector）という．

3.7.4 ソフトマージン

ハードマージンを，現実の多くの問題に仮定することは強すぎる．そこで，SV 分類を分離可能ではないデータに適用する場合，ソフトマージン（soft margin）を利用する．これは，SV 分類のもつ制約条件 $w^\top x_i + b \geq 1$ を緩和することで導かれる．ここで，制約条件を

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad i \in [n]$$

と変更することが可能である．なお，$y_i(w^\top x_i + b)$ は 1 よりも ξ_i だけ小さくなっていることとなる．これにより，x_i がマージンを超えて異なるクラス側にはいることを許容することとなる．これを**図 3.11** に示す．

ハードマージンでは分類境界に最も近い x_i までの距離を使ってマージンを定義したが，ソフトマージンでは $f(x) = -1$ と $f(x) = 1$ の間の距離をマージンとする．

なお，誤分類が発生した場合，制約条件を満たすためには $\xi_i > 1$ である必要がある．つまり，$\Sigma_{i \in [n]} \xi_i$ を小さく保つことで誤分類を抑制することができ，サポートベクトルマシンの最適化問題は

$$\begin{aligned} &\max_{w,b,M} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i \in [n]} \xi_i \\ s.t. \quad &y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad i \in [n] \\ &\xi_i \geq 0, \quad i \in [n] \end{aligned} \tag{3.36}$$

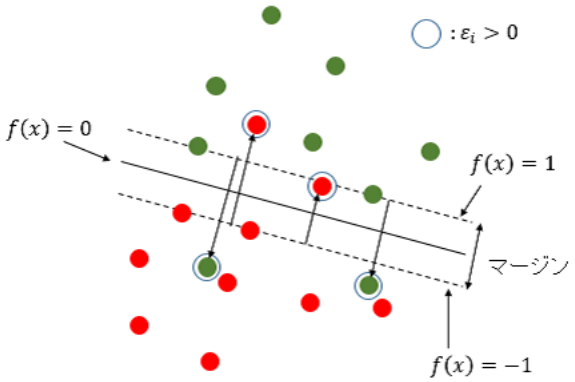

図 3.11: ソフトマージン

となる．なお，係数 C は正則化係数（regularization parameter）といい，事前に正の定数の値を与える必要があり，抑制の度合いを調整するためのパラメータである．C を大きくするとハードマージンに近づき，$C = \infty$ においてソフトマージンはハードマージンと同等となる．逆に，C を小さくすると，誤分類が許容されやすくなる．**図 3.12** は異なる C の値を用いたソフトマージンの分類結果を比較したものである．なお，C の値は交差検証法 (cross validation) を利用して評価を行い，最適な値を選択する．

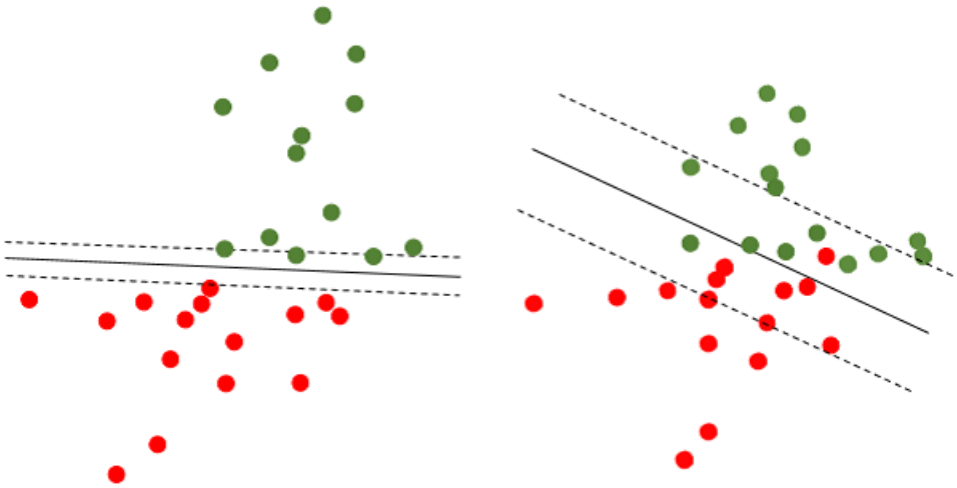

図 3.12: ソフトマージンによる分類

ソフトマージンでは，各訓練集合を $y_i(w^\top x_i + b)$ の値によって 3 種類に分類可能である．なお，**図 3.13** には，マージンの内側および上および外側の図を示す．

- マージンの外側

  $y_i(w^\top x_i + b) > 1$ となるような x_i をマージンの外側という．なお，図 3.13 において，四角で囲まれた点のことである．マージンの外側の点は分類境界の形成に影響

を与えない．

- マージン上

  $y_i(w^\top x_i + b) = 1$ となるような x_i をマージン上という．なお，図 3.13 において，三角で囲まれた点のことである．マージン上の点はハードマージンに対応するものであり，分類境界の形成に影響を与える．

- マージンの内側

  $y_i(w^\top x_i + b) < 1$ となるような x_i をマージンの内側という．なお，図 3.13 において，丸で囲まれた点のことである．マージンの内側の点は，ハードマージンでは存在せず，ソフトマージンにおいては分類境界の形成に影響を与える．

つまり，ソフトマージンでは，マージン上の点とマージンの内側の点が分類境界の形成に影響を与えており，これらをサポートベクトルという．

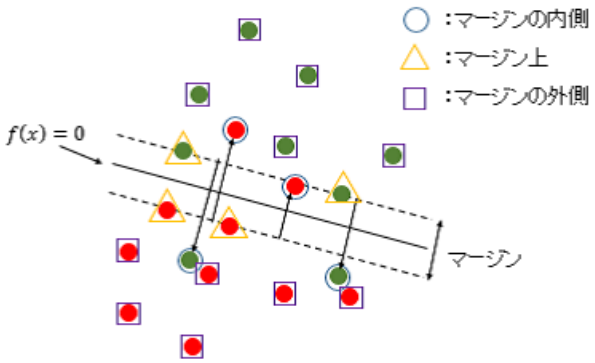

図 3.13: マージンの内側・上・外側

3.7.5 双対表現

ここまでの**式 3.35** や **式 3.36** は SV 分類の主問題（primal problem）である．一方，これに対して双対問題（dual problem）を導くことで，最適化問題に対して違った見解となることがある．SVM の場合，主問題の代わりに双対問題を解くことで分類器を得ることができる．これには，SVM において双対問題が主問題より解きやすい場合や，分類境界の非線形化を考えるうえで有用であることがある．

3.7.6 双対問題

まず，式 3.36 より，

$$\max_{w,b,\xi} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i \in [n]} \xi_i$$
$$s.t. \quad -(y_i(w^\top x_i + b) - 1 + \xi_i) \leq 0, \quad i \in [n] \tag{3.37}$$
$$-\xi_i \leq 0, \quad i \in [n]$$

と置き換える．ここで，双対問題では，新たに $\alpha_i \in R^+, \quad i \in [n]$ と $\mu_i \in R^+, \quad i \in [n]$ という非負の変数を導入する．α_i と 1 つ目の制約条件の左辺 $-(y_i(w^\top x_i + b) - 1 + \xi_i), \mu_i$ と，2 つ目の制約条件の左辺 $-\xi_i$ を乗算したものをそれぞれの目的関数に足し合わせた関数をラグランジュ関数（Lagrange function）といい，

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2} \parallel w \parallel^2 + C \sum_{i \in [n]} \xi_i - \sum_{i \in [n]} \alpha_i (y_i(w^\top x_i + b) - 1 + \xi_i) - \sum_{i \in [n]} \mu_i \xi_i$$

となる．ただし，添字のない α と μ はベクトル $\alpha = (\alpha_1, \cdots, \alpha_n)^\top, \mu = (\mu_1, \cdots, \mu_n)^\top$ となる．また，w, b, ξ を主変数（primal variable），α, μ を双対変数（dual variable）という．

ここで，ラグランジュ関数を双対変数について最大化した関数を $P(w, b, \xi)$ と定義すると，

$$P(w, b, \xi) = \max_{\alpha \geq 0, \mu \geq 0} L(w, b, \xi, \alpha, \mu)$$

となり，主変数について最小化するには，

$$\min_{w,b,\xi} P(w, b, \xi) = \min_{w,b,\xi} \quad \max_{\alpha \geq 0, \mu \geq 0} L(w, b, \xi, \alpha, \mu) \tag{3.38}$$

の最適化を考える必要がある．

この問題は **式 3.37** と同等であり，

$$\begin{aligned} P(w, b, \xi) &= \frac{1}{2} \parallel w \parallel^2 + C \sum_{i \in [n]} \xi_i \\ &\quad + \max_{\alpha \geq 0, \mu \geq 0} (- \sum_{i \in [n]} \alpha_i (y_i(w^\top x_i + b) - 1 + \xi_i) - \sum_{i \in [n]} \mu_i \xi_i) \\ &= \begin{cases} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i \in [n]} \xi_i & \text{主変数が実行可能な場合} \\ \text{定義なし} \quad \text{主変数が実行可能でない場合} \end{cases} \end{aligned}$$

となる．ここで，最適化問題の制約条件を満たしていることを実行可能（feasible）といい，主変数が実行可能でない場合，$-(y_i(w^\top x_i + b) - 1 + \xi_i) > 0$ か $-\xi_i > 0$ となっている i が存在する．一方，主変数が実行可能な場合，全ての i で $-(y_i(w^\top x_i + b) - 1 + \xi_i) \leq 0$

かつ $-\xi_i \leq 0$ となり，双対変数 α_i または μ_i との積の項の最大値は 0 となる．そのため，**式 3.38** はラグランジュ関数により主問題と同等であることがいえる．

次に，ラグランジュ関数を主変数について

$$D(\alpha, \mu) = \min_{w,b,\xi} L(w, b, \xi, \alpha, \mu)$$

のような最小化した関数を定義する．

また，$D(\alpha, \mu)$ を双対変数 α, μ について最大化する

$$\max_{\alpha \geq 0, \mu \geq 0} D(\alpha, \mu) = \max_{\alpha \geq 0, \mu \geq 0} \min_{w,b,\xi} L(w, b, \xi, \alpha, \mu) \tag{3.39}$$

のような問題を双対問題という．また，最小化について考える際，L の w, b, ξ_i の偏微分を 0 とすると，

$$\frac{\partial L}{\partial w} = w - \sum_{i \in [n]} \alpha_i y_i x_i = 0 \tag{3.40}$$

$$\frac{\partial L}{\partial b} = -\sum_{i \in [n]} \alpha_i y_i = 0 \tag{3.41}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad i \in [n] \tag{3.42}$$

となる．**式 3.40**，**式 3.41**，**式 3.42** の条件を満たしていると，ラグランジュ関数から主変数を消去することができ，L は

$$\begin{aligned} L &= \frac{1}{2} \parallel w \parallel^2 - \sum_{i \in [n]} \alpha_i y_i w^\top x_i - b \sum_{i \in [n]} \alpha_i y_i + \sum_{i \in [n]} \alpha_i + \sum_{i \in [n]} (C - \alpha_i - \mu_i) \xi_i \\ &= -\frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y_i y_j {x_i}^\top x_j + \sum_{i \in [n]} \alpha_i \end{aligned}$$

となる．式 3.42 を変形すると，$C - \alpha_i = \mu_i \geq 0$ となり，$C - \alpha_i \geq 0$ という制約が得られるので，サポートベクトルマシンの双対問題は

$$\begin{aligned} &\max_{\alpha} -\frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y_i y_j {x_i}^\top x_j + \sum_{i \in [n]} \alpha_i \\ &s.t. \quad \sum_{i \in [n]} \alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C, \quad i \in [n] \end{aligned} \tag{3.43}$$

となる．

3.7.7 双対性と鞍点

ここでは，主問題と双対問題の関係性について考える．主問題における式 3.38 の最適解を w^*, b^*, ξ^*，双対問題における**式 3.39** の最適解を α^*, μ^* とすると，

$$\begin{aligned} D(\alpha^*, \mu^*) &= \min_{w,b,\xi} L(w, b, \xi, \alpha^*, \mu^*) \\ &\leq L(w^*, b^*, \xi^*, \alpha^*, \mu^*) \\ &\leq \max_{\alpha \geq 0, \mu \geq 0} L(w^*, b^*, \xi^*, \alpha, \mu) = P(w^*, b^*, \xi^*) \end{aligned} \tag{3.44}$$

であり，

$$D(\alpha^*, \mu^*) \leq P(w^*, b^*, \xi^*)$$

の関係となる．この性質を弱双対性（weak duality）といい，どのような最適化問題でも成り立つ．サポートベクトルマシンの場合は，

$$D(\alpha^*, \mu^*) = P(w^*, b^*, \xi^*) \tag{3.45}$$

のようなより強い強双対性（strong duality）となる．つまり，主問題と双対問題の目的関数値が最適解において一致する．強双対性の**式 3.45** となるとき，**式 3.44** は，

$$P(w^*, b^*, \xi^*) = L(w^*, b^*, \xi^*, \alpha^*, \mu^*) = D(\alpha^*, \mu^*) \tag{3.46}$$

となる．さらに，定義より

$$\begin{aligned} P(w^*, b^*, \xi^*) &= \max_{\alpha \geq 0, \mu \geq 0} L(w^*, b^*, \xi^*, \alpha, \mu) \geq L(w^*, b^*, \xi^*, \alpha, \mu) \\ D(\alpha^*, \mu^*) &= \min_{w,b,\xi} L(w, b, \xi, \alpha^*, \mu^*) \leq L(w, b, \xi, \alpha^*, \mu^*) \end{aligned}$$

となり，

$$L(w^*, b^*, \xi^*, \alpha, \mu) \leq L(w^*, b^*, \xi^*, \alpha^*, \mu^*) \leq L(w, b, \xi, \alpha^*, \mu^*)$$

となる，主変数 w, b, ξ について極小値であり，双対変数 α, μ については最大値であることを示している．

3.7.8 最適性条件

SV 分類の最適問題の解を得るためには，

$$\frac{\partial L}{\partial w} = w - \sum_{i\in[n]} \alpha_i y_i x_i = 0 \tag{3.47}$$

$$\frac{\partial L}{\partial b} = w - \sum_{i\in[n]} \alpha_i y_i = 0 \tag{3.48}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad i \in [n] \tag{3.49}$$

$$-(y_i(w^\top x_i + b) - 1 - \xi_i) \leq 0, \quad i \in [n] \tag{3.50}$$

$$-\xi_i \leq 0, \quad i \in [n] \tag{3.51}$$

$$\alpha_i \geq 0, \quad i \in [n] \tag{3.52}$$

$$\mu_i \geq 0, \quad i \in [n] \tag{3.53}$$

$$\alpha_i(y_i(w^\top x_i + b) - 1 - \xi_i) = 0, \quad i \in [n] \tag{3.54}$$

$$\mu_i \xi_i = 0, \quad i \in [n] \tag{3.55}$$

のようにあらわされるカルーシュ･クーン･タッカー条件（KKT 条件：Karush-Kuhn-Tucker condition）が必要十分条件であり，SVM の計算や解がもつ性質を考えるうえで重要である．なお，**式 3.54** および**式 3.55** は，相補性条件（complementarity condition）という．

式 3.47 より，決定関数は

$$f(x) = \sum_{i\in[n]} \alpha_i y_i {x_i}^T x + b \tag{3.56}$$

となる．

相補性条件より，マージンと各事例 i の位置関係性を**表 3.1** に示す．

表 3.1: マージンと特徴ベクトルの位置関係と双対変数

マージンの外側	$y_i(w^\top x_i + b) - 1 > 0 \quad \alpha_i = 0$
マージン上	$y_i(w^\top x_i + b) - 1 = 0 \quad \alpha_i \in [0, C]$
マージンの内側	$y_i(w^\top x_i + b) - 1 < 0 \quad \alpha_i = C$

3.7.9 カーネルによる一般化

双対問題は，SVM の非線形化を考えるうえで重要な役割となる．ここで，入力 x を特徴空間 F へ写像する関数 $\phi : R^d \to F$ を考える．$\phi(x)$ を新たな特徴ベクトルとすると，$f(x)$ は

$$f(x) = w^\top \phi(x) + b$$

となり，その関数による変化を図 **3.14** に示す．

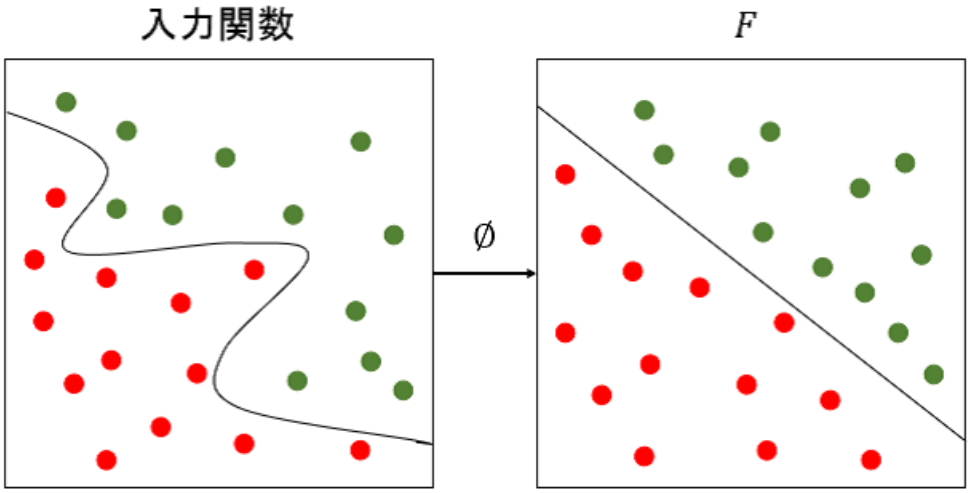

図 3.14: 非線形な分類境界および線形な分類境界

変化後の $\phi(x)$ を新たな特徴ベクトルとすると， x と置き換えることができる．よって，式 **3.43** より，

$$\max_{\alpha} -\frac{1}{2}\sum_{i,j\in[n]} \alpha_i\alpha_j y_i y_j \phi(x_i)^\top \phi(x_j) + \sum_{i\in[n]} \alpha_i$$
$$s.t. \quad \sum_{i\in[n]} \alpha_i y_i = 0$$
$$0 \le \alpha_i \le C, \quad i \in [n]$$

となる．この最適化問題を解くには，$\phi(x)$ ではなく，内積 $\phi(x_i)^\top \phi(x_j)$ を計算する必要がある．よって，

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

と定義できる．また，よく用いられている RBF（radial basis function）カーネル関数は，

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2)$$

である．なお，$\gamma > 0$ は事前に設定する必要がある．これにより，双対問題は，

$$\max_{\alpha} -\frac{1}{2}\sum_{i,j\in[n]}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_{i\in[n]}\alpha_i$$
$$s.t. \quad \sum_{i\in[n]}\alpha_i y_i = 0$$
$$0 \le \alpha_i \le C, \quad i \in [n]$$

となる．式 **3.56** より，$f(x)$ は，

$$f(x) = \sum_{i\in[n]}\alpha_i y_i K(x_i,x) + b \tag{3.57}$$

とあらわすことができる．これにより，最適化や決定関数の計算では $\phi(x)$ を明示的に計算する必要がなくなる．図 **3.15** に 3 つの異なる C を用いた RBF カーネルによる非線形 SV 分類の例を示す．

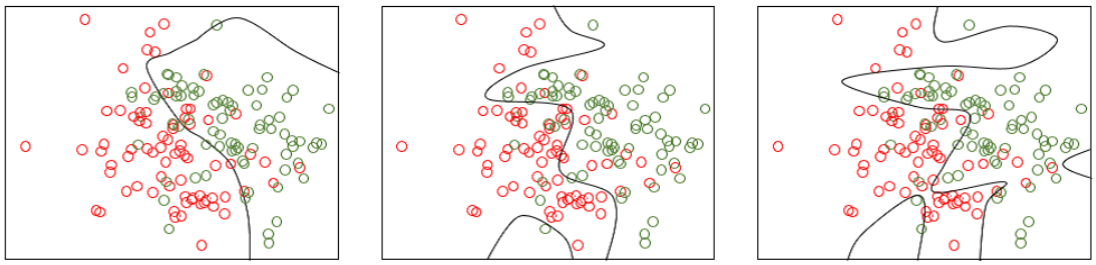

図 3.15: RBF カーネルによる SV 分類

図 3.15 より，非線形な分類境界が形成されていることがわかる．また，C の値が大きくなるにつれて，ハードマージンに近づいていき，訓練集合の境界線が正確になっていく．

3.7.10 計算上の特徴

SV 分類が普及した理由は，予測精度の高さだけでなく，以下の計算上の性質も関連している．

- 凸 2 次最適化問題への帰着

  SV 分類の最適化問題は凸 2 次最適化問題（convex quadratic optimization problem）という種類の最適化問題に属しており，どのような初期値から最適化をはじめても大域最適解にたどり着くことが保証されている．

- 双対変数のスパース性

  双対変数の一部が 0 になると計算効率が向上する場合がある．この特徴は，双対問題を解くうえで利用できる場合があり，最適化における計算を高速化することも場合により可能である．

- 内積によるデータの表現

  SV 分類の双対問題では特徴ベクトルに対して，内積の形に依存する．これにより，カーネル関数での非線形化が可能となる．特定の問題が特徴ベクトルの内積で表現されている場合，内積をカーネル関数で置き換え，カーネルトリック（kernel trick）という非線形化する考え方がある．この考え方は，主成分分析（principal component analysis）でも扱われている．

  また，双対問題を扱う場合，内積のみが分かっていれば x_i を保持する必要はない．そのため，訓練集合のサイズ n が次元数 d に対して小さいと，内積を保持する際にメモリが小容量で済む．

3.7.11 期待損失最小化

入力とラベルを確率変数であらわすと X と Y となる．このとき，実際の訓練データはこの確率変数の実現値となる．データが確率密度関数 $p(X,Y)$ に基づいて生成されていると考えたとき，分類器 $g(x)$ のよさを測るために，損失関数（loss function） $l(y,g(x))$ という関数を扱う．なお，2 クラス分類問題の際には，0-1 損失（0-1 loss） という関数を用いる．

$$l(y,g(x)) = \begin{cases} 0 & y = g(x) \text{ の場合} \\ 1 & y \neq g(x) \text{ の場合} \end{cases}$$

3.7.12 損失関数と正則化

X と Y に関する期待値を含む期待損失を最小化する分類器を求めることは難しい．そこで，訓練集合により期待値を近似した

$$\frac{1}{n}\sum_{i\in[n]} l(y_i, g(x_i))$$

の経験損失を考える．この値は，分類器の精度を図る基準として適している．SV 分類のように決定関数 $f(x)$ を用いると，0-1 損失は

$$l(y,f(x)) = \begin{cases} 1 & yf(x) < 0 \text{ の場合} \\ 0 & \text{それ以外} \end{cases}$$

となる．しかし，この場合，連続関数の最適化に比べ計算が困難となる．そこで，ヒンジ損失という **式 3.58** の損失関数を定義する．

$$l(y, f(x)) = \max\{0, 1 - yf(x)\} \tag{3.58}$$

図 3.16 にはヒンジ損失と 0-1 損失を示す．ヒンジ損失は 0-1 損失よりもずっと扱いやすくなる．

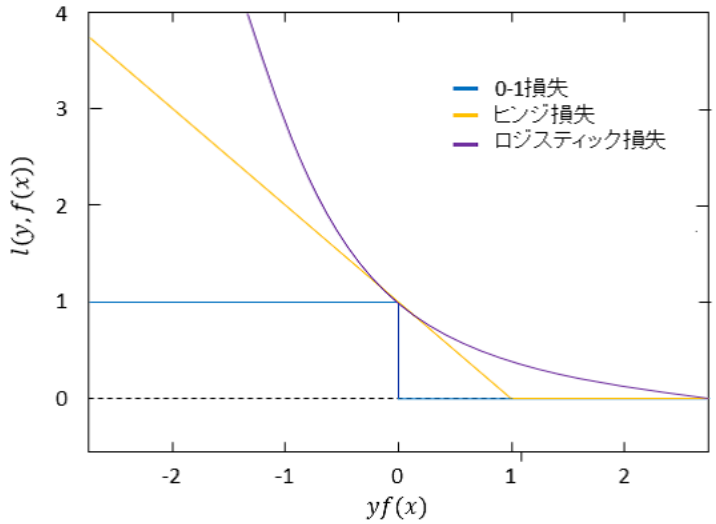

図 3.16: 損失関数の比較

決定関数を $f(x) = w^\top \phi(x) + b$ とすると，訓練集合に対してヒンジ損失を最小化する最適化問題は

$$\min_{w,b} \sum_{i \in [n]} \max\{0, 1 - y_i(w^\top \phi(x_i) + b)\} \tag{3.59}$$

となり，変数 ξ_i を導入すると，

$$\begin{aligned} \max\{0, 1 - y_i(w^\top \phi(x_i) + b)\} &= \min_{\xi_i} \xi_i \\ s.t. \quad \xi_i \geq 0, \quad \xi_i &\geq 1 - y_i(w^\top \phi(x_i) + b) \end{aligned}$$

となる．よって，**式 3.59** より，

$$\begin{aligned} &\min_{w,b,\xi} \sum_{i \in [n]} \xi_i \\ s.t. \quad y_i(w^\top \phi(x_i) + b) &\geq 1 - \xi_i, \quad i \in [n] \\ \xi_i &\geq 0, \quad i \in [n] \end{aligned} \tag{3.60}$$

となる．

$\sum_{i\in[n]} \xi_i \leq K$ であるとすると，誤分類の数は K 以下となり，訓練データをできるだけ分類できるような決定境界が得られる．しかし，訓練集合の分類を追求により期待損失を最小限に押さえることができるとは限らない．**図 3.17** には，ベイズ決定境界と訓練データの誤分類を最小化するように学習した分類境界を示す．

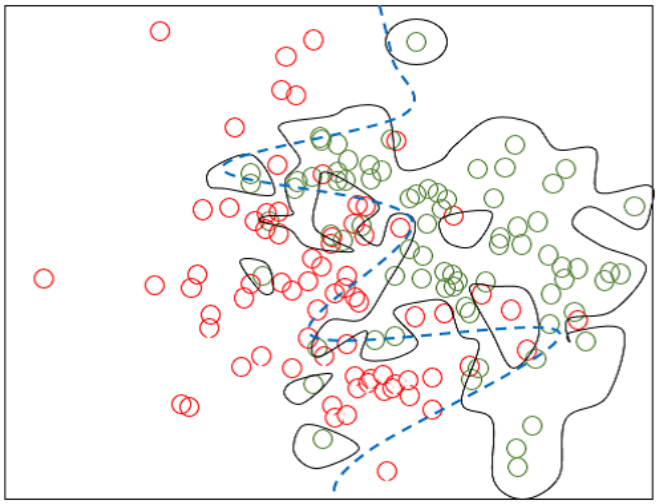

図 3.17: 訓練データに過学習した分類境界（黒実線）とベイズ決定境界（青破線）の比較

黒い実線の分類境界は訓練データを分類しているが，青破線のベイズ決定境界とは異なり，最適な分類精度となっていない．よって，学習アルゴリズムは訓練集合に過度に適合する現象を過学習（over-fitting）といい，これを防ぐためには，正則化（regularization）する必要がある．そこで，パラメータ w の L_2 ノルムが大きくなりすぎないように制限することを考える際，

$$\begin{aligned} &\min_{w,b,\xi} \sum_{i\in[n]} \xi_i + \frac{\lambda}{2} \parallel w \parallel^2 \\ s.t. \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \quad i \in [n] \\ & \xi_i \geq 0, \quad i \in [n] \end{aligned} \tag{3.61}$$

のような問題がある．L_2 ノルムを用いた場合，計算は簡単となり，さらには実用上高い精度を出すこともあるため，広く用いれている．これにより，導かれた最適化問題（**式 3.62**）とソフトマージン SV 分類は等価といえる．

図 3.18 に λ の値を変えて学習した 3 つの分類境界をとベイズ決定境界を示す．λ を大きくすると分類境界が制限され，滑らかになる．しかしながら，データにより異なるので，λ の値は交差検証誤差などを用いて適切に選択する必要がある．

表 3.2: 損失関数と，期待損失の最小化によって得られる解

損失関数	$l(y, f(x))$	期待損失の最小化による解
二乗誤差関数	$(y - f(x))^2$	$f(x) = 2p(Y = 1\|X = x) - 1$
ロジスティック損失	$\log(1 + e^{-yf(x)})$	$f(x) = \log \frac{p(Y=1\|X=x)}{p(Y=-1\|X=x)}$
ヒンジ損失	$\max\{0, 1 - yf(x)\}$	$f(x) = sgn(p(Y = 1\|X = x) - \frac{1}{2})$

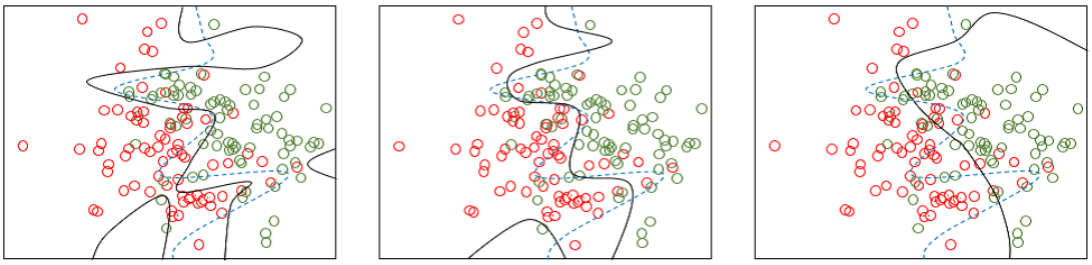

図 3.18: 正則化の度合いの異なる分類境界（黒実線）とベイズ決定境界（青破線）

損失関数と正則化項の和を最小化する枠組みにおいてヒンジ損失を用いると，SV 分類が導かれる．しかし，損失関数に二乗誤差損失（squared error loss）とロジスティック損失（logistic loss）等の異なる関数を用いると異なる手法が導かれる．しかし，図 3.16 に示しているように， $yf(x) > 1$ のような損失が発生したり，外れ値（outlier）による問題点がある．ロジスティック損失はロジスティック回帰（logistic regression）という．二次誤差損失およびロジスティック損失と，ヒンジ損失を比較すると，左側の裾の損失の値の増加は小さくなっており，外れ値の影響は受けにくい．右側の裾においてヒンジ損失の値は 0 になり，他の損失関数と異なっている．他の損失関数では全ての訓練集合が分類境界に何らかの影響がある．

3.8 条件付き確率推定

0-1 損失はベイズ分類器と関連付けることができるが，最適化計算には 0-1 損失の代理としてヒンジ損失が有効である．各損失関数において，期待損失 $E_{X,Y}\left[l(Y, f(X))\right]$ を最小化する $f(x)$ を比較したものを**表 3.2** に示す．

二乗誤差損失は条件付き確率の線形関数を，ロジスティック損失は条件付確率の比を，ヒンジ損失は条件付き確率を離散値に変換したものを推定している．

条件付き確率 $p(Y|X)$ が得られればベイズ分類となり，より直線的に $p(Y|X)$ の確率密度関数を推定することで分類を考えることも可能である．その際に，ベイズの定理（Bayes' theorem）という

$$p(Y|X) = \frac{p(Y|X)p(Y)}{p(X)} \tag{3.62}$$

を用いる．多くの分類手法は何らかの方法でベイズ分類器を推定しようとしているが，手法によりアプローチが異なっている．SV 分類の特徴は分類に必要な最小限のものを直接推定していることである．確率密度関数により，ベイズ分類器を得ることは可能であるが，分類規則がわかっても確率密度関数を得ることはできない．そのため，確率密度関数の推定は分類問題より一般性のある問題であると考えられる．

第4章

提案手法

4.1 提案概要

本稿では，Kinect を用いてピッキングを行う不審者の検知を提案する．本研究では，Kinect を用いて取得した骨格座標のみを使用するため，データをサーバに送信した場合でも映像によるプライバシ侵害は起きない．また，監視カメラは簡易なものでも 20,000 円以上するものが多く，現実的な警備に耐えうる機器になると，さらに高額になる．それに対し，Kinect は 15,000 円程と安価である．Kinect は陰になっている部分の骨格座標もある程度は予測により再現することができる．

本論文は，家に不審者が侵入する際に，その時点で侵入を検知することを目的としているが，本章では不審者によるピッキングと家人による鍵開けを区別する手法のみを提案する．1 章でも述べたが，家人が通常出入りしない窓であれば，簡単なセンサで侵入を検知できる．また，2.1 節や 2.2 節で述べたように，ドアを破壊するような大きな動作をドアの前で行う場合には，Pang らの研究や Horiuchi らの研究から高い精度で検知が可能である．

なお，防犯性を高めるだけであれば，ドアをツーロックにするという方法もあるが，それは不審者が 2 つの鍵をピッキングするという手間が増えるだけであり，ピッキング可能であることには変わりがなく，家人も常に 2 つの鍵を開け閉めしなければならないというデメリットも生じるため，本研究の対象外とする．原理としては，鍵が 2 つになっても本研究の提案手法で対応可能である．また，指紋やカードキーを使った電子的な鍵にするという方法もあるが，バッテリーが切れた際や通電しなくなった際に開いたままになるか開かなくなってしまい，鍵を親戚などに簡単に渡せないというデメリットも生じるため，これも本研究の対象外とする．コスト面からみても，スマートロックへの交換は 3〜6 万円，暗証番号鍵は 5〜10 万円，リモートキーは 5〜10 万円，カードキーは 5〜10 万円，生体認証鍵は 8〜20 万円と Kinect の設置よりも高額であり，暗証番号や鍵の紛失などを考慮すると，普通の鍵よりもコストがかかってしまう [28]．さらに，これらの電子的な鍵にしても，侵入時に侵入を検知できないため，本研究の目的とは合致しない．もちろん，これらの手段を講じたり，窓ガラスにフィルムを貼って割られにくくするなどの，家のセキュリティを高める行為自体を本研究は否定するものではなく，むしろ併用することは大切であると考えている．

本章では，Kinect を用いて骨格座標を取得し，家人による鍵開けと不審者によるピッキ

ングという，動作の違いが少ないものを分類する手法を提案する．分類には機械学習を使用するが，少ない被験者数で精度を高める工夫を行っている．

本研究により，類似の行動を区別することで，警備員がモニタを監視する際の負荷が軽減され，業務が効率的になると考えられる．まず，本技術がない場合は，全ての鍵開け動作にセンタにいる警備員は監視する必要がある．国内におけるピッキングと鍵開けの想定比率が $11 : 1.3 \times 10^{11}$ となり，ピッキングの発生件数は鍵開けの発生件数より十分に小さいことから，ピッキングの発生件数および監視時間は無視することとする．なお，本技術がある場合も同様とする．鍵開けおよびピッキング動作とそれ以外の動作（人物を検出し扉に近づいてくる動作）は大きく異なる動作とすると，既存技術で見分けられることとする．ここで，まず，セコムにおけるホームセキュリティの契約件数は，238 万件（2020 年 9 月末時点）である [29]．一戸建て住宅に 2.71 人居住しており，1 週間における移動回数（玄関の出入り回数とすると）を 14.21 回，1 日にすると 5.5 回となる．鍵を取り出し鍵開け動作が終了するまでに 1 秒かかるとすると，本技術がない場合の日本国内における 1 日あたりの監視時間は，

$$\begin{aligned} &\text{ホームセキュリティ契約件数} \times 1 \text{ 日あたりの移動回数} \times \text{鍵開け動作時間} \\ &= 2.38 \times 10^6 \cdot 5.5 \cdot 1 = 1.3 \times 10^7 [s] = 3.6 \times 10^3 [h] \end{aligned} \tag{4.1}$$

となる．一方，本技術がある場合は，正しく検知できるとすると，国内における年間のピッキング件数の 11 回，開錠におけるピッキング時間は早くて 10 秒程度であることから，110 秒となる．しかし，ピッキングの発生件数および監視時間は無視するので，センタにいる警備員は学習済みモデルによりピッキングと誤検知された鍵開け動作のみを監視することとなる．よって，本技術がある場合の日本国内における 1 日あたりの鍵開け動作の監視時間は，

$$\begin{aligned} &1 \text{ 日あたりの本技術がない場合の監視時間} \times \text{過検知率} \\ &= 3.6 \times 10^3 \cdot \text{過検知率} \ [h] \end{aligned} \tag{4.2}$$

となる．**図 4.1** には，学習済みモデルによるピッキング検出時のフローを示す．そして，センタにいる警備員は，カメラによりピッキング動作であるかを最終判断し，ピッキングであればアラート通知を行い自宅に出動する．また，家主にもアラートを通知する．

次に，精度と鍵開けをピッキングと誤検知された監視時間について考える．その際，国内におけるピッキングと鍵開けの想定比率が $11 : 1.3 \times 10^{11}$ となり，ピッキングの発生件数は鍵開けの発生件数より十分に小さいことから，ピッキングの発生件数および監視時間は無視することとする．ここで，Positive と Negative は，クラスの値（0，1）により決定する．本研究では，鍵開けを 0，ピッキングを 1 としていることから，ピッキングを Positive，鍵開けを Negative としている．なお，Positive と Negative はクラスの値を入れ替えることで，変更可能である．他クラス分類において，クラスごとに使用しているデータ数による偏りやどのクラスを陽性にしているかにより，評価が不適切になってしまう場合がある．その際には，陽性と陰性を入れ替えて平均値をとるマクロ平均，マイクロ平均

の指定も可能である．Accuracy においては，Average Accuracy がその平均値にあたる．ここでは，accuracy_score() メソッドにおける一般的な精度（Accuracy）である

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

となる．TP と FN はピッキング回数であるため無視できるほど小さいとするので，

$$Accuracy = \frac{TN}{FP + TN}$$

となる．よって，FP と TN に着目する．精度が 70%の場合，$FP + TN = 100(= 1.0)$ とすると，$TN = 70(= 0.7)$ ，$FP = 30(= 0.3)$ となる．そして，本システムがない場合，警備員はモニタを 100 回 (=1.0) 確認する必要がある．一方，本システムがある場合，警備員は FP 時の 30 回 (=0.3) モニタを確認する必要がある．監視時間にすると，本システムがない場合，**式 4.1** より，$3.6 \times 10^3[h]$ となる．本システムがある場合，**式 4.2** より，$1.1 \times 10^3[h](= 3.6 \times 10^3 \times 0.3[h])$ となる．以上より，精度が 70%の場合，本システムがある場合では，本システムがない場合と比較して警備員による監視時間は 70%減になると考えられる．また，精度が 80%の場合は 80%減，90%の場合は 90%減となる．よって，これによりセンタの警備員の業務も効率的になる．

センタと各家庭に設置されるデバイスの関係について述べる．東日本地域でいえば，具体的に，センタは東日本および東日本に存在する遠隔統制席（監視センタ）やデータセンタであり，鍵開けとピッキング動作における骨格座標や画像データ，および学習済みモデルを管理している機関である．各家庭に設置されるデバイスは，インターネット経由でセンタに接続されており，設置時には鍵開けとピッキングを区別できる学習済みモデルがインストールされている状態である．鍵開けもしくはピッキングが行われる度に，その骨格座標がデバイスからセンタに送信される．確率は低いが，実際にピッキングが行われて家屋に侵入された場合，ピッキング時の骨格座標をデバイスが取得できる．ピッキング時骨格座標をセンタが一定程度集める度に，ピッキングのトレーニングに必要なデータをセンタ側で更新する．更新されたデータを用い，一定期間ごとにセンタで機械学習によるモデルのトレーニングが行われる．このようにして更新された学習済みモデルは，インターネットを通して各家庭のデバイスに送られ，モデルがアップデートされる．ここで，Kinect の設置場所である各家庭にある機器内の学習済みモデルとデータセンタにある学習済みモデルは同一のものである．そして，一定期間後にデータセンタの学習済みモデルがトレーニングにより更新されると，各家庭にある学習済みモデルも更新される．つまり，各家庭にある学習済みモデルは，データセンタにある学習済みモデルと同期していることになる．また，各家庭のデバイスから，センタを通して遠隔統制席に，ピッキングが行われる度に，その動画が送られる．警備員は手動で動画の内容を確認し，誤りでなければその家庭に急行する．

なお，家庭にてピッキング動作をし始めピッキングと識別された場合から警備員が動画確認を終えるまでの時間は，ほぼ警備員の動画確認時間の数秒程である．具体的には，ネットワーク遅延がないとすると，実データにおけるテスト時間はほぼ 0 秒で完了し，ピッキング動作開始直後の動画が監視センタに送信される．そして，警備員は現在行われている

鍵開けおよびピッキング動作の数十ミリ秒後以内の動画をみていることとなりリアルタイム性はあると考えられる．なお，本研究におけるタイムラグの影響は動画送信のみである．ここで，ピッキング検出時には骨格座標と動画がセンタに送られ，1 サンプル 2 秒間における合計のデータ量は 804KB となり，1 秒換算すると 402KB となる．よって，データ量 400KB/s における遅延が発生してもせいぜい数秒程だと想定され，数秒程のタイムラグは許容の範囲内であると考えられる．

次に，骨格座標の場合と，画像の場合におけるデータ量の違いについてである．1 家庭において 1 日 5.5 回玄関を出入りし， 歩いて扉に近づいて鍵開け動作を行うまでの時間を 4[s/回] とすると，Kinect において 1 日 1 家庭で最低でも 22 秒間程度動作検出することとなる．1 サンプルあたり 2 秒間取得を行うため，最低 11 サンプルの取得となる．よって，1 サンプルあたり骨格座標では 27KB（26〜28KB），画像では 375KB（350〜400KB）となり，1 家庭において骨格座標では 297KB，画像では 4,125KB となる．つまり，骨格座標のデータ量は，画像のデータ量と比較して 93%減となる． 次に最初のデータが 10 人分の 2,000 サンプルとすると，骨格座標では 54MB，画像では 750MB となる．アップデートする際，ピッキング動作を検出した場合となるので，年間 11 件とすると，骨格座標では 1 件につき 54KB となり，11 件で 594KB となる．一方で，画像では 1 件につき 750KB となり，11 件で 8.25MB となる．

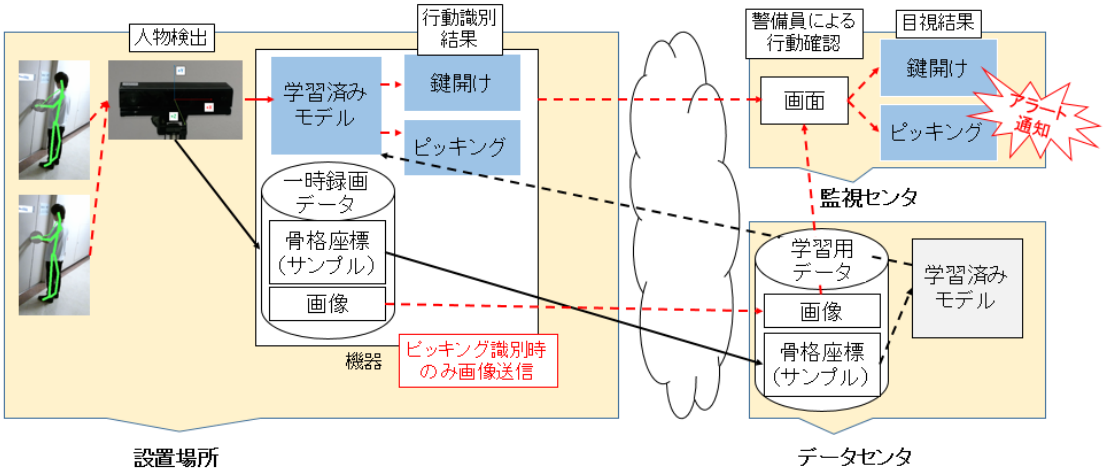

図 4.1: 学習済みモデルによるピッキング検出時の動作フロー

まず，大手セキュリティ会社 2 社におけるホームセキュリティの契約件数は，セコムで 238 万件（2020 年 9 月末時点），ALSOK で 117 万件（2019 年 3 月時点）となっている [29] [30]．鍵開けの場合，通常 1 秒以内に動作が終わると考えられる．一方、ピッキングは早くて 10 秒程度である [31]．また一戸建て住宅に 2.71 人居住しており，1 週間における移動回数が 14.21 回とすると，1 日あたり 5.5 回玄関を出入りすることとなる．以上より，セコムの契約件数の場合，24 時間監視する際 1.3×10^7 秒間（3,636 時間）の監視が必要となる．日本は 8 地方区分に分けられ，それにより監視場所も 8ヶ所あるとすると，1 拠点あたり 1.6×10^6 秒間（454 時間）の監視が必要となる．しかしながら，精度によりピッキング検出件数が変わってくる．50%の場合，1 拠点あたり 227 時間監視が必要となる．次に 60%の場合、1 拠点あたり 182 時間監視が必要となる．次に 70%の場合，1 拠点

あたり 136 時間監視が必要となる。次に 80%の場合，1 拠点あたり 91 時間監視が必要となる．次に 90%の場合，1 拠点あたり 45 時間監視が必要となる．監視の際，1 人による監視ではなく最低 2 人による監視や複数画面分割の監視が想定される．ここで，監視人数 2 人，画面分割による 1 人あたりの監視画面 8 つの場合，精度 50%では 14 時間，精度 60%では 11 時間，精度 70%では 8.5 時間，精度 80%では 5.6 時間，精度 90%では 2.8 時間となる．精度 70%において，8.5 時間の監視が必要となり，24 時間のうちの 3 分の 1 程となり，センタにいる警備員の負担が軽減ざれることとなる．よって，実用的な使用は可能であると思われる．

次に，家 1 件あたりにおける年間のピッキングと鍵開けの比率を記述する．一戸建て住宅における年間のピッキングによる侵入件数は 11 件である [32]．次に 1 戸建て住宅における年間鍵開け回数を予測する．1 戸あたり居住室数は 5.77 室あり，1 室あたり人数は 0.47 人である [33]．よって，1 戸あたり 2.71 人居住していることとなる．また，1 日あたりの移動回数は平日 2.17 回，休日 1.68 回である [34]．1 週間換算にすると，移動回数は 14.21 回となり，1 年間の移動回数は 741.09 回となる．なお，祝日は考慮しないこととする．次に 1 戸あたり 2.71 人居住しているので，1 戸あたりの年間移動回数は 2008.35 となり，住宅数 62,407,000 戸を掛けると 1.25×10^{11} となる．よって，ピッキングと鍵開けの比率は，$11 : 1.3 \times 10^{11}$ となる．従って，鍵開けをほぼ正しく検知できるとすると，センタの警備員の業務の効率を図ることができる．

4.2 Kinect の設置環境

本研究では Kinect v2 を使用している．Kinect v2 の設置環境には次に述べる 2 つを想定して実験を行った．

Kinect v2 の設置環境には以下 2 つの場合を考え定めた．またドアを配慮し被験者の真正面からの関節座標の取得は不可能なので斜め背後からの取得を考えた．なお，建物の都合上，斜め背後に設置できない場合もあると考え，真横からの取得も試みた．背後からの取得は，建物からアームを伸ばして Kinect を設置することを考えると，困難であると考え見送った．

まず，Kinect の高さは 185cm とした．Kinect は被験者に正面に設置するのが推奨されており，Dehbandi らの研究などでも Kinect を正面の最適な距離に設置して評価を行っているが，本研究の環境では，被験者の正面には常に扉がありそこには設置できない．そこで，一般的な監視カメラを設置するのと同様に，常識的に考えて，扉の斜め横から斜め下向きに撮影されるようにした．高さを 185cm としたのは，推奨では被験者の胸の位置くらいになる高さが望ましいが，その位置は出入り時に被験者とぶつかるため，常識的に考えて困難としたためである．位置が高ければ高いほど推奨の高さから外れていくため，極端に高身長の人でなければぎりぎり頭が接触しない程度の高さとした．

次に，Kinect の水平方向の角度は，被験者の斜め右後ろでドアに対して 30 度の位置と，ドアに対して被験者の真横の 2 つとした．被験者の正面から撮影できないのであれば，被験者の真後ろが次に望ましいが，扉からアームを伸ばしてそのように監視カメラを設置するのは常識的に考えて不自然である．さらに，本研究では，体の前にあるドアノブを操作するため，真後ろでは手元が胴体の陰になってしまう．そこで，なるべく被験者の右手側

と左手側を同時に捉えられる位置で，ドアからそれほどアームを伸ばさなくてもよい範囲として 30 度の位置を選択した．ただし，家の構造によっては，ドアの前にそれほどアームを伸ばせない場合もある．そこで，被験者の真横の場合も選択している．なお，被験者の真横から撮影する場合には，被験者の左手側は胴体の陰になり撮影できない．被験者の右手側から撮影するのは，人間は右利きが多く，ドアは基本的に右手で開けることが想定されており，右手側を映す必要があるためである．

仕様上，Kinect v2 の人物の検出範囲は 0.5m〜4.5m であり，近すぎたり遠すぎたりすると骨格座標が取得できなくなる．そこで，本研究では，Kinect と人物の水平方向の距離が約 300cm となるようにしている．詳細に説明すると，人間には身長があるため，例えば，身長 160cm で骨格の中心が上から 80cm の位置とすると，真上から骨格の中心までが 300cm の場合，頭の位置は 220cm(300-80) となり，足の位置は 380cm(300+80) となる．実際には真上ではなく斜め上から撮影しているが，手を挙げれば頭より上に手が来ることもあり，ドアの前で行動する人間の全体が Kincet v2 の検出範囲の中央付近にはいるようにすることを考えてこの距離としている．

水平方向の角度について補足しておくと，実験場所の廊下の幅が 192cm であり，被験者から 300cm の距離を取るとこれ以上の角度を付けることはできない．よって，実験に使用した角度は上記の 2 つである．なお，具体的な評価環境は 6.7 節の**図 6.1** に示す．

4.3 関節座標取得データ

Kinect のスケルトントラッキングによって骨格座標を取得するが，本研究で扱う Kinect v2 は人体を 25 個の関節としてあらわしており，それらの関節 1 つ 1 つを X 座標，Y 座標，Z 座標で立体的にあらわすことができる．そして，X 座標，Y 座標，Z 座標の値をカンマ区切りして 25 個の骨格座標を 1 行に並べて記録した．これを 1 フレームと呼ぶ．1 フレームを 40 コマ，つまり 40 行取得し，それを機械学習における 1 サンプルとしている．

4.4 関節座標データの整形

提案手法においては取得されたデータの整形を行っている．取得したデータをそのまま使用した場合，身長や腕の長さなどの個人の特徴量もデータに入ってしまう．そこで，同じ身長になるようにデータを整形してプライバシを保護している．具体的には，ドアノブに接している骨格座標と床に接している骨格座標の値を固定し，それ以外の骨格座標の値を，被験者がおよそ平均的な身長になるように変更している．これは，例えば，人体の中心の座標を固定して身長を変更した場合，足の座標が地面にめり込んでしまったり宙に浮いてしまったりすることになり，地面と足の座標の距離からもとの身長が判明してしまうためでもある．

4.5 線形補間

本研究では，被験者データをもとに生成したデータを扱う際に，線形補間を行っている．その際の処理について説明する．例えば，40 フレームのサンプルがあるとして，これを

41 フレーム以上に線形補間して 40 フレームまでを使用する場合，41 フレーム以降のデータは削除される．逆に，39 フレーム以下に線形補間して 40 フレームまでを使用する場合，不足するフレームは 1 フレーム目から不足するフレーム数だけコピーして使用する．線形補間して 37 フレームになったのであれば，38〜40 フレームは 1〜3 フレームのコピーとなる．

4.6 一部の骨格座標取得におけるサンプルの整形

本研究では，一部の骨格座標を取得し，遮蔽物により骨格座標が隠れた場合を再現したサンプルの整形を行っている．その処理について説明する．例えば，取得できていない骨格座標がある場合，その骨格座標の値は 0 に置き換えている．一方，取得できている骨格座標の場合，4.4 節で整形した骨格座標の値をそのまま使用している．

4.7 機械学習モデル

畳込みニューラルネットワーク（CNN）およびサポートベクターマシン (SVM) により評価を行う．CNN を選んだ理由として，視覚的イメージの分析に向いており [35] [36] [37]，複数回の畳込みにより小さな特徴量を捉えることができるとともに，移動不変性によりどこにあっても特徴抽出ができる．よって，骨格座標の分類にも向いていると考えた．その他にも回帰型ニューラルネットワーク（RNN）や LSTM，NB がある．RNN はテキストや数値時系列データからどのようなことが起こるのかを予測することに主に使用されている [38] [39] [40]．次に，LSTM は RNN の一種と捉えることができる．ナイーブベイズ（NB）は，テキスト分類に主に使用されている [41] [42] [43]．しかしながら，RNN や LSTM，NB は，それらの分類に向いていることが多いだけで，データによってはクラス分類が可能である場合は存在する．

次に SVM を選んだ理由は，汎化能力を高めることや高いパターン識別されるため，骨格座標の分類に使用されている [44] [45] [46]．また，Sebastian 氏の経験則として，ランダムフォレスト（RF）は総合的に優れているが，SVM は小規模データセットの場合最適であると思われた [47]．なお，頻脈性不整脈を検知により心臓突然死を防止する研究や，テキストによる年齢推定する研究，表情認識を行う研究では，少量のデータセットにおいて SVM で高精度を示している [48] [49] [50]．そして，本研究で使用するデータセットは最大で 83,000 サンプルであり，最小で 2,000 サンプルと少量である．そこで，CNN および SVM により評価を行う．

4.7.1 CNN

畳込みニューラルネットワークにおいては，1 次元の畳込みを行う．このネットワークの構造は，畳込み層，ReLU 層，プーリング層，畳込み層，ReLU 層，プーリング層，全結合層，ReLU 層，全結合層，ReLU 層，ソフトマックス層とする．また，関節座標データは，X 座標，Y 座標，Z 座標の 3 軸座標であり，人体をあらわす関節座標は 25 個である．これを 1 行にカンマ区切りに並べたものを 40 行並べたものを 1 サンプルとする．よっ

て，入力ユニット数は 1 であり，またソフトマックス層によって出力されるユニット数は，ピッキングか鍵開けかの 2 値分類を行うので 2 とする．このときに使用するバッチサイズは 32 から 1,024 までの値の間で 2 を n 乗した値または 1,800 の値とし，エポック数は 50 から 50 ずつ増加させていき，値を変えていったときに F 値が 90%を超えたものを使用する．F 値は**式 4.3** のとおりである．

$$F = 2\frac{P * R}{P + R} \tag{4.3}$$

提案手法においては，F 値が 90%以上であり，かつ標準偏差の値が小さいエポック数とバッチサイズの組合せを選択して評価を行っている．また，2.8 節で述べたように，エポック数とバッチサイズを決定する際は，10 交差検証を用いる．取り扱う両者の訓練用サンプルは全サンプルの 9 割とし，残りの 1 割をテスト用サンプルとする．テスト用サンプルに 1 名の被験者，訓練用サンプルに残り全員の被験者（被験者 1〜10 の中からテスト用サンプルに使用していない被験者，あるいは被験者 11〜20 の 10 名の被験者）のデータを使用する場合には，テスト用サンプルに選ぶ被験者を入れ替えて 10 回ずつ評価を行う．

4.7.2 SVM

サポートベクターマシンにおける構造は，入力層，中間層，出力層とする．また，関節座標データは 4.7.1 項と同様に 25 個の関節座標を 1 行にカンマ区切りに並べ，これを 40 行並べたものを 1 サンプルとする．よって，ピッキングか鍵開けかの 2 値分類を行う．精度評価を行うときは F 値を用いることとする．テスト用サンプルに被験者 1〜10 の中から 1 名を，訓練用サンプルに残り全員の被験者（被験者 1〜10 の中からテスト用サンプルに使用していない被験者，あるいは被験者 11〜20 の 10 名の被験者）のデータを使用し，テスト用サンプルに選ぶ被験者を入れ替えて 10 回ずつ評価を行う．

4.8 実験方法

まず，ピッキングと鍵開けの分類を行うとともに，平均精度が 90%を超えるバッチサイズとエポック数を確認する．ピッキングと鍵開けの動作は次のとおりとした．被験者をドアの前に立たせ，ピッキングの場合には右手で棒状のものをもち鍵穴付近を上下に動かし，鍵開けの場合には右手で鍵穴に鍵を入れて開け閉めを行う．いずれの場合にも左手はドアノブに手を添えておく．

Kinect は 4.2 節にある設置環境のとおりに設置する．被験者の行動を記録する前に，まず，Kinect の方向を向き，骨格座標が取得されているのを確認してからピッキングや鍵開けの行動を行う．これらの行動を行うまでの時間に取得されたデータはサンプルとして使用せず，これらの行動が開始されてからのデータをサンプルに使用する．

第5章

実装

5.1 スケルトントラッキング

4.3 節で述べたように，スケルトントラッキングにより骨格座標の取得を次に示す環境で行った．OS を Windows8.1，IDE を Visual Studio2015，プラットフォームを x64，OpenCV のバージョンを 3.1.0，プログラミング言語を C++とした．また本稿でスケルトントラッキングに用いる Kinect は第二世代の Kinect v2 である．これによって，骨格座標を取得していくわけであるが，そのためには Sensor，Source，Reader，Frame，Data を順に取得していく必要がある [51]．その Kinect v2 におけるデータ取得する際のメソッドの取得の流れを**図 5.1** に示す．

図 5.1: Kinect v2 におけるデータ取得の流れ

始めに Sensor では，まず Kinect を扱うための Sensor インターフェースを定義し，その後，Sensor を取得し，Sensor を開く．次に Source では，Color フレームのための Source インターフェースを定義し，その後 Sensor から Source を取得する．次に Reader では，Color フレームのための Reader インターフェースを定義し，Source から Reader を開く．次に Frame と Data では，Color 画像のサイズ，データサイズの設定をし，Color 画像を扱うために Open CV の準備を行う．また，Color 画像を取得するための Frame インターフェースを定義し，Reader から Frame を取得し，Frame から Color 画像を取得する．

そして，動画ファイルに動画を書き出す．この動画ファイルは被験者が Kinect 内に収まっているかを確認するためのものである．Color 画像を表示するにはサイズが大きいので cv::resize() によってサイズを半分にし, 最後に Frame を解放した．

次に，被験者の骨格座標を取得する．まず，Body のための Frame インターフェースを定義し，Reader から Frame を取得する．そして Frame から Body を取得することにより準備が完了し，被験者から骨格座標をそれぞれ取得していく．

Kinect v2 では，同時に 6 人まで骨格座標を取得可能なので，場合分けによってそれぞ

れのテキストファイルに骨格座標を記述するようにしている．ここでは，1 フレーム分で取得した人体の骨格座標 25 個を X 座標，Y 座標，Z 座標をカンマ区切りにして 1 行に並べ，それを 40 フレーム並べたものを 1 サンプルとしており，1 サンプルごとに出力ファイルを変更している．異なる被験者のサンプルは異なるファイルに出力される．フレームごとの骨格座標の取得間隔は 0.025 秒とし，1 サンプルは 2 秒間のデータで構成されている．これは，Kinect v2 の性能上，骨格座標の値が更新されてから取得される最小のタイミングを考慮して決定している．1 サンプルのファイルサイズは 26KB から 28KB である．

5.2　関節座標データの整形

本研究では，4.4 節で述べたようにデータの整形を行う．データ整形は次に示す環境で行った．OS を Windows8.1，IDE を NetBeans8.0.1，プログラミング言語を Java とした．本研究で用いる Kinect によって，あらわされる関節座標の名称を**図 5.2** に示す．なお，Kinect v2 における関節の定義を**表 5.1** に示す．

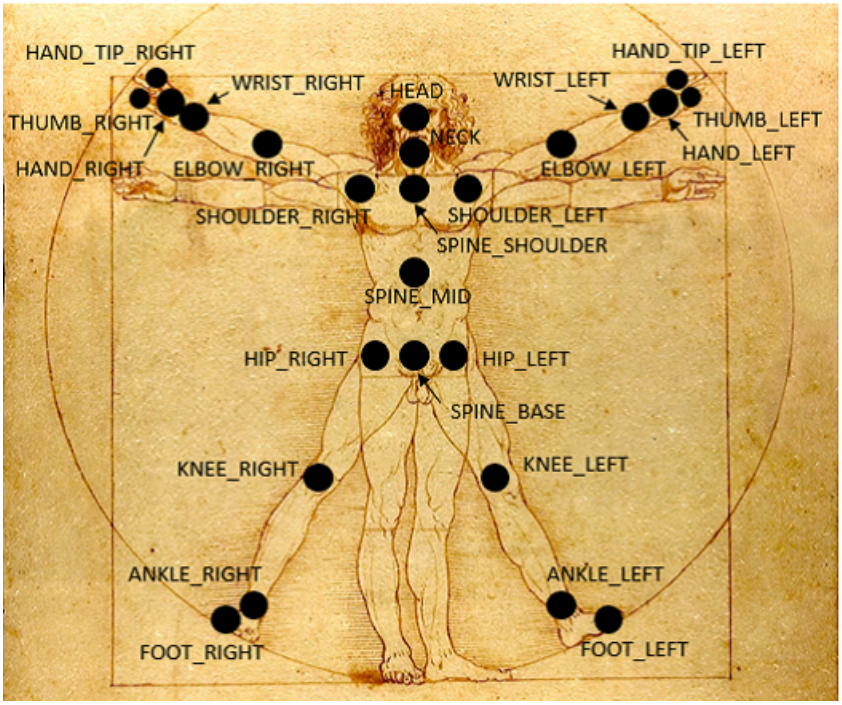

図 5.2: 全骨格座標の位置と名称[1]

[1]https://adinora.com/2018/03/14/die-liebe-der-tod-und-die-zeit-danach/

表 5.1: Kinect v2 における関節の定義

関節の種類	関節番号
SpineBase	0
SpineMid	1
Neck	2
Head	3
ShoulderLeft	4
ElbowLeft	5
WristLeft	6
HandLeft	7
ShoulderRight	8
ElbowRight	9
WristRight	10
HandRight	11
HipLeft	12
KneeLeft	13
AnkleLeft	14
FootLeft	15
HipRight	16
KneeRight	17
AnkleRight	18
FootRight	19
SpineShoulder	20
HandTipLeft	21
ThumbLeft	22
HandTipRight	23
ThumbRight	24

まず，1 サンプル分の値である 3,000 個の骨格座標を格納する配列を入力と出力の 2 種類で定義した．また，1 サンプル分の行数を格納するための配列を定義した．その後にテキストデータを読み込み，各行を 1 つの配列に格納し，split() メソッドを用いてカンマごとに値を取り出し，各値を配列に格納した．さらに，被験者の身長が 170cm になるように拡大縮小する．この拡大縮小では，HEAD と，FOOT RIGHT および FOOT LEFT 間の平均の値を求め，それが 170cm になるように特定の値で割っている．この値を出力する側の配列に格納し，読み込んだものとは別のテキストファイルとして出力した．同様の手順で，骨格座標を 40 行に並べたものを 1 サンプルとして，被験者 1 名あたりのデータは 50×4 通りであるため，テキストファイル名を 4 回変更しながら各 50 回のデータ整形を行う．

このデータ整形を具体的に述べると，各被験者の身長を同じにする際，床に接してい

る骨格座標とドアノブに接している骨格座標の値を固定し，それ以外の骨格座標を動かしている．つまり，WRIST RIGHT，HAND RIGHT，WRIST LEFT，HAND LEFT，THUMB RIGHT，HAND TIP RIGHT，THUMB LEFT，HAND TIP LEFT，ANKLE RIGHT，FOOT RIGHT，ANKLE LEFT，FOOT LEFT は座標を固定する．足のサイズに関しては，身長と足のサイズの間には必ずしも相関がないため本実験では固定した．ELBOW RIGHT と ELBOW LEFT に関しては，HAND RIGHT と HAND LEFT の骨格座標を規準にし，それ以外は FOOT RIGHT と FOOT LEFT の位置を規準にして，Y 座標のみを変更した．Y 座標が鉛直方向である．このデータ整形を行った後の 1 サンプルのファイルサイズは 27KB から 29KB となった．

5.3 被験者データをもとにしたデータ生成

7.6 節で記述しているように，整形したデータをもとに，被験者をペアにして追加のサンプルを生成した．OS は Windows8.1，IDE は NetBeans8.0.1，プログラミング言語は Java である．はじめに，2 名の被験者の同じ行，同じ骨格座標の値の平均値を取る．次に，それぞれの被験者の骨格座標の値からこの平均値を引き，2 で割る．この値をそれぞれの被験者の骨格座標の値に足したもの，および先ほど求めた平均値の値を新たな骨格座標の値としてサンプルに追加した．なお，この手順により追加されたサンプルの，1 サンプルのファイルサイズは 30KB から 32KB となった．

5.4 線形補間プログラム

4.5 節で記述したように，被験者ごとの鍵およびピッキングによる開錠動作における周期を同じにあわすために線形補間を行う．まず，1 サンプル分の値である 3,000 個の骨格座標を入力した際に格納する配列を定義した．なお，入力した値が格納されている配列は X 座標または Y 座標または Z 座標のいずれかが入っている．また 1 サンプル分の行数を格納するための配列を定義した．その後，入力した値が格納されている配列を 1 つの骨格座標の X 座標，Y 座標，Z 座標の 3 軸の値を 1 つの要素が保持している配列を入力と出力の 2 種類で定義した．

次にテキストデータを読み込み，各行を 1 つの配列に格納し，split() メソッドを用いてカンマごとに値を取り出し，各値を配列に格納した．この配列は X 座標，Y 座標，Z 座標いずれかの値が格納されているため，1 個の骨格座標の 3 軸座標の値が 1 つの要素に格納した．次に，周期変更を行う．1 サンプルは 40 フレームからなっているので下記で記述しているように，周期を 0.7 倍したものと 1.3 倍にしたものを作り出す．なお，0.7 倍にした場合，1 サンプルが 28 フレームとなるので，1 フレーム目から 12 フレーム目までの 12 フレームを入れることで 40 フレームになるようにした．また，1.3 倍した場合，1 サンプルが 52 フレームになるので 41 フレーム以降は切り捨てることとした．この周期変更後の値を出力するために用意した配列に格納し，テキストデータに書き込み，周期変更を行ったデータの生成を行った．これにより，被験者行動における分類が行えるかを確かめる．そのために，被験者 10 名分の鍵とピッキングにおける周期をそれぞれ**表 5.2** と**表 5.3** に示す．

表 5.2: 鍵による開錠動作における行動周期

被験者番号	回数 [回数/秒]					標準偏差
	1	2	3	4	5	
被験者 1	0.70	0.70	0.75	0.75	0.70	0.0245
被験者 2	0.65	0.80	0.70	0.75	0.80	0.0245
被験者 3	0.70	0.65	0.75	0.70	0.70	0.0583
被験者 4	1.10	1.10	1.05	1.10	1.15	0.036
被験者 5	0.45	0.35	0.50	0.35	0.35	0.0632
被験者 6	0.70	0.65	0.70	0.65	0.65	0.0245
被験者 7	0.40	0.35	0.35	0.35	0.40	0.0245
被験者 8	1.10	1.00	1.10	1.05	1.00	0.0447
被験者 9	0.50	0.60	0.50	0.50	0.45	0.0490
被験者 10	0.85	0.85	0.80	0.75	0.85	0.0400
被験者 11	0.60	0.60	0.80	0.85	0.85	0.116
被験者 12	0.45	0.50	0.50	0.30	0.25	0.105
被験者 13	0.40	0.55	0.50	0.50	0.60	0.0663
被験者 14	0.30	0.40	0.55	0.65	0.60	0.130
被験者 15	0.35	0.40	0.90	0.85	0.65	0.105
被験者 16	0.60	0.65	0.75	0.85	0.90	0.114
被験者 17	0.30	0.45	0.55	0.55	0.55	0.0980
被験者 18	0.75	0.85	0.75	0.85	0.80	0.0447
被験者 19	0.40	0.55	0.70	0.85	0.90	0.186
被験者 20	0.80	0.85	0.85	0.70	0.85	0.0583
被験者 1〜10			0.71			0.235
被験者全体			0.66			0.215

表 5.2 から鍵による開錠動作における周期の最小値は 0.35[回/秒] であり，最大値は 1.15[回/秒]，平均値は 0.71[回/秒] であった．また表 5.3 からピッキングによる開錠動作における周期の最小値は 0.5[回/秒] であり，最大値は 1.55[回/秒]，平均値は 0.93[回/秒] であった．以上の結果から，それぞれのデータにおける周期を 0.7 倍，1.3 倍のデータを作り出すこととした．

表 5.3: ピッキングによる開錠動作における行動周期

被験者番号	回数 [回数/秒]					標準偏差
	1	2	3	4	5	
被験者 1	1.15	1.20	1.20	1.40	1.35	0.0245
被験者 2	1.15	1.45	1.10	1.35	1.40	0.0583
被験者 3	0.55	0.55	0.55	0.55	0.60	0.0316
被験者 4	1.15	1.15	0.75	1.05	1.10	0.0316
被験者 5	0.50	0.50	0.90	0.85	0.65	0.0632
被験者 6	0.55	0.80	0.90	0.75	0.85	0.0245
被験者 7	0.55	0.50	0.55	0.60	0.55	0.0245
被験者 8	0.85	0.85	1.00	1.05	1.20	0.0447
被験者 9	0.65	0.60	0.75	0.70	0.80	0.0490
被験者 10	1.35	1.30	1.55	1.60	1.55	0.0400
被験者 11	1.95	1.75	2.20	2.25	1.95	0.183
被験者 12	0.45	0.40	0.35	0.35	0.40	0.0374
被験者 13	0.75	0.70	0.65	0.65	0.70	0.0374
被験者 14	0.80	1.20	1.45	1.40	1.40	0.241
被験者 15	0.40	0.50	0.55	0.65	0.60	0.0860
被験者 16	0.85	1.50	1.55	1.95	1.95	0.403
被験者 17	0.65	0.65	0.65	0.60	0.60	0.0245
被験者 18	1.00	1.15	1.00	1.40	1.55	0.220
被験者 19	0.60	0.70	0.60	0.70	0.70	0.0490
被験者 20	0.55	0.50	0.50	0.70	0.65	0.0812
被験者 1〜10			0.93			0.332
被験者全体			0.94			0.446

5.5 被験者データにおける骨格座標のスライド

7.9 節で述べたように，ここでは被験者データの骨格座標の値を近くするために，骨格座標のスライドを行う．

まず，1 サンプル分の値である 3,000 個の骨格座標を格納する入力するための配列を 1 種類定義した．また，1 サンプル分の行数を格納するための配列を定義した．そして，骨格座標をスライドするために必要な基準値となる座標を格納する変数を 3 種類と，3 軸上をスライドさせるための配列を 1 種類定義した．なお，この変数には Kinect の設置環境により基準となる座標値が変わるため，その都度変更するものとする．その後，テキストデータを読み込み，各行を 1 つの配列に格納し，split() メソッドを用いてカンマごとに値を取り出し，各値を配列に格納した．各行の基準となる入力座標を被験者 1〜10 の 10 名分の基準となる座標の平均値で引き，スライドさせるための座標値を求めた．その後，各行の入力値と先ほど求めたスライドさせるための座標値を足し，この値をテキストファイ

表 5.4: 被験者の立ち位置

動作	Kinect 設置環境	斜め後ろ			真横		
	被験者番号	X 座標	Y 座標	Z 座標	X 座標	Y 座標	Z 座標
鍵開け	被験者 1	0.9247	-1.0667	2.9757	0.9099	-1.0790	2.9659
	被験者 2	0.4287	-1.2112	2.8947	0.4796	-1.2008	2.9187
	被験者 3	0.0814	-0.8737	2.8640	0.1193	-0.2989	2.7150
	被験者 4	0.0413	-0.9253	3.1992	-0.0390	-0.9775	3.2692
	被験者 5	0.3523	-0.9205	3.2871	0.2958	-0.9338	3.2355
	被験者 6	0.2813	-0.9017	3.2669	0.2956	-0.8939	3.2631
	被験者 7	0.2839	-0.9309	3.2935	0.2904	-0.9077	3.2858
	被験者 8	0.3111	-0.9817	3.2241	0.2821	-0.9467	3.2333
	被験者 9	0.3199	-0.9786	3.2191	0.3045	-0.9739	3.2419
	被験者 10	0.7225	-0.9758	3.2237	0.7099	-0.9711	3.2476
ピッキング	被験者 1	0.9100	-1.0790	2.9659	-0.0494	-0.9781	2.7824
	被験者 2	0.4796	-1.2008	2.9187	-0.2822	-1.3092	2.9505
	被験者 3	0.1193	-0.2989	2.7150	-0.2406	-0.5647	2.8951
	被験者 4	-0.0390	-0.9775	3.2692	-0.2077	-1.1048	2.6665
	被験者 5	0.2958	-0.9338	3.2355	0.0721	-0.9950	2.9363
	被験者 6	0.2956	-0.8939	3.2631	0.0337	-1.0079	2.8723
	被験者 7	0.2904	-0.9077	3.2858	0.1399	-0.9950	2.9471
	被験者 8	0.2821	-0.9467	3.2333	0.0718	-1.0571	2.8050
	被験者 9	0.3045	-0.9739	3.2419	0.0537	-0.9768	2.9667
	被験者 10	0.7099	-0.9711	3.2476	0.2634	-1.0395	2.9153
	全被験者平均	0.3698	-0.9475	3.1412	-0.0068	-1.0054	2.8940

ルとして出力した．なお，基準とした骨格座標には基準となる座標の平均値を代入した．この処理を被験者 1〜20 に対して行った．

スライドを行う際，基準となる骨格座標が必要となる．そこで，動作を行っているときに最も座標の値がほぼ一定である足のかかと，左のかかとを基準とした．なお，Kinect の設置環境により骨格座標の値が異なってくるため，斜め後ろからと，真横からで座標の平均値を求めた．そのときに使用する座標は被験者 1〜10 の 10 名における，鍵開けとピッキングによるものである．また，被験者 10 名の左かかとの座標値を**表 5.4** に示す．

表 5.4 の結果，斜め後ろからの場合は，X 座標 0.3698[m]，Y 座標-0.9475[m]，Z 座標 3.1412[m] であった．また真横からの場合は，X 座標-0.0068[m]，Y 座標-1.0054[m]，Z 座標 2.8940[m] であった．よって，それぞれの Kinect の設置環境において，それぞれの座標を基準に骨格座標をスライドするものとする．

5.6　被験者データをもとにした隠れた骨格座標の再現

4.6 節で述べているように，ここでは遮蔽物により被験者の骨格座標が隠れた場合を再現する．そこで，5.2 で整形したデータをもとにサンプルを生成した．OS は Windows8.1，IDE は NetBeans8.0.1，プログラミング言語は Java である．

まず，1 サンプル分の値である 3,000 個の骨格座標を格納し，入力および出力するための配列を 1 種類定義した．また，1 サンプル分の行数を格納するための配列を定義した．

次に，サンプルが格納されているファイルから各被験者のサンプルを取得する．その後，テキストデータを読み込み，各行を 1 つの配列に格納し，split() メソッドを用いてカンマごとに値を取り出し，各値を配列に格納した．そして，遮蔽物により隠れている骨格座標の数値を 0 とする．なお，遮蔽物により隠れていない骨格座標の数値は変更しないものとする．数値変更後に，格納ファイルを指定し，サンプルを指定ファイルに生成する．

本研究では，遮蔽物により右腕以外の骨格座標が隠れた場合のサンプルを生成を行う．また，右腕以外の骨格座標により行動判別されているのかをみるため，遮蔽物により右腕の骨格座標が隠れた場合のサンプルの生成も行う．なお，右腕の指先から肩までの骨格座標を取得した場合の位置を**図 5.3** に示し，右腕の指先から肩まで以外の骨格座標を取得した場合の位置を**図 5.4** に示す．また，右腕の指先から肘までの骨格座標を取得した場合の位置を**図 5.5** に示し，右腕の指先から手首までの骨格座標を取得した場合の位置を**図 5.6** に示す．

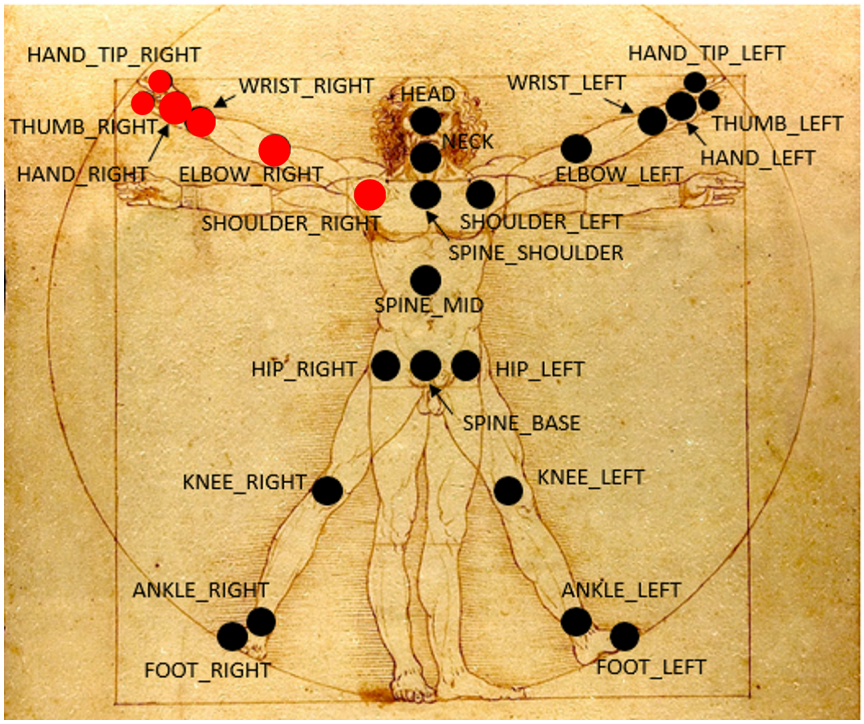

図 5.3: 右腕の指先から肩までの骨格座標を取得した場合の位置

図 5.3 より，右腕の指先から肩までの骨格座標を取得した場合には，右腕の骨格座標である Shouler_Right，Elbow_Right，Wrist_Right，Hand_Right，Hand_Tip_Right，Thumb_Right

以外の数値を 0 に置き換えている．

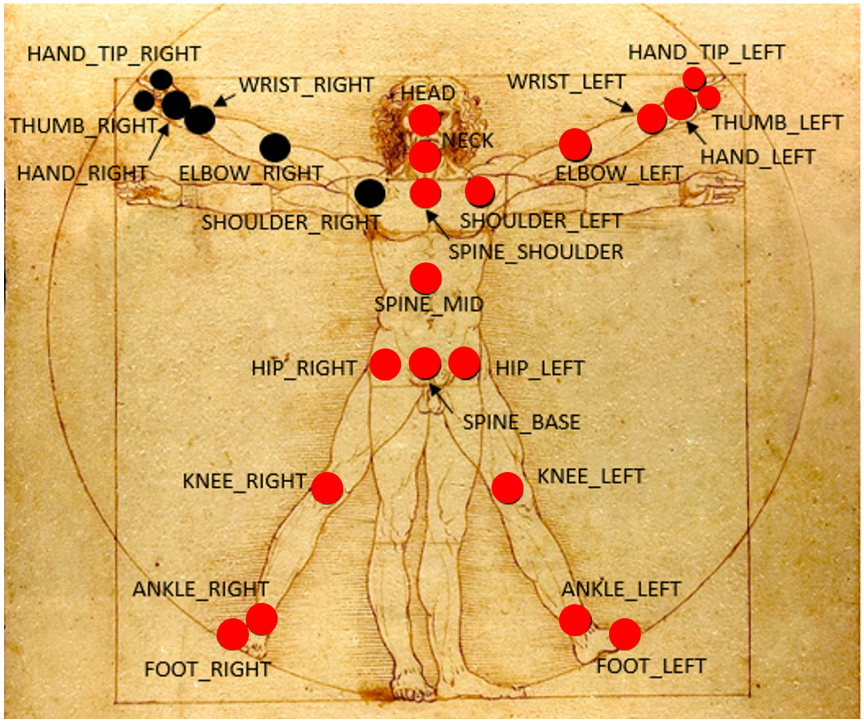

図 5.4: 右腕の指先から肩まで以外の骨格座標を取得した場合の位置

次に，図 5.4 より，右腕の指先から肩まで以外の骨格座標を取得した場合には，右腕の骨格座標である Shouler_Right，Elbow_Right，Wrist_Right，Hand_Right，Hand_Tip_Right，Thumb_Right の数値を 0 に置き換えている．

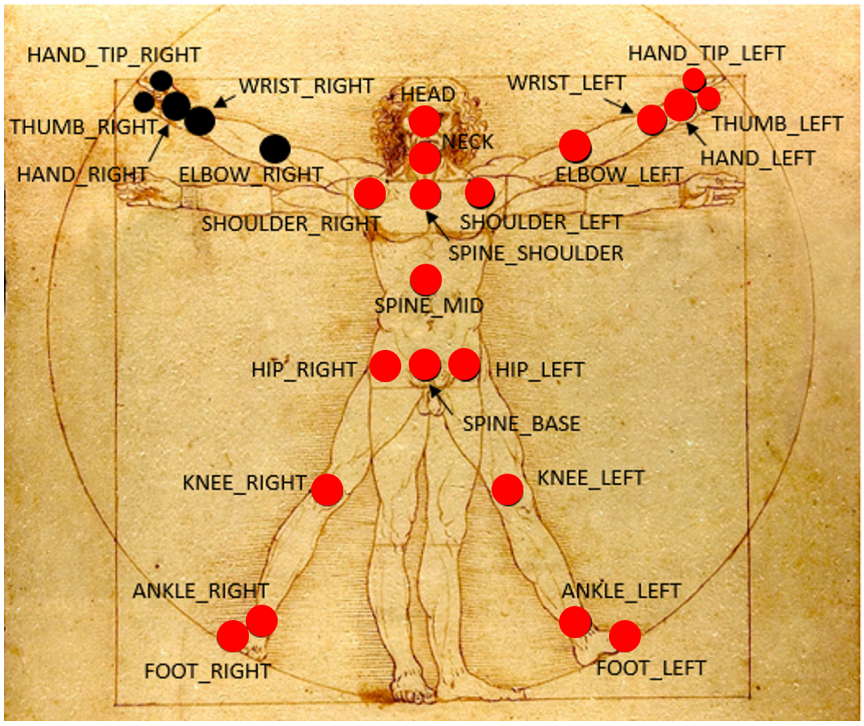

図 5.5: 右腕の指先から肘まで以外の骨格座標を取得した場合の位置

次に，図 5.5 より，右腕の指先から肘までの骨格座標を取得した場合には，右腕の骨格座標である Elbow_Right，Wrist_Right，Hand_Right，Hand_Tip_Right，Thumb_Right の数値を 0 に置き換えている．

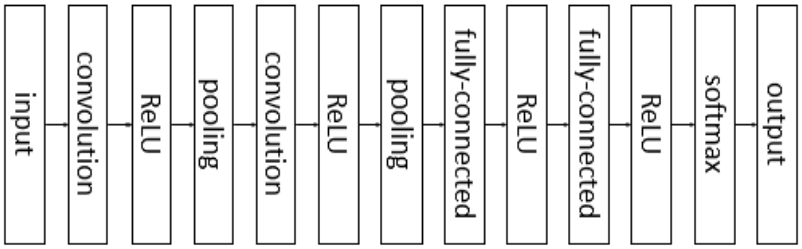

図 5.7: CNN のネットワーク構造

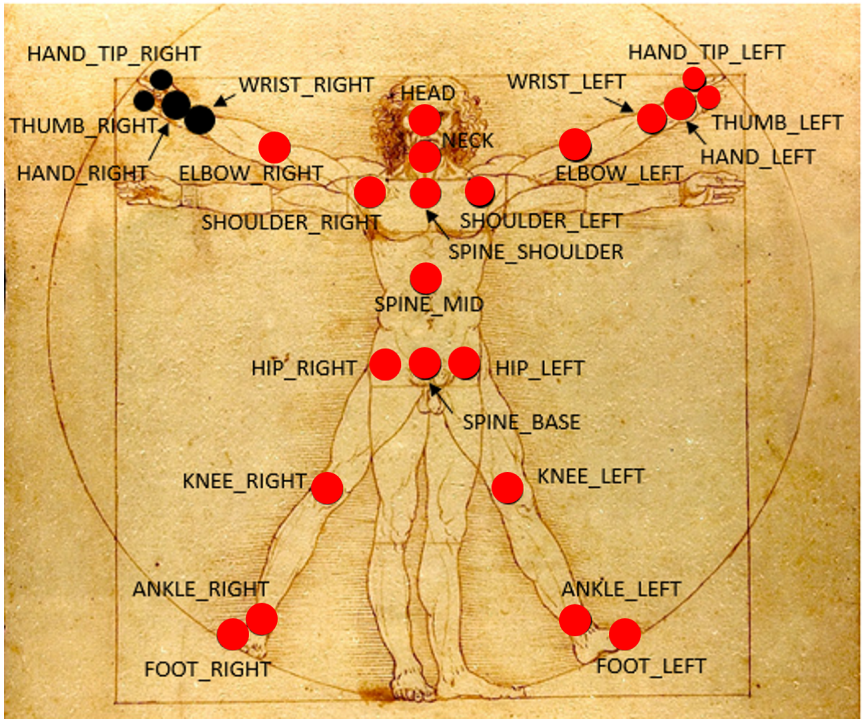

図 5.6: 右腕の指先から手首まで以外の骨格座標を取得した場合の位置

次に，図 5.6 より，右腕の指先から手首までの骨格座標を取得した場合には，右腕の骨格座標である Wrist_Right，Hand_Right，Hand_Tip_Right，Thumb_Right の数値を 0 に置き換えている．

5.7　機械学習モデル

5.7.1　畳込みニューラルネットワーク

4.7.1 項で述べたように，畳込みニューラルネットワークでは，一次元の畳込みを行う．本研究で用いる畳込みニューラルネットワークの構造を**図 5.7** に示す．

このネットワークは 11 層となっている．入力されると，畳込み層，ReLu 層，プーリング層，畳込み層，ReLu 層，プーリング層，全結合層，ReLu 層，全結合層，ReLu 層，ソ

フトマックス層を通り，出力されるという構造になっている．具体的には，入力されるユニット数を 1 サンプル分の 1 とし，1 サンプルには 25 関節を 3 軸座標の値であらわしたものを 1 行に並べ，それを 40 行並べたものが格納されており，これがネットワークの入力となる．畳込みには chainer.links の ConvolutionND を用いた．1 層目の畳込み層では，入力チャネルはサンプル分なので 1，出力チャネルは 1 サンプル分の行数である 40，フィルタサイズは 3，ストライドを 3，パッドを 1 とした．2 層目は ReLu 層である．3 層目は平均プーリング層であり，プーリングサイズを 2 とした．4 層目の畳込みでは入力チャネルを 40，出力チャネルを 20，フィルタサイズを 3，ストライドを 1，パッドを 1 とした．5 層目は ReLu である．6 層目は平均プーリング層であり，プーリングサイズを 2 とした．7 層目は全結合層であり，入力ユニット数 360 として，出力ユニット数を 1 サンプル分の数値の個数である 1,000 とした．8 層目は ReLu 層である．9 層目は全結合層であり，入力ユニット数 1,000 として出力ユニット数を 1,000 とした．10 層目は ReLu 層である．11 層目はソフトマックス層であり，入力ユニット数が 1,000 であり，二値分類を行うので出力ユニット数は 2 である．これによってピッキング，鍵開けのどちらに分類されるかを求める．

次に，全サンプルが存在するディレクトリを指定し，そこから全ファイルのリスト名を取得する．次に，テスト用サンプルが全体の 1 割りとなるように選択し，そのリスト名を取得する．そして，残りを訓練用サンプルとして，そのリスト名を取得する．その後，それぞれのサンプルのデータを float 型の 32 ビットの配列に格納する．それぞれのサンプルのリスト名は int 型の 32 ビットの配列に格納する．そして，それぞれのサンプルのデータ配列の次元数を，reshape() メソッドを用いて 1 × 40 × 75 の 3 次元に変更する．さらに，TupleDataset() メソッドによりデータセットを作成し，SerialIterator() メソッドによりデータを順番に取り出す．また，epoch と，学習もしくはテスト時の誤差関数 precision を表示し，学習の最後にそれぞれのディレクトリの precision，recall，F 値を表示する．

2.8 節で記述したように，評価の際に，訓練用サンプルとテスト用サンプルの両方に同一被験者のデータが含まれる場合は，10 交差検証を用いて精度の平均をとって二値分類を行う．訓練用サンプルは全サンプルの 9 割，テスト用サンプルは 1 割であり，同じサンプルがテスト用サンプルに重複して選ばれないよう，10 回入れ替えながら学習とテストを行うので，各サンプルに番号を振り，その番号を 10 で割った余りによってテスト用サンプルを区別している．一方，テスト用サンプルに 1 名の被験者のデータを入れ，残りを訓練用サンプルとする場合は，被験者を入れ替えながら 10 回の学習とテストを行った．

5.7.2 サポートベクターマシン

4.7.2 項で記述したように，入力層，中間層，出力層という構造になっている．また本研究で用いたサポートベクターマシンは線形 SVC（LinearSVC）により精度評価を行う．LinearSVC を用いた理由として，scikit-learn algorithm cheat-sheet に記載してある条件をもとに決定した [52] [53]．**図 5.8** には，scikit-learn algorithm cheat-sheet における分類器の決定フローを示す．まず，分類識別において，学習モデルは SGD Classifier や kernel approximation，LinearSVC，Naive Bayes，KNeighbors Classifier，SVC または Ensemble Classifier がある．SGD Classifier は 10 万サンプル以上のときに使用し，上手く学習できないときには kernel approximation を用いる．10 万サンプル以下のときには LinearSVC

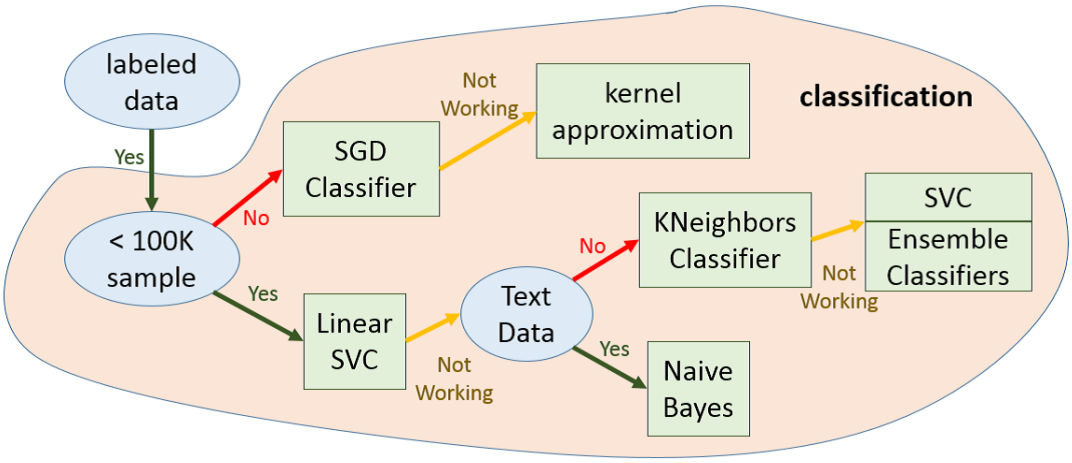

図 5.8: scikit-learn algorithm cheat-sheet における分類器の決定フロー

を使用し，上手く学習できないときにはテキストデータであれば Naive Bayes を，テキストデータでなければ KNeighbors Classifier を使用する．KNeighbors Classifier で上手く学習できない場合は SVC や Ensemble Classifier を用いることとなる．本研究では，最大で 83,000 サンプルを使用しており，10 万サンプル以下となるため LinearSVC が有効であると思われた．

そして，分類識別において，カーネルを使用した SVC や LinearSVC の他に，カーネルを使用していない KNeighbors Classifier や Ensemble Classifier などがある．また，SVC カーネルには linear や poly，rbf，sigmoid，precomputed が存在し，Ensemble Classifier には Random Forest(RF) が存在するが，今回は scikit-learn algorithm cheat-sheet に従うこととした．

まず最初に，全サンプルが存在するディレクトリを指定し，そこから全ファイルのリスト名を取得する．次に，テスト用サンプルが被験者 1〜10 のうち 1 人となるように選択し，そのリスト名を取得する．つまり，テスト用サンプルは被験者 1〜10 のうち 1 名とし，それぞれの被験者データを入れ替えながら行う．そして，残りを訓練用サンプルとして，そのリスト名を取得する．なおテスト用サンプルは，被験者 11〜20 における全てのものである．その後，それぞれのサンプルのデータを float 型の 32 ビットの配列に格納する．それぞれのサンプルのリスト名は int 型の 32 ビットの配列に格納する．次に，線形 SVC の分類器にパラメータを与える．ここでは，パラメータの値はデフォルト値とし，学習は fit() メソッドに引数として，訓練用サンプルと訓練用サンプルのリスト名を指定した．最後に，ピッキングと鍵開けにおけるそれぞれのディレクトリの precision，recall，F 値を表示する．

第6章

評価手法

6.1 ドアと被験者の位置関係

被験者がピッキングと鍵開けを行う際に，自由にどのように行ってもよいことにしてしまうと，極端な例としては右手と左手の動作を逆にしたり，不自然な体勢で鍵を開けることもでき，正しく学習や評価が行えない．そこで，鍵開けを行う際に，自然に行うとどうなるのか，足のかかとの開き幅やドアとかかととの距離を調査した．かかとを基準に測定した理由は，足は最も体重がかかる場所であり体を支える場所であることと，つま先は足を開く幅に個人差があるのではないかと考えたためである．土踏まずも，足の中央にあるため，足が開く場合の影響をかかとよりも受ける．また，被験者により足のサイズも異なるため，脚が足と接続されているかかとがこれらの影響を一番受けない箇所である．

そこで，本研究では，鍵開けを行う被験者 1〜10 の 10 名に対して，本人が最も自然に鍵開けを行う際の足の開き幅とドアと被験者との距離の関係を測定し，その平均を一般的な人物のドアを開錠するときの立ち位置とした．**表 6.1** にドアと被験者の位置関係を示す．

表 6.1 の結果から，被験者 10 名におけるかかとの開き幅の平均は約 14cm となり，ドアからかかとの距離の平均は約 56cm となった．よって，本研究の実験においては，被験者 10 名の平均である，かかとの開き幅 14cm，ドアからのかかとの距離 56cm を使用して被験者をドアの前に配置し，データ収集を行った．

6.2 関節座標データの複数同時取得

4.3 節で記述したように，提案手法では，骨格座標を X 座標，Y 座標，Z 座標に分けて並べ，これを 1 組として 25 箇所の骨格座標を 1 行にあらわし，それを 40 行取得したものを 1 サンプルとしている．実験においては，被験者が継続してピッキングや鍵開けの動作を行っている際に，連続して複数のサンプルを取得した．

本実装においては，Kinect を用いたプログラムを実行すると，サンプル番号と取得するサンプル数を入力する画面が表示されるので，数値を入力することで Kinect が動作し，骨格座標が取得され始める．その際，骨格座標が認識されていない場合があると実験に支障があるため，被験者は体を Kinect に向けてからピッキングや鍵開けの動作に移るようにした．そのため，データ取得直後のサンプルは評価に用いず，骨格座標が取得され始め

表 6.1: ドアと被験者の位置関係

被験者番号	かかとの開き幅 [cm]	ドアからののかかとの距離 [cm]
被験者 1	14	55
被験者 2	19	51
被験者 3	17	65
被験者 4	7	56
被験者 5	15	48
被験者 6	16	61
被験者 7	12	60
被験者 8	12	52
被験者 9	12	55
被験者 10	15	59
平均	14	56

てから約 5 秒後のサンプルから評価に使用している．これは，5 秒あれば，被験者が扉のほうに向き直り，指定された位置でピッキングや鍵開けの動作を行い始めるのに十分だからである．ここで，サンプルの取得間隔は 2 秒，1 サンプル分の取得に必要時間は 1 秒間強である．それを踏まえて本研究では，1 回あたりの取得サンプル数を 25 とし，それを 2 回行い 50 サンプル分のデータを取得した．従って，Kinect による関節データの取得回数を 27 とし，始めの 2 サンプルは評価に使用せず，それ以降の 25 サンプルを評価に使用している．なお，本研究では，各被験者におけるサンプルの取得日は，動作識別であり，日時による被験者の動作の違いは影響しないため，同日に取得している．一方，個人識別の場合には，日時による被験者の行動の違いを学習させることが必要である．そのため，各被験者のサンプルを分割して別日に取得することとなる．

6.3 被験者データをもとにしたデータ生成

5.3 節で記述したように，被験者 10 名分のデータをもとに新たなサンプルを生成し，学習による精度を向上させている．被験者 10 名をペアにし，各ペアに対してこの作業を行った．具体的には，被験者 1 と被験者 2 のペア，被験者 1 と被験者 3 のペア，被験者 1 と被験者 4 のペアというように組み合わせ，最後は被験者 8 と被験者 9 のペア，被験者 8 と被験者 10 のペア，被験者 9 と被験者 10 のペアと，全ての被験者の組み合わせで 45 通りのペアに対してこの作業を行った．生成されるサンプルは，オリジナルの被験者データである 2,000 サンプルに対し，27,000 サンプルである．

6.4 被験者データをもとにした線形補間によるデータ生成

5.4 節で記述したように，周期によりピッキングか鍵開けかを分類しているのではないかと推測し，線形補間により周期を 0.7 倍したデータと，1.3 倍したデータを生成した．つ

まり，6.3 節で生成したデータをもとに周期が 0.7 倍と 1.3 倍のデータを生成した．ここで生成されるサンプルは，オリジナルの被験者データである 2,000 サンプルに対し，56,000 サンプルである．

6.5 新たな被験者のデータ取得

7.9 節で記述したように被験者数を 20 名に増やした．そして，訓練用サンプルに被験者 11〜20 の 10 名分のデータを，テスト用サンプルに被験者 1〜10 の中から 1 名分のデータを使用する．なお，テスト用サンプルは被験者をそれぞれ入れ替えて評価を行う．なお，評価には畳込みニューラルネットワークおよびサポートベクターマシンを用いる．

次に，5.5 節で記述したように被験者の骨格座標の値が近くなるようにスライドする．なお，骨格座標をスライドさせるとき，5.5 節および表 5.4 で記述したように，それぞれの Kinect の設置環境にあわせて基準となる座標により，骨格座標をスライドさせ，その被験者データをもとに精度評価を行う．訓練用サンプルに被験者 11〜20 の 10 名分のデータを，テスト用サンプルに被験者 1〜10 の中から 1 名分のデータを使用する．なお，テスト用サンプルは被験者をそれぞれ入れ替えて評価を行う．なお，評価には畳込みニューラルネットワークおよびサポートベクターマシンを用いる．

6.6 被験者データをもとにした一部の骨格座標が隠れた場合のデータ生成

5.6 節で記述したように，ピッキングと鍵開けの動作を行っている右手のみで分類できることを確認するため新たにデータを生成した．なお，遮蔽物により骨格座標が隠れたことを再現するため隠れた骨格座標の値を 0 で置き換えることで再現する．ここで生成されるサンプルは，オリジナルの被験者データである 2,000 サンプルに対し，同じく 2,000 サンプルである．

6.7 実験環境

本研究で，ピッキングと鍵開けを行う際の環境としての，人物と扉と Kinect の位置関係を図 6.1 に示す．

6.1 節で記述したように，被験者がドアの前に立った場合のかかとの開き幅を 14cm，立つ位置としては扉と人物のかかとの距離が 56cm 離れたところに立つ．そして，背筋は伸ばし，頭は少し下を向くようにする．Kinect 環境に関しては 4.2 節で述べたように，被験者からの距離を 300cm とし，角度を被験者の右斜め後ろ，つまり扉に対して 30 度，また被験者の右真横の 2 パターンで設置した．骨格座標取得後のデータ整形に関して，基準とする平均身長は，総務省が発表している 2018 年度日本人男性の平均身長を参考に 170cm とした [54] [55]．

畳込みニューラルネットワークによる学習の際には，ディレクトリ「0」に正常行動のデータを，ディレクトリ「1」に不審行動のデータを入れてラベリングを行った．この際

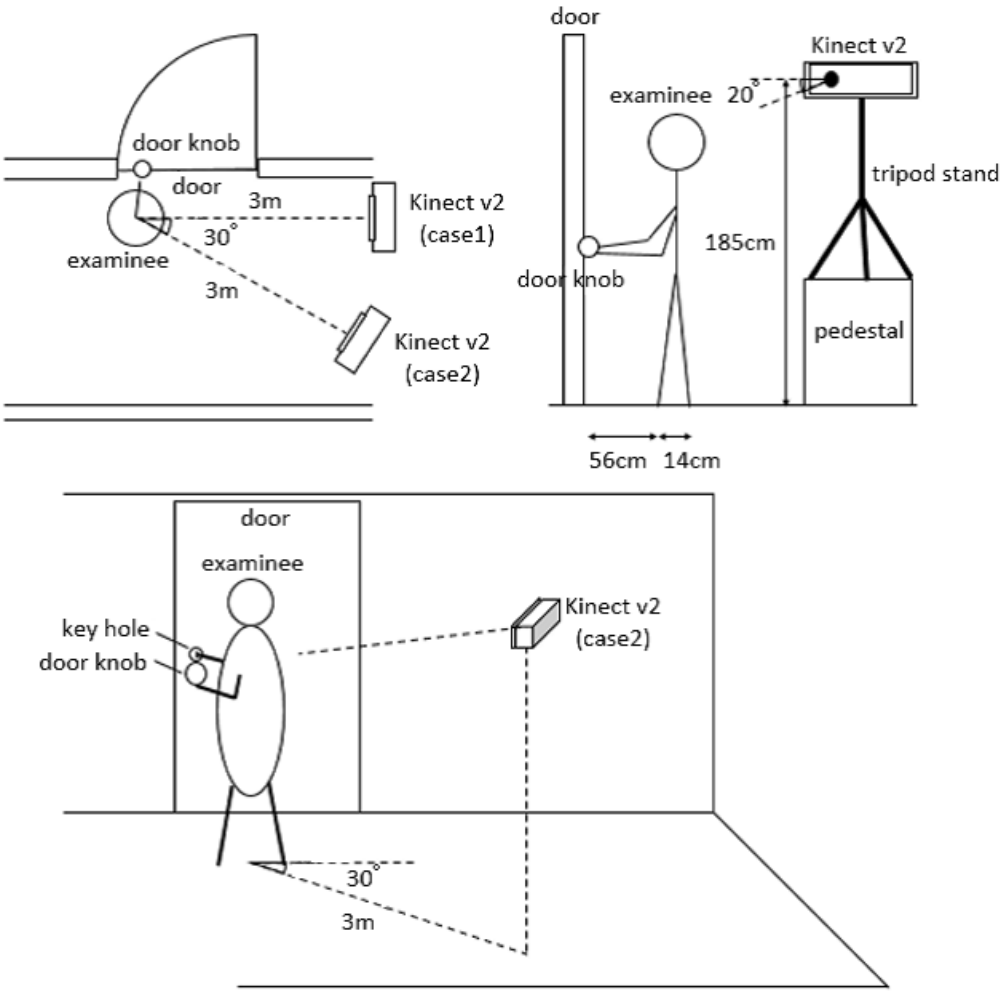

図 6.1: Kinect により骨格座標を取得する実験環境

のバッチサイズは 32 から倍々に増加させ，またエポック数は 50 から 50 ずつ増加させていき，F 値が 90%を超えたものを，本実験で使用可能なバッチサイズ，エポック数として抽出した．さらに，そのなかで標準偏差の値が小さいものを採用している．その際に，10 交差検証を用いて精度評価を行い，それを 5 回繰り返した平均精度をとっている．

次に，テスト用サンプルに被験者 1 名分のデータを入れ，訓練用サンプルに残りの被験者のデータを入れ，学習に使用していない被験者の行動が正しく分類されるのか評価を行った．また，被験者データをもとにサンプルの生成を行い，同様の方法で学習に使用していない被験者の行動が正しく分類されるのか評価を行った．なお，この評価は畳込みニューラルネットワークを用いて行い，それぞれの被験者に対して 10 回繰り返し，その平均値をとっている．

次に，線形補間により生成したデータを訓練用サンプルに加え，被験者 1〜10 の 10 名に対して，テスト用サンプルに被験者 1 名分のデータを入れ，訓練用サンプルに残りの被験者のデータを入れ，学習に使用していない被験者の行動が正しく分類されるのか評価を行った．なお，この評価は畳込みニューラルネットワークを用いて行い，それぞれの被験

者に対して 10 回繰り返し，その平均値をとっている．

次に，被験者 20 名に対して，テスト用サンプルに被験者 1〜10 の中から 1 名分のデータを入れ，訓練用サンプルに被験者 11〜20 の 10 名分のデータを入れ，学習に使用していない被験者の行動が正しく分類されるのか評価を行った．なお，この評価は畳込みニューラルネットワークおよびサポートベクターマシンを用いて行い，それぞれの被験者に対して 10 回繰り返し，その平均値をとっている．

次に，遮蔽物により骨格座標の一部が隠れた場合を再現し，被験者の行動が正しく分類されるのか評価を行った．本研究では，右腕の指先から肩までの骨格座標を取得した場合，右腕の指先から肩まで以外の骨格座標を取得した場合，右腕の指先から肘まで以外の骨格座標を取得した場合，右腕の指先から手首まで以外の骨格座標を取得した場合，の 4 つの場合において評価を行った．なお，この評価は畳込みニューラルネットワークおよびサポートベクターマシンを用いて行い，それぞれの被験者に対して 10 回繰り返し，その平均値をとっている．畳込みニューラルネットワークによる学習の際には，F 値が 90%を超えたものを，本実験で使用可能なバッチサイズ，エポック数として抽出した．さらに，そのなかで標準偏差の値が小さいものを採用している．その際に，10 交差検証を用いて精度評価を行い，それを 5 回繰り返した平均精度をとっている．

第7章

評価と考察

7.1 被験者情報

本研究では，被験者 20 名に対して骨格座標データの取得を行った．**表 7.1** はその被験者 20 名の身長を記したものである．なお，被験者 11〜20 の 10 名に関して，線形補間による精度評価を行った後に取得したものである．

表 7.1: 被験者の身長

被験者番号	身長 [cm]
被験者 1	169
被験者 2	165
被験者 3	163
被験者 4	172
被験者 5	161
被験者 6	171
被験者 7	170
被験者 8	170
被験者 9	170
被験者 10	163
被験者 11	177
被験者 12	165
被験者 13	171
被験者 14	166
被験者 15	171
被験者 16	166
被験者 17	164
被験者 18	182
被験者 19	172
被験者 20	168

7.2　鍵開けおよびピッキング動作の再現性

ここで，本研究で取得したサンプルにおいて，鍵開けおよびピッキング動作が再現できているのかを確認した．そのうえで，実際に犯人がピッキングを行っている動画をインターネット上で検索したが，存在しないように思われた．そこで，鍵屋のホームページに存在するピッキング動作を参照した [56]．ピッキングでは，鍵穴にあるピンが揃った場合にまわすための針金と，ピンを正しい位置にするための針金の 2 種類を用いて行っていた．右手でピンを正しい位置にするための針金をもつ．そして，鍵穴上部にあるピンを一つずつ適正な位置にするために，右手首を上下に動かす．なお，鍵穴の中にピンに針金を当てる際，針金を斜め上に向くように動かしていた．また，ピッキングの可動域や動作速度は不規則であった．よって，本研究で行ったピッキング動作は，鍵屋の方が行っているピッキング動作と異なっており，犯人が行うピッキング動作と違うことが考えられる．つまり，ピッキングは再現されていなかった．

一方，本研究が目的としているのは，完全なピッキング動作を再現することではなく，類似の動作を機械学習で区別する際のデータの重要性である．つまり，完全なピッキング動作を再現できていなくても，ピッキングのような動作と鍵開けの動作に相違点があり，かつ類似であればよい．そこで，本研究で行ったピッキングの模倣動作と鍵開け動作が，どのような動きとなっているのか確認した．**図 7.1** に各動作の比較を示す．なお，左側に鍵開け動作，右側にピッキングの模倣動作を示している．上段には被験者が動作を行っている際の手元を拡大したものである．なお，本研究では，6.1 節で記述したように，動作を行っている右手以外に違いが出ないよう，被験者の立ち位置や姿勢は同じになるようにした．そのため，動作の違いがあらわれる手元が分かりやすいよう画像を拡大した．上から二段目に右手の動作を示し，上から三段目には右手の先端を示し，三段目には右手の親指を示している．

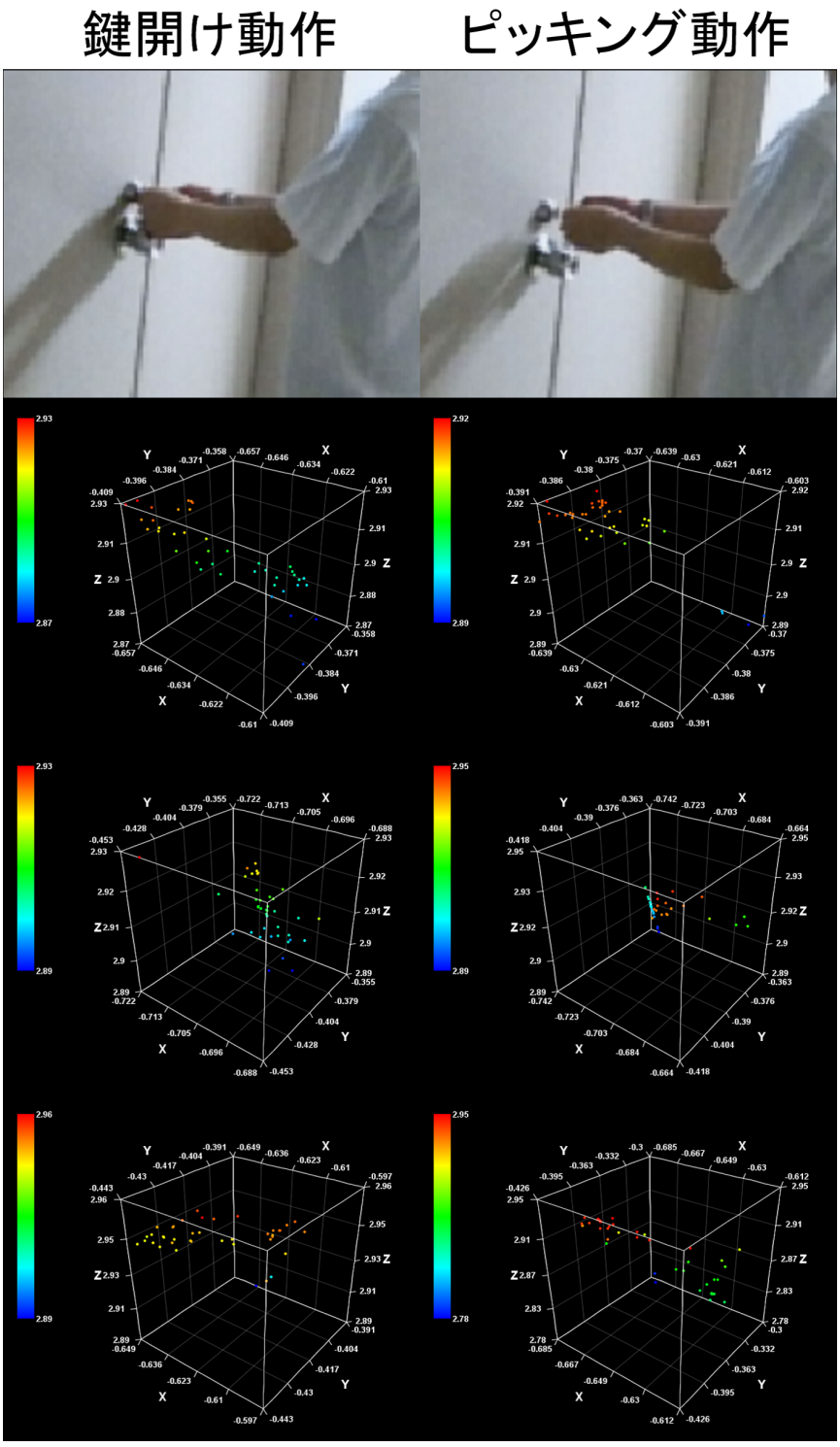

図 7.1: 鍵開け動作（左側）およびピッキング動作（右側）の再現性

図 7.1 の結果，上段より，画像は類似動作しているが，視覚による判断は可能であると思われる．骨格座標の値における判別も特徴があらわれており可能であると思われる．な

お，鍵開け動作における骨格座標の値の特徴として，HandRight は Z 軸のさまざまな値に一直線上に連なっており，HandTipRight は一点に集中しているもののばらつきがある．ThumbRight は Z 軸の同値周辺に一直線上に連なっている．手先の回転により鍵開け動作が再現されていると思われた．一方，ピッキング動作における骨格座標の値の特徴として，HandRight は一点に集中しているものの若干ではあるがばらつきがある．HandTipRight は一点に集中している．ThumbRight は Z 軸のさまざまな値に一直線上に連なっている．手元の小さな動作によりピッキング動作が再現されていると思われた．よって，鍵開けとピッキングの模倣動作は異なる動きであり，区別することは可能だと考えられた．

しかし，手元に特徴があらわれていると考えられるため，手元が隠れた場合に動作を行っている右手のどの骨格座標がみえていればどのくらいの精度で判別可能か不明である．そのため，右手の骨格座標を徐々にみえるようにしていったときに精度がどのようになるかをあわせて調べた．

7.3 マシンスペック

本研究で使用する機械学習用サーバのスペックを以下に示す．プロセッサは Intel(R) Core(TM) i5-7600 CPU，クロック周波数は 3.50GHz，コア数は 4 となっている．メインメモリのサイズは 64GB である．また，ビデオカードは Intel HD Graphics 630 である．

本研究で使用した最大のデータ数，データ容量はそれぞれ 6.4 節における線形補間データを生成した際であり，それぞれ 66,800 データで，約 2.11GB である．なお，データの容量上，約 3GB を超えてしまうと機械学習を行えなくなってしまう．そのため，被験者 20 名に関して，データ生成は行わなかった．

7.4 全骨格座標を取得した場合の評価

全骨格座標を取得した場合の評価を行う前に，エポック数とバッチサイズを決定する必要がある．そこで，被験者 1〜10 のサンプルを用いて，10 分割交差検証により最適なエポック数とバッチサイズを調べた．本実験の事前実験では 10 分割交差検証を 5 セット行い，評価をとった．

被験者 1 名あたりのサンプル数は 200 であり，全被験者の合計サンプル数は 2,000 である．また，テスト用サンプルと訓練用サンプルにそれぞれの被験者のデータがはいるように設定しており，テスト用サンプルに含まれるピッキングと鍵開けのサンプルの割合はそれぞれ 100 である．エポック数は 50 から 50 ずつ増加させ，バッチサイズは 32 から倍々にしていき，1,800 になるまでの値について精度評価を行った．本研究で用いるエポック数とバッチサイズは F 値が 90%を超えたときに標準偏差が 5%以下のものを用いるものとした．**表 7.2** は，エポック数 50 のときにバッチサイズを変更した場合の精度の結果である．また**表 7.3** は，エポック数 100 のときにバッチサイズを変更した場合の精度の結果である．

表 7.2: 全骨格座標におけるエポック 50 とバッチサイズとの関係

エポック数, バッチサイズ		precision	recall	F 値
epoch 50, batchsize 32	平均	0.85	0.83	0.83
	標準偏差	0.0779	0.0962	0.115
epoch 50, batchsize 64	平均	0.84	0.82	0.82
	標準偏差	0.0669	0.0771	0.0853
epoch 50, batchsize 128	平均	0.84	0.83	0.83
	標準偏差	0.0611	0.0670	0.0677
epoch 50, batchsize 256	平均	0.80	0.79	0.78
	標準偏差	0.0558	0.0674	0.0727
epoch 50, batchsize 512	平均	0.74	0.72	0.72
	標準偏差	0.0496	0.0495	0.0540
epoch 50, batchsize 1024	平均	0.70	0.67	0.65
	標準偏差	0.0328	0.0387	0.0528
epoch 50, batchsize 1800	平均	0.65	0.62	0.60
	標準偏差	0.0546	0.0426	0.0541

表 7.3: 全骨格座標におけるエポック 100 とバッチサイズとの関係

エポック数, バッチサイズ		precision	recall	F 値
epoch 100, batchsize 32	平均	0.90	0.89	0.88
	標準偏差	0.121	0.122	0.152
epoch 100, batchsize 64	平均	0.90	0.89	0.89
	標準偏差	0.0624	0.0749	0.0791
epoch 100, batchsize 128	平均	0.91	0.91	0.90
	標準偏差	0.0485	0.0538	0.0549
epoch 100, batchsize 256	平均	0.91	0.90	0.90
	標準偏差	0.0381	0.0418	0.0416
epoch 100, batchsize 512	平均	0.84	0.83	0.82
	標準偏差	0.0547	0.0589	0.0597
epoch 100, batchsize 1024	平均	0.77	0.76	0.75
	標準偏差	0.0523	0.0555	0.0578
epoch 100, batchsize 1800	平均	0.72	0.70	0.69
	標準偏差	0.0397	0.0428	0.0492

表 7.2 の結果から，エポック数が 50 のときにバッチサイズを変更しても F 値が 90%を超える場合は存在しなかった．表 7.3 の結果から，エポック数が 100 のとき，バッチサイズが 128，256 の場合に F 値が 90%超を示した．そのなかで，標準偏差が 5%以下となったのはバッチサイズが 256 の場合で，その標準偏差は 4.16%であった．以上の結果から，

本研究ではエポック数を 100，バッチサイズを 256 とし，以降の実験を行っていく．

なお，本実験では訓練用サンプルが少ないため，過学習を起こしている可能性がある．そこで，訓練（train）と検証（validation）の loss を取得し，**図 7.2** に示した．

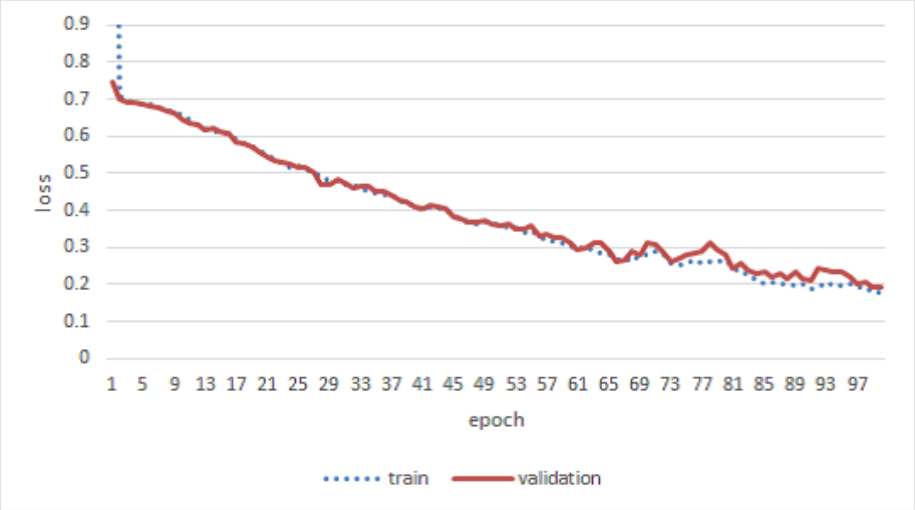

図 7.2: 全骨格座標におけるエポック数とバッチサイズを決定する実験における loss

その結果，訓練においては，85〜96 エポックで loss がおよそ 0.20 となり，100 エポックで 0.18 まで下がった．検証においては，84〜96 エポックで loss が 0.21 から 0.24 の間を行き来し，100 エポックで 0.19 まで下がった．loss は減少し続けているが，過学習は起きていないことが確認できた．

7.5　全骨格座標取得における被験者 10 名での精度評価

7.4 節で述べたように，エポック数に 100 を，バッチサイズに 256 を設定した．そして，被験者 1 名のデータをテスト用サンプルに，残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．なお，ここで使用するサンプルは被験者 1〜10 の 10 名分のデータである．CNN における結果を**表 7.4** に，SVM における結果を**表 7.5** に示す．なお，それぞれの被験者に対して 10 回精度評価を行い，その平均と標準偏差を示している．

表 7.4: 全骨格座標における CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.64	0.58	0.54
	標準偏差	0.128	0.0857	0.0954
被験者 2	平均	0.68	0.62	0.60
	標準偏差	0.100	0.0673	0.0673
被験者 3	平均	0.56	0.55	0.54
	標準偏差	0.149	0.127	0.132
被験者 4	平均	0.64	0.53	0.44
	標準偏差	0.201	0.0736	0.0766
被験者 5	平均	0.76	0.54	0.42
	標準偏差	0.013	0.041	0.0811
被験者 6	平均	0.85	0.79	0.76
	標準偏差	0.0623	0.114	0.148
被験者 7	平均	0.56	0.50	0.42
	標準偏差	0.155	0.0518	0.0891
被験者 8	平均	0.59	0.58	0.58
	標準偏差	0.0724	0.0619	0.0604
被験者 9	平均	0.60	0.56	0.54
	標準偏差	0.0949	0.0321	0.0264
被験者 10	平均	0.80	0.76	0.75
	標準偏差	0.102	0.0978	0.0968
被験者全体	平均	0.67	0.60	0.56
	標準偏差	0.154	0.122	0.149

表 7.4 の結果から，被験者全体の平均 F 値は 56%と低く，その標準偏差も被験者平均で 14.9%とばらつきがあることがわかる．被験者ごとに平均 F 値をみると，40%台のものから 80%近くのものまであり，一部の被験者に関してはある程度高い精度で分類が行えている．しかし，ほとんどの被験者の分類精度が低い．

表 7.4 をみると，被験者全体の平均 F 値は 56%となっており，その標準偏差も 14.9%とばらつきがあり，うまく二値分類できていない．ピッキングと鍵開けの二値分類であるため，当てずっぽうで答えても 50%で正解する．被験者ごとにみると被験者 3 の F 値は 54%，被験者 7 の F 値は 42%となっている一方で，被験者 10 の F 値は 75%である．このように被験者ごとのばらつきが大きい．

表 7.5: 全骨格座標における SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.74	0.60	0.54
	標準偏差	0.0156	0.0498	0.0888
被験者 2	平均	0.82	0.75	0.72
	標準偏差	0.0544	0.119	0.149
被験者 3	平均	0.75	0.52	0.37
	標準偏差	0.00640	0.0202	0.0445
被験者 4	平均	0.23	0.27	0.24
	標準偏差	0.00600	0.00539	0.00458
被験者 5	平均	0.76	0.52	0.38
	標準偏差	0.00671	0.0176	0.0338
被験者 6	平均	0.75	0.72	0.70
	標準偏差	0.0425	0.0560	0.0639
被験者 7	平均	0.77	0.58	0.47
	標準偏差	0.0253	0.0856	0.139
被験者 8	平均	0.76	0.71	0.69
	標準偏差	0.0241	0.0781	0.114
被験者 9	平均	0.76	0.65	0.60
	標準偏差	0.0249	0.0749	0.114
被験者 10	平均	0.70	0.68	0.66
	標準偏差	0.0490	0.0196	0.0205
被験者全体	平均	0.70	0.60	0.54
	標準偏差	0.163	0.148	0.182

また，表 7.5 の結果から，被験者全体の平均の F 値は 54%と低く，その標準偏差も被験者平均で 18.2%とばらつきがあることがわかる．被験者ごとに平均の F 値をみると，20%台のものから 70%近くのものまであり，一部の被験者に関してはある程度高い精度で分類が行えている．しかし，ほとんどの被験者の分類精度が低い．

表 7.5 をみると，被験者全体の平均 F 値は 54%となっており，その標準偏差も 18.2%とばらつきがあり，うまく二値分類できていない．ピッキングと鍵開けの二値分類であるため，当てずっぽうで答えても 50%で正解する．被験者ごとにみると被験者 3 の F 値は 37%，被験者 4 の F 値は 24%となっている一方で，被験者 2 の F 値は 72%である．このように被験者ごとのばらつきが大きい．

この原因として，被験者 9 名分のデータを訓練用サンプルとして，被験者 1 名のデータをテスト用サンプルとした際に，訓練用サンプルの被験者数が少ないことから，テスト用サンプルの被験者と骨格座標がほぼ一致するサンプルが現れず，うまく畳込みが行われなかったのではないかと考えた．訓練用サンプルの被験者数を増やせばこの問題が解決する可能性があるが，7.3 節で述べたように，被験者数を増やすと本環境では学習が行えなくなること，非常に多くの被験者を集めることは本実験では困難であることを考慮し，少な

い被験者数で精度の向上を確認できる方法を考えた．その方法の一つが，6.3 節で述べた，被験者データをもとにしたサンプル生成である．

7.6 全骨格座標取得における被験者データをもとに生成したサンプルを追加した場合の被験者 10 名での精度評価

訓練用サンプルのそれぞれの被験者のデータを用い，仮想的に体格の異なる被験者を作り出した．5.3 節に述べた方法で仮想的にサンプル数を増加させている．なお，テスト用サンプルの被験者のデータは訓練用サンプルのデータを作り出すのに使用していない．

この方法でサンプル追加し，被験者 1 名のデータをテスト用サンプルに，残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．ここで使用したサンプルは被験者 1〜10 の 10 名分である．CNN における結果を**表 7.6** に，SVM における結果を**表 7.7** に示す．なお，それぞれの被験者に対して 10 回精度評価を行い，その平均と標準偏差を示している．

表 7.6: 全骨格座標における生成したサンプルを追加した場合の CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.58	0.51	0.46
	標準偏差	0.109	0.172	0.0543
被験者 2	平均	0.66	0.62	0.61
	標準偏差	0.136	0.115	0.115
被験者 3	平均	0.72	0.60	0.55
	標準偏差	0.0814	0.0617	0.0926
被験者 4	平均	0.59	0.50	0.44
	標準偏差	0.215	0.124	0.116
被験者 5	平均	0.67	0.53	0.41
	標準偏差	0.129	0.0318	0.0730
被験者 6	平均	0.93	0.91	0.91
	標準偏差	0.0377	0.0545	0.0572
被験者 7	平均	0.61	0.57	0.53
	標準偏差	0.125	0.0754	0.0851
被験者 8	平均	0.77	0.66	0.62
	標準偏差	0.0605	0.104	0.140
被験者 9	平均	0.74	0.65	0.61
	標準偏差	0.105	0.114	0.155
被験者 10	平均	0.84	0.82	0.82
	標準偏差	0.0453	0.0564	0.0599
被験者全体	平均	0.71	0.64	0.59
	標準偏差	0.158	0.152	0.182

表 7.6 の結果から，被験者 10 名の平均 F 値は 59%で分類は行えていない．なお，F 値の平均標準偏差も 18.2%と，表 7.4 の 14.9%と比較して大きくなっている．しかし，被験者ごとに F 値をみると，被験者 6 と被験者 10 においてそれぞれ 91%と 82%となっており，分類されている被験者もいる．

表 7.6 をみると，被験者全体の平均 F 値は 59%となっており，標準偏差は 18.2%とばらつきが表 7.4 の同一箇所の値より大きくなっている．被験者ごとにみても，一番低い F 値は被験者 5 の 41%であり，一番高い F 値は被験者 6 の 91%で分類できている被験者もいる．また，表 7.4 と 表 7.6 を比較すると，被験者 6 は F 値が 76%から 91%と，被験者 10 は 75%から 82%と向上しており，被験者の擬似的な増加とサンプルの水増しは CNN における F 値の向上に有効であった．

表 7.7: 全骨格座標における生成したサンプルを追加した場合の SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.76	0.56	0.45
	標準偏差	0.00640	0.0257	0.0484
被験者 2	平均	0.84	0.80	0.80
	標準偏差	0.0218	0.0385	0.0400
被験者 3	平均	0.76	0.52	0.39
	標準偏差	1.11E-16	0.00663	0.0102
被験者 4	平均	0.21	0.24	0.22
	標準偏差	0.00458	0.00490	0.00458
被験者 5	平均	0.65	0.51	0.36
	標準偏差	0.00490	0.00640	0.0143
被験者 6	平均	0.74	0.74	0.74
	標準偏差	0.0280	0.0310	0.0310
被験者 7	平均	0.75	0.52	0.36
	標準偏差	0.000	0.00500	0.00900
被験者 8	平均	0.81	0.81	0.81
	標準偏差	0.0175	0.0324	0.0366
被験者 9	平均	0.80	0.66	0.62
	標準偏差	0.0168	0.0569	0.0827
被験者 10	平均	0.69	0.68	0.67
	標準偏差	0.0215	0.0195	0.0253
被験者全体	平均	0.71	0.61	0.54
	標準偏差	0.172	0.165	0.203

次に，表 7.7 の結果から，被験者 10 名の平均 F 値は 54%で分類は行えていない．なお，F 値の平均標準偏差も 20.3%と，表 7.5 の 18.2%と比較して大きくなっている．しかし，被験者ごとに F 値をみると，被験者 2 と被験者 8 においてそれぞれ 80%と 81%となっており，分類されている被験者もいる．

表 7.7 をみると，被験者全体の平均 F 値は 54%となっており，標準偏差は 20.3%とばらつきが表 7.5 の同一箇所の値より大きくなっている．被験者ごとにみても，一番低い F 値は被験者 4 の 22%であり，一番高い F 値は被験者 8 の 81%で分類できている被験者もいる．また，表 7.5 と 表 7.7 を比較すると，被験者 2 は F 値が 72%から 80%と，被験者 6 は F 値が 70%から 74%と，被験者 8 は 69%から 81%と向上しており，SVM における F 値が 70%超となった被験者は 2 名から 3 名となり，被験者の擬似的な増加とサンプルの水増しは SVM における F 値の向上に有効であった．

表 7.4 および表 7.6，並びに表 7.5 および表 7.7 の精度の違いは，被験者 10 名のデータの他に，そこから生成した擬似的な 45 名分のデータを追加したことによるものである．これは，55 名程度の被験者を集めて実験を行えば，一部の高精度を示した被験者に関しては

精度を向上させることができるが，工夫をすれば 10 名の被験者でも同程度の精度を出せるということを示しており，2 者間の擬似的なサンプル生成だけではなく，3 者間の擬似的なサンプル生成をすることで，さらなる F 値の向上につながると考えられる．

この時点で，まだ精度が低いと感じられたため，次に，5.4 節の表 5.2，表 5.3 で示したように，鍵開けとピッキングの行動周期に注目した．周期が類似する訓練用サンプルが存在すれば，テスト用サンプルをうまく分類できるのではないかということである．そこで，4.5 節で述べた線形補間によって，異なる周期をもつサンプルを仮想的に作成した．

7.7　線形補間データを追加した場合の被験者ごとの精度評価

4.5 節で述べた線形補間によって，異なる周期をもつサンプルを仮想的に作成し，被験者 1 名のデータをテスト用サンプルに，残りの被験者のデータを訓練用サンプルにして，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．なお，ここで使用するサンプルは被験者 1〜10 の 10 名分のデータである．CNN における結果を**表 7.8** に，SVM における結果を**表 7.9** に示す．なお，それぞれの被験者に対して 10 回精度評価を行い，その平均と標準偏差を示している．

表 7.8: 全骨格座標における線形補間により生成したサンプルを追加した場合の CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.64	0.55	0.48
	標準偏差	0.114	0.0593	0.0969
被験者 2	平均	0.59	0.57	0.55
	標準偏差	0.0868	0.0707	0.0701
被験者 3	平均	0.61	0.53	0.46
	標準偏差	0.166	0.0981	0.109
被験者 4	平均	0.35	0.39	0.35
	標準偏差	0.0502	0.0422	0.0473
被験者 5	平均	0.61	0.51	0.39
	標準偏差	0.152	0.0435	0.0522
被験者 6	平均	0.95	0.94	0.94
	標準偏差	0.0437	0.0582	0.0641
被験者 7	平均	0.65	0.56	0.51
	標準偏差	0.125	0.0756	0.103
被験者 8	平均	0.73	0.62	0.57
	標準偏差	0.0731	0.0707	0.109
被験者 9	平均	0.70	0.62	0.58
	標準偏差	0.125	0.117	0.151
被験者 10	平均	0.87	0.83	0.83
	標準偏差	0.0492	0.0809	0.0861
被験者全体	平均	0.67	0.61	0.57
	標準偏差	0.188	0.170	0.199

表 7.8 の結果から，被験者 10 名の平均 F 値は 57%で分類は行えていない．なお，F 値の平均標準偏差も 19.9%と，表 7.4，表 7.6 の 14.9%，18.2%と比較して大きくなっている．しかし，被験者ごとに F 値をみると，被験者 6 と被験者 10 においてそれぞれ 94%と 83%となっており，分類されている被験者もいる．

表 7.8 をみると，被験者全体の平均 F 値は 57%となっており，標準偏差は 19.9%とばらつきが表 7.4，表 7.6 の同一個所の値より大きくなっている．被験者ごとにみると，一番低い F 値は被験者 4 の 35%であり，一番高い F 値は被験者 6 の 94%で分類できている被験者もいる．表 7.6 と表 7.8 を比較すると，平均精度は 2%しか変わっていない．しかし，最小値と最大値が 41%から 35%と，91%から 94%とレンジが拡大しており，ばらつきが大きくなっている．高精度を示した被験者 6 と被験者 10 に関してはそれぞれ 91%から 94%，82%から 83%と精度は向上しているが，標準偏差を考慮するとばらつきの範囲内である．よって周期による差は影響がないことが判明した．

表 7.9: 全骨格座標における線形補間により生成したサンプルを追加した場合の SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.69	0.55	0.43
	標準偏差	0.0266	0.0175	0.0400
被験者 2	平均	0.88	0.86	0.85
	標準偏差	0.0301	0.0452	0.0441
被験者 3	平均	0.76	0.54	0.42
	標準偏差	1.11E-16	0.00539	0.0112
被験者 4	平均	0.22	0.25	0.23
	標準偏差	0.00400	0.00400	0.00400
被験者 5	平均	0.75	0.51	0.35
	標準偏差	0.00	0.00447	0.0110
被験者 6	平均	0.79	0.79	0.78
	標準偏差	0.0271	0.0257	0.0250
被験者 7	平均	0.75	0.53	0.39
	標準偏差	0.0179	0.0112	0.0233
被験者 8	平均	0.84	0.82	0.81
	標準偏差	0.0185	0.0380	0.0420
被験者 9	平均	0.80	0.65	0.60
	標準偏差	0.0162	0.0482	0.0652
被験者 10	平均	0.68	0.66	0.65
	標準偏差	0.0194	0.00663	0.0101
被験者全体	平均	0.71	0.51	0.55
	標準偏差	0.176	0.174	0.210

表 7.9 の結果から，被験者 10 名の平均 F 値は 55%で分類は行えていない．なお，F 値の平均標準偏差も 21.0%と，表 7.5，表 7.7 の 18.2%，20.3%と比較して大きくなっている．しかし，被験者ごとに F 値をみると，被験者 2 と被験者 8 においてそれぞれ 85%と 81%となっており，分類されている被験者もいる．

表 7.8 をみると，被験者全体の平均 F 値は 55%となっており，標準偏差は 21.0%とばらつきが表 7.5，表 7.7 の同一個所の値より大きくなっている．被験者ごとにみると，一番低い F 値は被験者 4 の 23%であり，一番高い F 値は被験者 6 の 85%で分類できている被験者もいる．表 7.7 と表 7.9 を比較すると，平均精度は 1%しか変わっていない．しかし，最小値と最大値が 22%から 23%と，81%から 85%とレンジが拡大しており，ばらつきが大きくなっている．高精度を示した被験者 2 と被験者 8 に関してはそれぞれ 80%から 85%，81%から 81%と精度は向上しているが，標準偏差を考慮するとばらつきの範囲内である．よって周期による差は影響がないことが判明した．

精度を向上させる残された方法は，単純な被験者増である．7.3 節で述べたように，本環境では大量の被験者のデータを学習させることはできず，また，大量の被験者を集める

ことも困難であるため，被験者数を 10 名から 20 名に増やし，そのうち被験者 11〜20 の 10 名を訓練用サンプルとし，被験者 1〜10 の 10 名の中から 1 名ずつ入れ替えながらテスト用サンプルを作成し，精度評価を行った．

7.8　全骨格座標における被験者 20 名による被験者ごとの精度評価

被験者 10 名分のデータを追加で取得し，被験者 20 名分のデータを用いて精度評価を行った．被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20 の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.10** に，SVM による結果を**表 7.11** に示す．

表 7.10: 全骨格座標における CNN による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.63	0.53	0.48
	標準偏差	0.113	0.0225	0.0535
被験者 2	平均	0.52	0.52	0.51
	標準偏差	0.0374	0.0347	0.0332
被験者 3	平均	0.68	0.59	0.54
	標準偏差	0.126	0.0931	0.122
被験者 4	平均	0.72	0.57	0.48
	標準偏差	0.111	0.0545	0.0752
被験者 5	平均	0.63	0.60	0.57
	標準偏差	0.0889	0.0638	0.0776
被験者 6	平均	0.70	0.59	0.52
	標準偏差	0.134	0.115	0.160
被験者 7	平均	0.53	0.52	0.52
	標準偏差	0.0490	0.0452	0.0435
被験者 8	平均	0.66	0.56	0.49
	標準偏差	0.109	0.0681	0.115
被験者 9	平均	0.58	0.57	0.57
	標準偏差	0.0454	0.0424	0.0406
被験者 10	平均	0.65	0.56	0.51
	標準偏差	0.100	0.0471	0.0986
被験者全体	平均	0.63	0.56	0.52
	標準偏差	0.117	0.0967	0.0963

表 7.10 の結果から，CNN による被験者 10 名の平均 F 値は 52%で分類は行えていない．なお，その標準偏差も被験者平均で 9.63%とばらつきがあることがわかる．

表 7.11: 全骨格座標における SVM による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.86	0.80	0.79
	標準偏差	0.0316	0.0591	0.0688
被験者 2	平均	0.95	0.94	0.94
	標準偏差	0.0316	0.0441	0.0472
被験者 3	平均	0.82	0.73	0.70
	標準偏差	0.0367	0.0805	0.101
被験者 4	平均	0.74	0.51	0.37
	標準偏差	0.0432	0.00490	0.0128
被験者 5	平均	0.82	0.75	0.73
	標準偏差	0.0125	0.0500	0.0654
被験者 6	平均	0.93	0.91	0.91
	標準偏差	0.0530	0.0712	0.0741
被験者 7	平均	0.76	0.71	0.69
	標準偏差	0.0161	0.0624	0.0815
被験者 8	平均	0.86	0.78	0.76
	標準偏差	0.0512	0.106	0.126
被験者 9	平均	0.86	0.83	0.82
	標準偏差	0.0324	0.0394	0.0408
被験者 10	平均	0.90	0.88	0.86
	標準偏差	0.0502	0.111	0.142
被験者全体	平均	0.85	0.78	0.76
	標準偏差	0.0743	0.135	0.174

表 7.11 の結果から，SVM による被験者 10 名の平均 F 値は 76%となっている．しかし，その標準偏差も被験者平均で 17.4%とばらつきがある．これは被験者 4 において，F 値の平均が 37%となっていることもあるが，一方で，F 値が 60%台が 1 名，70%台が 4 名，80%台が 2 名，90%台が 2 名となっている．

以上より，被験者数を増加させれば，分類アルゴリズムによっては精度が高くなることが示された．この場合，CNN より SVM のほうがうまく二値分類できている．ここで，畳込みニューラルネットワークによる精度評価がうまくいかなかったのは，被験者一人ひとりの行動する位置がずれているのではないかと考え，骨格座標の値が近くなるように座標をスライドさせ，再度評価を行った．

7.9 全骨格座標における座標をスライドした場合の被験者 20 名での精度評価

全被験者の骨格座標の値が近いものになるよう，左かかとを基準にして骨格座標をスライドさせ，精度評価を行った．7.8 節と同様に，被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20 の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.12** に，SVM による結果を**表 7.13** に示す．

表 7.12: 骨格座標のスライドにおける CNN による被験者 20 名における被験者ごとの行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.62	0.54	0.51
	標準偏差	0.126	0.0388	0.0386
被験者 2	平均	0.51	0.50	0.50
	標準偏差	0.0102	0.00800	0.0111
被験者 3	平均	0.60	0.55	0.52
	標準偏差	0.141	0.0940	0.0923
被験者 4	平均	0.67	0.56	0.48
	標準偏差	0.0689	0.0252	0.0618
被験者 5	平均	0.45	0.44	0.38
	標準偏差	0.188	0.0911	0.0954
被験者 6	平均	0.79	0.67	0.62
	標準偏差	0.0804	0.125	0.167
被験者 7	平均	0.60	0.56	0.50
	標準偏差	0.0742	0.0550	0.0819
被験者 8	平均	0.73	0.59	0.51
	標準偏差	0.0498	0.0623	0.119
被験者 9	平均	0.70	0.61	0.57
	標準偏差	0.124	0.0703	0.0785
被験者 10	平均	0.61	0.56	0.53
	標準偏差	0.122	0.0735	0.0914
被験者全体	平均	0.63	0.56	0.51
	標準偏差	0.145	0.0934	0.110

表 7.12 の結果から，骨格座標のスライドにおける CNN による被験者 10 名の平均 F 値は 51%で分類は行えていない．なお，その標準偏差も被験者平均で 11.0%とばらつきがある．

表 7.13: 骨格座標のスライドにおける SVM による被験者 20 名における被験者ごとの行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.76	0.60	0.53
	標準偏差	0.0323	0.0346	0.0643
被験者 2	平均	0.95	0.94	0.94
	標準偏差	0.0341	0.0385	0.0406
被験者 3	平均	0.80	0.66	0.60
	標準偏差	0.0349	0.0887	0.115
被験者 4	平均	0.42	0.48	0.36
	標準偏差	0.0356	0.0150	0.00831
被験者 5	平均	0.79	0.71	0.69
	標準偏差	0.0686	0.0402	0.0356
被験者 6	平均	0.95	0.94	0.94
	標準偏差	0.0422	0.0558	0.0567
被験者 7	平均	0.69	0.67	0.65
	標準偏差	0.0429	0.0196	0.0210
被験者 8	平均	0.92	0.91	0.91
	標準偏差	0.0276	0.0385	0.0410
被験者 9	平均	0.85	0.82	0.81
	標準偏差	0.0199	0.00917	0.0118
被験者 10	平均	0.88	0.84	0.83
	標準偏差	0.0310	0.0633	0.0682
被験者全体	平均	0.80	0.75	0.73
	標準偏差	0.155	0.157	0.192

表 7.13 の結果から，骨格座標のスライドにおける SVM による被験者 10 名の平均 F 値は 73%となっている．しかし，その標準偏差も被験者平均で 19.2%とばらつきがある．これは被験者 4 において，F 値の平均が 36%となっていることもあるが，一方で，F 値が 60%台が 3 名，80%台が 2 名，90%台が 3 名となっている．

表 7.11 と比較し，骨格座標をスライドさせ，あわせれば，SVM においては F 値が向上しているようにみえるが，標準偏差を考慮すると誤差の範囲といえる．また，CNN には影響はみられなかった．

7.10 右腕の指先から肩までの骨格座標を取得した場合

次に，4.6 節で述べているように，右腕の指先から肩までの骨格座標を取得した場合の評価を行う．まず，エポック数とバッチサイズを決定する必要がある．そこで，被験者 1〜20 にあたる 20 名分のサンプルを用いて，10 分割交差検証により最適なエポック数と

バッチサイズを調べた．本実験の事前実験では，10 分割交差検証を 5 回セット行い評価をとった．

なお，被験者 1 名あたりのサンプル数は 200 であり，全被験者の合計サンプル数は 4,000 である．また，テスト用サンプルと訓練用サンプルにそれぞれの被験者のデータがはいるように設定しており，テスト用サンプルに含まれるピッキングと鍵開けのサンプルの割合はそれぞれ 200 である．なお，本研究で用いるエポック数とバッチサイズは，上記の決定条件と同様である．**表 7.14** は，遮蔽物による右腕の指先から肩まで以外の骨格座標が遮られた場合におけるエポック数 50 のときにバッチサイズを変更した場合の精度の結果である．

表 7.14: 右腕の指先から肩までの骨格座標におけるエポック 50 とバッチサイズとの関係

エポック数, バッチサイズ		precision	recall	F 値
epoch 50, batchsize 32	平均	0.92	0.91	0.91
	標準偏差	0.0761	0.0815	0.0829
epoch 50, batchsize 64	平均	0.94	0.94	0.94
	標準偏差	0.0512	0.0549	0.0550
epoch 50, batchsize 128	平均	0.96	0.95	0.95
	標準偏差	0.0351	0.0380	0.0392
epoch 50, batchsize 256	平均	0.94	0.93	0.93
	標準偏差	0.0268	0.0303	0.0317
epoch 50, batchsize 512	平均	0.90	0.88	0.88
	標準偏差	0.0387	0.0570	0.0608
epoch 50, batchsize 1024	平均	0.82	0.78	0.77
	標準偏差	0.0436	0.0780	0.0917
epoch 50, batchsize 1800	平均	0.72	0.66	0.63
	標準偏差	0.0524	0.0719	0.102

表 7.14 の結果から，エポック数が 50 のとき，バッチサイズが 32，64，128，256 の場合に F 値が 90%超を示した．そのなかで，標準偏差が 5%以下となったのはバッチサイズが 128，256 の場合で，その標準偏差はそれぞれ 3.92%，3.17%であった．以上の結果から，本研究ではエポック数を 50，バッチサイズを 128 とし，遮蔽物による骨格座標における実験を行っていく．

なお，本実験では訓練用サンプルが少ないため，過学習を起こしている可能性がある．そこで，訓練（train）と検証（validation）の loss を取得し，**図 7.3** に示した．

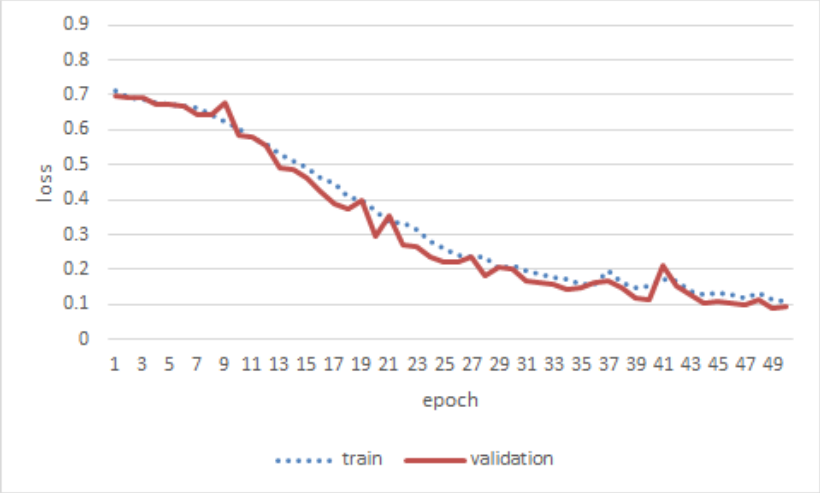

図 7.3: 右腕の指先から肩までの骨格座標におけるエポック数とバッチサイズを決定する実験における loss

その結果，訓練においては，35〜42 エポックで loss がおよそ 0.14 から 0.17 の間を行き来し，50 エポックで 0.11 まで下がった．検証においては，44〜50 エポックで loss がおよそ 0.10 となり，50 エポックで 0.093 まで下がった．loss は減少し続けているが，過学習は起きていないことが確認できた．

7.10.1 右腕の指先から肩までの骨格座標における被験者 10 名での精度評価

次に，右腕の指先から肩までの骨格座標を取得し，被験者 10 名における精度評価を行った．そして，被験者 1〜10 の中から 1 名のデータをテスト用サンプルにし，テスト用サンプルに使用していない残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.15** に，SVM による結果を**表 7.16** に示す．

表 7.15: 右腕の指先から肩までの骨格座標における CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.71	0.65	0.62
	標準偏差	0.108	0.0850	0.107
被験者 2	平均	0.64	0.61	0.60
	標準偏差	0.153	0.114	0.106
被験者 3	平均	0.67	0.64	0.63
	標準偏差	0.0942	0.0702	0.0683
被験者 4	平均	0.28	0.32	0.26
	標準偏差	0.177	0.117	0.00875
被験者 5	平均	0.43	0.46	0.44
	標準偏差	0.103	0.0562	0.0864
被験者 6	平均	0.75	0.63	0.56
	標準偏差	0.0889	0.0897	0.136
被験者 7	平均	0.64	0.59	0.56
	標準偏差	0.130	0.0888	0.0926
被験者 8	平均	0.57	0.57	0.57
	標準偏差	0.0602	0.0551	0.0544
被験者 9	平均	0.80	0.67	0.61
	標準偏差	0.0310	0.00907	0.135
被験者 10	平均	0.74	0.71	0.70
	標準偏差	0.0904	0.0695	0.0680
被験者全体	平均	0.63	0.58	0.55
	標準偏差	0.188	0.138	0.152

表 7.15 の結果から，右腕の指先から肩までの骨格座標における CNN による被験者 10 名の平均 F 値は 55%で分類は行えていない．なお，その標準偏差も被験者平均で 15.2%とばらつきがある．しかし，被験者 10 において平均 F 値は，70%となっている．

表 7.16: 右腕の指先から肩までの骨格座標における SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.71	0.64	0.61
	標準偏差	0.0953	0.0560	0.0622
被験者 2	平均	0.79	0.72	0.71
	標準偏差	0.0597	0.0614	0.0727
被験者 3	平均	0.77	0.58	0.49
	標準偏差	0.0225	0.0709	0.114
被験者 4	平均	0.18	0.21	0.20
	標準偏差	0.0150	0.00943	0.0102
被験者 5	平均	0.77	0.60	0.52
	標準偏差	0.0174	0.0744	0.121
被験者 6	平均	0.84	0.76	0.73
	標準偏差	0.0474	0.127	0.174
被験者 7	平均	0.74	0.60	0.52
	標準偏差	0.0147	0.0925	0.165
被験者 8	平均	0.88	0.82	0.79
	標準偏差	0.0737	0.153	0.188
被験者 9	平均	0.85	0.83	0.83
	標準偏差	0.0623	0.0704	0.0736
被験者 10	平均	0.80	0.71	0.67
	標準偏差	0.0284	0.0976	0.145
被験者全体	平均	0.73	0.65	0.60
	標準偏差	0.196	0.191	0.216

表 7.16 の結果から，右腕の指先から肩までの骨格座標における SVM による被験者 10 名の平均 F 値は 60%となっている．しかし，その標準偏差も被験者平均で 21.6%とばらつきがある．これは被験者 4 において，F 値の平均が 20%となっていることもあるが，一方で，F 値が 60%台が 2 名，70%台が 3 名，80%台が 1 名となっている．

表 7.5 と比較し，表 7.5 で平均 F 値が 70%を超えたのは，被験者 2，被験者 6 の 2 名であった．一方，表 7.16 で平均 F 値が 70%を超えたのは，被験者 2，被験者 6，被験者 8，被験者 9 の 4 名であった．よって，右腕の指先から肩までの骨格座標にピッキングおよび鍵開け動作の特徴が入っていることがわかった．

7.10.2 右腕の指先から肩までの骨格座標における被験者 20 名での行動評価

次に，右腕の指先から肩までの骨格座標を取得し，被験者 20 名における精度評価を行った．被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20 の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.17** に，SVM による結果を**表 7.18** に示す．

表 7.17: 右腕の指先から肩までの骨格座標における CNN による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.54	0.51	0.48
	標準偏差	0.115	0.0422	0.0735
被験者 2	平均	0.56	0.54	0.52
	標準偏差	0.110	0.0810	0.0946
被験者 3	平均	0.79	0.63	0.56
	標準偏差	0.0230	0.0769	0.120
被験者 4	平均	0.78	0.60	0.49
	標準偏差	0.0390	0.102	0.158
被験者 5	平均	0.56	0.54	0.47
	標準偏差	0.194	0.0959	0.125
被験者 6	平均	0.70	0.57	0.50
	標準偏差	0.162	0.112	0.133
被験者 7	平均	0.64	0.54	0.47
	標準偏差	0.123	0.0388	0.0604
被験者 8	平均	0.52	0.48	0.44
	標準偏差	0.132	0.0595	0.0826
被験者 9	平均	0.52	0.48	0.40
	標準偏差	0.186	0.0808	0.0804
被験者 10	平均	0.52	0.51	0.50
	標準偏差	0.124	0.0810	0.0813
被験者全体	平均	0.61	0.54	0.48
	標準偏差	0.166	0.0925	0.113

表 7.17 の結果から，右腕の指先から肩までの骨格座標における CNN による被験者 10 名の平均 F 値は 48%で分類は行えていない．なお，その標準偏差も被験者平均で 11.3%とばらつきがある．

表 7.18: 右腕の指先から肩までの骨格座標における SVM による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.77	0.70	0.67
	標準偏差	0.0415	0.0611	0.0866
被験者 2	平均	0.90	0.86	0.85
	標準偏差	0.0508	0.0774	0.0845
被験者 3	平均	0.98	0.97	0.97
	標準偏差	0.00943	0.00943	0.00943
被験者 4	平均	0.23	0.46	0.32
	標準偏差	0.0298	0.0164	0.00640
被験者 5	平均	0.77	0.63	0.56
	標準偏差	0.00917	0.0671	0.0986
被験者 6	平均	0.90	0.87	0.86
	標準偏差	0.0496	0.0781	0.0845
被験者 7	平均	0.70	0.64	0.61
	標準偏差	0.0302	0.0492	0.0852
被験者 8	平均	0.78	0.74	0.73
	標準偏差	0.0645	0.0977	0.118
被験者 9	平均	0.92	0.90	0.89
	標準偏差	0.0601	0.0961	0.106
被験者 10	平均	0.91	0.87	0.86
	標準偏差	0.0689	0.111	0.122
被験者全体	平均	0.79	0.76	0.73
	標準偏差	0.203	0.166	0.206

表 7.18 の結果から，右腕の指先から肩までの骨格座標における SVM による被験者 10 名の平均 F 値は 73%となっている．しかし，その標準偏差も被験者平均で 20.6%とばらつきがある．これは被験者 4 において，F 値の平均が 32%となっていることもあるが，一方で，F 値が 60%台が 2 名，70%台が 1 名，80%台が 4 名，90%台が 1 名となっているからである．

表 7.11 と比較し，表 7.11 で平均 F 値が 70%を超えたのは，被験者 1，被験者 2，被験者 3，被験者 5，被験者 6，被験者 8，被験者 9，被験者 10 の 8 名であった．一方，表 7.18 で平均 F 値が 70%を超えたのは，被験者 2，被験者 3，被験者 6，被験者 8，被験者 9，被験者 10 の 6 名であった．よって，右腕の指先から肩までの骨格座標にピッキングおよび鍵開け動作の特徴が入っていることがわかった．

以上より，右腕の指先から肩までの骨格座標を取得した場合，CNN より SVM のほうがうまく二値分類できている．

7.11　右腕の指先から肩まで以外の骨格座標を取得した場合

次に，遮蔽物により右腕の指先から肩までの骨格座標が隠れていることを再現し評価を行うために，右腕以外の骨格座標を取得した場合の評価を行う．まず，エポック数とバッチサイズを決定する必要がある．そこで，被験者 1〜20 にあたる 20 名分のサンプルを用いて，10 分割交差検証により最適なエポック数とバッチサイズを調べた．本実験の事前実験では，10 分割交差検証を 5 回セット行い評価をとった．

なお，被験者 1 名あたりのサンプル数は 200 であり，全被験者の合計サンプル数は 4,000 である．また，テスト用サンプルと訓練用サンプルにそれぞれの被験者のデータがはいるように設定しており，テスト用サンプルに含まれるピッキングと鍵開けのサンプルの割合はそれぞれ 200 である．なお，本研究で用いるエポック数とバッチサイズは，上記の決定条件と同様である．

表 7.19 は，遮蔽物による右腕の指先から肩までの骨格座標が遮られた場合におけるエポック数 50 のときにバッチサイズを変更した場合の精度の結果である．また，**表 7.20** は，遮蔽物による右腕の指先から肩までの骨格座標が遮られた場合におけるエポック数 100 のときにバッチサイズを変更した場合の精度の結果である．

表 7.19: 右腕の指先から肩まで以外の骨格座標におけるエポック 50 とバッチサイズとの関係

エポック数, バッチサイズ		precision	recall	F 値
epoch 50, batchsize 32	平均	0.81	0.73	0.68
	標準偏差	0.0600	0.155	0.228
epoch 50, batchsize 64	平均	0.80	0.78	0.77
	標準偏差	0.0740	0.107	0.140
epoch 50, batchsize 128	平均	0.80	0.79	0.78
	標準偏差	0.0407	0.0507	0.0525
epoch 50, batchsize 256	平均	0.75	0.75	0.74
	標準偏差	0.106	0.0490	0.0521
epoch 50, batchsize 512	平均	0.73	0.71	0.71
	標準偏差	0.0410	0.0447	0.0523
epoch 50, batchsize 1024	平均	0.69	0.67	0.66
	標準偏差	0.0352	0.0340	0.0427
epoch 50, batchsize 1800	平均	0.66	0.64	0.63
	標準偏差	0.0345	0.0341	0.0434

表 7.20: 右腕の指先から肩まで以外の骨格座標におけるエポック 100 とバッチサイズとの関係

エポック数, バッチサイズ		precision	recall	F 値
epoch 100, batchsize 32	平均	0.84	0.77	0.72
	標準偏差	0.0885	0.179	0.250
epoch 100, batchsize 64	平均	0.90	0.88	0.87
	標準偏差	0.0642	0.120	0.162
epoch 100, batchsize 128	平均	0.91	0.91	0.90
	標準偏差	0.0307	0.0311	0.0307
epoch 100, batchsize 256	平均	0.88	0.88	0.88
	標準偏差	0.0357	0.0393	0.0390
epoch 100, batchsize 512	平均	0.82	0.81	0.80
	標準偏差	0.0441	0.0446	0.0457
epoch 100, batchsize 1024	平均	0.77	0.75	0.74
	標準偏差	0.0416	0.0485	0.0517
epoch 100, batchsize 1800	平均	0.71	0.69	0.68
	標準偏差	0.107	0.105	0.106

表 7.19 の結果から，エポック数が 50 のときにバッチサイズを変更しても F 値が 90%を超える場合は存在しなかった．表 7.20 の結果から，エポック数が 100 のとき，バッチサイズが 128 の場合に F 値が 90%超を示した．そのなかで，標準偏差が 5%以下となったのはバッチサイズが 128 の場合で，その標準偏差は 3.07%であった．以上の結果から，本研究ではエポック数を 100，バッチサイズを 128 とし，以降の実験を行っていく．

なお，本実験では訓練用サンプルが少ないため，過学習を起こしている可能性がある．そこで，訓練（train）と検証（validation）の loss を取得し，**図 7.4** に示した．

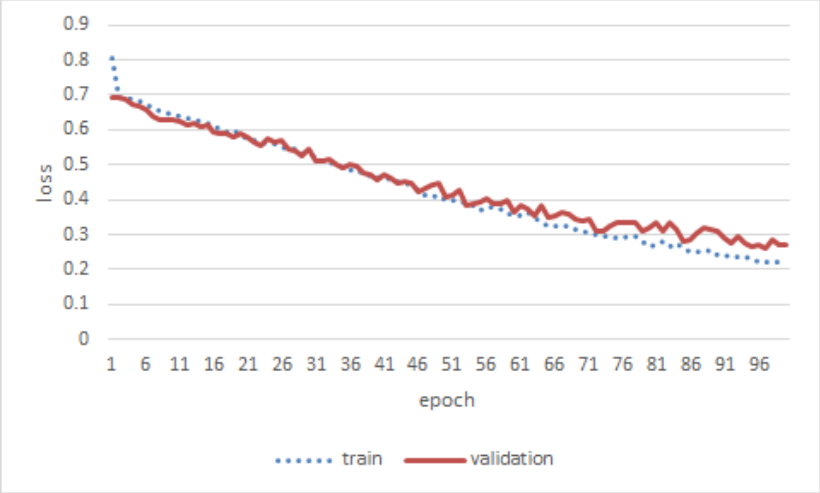

図 7.4: 右腕の指先から肩まで以外の骨格座標におけるエポック数とバッチサイズを決定する実験における loss

その結果，訓練においては，92〜95 エポックで loss がおよそ 0.23 となり，100 エポックで 0.22 まで下がった．検証においては，92〜99 エポックで loss が 0.26 から 0.29 の間を行き来し，100 エポックで 0.27 まで下がった．loss は減少し続けているが，過学習は起きていないことが確認できた．

7.11.1 右腕の指先から肩まで以外の骨格座標における被験者 10 名での行動評価

次に，右腕の指先から肩まで以外の骨格座標を取得し，被験者 10 名における精度評価を行った．そして，被験者 1〜10 の中から 1 名のデータをテスト用サンプルにし，テスト用サンプルに使用していない残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.21** に，SVM による結果を**表 7.22** に示す．

表 7.21: 右腕の指先から肩まで以外の骨格座標における CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.75	0.55	0.45
	標準偏差	0.0240	0.0448	0.0749
被験者 2	平均	0.53	0.52	0.48
	標準偏差	0.145	0.0825	0.104
被験者 3	平均	0.59	0.58	0.55
	標準偏差	0.134	0.118	0.135
被験者 4	平均	0.65	0.56	0.49
	標準偏差	0.0953	0.0565	0.104
被験者 5	平均	0.52	0.50	0.42
	標準偏差	0.116	0.0356	0.0674
被験者 6	平均	0.67	0.64	0.62
	標準偏差	0.101	0.0865	0.0947
被験者 7	平均	0.36	0.39	0.36
	標準偏差	0.0942	0.0891	0.0816
被験者 8	平均	0.60	0.58	0.55
	標準偏差	0.0756	0.0693	0.0956
被験者 9	平均	0.44	0.48	0.44
	標準偏差	0.110	0.0534	0.0923
被験者 10	平均	0.71	0.66	0.64
	標準偏差	0.0502	0.0645	0.0985
被験者全体	平均	0.58	0.55	0.50
	標準偏差	0.153	0.106	0.129

表 7.21 の結果から，右腕の指先から肩まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 50%で分類は行えていない．なお，その標準偏差も被験者平均で 12.9%とばらつきがある．

表 7.22: 右腕の指先から肩まで以外の骨格座標における SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.54	0.53	0.52
	標準偏差	0.159	0.121	0.118
被験者 2	平均	0.62	0.55	0.48
	標準偏差	0.103	0.0430	0.0607
被験者 3	平均	0.18	0.21	0.19
	標準偏差	0.0262	0.0191	0.0218
被験者 4	平均	0.54	0.52	0.43
	標準偏差	0.0233	0.00781	0.0498
被験者 5	平均	0.25	0.29	0.25
	標準偏差	0.0685	0.107	0.0680
被験者 6	平均	0.49	0.50	0.48
	標準偏差	0.0744	0.0499	0.0674
被験者 7	平均	0.54	0.52	0.44
	標準偏差	0.0651	0.0352	0.0618
被験者 8	平均	0.54	0.53	0.46
	標準偏差	0.0490	0.0385	0.0965
被験者 9	平均	0.40	0.49	0.41
	標準偏差	0.152	0.0189	0.0703
被験者 10	平均	0.60	0.54	0.45
	標準偏差	0.0562	0.0196	0.0580
被験者全体	平均	0.47	0.47	0.41
	標準偏差	0.159	0.121	0.118

表 7.22 の結果から，右腕の指先から肩まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 41%となっている．なお，その標準偏差も被験者平均で 11.8%とばらつきがある．

表 7.16 と比較し，表 7.16 で平均 F 値が 70%を超えたのは，被験者 2，被験者 6，被験者 8，被験者 9 の 4 名であった．一方，表 7.22 で平均 F 値が 70%を超えたのは，1 名も存在しなかった．よって，右腕の指先から肩まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が入っていないことがわかった．

7.11.2 右腕の指先から肩まで以外の骨格座標における被験者 20 名での行動評価

次に，右腕の指先から肩まで以外の骨格座標を取得し，被験者 20 名における精度評価を行った．被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20 の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.23** に，SVM による結果を**表 7.24** 示す．

表 7.23: 右腕の指先から肩まで以外の骨格座標における CNN による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.56	0.51	0.48
	標準偏差	0.103	0.0220	0.0367
被験者 2	平均	0.52	0.52	0.52
	標準偏差	0.0508	0.0478	0.0450
被験者 3	平均	0.35	0.38	0.36
	標準偏差	0.0678	0.0374	0.0560
被験者 4	平均	0.63	0.60	0.58
	標準偏差	0.0723	0.0486	0.0626
被験者 5	平均	0.50	0.49	0.46
	標準偏差	0.113	0.0509	0.0683
被験者 6	平均	0.54	0.49	0.46
	標準偏差	0.107	0.0246	0.0553
被験者 7	平均	0.49	0.49	0.48
	標準偏差	0.0531	0.0390	0.0570
被験者 8	平均	0.64	0.53	0.45
	標準偏差	0.118	0.0326	0.0780
被験者 9	平均	0.51	0.51	0.51
	標準偏差	0.0179	0.0179	0.0150
被験者 10	平均	0.57	0.51	0.45
	標準偏差	0.141	0.0777	0.0937
被験者全体	平均	0.53	0.50	0.48
	標準偏差	0.120	0.0660	0.0811

表 7.23 の結果から，右腕の指先から肩まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 48%で分類は行えていない．なお，その標準偏差も被験者平均で 8.11%とばらつきがある．

表 7.24: 右腕の指先から肩まで以外の骨格座標における SVM による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.76	0.69	0.66
	標準偏差	0.0349	0.0225	0.0337
被験者 2	平均	0.76	0.55	0.43
	標準偏差	0.0102	0.0559	0.102
被験者 3	平均	0.26	0.30	0.27
	標準偏差	0.0217	0.0417	0.0236
被験者 4	平均	0.40	0.46	0.38
	標準偏差	0.0723	0.0185	0.0415
被験者 5	平均	0.51	0.51	0.49
	標準偏差	0.0226	0.0210	0.0283
被験者 6	平均	0.55	0.53	0.51
	標準偏差	0.0592	0.0260	0.0180
被験者 7	平均	0.43	0.46	0.43
	標準偏差	0.0626	0.0340	0.0628
被験者 8	平均	0.46	0.49	0.41
	標準偏差	0.0809	0.0162	0.0458
被験者 9	平均	0.54	0.53	0.52
	標準偏差	0.0622	0.0448	0.0478
被験者 10	平均	0.49	0.50	0.42
	標準偏差	0.0468	0.0341	0.0735
被験者全体	平均	0.52	0.50	0.45
	標準偏差	0.153	0.0963	0.111

表 7.24 の結果から，右腕の指先から肩まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 45%となっている．なお，その標準偏差も被験者平均で 11.1%とばらつきがある．

表 7.18 と比較し，表 7.18 で平均 F 値が 70%を超えたのは，被験者 2，被験者 3，被験者 6，被験者 8，被験者 9，被験者 10 の 6 名であった．一方，表 7.24 で平均 F 値が 70%を超えたのは，1 名も存在しなかった．よって，右腕の指先から肩まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が入っていないことがわかった．

以上より，右腕の指先から肩まで以外の骨格座標を取得した場合，CNN および SVM の両者において，うまく二値分類できない．

7.12　右腕の指先から肘まで以外の骨格座標を取得した場合

次に，右腕の指先から肘まで以外の骨格座標を取得した場合の評価を行う．まず，エポック数とバッチサイズを決定する必要がある．そこで，被験者 1〜20 にあたる 20 名分のサンプルを用いて，10 分割交差検証により最適なエポック数とバッチサイズを調べた．本実験の事前実験では，10 分割交差検証を 5 回セット行い評価をとった．

なお，被験者 1 名あたりのサンプル数は 200 であり，全被験者の合計サンプル数は 4,000 である．また，テスト用サンプルと訓練用サンプルにそれぞれの被験者のデータがはいるように設定しており，テスト用サンプルに含まれるピッキングと鍵開けのサンプルの割合はそれぞれ 200 である．なお，本研究で用いるエポック数とバッチサイズは，上記の決定条件と同様である．

表 7.25 は，右腕の指先から肘まで以外の骨格座標を取得した場合におけるエポック数 50 のときにバッチサイズを変更した場合の精度結果である．また，**表 7.26** は，右腕の指先から肘まで以外の骨格座標を取得した場合におけるエポック数 100 のときにバッチサイズを変更した場合の精度結果である．

表 7.25: 右腕の指先から肘まで以外の骨格座標におけるエポック 50 とバッチサイズとの関係

batch		precision	recall	F 値
32	平均	0.77	0.70	0.64
	標準偏差	0.0914	0.156	0.222
64	平均	0.77	0.74	0.72
	標準偏差	0.0886	0.117	0.155
128	平均	0.78	0.77	0.77
	標準偏差	0.0613	0.0678	0.0761
64	平均	0.77	0.76	0.76
	標準偏差	0.0422	0.0461	0.0476
512	平均	0.73	0.71	0.71
	標準偏差	0.0367	0.0398	0.0430
1,024	平均	0.69	0.67	0.66
	標準偏差	0.0336	0.0381	0.0467
1,800	平均	0.66	0.63	0.60
	標準偏差	0.0371	0.0370	0.0559

表 7.26: 右腕の指先から肘まで以外の骨格座標におけるエポック 100 とバッチサイズとの関係

batch		precision	recall	F 値
32	平均	0.82	0.72	0.65
	標準偏差	0.0885	0.189	0.266
64	平均	0.86	0.84	0.83
	標準偏差	0.0859	0.119	0.150
128	平均	0.90	0.90	0.90
	標準偏差	0.0465	0.0478	0.0474
64	平均	0.87	0.86	0.86
	標準偏差	0.0477	0.0531	0.0518
512	平均	0.83	0.81	0.81
	標準偏差	0.0441	0.0550	0.0591
1,024	平均	0.77	0.75	0.75
	標準偏差	0.0443	0.0469	0.0499
1,800	平均	0.72	0.70	0.69
	標準偏差	0.0394	0.0420	0.0491

表 7.25 の結果から，エポック数が 50 のときにバッチサイズを変更しても F 値が 90%を超える場合は存在しなかった．表 7.26 の結果から，エポック数が 100 のとき，バッチサイズが 128 の場合に F 値が 90%超を示した．なお，バッチサイズが 128 の場合，標準偏差は 4.74%となり，5%以下となった．以上の結果から，本研究ではエポック数を 100，バッチサイズを 128 とし，以降の実験を行っていく．

なお，本実験では訓練用サンプルが少ないため，過学習を起こしている可能性がある．そこで，訓練（train）と検証（validation）の loss を取得し，**図 7.5** に示した．

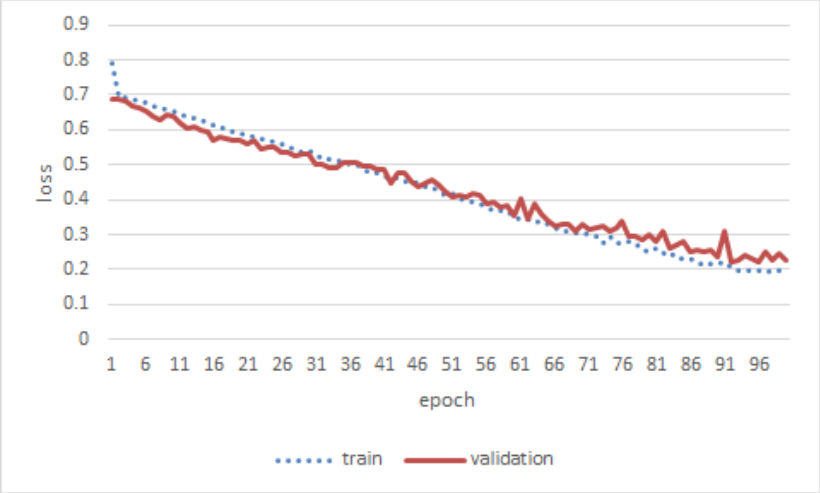

図 7.5: 右腕の指先から肘まで以外の骨格座標におけるエポック数とバッチサイズを決定する実験における loss

その結果，訓練においては，92〜99 エポックで loss がおよそ 0.20 となり，100 エポックで 0.18 まで下がった．検証においては，92〜99 エポックで loss が 0.22 から 0.26 の間を行き来し，100 エポックで 0.23 まで下がった．loss は減少し続けているが，過学習は起きていないことが確認できた．

7.12.1 右腕の指先から肘まで以外の骨格座標における被験者 10 名での行動評価

次に，右腕の指先から肘まで以外の骨格座標を取得し，被験者 10 名における精度評価を行った．そして，被験者 1〜10 の中から 1 名のデータをテスト用サンプルにし，テスト用サンプルに使用していない残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.27** に，SVM による結果を**表 7.28** に示す．

表 7.27: 右腕の指先から肘まで以外の骨格座標における CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.75	0.53	0.40
	標準偏差	0.0266	0.0128	0.0321
被験者 2	平均	0.59	0.55	0.52
	標準偏差	0.0989	0.0685	0.0893
被験者 3	平均	0.63	0.59	0.54
	標準偏差	0.151	0.0658	0.108
被験者 4	平均	0.57	0.52	0.46
	標準偏差	0.105	0.0500	0.0984
被験者 5	平均	0.55	0.50	0.43
	標準偏差	0.150	0.0604	0.0753
被験者 6	平均	0.72	0.66	0.63
	標準偏差	0.0644	0.0694	0.0936
被験者 7	平均	0.44	0.45	0.44
	標準偏差	0.0587	0.0486	0.0585
被験者 8	平均	0.57	0.55	0.53
	標準偏差	0.0565	0.0261	0.0224
被験者 9	平均	0.51	0.51	0.48
	標準偏差	0.104	0.0568	0.0893
被験者 10	平均	0.71	0.67	0.65
	標準偏差	0.0512	0.0621	0.0899
被験者全体	平均	0.60	0.55	0.51
	標準偏差	0.133	0.0860	0.113

表 7.27 の結果から，右腕の指先から肘まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 51%で分類は行えていない．なお，その標準偏差も被験者平均で 11.3%とばらつきがある．

表 7.28: 右腕の指先から肘まで以外の骨格座標における SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.54	0.53	0.53
	標準偏差	0.0720	0.0472	0.0377
被験者 2	平均	0.61	0.56	0.51
	標準偏差	0.0814	0.0433	0.0735
被験者 3	平均	0.21	0.25	0.23
	標準偏差	0.0240	0.0119	0.0169
被験者 4	平均	0.58	0.53	0.47
	標準偏差	0.0725	0.0187	0.0504
被験者 5	平均	0.29	0.35	0.29
	標準偏差	0.0804	0.101	0.0704
被験者 6	平均	0.41	0.47	0.42
	標準偏差	0.0929	0.0361	0.0827
被験者 7	平均	0.50	0.50	0.45
	標準偏差	0.0411	0.0269	0.0537
被験者 8	平均	0.60	0.54	0.44
	標準偏差	0.0646	0.0273	0.0729
被験者 9	平均	0.53	0.53	0.44
	標準偏差	0.106	0.0529	0.100
被験者 10	平均	0.55	0.53	0.46
	標準偏差	0.0626	0.0233	0.0764
被験者全体	平均	0.48	0.48	0.42
	標準偏差	0.148	0.105	0.112

表 7.28 の結果から，右腕の指先から肘まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 42%となっている．なお，その標準偏差も被験者平均で 11.2%とばらつきがある．

表 7.16 と比較し，表 7.16 で平均 F 値が 70%を超えたのは，被験者 2，被験者 6，被験者 8，被験者 9 の 4 名であった．一方，表 7.28 で平均 F 値が 70%を超えたのは，1 名も存在しなかった．よって，右腕の指先から肘まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が入っていないことがわかった．

7.12.2 右腕の指先から肘まで以外の骨格座標における被験者 20 名での行動評価

次に，右腕の指先から肘まで以外の骨格座標を取得し，被験者 20 名における精度評価を行った．被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20 の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.29** に，SVM による結果を**表 7.30** に示す．

表 7.29: 右腕の指先から肘まで以外の骨格座標における CNN による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.58	0.52	0.49
	標準偏差	0.117	0.0286	0.0332
被験者 2	平均	0.49	0.50	0.49
	標準偏差	0.0889	0.0556	0.0749
被験者 3	平均	0.41	0.42	0.40
	標準偏差	0.129	0.116	0.123
被験者 4	平均	0.62	0.56	0.52
	標準偏差	0.0807	0.0351	0.0665
被験者 5	平均	0.55	0.52	0.51
	標準偏差	0.0831	0.0405	0.0483
被験者 6	平均	0.53	0.49	0.45
	標準偏差	0.113	0.0312	0.0609
被験者 7	平均	0.53	0.52	0.50
	標準偏差	0.0390	0.0246	0.0458
被験者 8	平均	0.60	0.51	0.43
	標準偏差	0.115	0.0241	0.0569
被験者 9	平均	0.51	0.51	0.51
	標準偏差	0.0181	0.0181	0.0158
被験者 10	平均	0.56	0.53	0.48
	標準偏差	0.177	0.116	0.135
被験者全体	平均	0.54	0.51	0.48
	標準偏差	0.120	0.0694	0.0831

表 7.29 の結果から，右腕の指先から肘まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 48%で分類は行えていない．なお，その標準偏差も被験者平均で 8.31%とばらつきがある．

表 7.30: 右腕の指先から肘まで以外の骨格座標における SVM による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.67	0.64	0.61
	標準偏差	0.0420	0.0425	0.0681
被験者 2	平均	0.78	0.70	0.67
	標準偏差	0.0179	0.0778	0.112
被験者 3	平均	0.26	0.34	0.28
	標準偏差	0.0206	0.0372	0.0130
被験者 4	平均	0.43	0.47	0.40
	標準偏差	0.0697	0.0228	0.0487
被験者 5	平均	0.51	0.51	0.50
	標準偏差	0.00943	0.00943	0.00490
被験者 6	平均	0.53	0.52	0.51
	標準偏差	0.0320	0.0239	0.0237
被験者 7	平均	0.49	0.49	0.48
	標準偏差	0.0265	0.0265	0.0228
被験者 8	平均	0.49	0.52	0.46
	標準偏差	0.104	0.0439	0.0912
被験者 9	平均	0.50	0.50	0.49
	標準偏差	0.0140	0.0119	0.0224
被験者 10	平均	0.62	0.59	0.56
	標準偏差	0.0825	0.0571	0.0689
被験者全体	平均	0.53	0.53	0.50
	標準偏差	0.144	0.100	0.188

表 7.30 の結果から，右腕の指先から肘まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 50%となっている．なお，その標準偏差も被験者平均で 18.8%とばらつきがある．

表 7.18 と比較し，表 7.18 で平均 F 値が 70%を超えたのは，被験者 2，被験者 3，被験者 6，被験者 8，被験者 9，被験者 10 の 6 名であった．一方，表 7.30 で平均 F 値が 70%を超えたのは，1 名も存在しなかった．よって，右腕の指先から肘まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が入っていないことがわかった．

以上より，右腕の指先から肘まで以外の骨格座標を取得した場合，CNN および SVM の両者において，うまく二値分類できない．

7.13　右腕の指先から手首まで以外の骨格座標を取得した場合

次に，右腕の指先から手首まで以外の骨格座標を取得した場合の評価を行う．まず，エポック数とバッチサイズを決定する必要がある．そこで，被験者 1〜20 にあたる 20 名分のサンプルを用いて，10 分割交差検証により最適なエポック数とバッチサイズを調べた．本実験の事前実験では，10 分割交差検証を 5 回セット行い評価をとった．

なお，被験者 1 名あたりのサンプル数は 200 であり，全被験者の合計サンプル数は 4,000 である．また，テスト用サンプルと訓練用サンプルにそれぞれの被験者のデータがはいるように設定しており，テスト用サンプルに含まれるピッキングと鍵開けのサンプルの割合はそれぞれ 200 である．なお，本研究で用いるエポック数とバッチサイズは，上記の決定条件と同様である．

表 7.31 は，右腕の指先から手首まで以外の骨格座標を取得した場合におけるエポック数 50 のときにバッチサイズを変更した場合の精度結果である．

表 7.31: 右腕の指先から手首まで以外の骨格座標におけるエポック 50 とバッチサイズとの関係

batch		precision	recall	F 値
32	平均	0.84	0.77	0.72
	標準偏差	0.0722	0.176	0.248
64	平均	0.85	0.83	0.82
	標準偏差	0.109	0.136	0.172
128	平均	0.90	0.90	0.90
	標準偏差	0.0307	0.0314	0.0308
64	平均	0.82	0.82	0.81
	標準偏差	0.0422	0.0461	0.0476
512	平均	0.80	0.78	0.78
	標準偏差	0.0451	0.0526	0.0559
1,024	平均	0.73	0.71	0.71
	標準偏差	0.0390	0.0436	0.0508
1,800	平均	0.67	0.65	0.64
	標準偏差	0.0349	0.0371	0.0462

表 7.31 の結果から，エポック数が 50 のとき，バッチサイズが 128 の場合に F 値が 90% 超を示した．なお，バッチサイズが 128 の場合，標準偏差は 3.08%となり，5%以下となった．以上の結果から，本研究ではエポック数を 50，バッチサイズを 128 とし，以降の実験を行っていく．

なお，本実験では訓練用サンプルが少ないため，過学習を起こしている可能性がある．そこで，訓練（train）と検証（validation）の loss を取得し，**図 7.6** に示した．

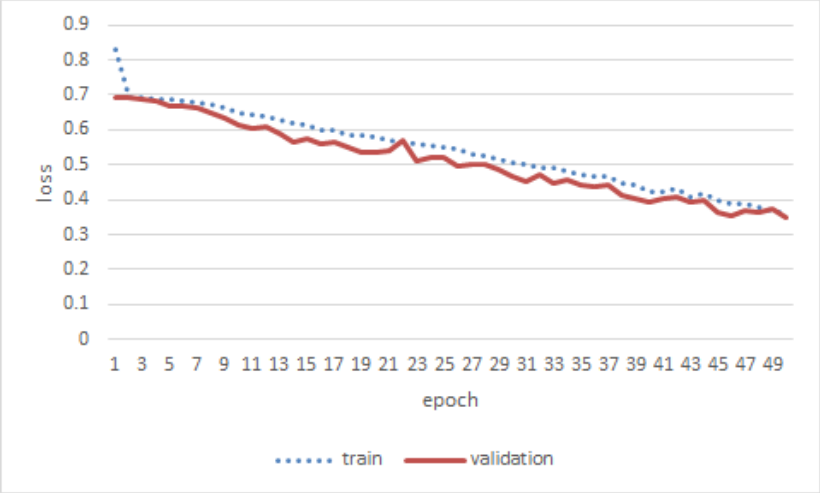

図 7.6: 右腕の指先から手首まで以外の骨格座標におけるエポック数とバッチサイズを決定する実験における loss

その結果，訓練においては，1 エポックで loss がおよそ 0.83 となり，徐々に減少していき，50 エポックで 0.36 まで下がった．検証においては，1 エポックで loss が 0.69 となり，徐々に減少していき，50 エポックで 0.35 まで下がった．loss は減少し続けているが，過学習は起きていないことが確認できた．

7.13.1 右腕の指先から手首まで以外の骨格座標における被験者 10 名での行動評価

次に，右腕の指先から手首まで以外の骨格座標を取得し，被験者 10 名における精度評価を行った．そして，被験者 1〜10 の中から 1 名のデータをテスト用サンプルにし，テスト用サンプルに使用していない残りの被験者のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.32** に，SVM による結果を**表 7.33** に示す．

表 7.32: 右腕の指先から手首まで以外の骨格座標における CNN による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.74	0.56	0.45
	標準偏差	0.0594	0.0432	0.0804
被験者 2	平均	0.61	0.56	0.53
	標準偏差	0.132	0.0853	0.0932
被験者 3	平均	0.56	0.53	0.49
	標準偏差	0.146	0.0822	0.107
被験者 4	平均	0.59	0.56	0.54
	標準偏差	0.0762	0.0444	0.0683
被験者 5	平均	0.68	0.55	0.45
	標準偏差	0.106	0.0544	0.0970
被験者 6	平均	0.72	0.68	0.66
	標準偏差	0.0864	0.0877	0.107
被験者 7	平均	0.50	0.49	0.46
	標準偏差	0.106	0.0541	0.0624
被験者 8	平均	0.63	0.58	0.54
	標準偏差	0.0988	0.0688	0.0935
被験者 9	平均	0.56	0.53	0.49
	標準偏差	0.0910	0.0342	0.0497
被験者 10	平均	0.75	0.73	0.73
	標準偏差	0.0709	0.0728	0.0733
被験者全体	平均	0.63	0.58	0.53
	標準偏差	0.129	0.0954	0.122

表 7.32 の結果から，右腕の指先から手首まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 53%で分類は行えていない．なお，その標準偏差も被験者平均で 12.2%とばらつきがある．しかし，被験者 10 においては，平均 F 値は 73%となってる．

表 7.33: 右腕の指先から手首まで以外の骨格座標における SVM による被験者 10 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.73	0.63	0.58
	標準偏差	0.0775	0.0665	0.109
被験者 2	平均	0.70	0.59	0.51
	標準偏差	0.0485	0.0704	0.110
被験者 3	平均	0.77	0.57	0.47
	標準偏差	0.0158	0.0478	0.0848
被験者 4	平均	0.55	0.53	0.44
	標準偏差	0.0432	0.0255	0.0566
被験者 5	平均	0.72	0.56	0.47
	標準偏差	0.0838	0.0505	0.0861
被験者 6	平均	0.72	0.69	0.68
	標準偏差	0.0314	0.00831	0.0117
被験者 7	平均	0.76	0.66	0.61
	標準偏差	0.0133	0.0803	0.135
被験者 8	平均	0.60	0.60	0.55
	標準偏差	0.115	0.0950	0.153
被験者 9	平均	0.76	0.68	0.64
	標準偏差	0.0282	0.0932	0.152
被験者 10	平均	0.79	0.72	0.70
	標準偏差	0.0238	0.0544	0.0742
被験者全体	平均	0.71	0.62	0.56
	標準偏差	0.0931	0.0882	0.138

表 7.33 の結果から，右腕の指先から手首まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 56%となっている．なお，その標準偏差も被験者平均で 13.8%とばらつきがある．しかし，被験者 10 においては，平均 F 値は 70%となっている．

表 7.16 と比較し，表 7.16 で平均 F 値が 70%を超えたのは，被験者 2，被験者 6，被験者 8，被験者 9 の 4 名であった．一方，表 7.33 で平均 F 値が 70%を超えたのは，被験者 10 の 1 名であった．よって，右腕の指先から手首まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が 10 名中 1 名入っていることがわかった．

7.13.2 右腕の指先から手首まで以外の骨格座標における被験者 20 名での行動評価

次に，右腕の指先から手首まで以外の骨格座標を取得し，被験者 20 名における精度評価を行った．被験者 1〜10 の中から 1 名のデータをテスト用サンプルに，被験者 11〜20

の 10 名分のデータを訓練用サンプルにし，被験者ごとにピッキングと鍵開けの分類が行えるか調べた．CNN による結果を**表 7.34** に，SVM による結果を**表 7.35** に示す．

表 7.34: 右腕の指先から手首まで以外の骨格座標における CNN による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.51	0.51	0.51
	標準偏差	0.0316	0.0285	0.0285
被験者 2	平均	0.56	0.54	0.53
	標準偏差	0.0781	0.0553	0.0625
被験者 3	平均	0.57	0.55	0.52
	標準偏差	0.178	0.125	0.133
被験者 4	平均	0.65	0.60	0.58
	標準偏差	0.0105	0.0635	0.0677
被験者 5	平均	0.60	0.53	0.47
	標準偏差	0.121	0.0500	0.0776
被験者 6	平均	0.56	0.51	0.48
	標準偏差	0.103	0.0395	0.0712
被験者 7	平均	0.55	0.53	0.52
	標準偏差	0.0885	0.0434	0.0247
被験者 8	平均	0.65	0.56	0.51
	標準偏差	0.105	0.0616	0.0947
被験者 9	平均	0.50	0.50	0.50
	標準偏差	0.000	0.000	0.000
被験者 10	平均	0.50	0.44	0.43
	標準偏差	0.133	0.142	0.0825
被験者全体	平均	0.57	0.53	0.50
	標準偏差	0.115	0.0831	0.0825

表 7.34 の結果から，右腕の指先から手首まで以外の骨格座標における CNN による被験者 10 名の平均 F 値は 50%で分類は行えていない．なお，その標準偏差も被験者平均で 8.25%とばらつきがある．

表 7.35: 右腕の指先から手首まで以外の骨格座標における SVM による被験者 20 名での行動評価

被験者番号		precision	recall	F 値
被験者 1	平均	0.25	0.50	0.33
	標準偏差	0.000	0.000	0.000
被験者 2	平均	0.25	0.50	0.33
	標準偏差	0.000	0.000	0.000
被験者 3	平均	0.86	0.82	0.81
	標準偏差	0.0147	0.0303	0.0327
被験者 4	平均	0.25	0.50	0.33
	標準偏差	0.000	0.000	0.000
被験者 5	平均	0.72	0.51	0.37
	標準偏差	0.0881	0.0170	0.0336
被験者 6	平均	0.82	0.71	0.67
	標準偏差	0.0250	0.0679	0.0936
被験者 7	平均	0.75	0.51	0.36
	標準偏差	0.000	0.00490	0.00872
被験者 8	平均	0.25	0.50	0.33
	標準偏差	0.000	0.000	0.000
被験者 9	平均	0.78	0.72	0.71
	標準偏差	0.0432	0.0355	0.0388
被験者 10	平均	0.76	0.53	0.39
	標準偏差	0.00490	0.0155	0.0323
被験者全体	平均	0.57	0.58	0.46
	標準偏差	0.265	0.117	0.182

表 7.35 の結果から，右腕の指先から手首まで以外の骨格座標における SVM による被験者 10 名の平均 F 値は 46%となっている．なお，その標準偏差も被験者平均で 18.2%とばらつきがある．しかし，被験者 3，被験者 9 において，平均 F 値はそれぞれ 81%，71%となっており，二値分類できている被験者もいる．よって，右腕の指先から手首まで以外の骨格座標には，ピッキングおよび鍵開け動作の特徴が入っていることがわかった．また，表 7.33 と表 7.35 より，被験者 10 名での行動評価より被験者 20 名での行動評価のほうが 56%と 46%と，平均 F 値が下がっている．これは被験者 10 名，被験者 20 名における訓練用サンプルはそれぞれ被験者 1〜10 のうち 9 名と，被験者 11〜20 の 10 名としているからである．つまり，被験者 10 名から 20 名にした際に，訓練用サンプルを単純に増加させたのではなく，他の被験者を訓練用サンプルにしたからである．

以上より，右腕の指先から肩までの骨格座標を取得した場合，CNN より SVM のほうがうまく二値分類できている．その理由として，CNN では，人物の骨格座標の長さによる人物識別精度は 98.7%，顔による人物識別精度は 93.67%，目による人物識別精度は 99.54%となっている [57] [58] [59]．よって，CNN では人物の骨格座標の長さや顔，目などにお

ける特徴を畳込むことで個人識別が有効であると考えられる．一方，SVM における，足長や足幅，足跡面積による男女識別精度は 90.4%，顔の表情による識別精度は 94.7%，行動識別による精度は 94.1%となっている [60] [61] [62]．よって，男女や顔の表情，行動識別といった個人識別以外のベクトルによる識別に有効であると考えられる．本研究では，行動識別を行っているため，SVM が有効であったと思われた．

しかしながら，本研究では，少量のデータにより学習を行っているため，大量のデータがあれば CNN でも上手く畳込みができ，SVM よりも高い行動識別となる可能性があると思われる．また，CNN はエポックやバッチサイズはチューニングを行ったが，チャネル数の増加やレイヤ数を変更することで，精度が向上する可能性が考えられる．

次に，全骨格座標における SVM による被験者 20 名での行動評価にて，F 値が 90%以上となった被験者 2，被験者 6 と，F 値が最も低かった被験者 4 との比較を行った．その際，HandRight と HandTipRight，ThumbRight の各骨格座標における 1 サンプル 40 個の点プロットにより比較を行う．**図 7.7**，**図 7.8** に鍵開け動作およびピッキング動作における被験者の点プロットの比較を示す．

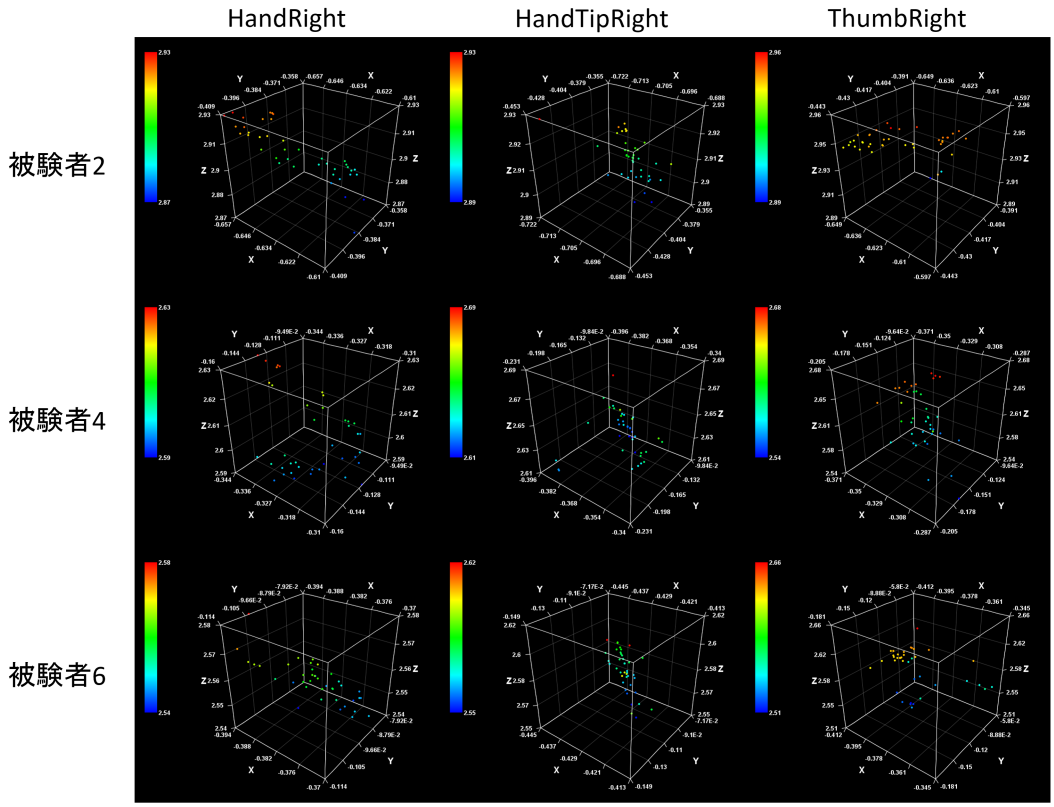

図 7.7: 鍵開け動作における被験者の比較

図 7.7 より，鍵開け動作において，低精度を示した被験者 4 は高精度を示した被験者 2，被験者 6 と比較すると，HandRight の点プロットに大きな違いがあるように思われた．HandRight では，低精度となった被験者 4 は大きくばらついているのに対し，高精度を示した被験者 2，被験者 6 は一直線上に点プロットがあった．HandTipRight では，違いは

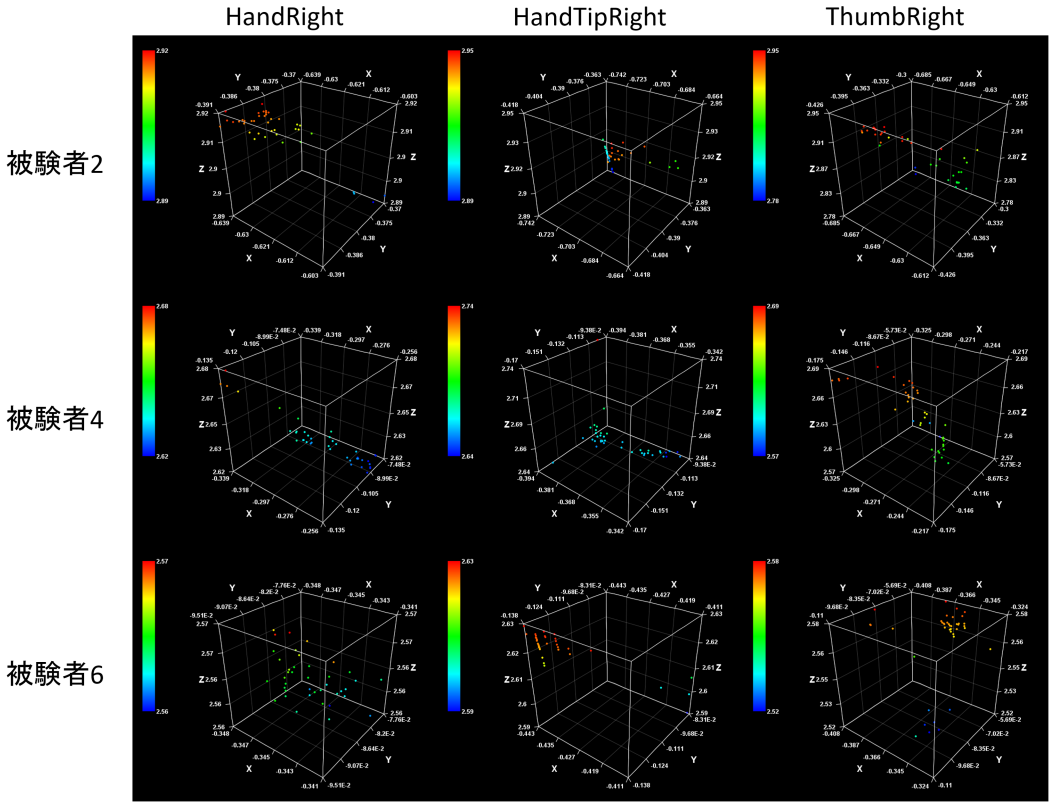

図 7.8: ピッキング動作における被験者の比較

ないように思われた．ThumbRight では，被験者 3 名とも一直線上に並んでいるが，低精度を示した被験者 4 は他の方向に傾いていると思われた．よって，低精度を示した被験者 4 の鍵開け動作は，高精度を示した被験者 2，被験者 6 の鍵開け動作と比較して，動作が大きくなっていると思われた．次に，図 7.8 より，ピッキング動作において，低精度を示した被験者 4 と高精度を示した被験者 2，被験者 6 と比較すると，HandRight の点プロットに大きな違いがあるように思われた．HandRight では，低精度を示した被験者 4 は X 軸方向に大きく直線的に動いているのに対し，高精度を示した被験者 2，被験者 6 は一点に固まって点がプロットされていた．よって，低精度を示した被験者 4 のピッキング動作は，高精度を示した被験者 2，被験者 6 のピッキング動作と比較して，動作が大きくなっていると思われた．なお，被験者 4 のピッキング動作には，X 軸方向に腕が動いていると思われた．つまり，上下する動作に加えて，手を手前や奥に動かしている動作が含まれていると考えられた．

以上より，低精度を示した被験者 4 は，高精度を示した被験者 2，被験者 6 と比較して，鍵開け動作では大きな動作になっており，ピッキング動作では手が手前や奥に動作していると考えられた．低精度を示した動作の F 値を向上させるためには，大きな動きの鍵開け動作や手を手前や奥に動かすピッキング動作を行っている類似の行動を訓練させることが必要だと思われる．

7.14 Wilcoxon の符合付順位和検定

最後に，取得する骨格座標を変更した際，サンプルに有意差があるかどうかを確認するために，Wilcoxon の符合付順位和検定を行った．[63]．その結果を**表 7.36** に示す．なお，それぞれの行動識別の上段にある両側有意確率は，斜め背後から骨格座標を取得した場合である．一方，下段にある両側有意確率は，真横から骨格座標を取得した場合である．

表 7.36: Wilcoxon の符合付順位和検定によるデータ比較

比較実験データ	行動種別	被験者番号									
		被験者 1	被験者 2	被験者 3	被験者 4	被験者 5	被験者 6	被験者 7	被験者 8	被験者 9	被験者 10
元データ 線形補間データ (0.7 倍)	鍵開け	0.625	0.836	0.237	0.786	0.849	0.373	0.650	0.464	0.938	0.511
		0.994	0.584	0.420	0.101	0.474	0.592	0.002	0.486	0.306	0.985
	ピッキング	0.302	0.529	0.054	0.404	0.783	0.854	0.016	0.710	0.427	0.553
		0.837	0.099	0.086	0.651	0.798	0.114	0.612	0.484	0.375	0.096
元データ 線形補間データ (1.3 倍)	鍵開け	0.004	0.004	0.305	0.883	0.468	0.329	0.247	0.941	0.522	0.582
		0.004	0.005	0.038	0.372	0.219	0.290	0.000	0.009	0.114	0.187
	ピッキング	0.048	0.743	0.044	0.794	0.867	0.312	0.000	0.908	0.684	0.049
		0.012	0.250	0.006	0.248	0.008	0.922	0.162	0.663	0.569	0.830
元データ スライドしたデータ	鍵開け	0.000	0.000	0.000	0.000	0.071	0.833	0.569	0.199	0.158	0.003
		0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.053	0.000
	ピッキング	0.000	0.000	0.359	0.000	0.503	0.365	0.292	0.389	0.137	0.000
		0.000	0.000	0.000	0.000	0.000	0.556	0.000	0.442	0.000	0.000
元データ 指先から肩までのデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
元データ 指先から肩まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
元データ 指先から肘まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.011	0.019	0.019	0.019	0.017	0.019	0.004	0.019	0.016	0.019
	ピッキング	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
		0.019	0.019	0.019	0.011	0.019	0.019	0.001	0.019	0.011	0.012
元データ 指先から手首まで以外のデータ	鍵開け	0.000	0.000	0.036	0.000	0.000	0.000	0.000	0.000	0.033	0.000
		0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
	ピッキング	0.000	0.000	0.036	0.036	0.000	0.000	0.000	0.000	0.008	0.000
		0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
指先から肩までのデータ 指先から肩まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
指先から肩までのデータ 指先から肘まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
指先から肩までのデータ 指先から手首まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
指先から肩まで以外のデータ 指先から肘まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
指先から肩まで以外のデータ 指先から手首まで以外のデータ	鍵開け	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ピッキング	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
指先から肘まで以外のデータ 指先から手首まで以外のデータ	鍵開け	0.000	0.000	0.307	0.277	0.000	0.000	0.000	0.000	0.000	0.000
		0.002	0.307	0.307	0.307	0.144	0.307	0.000	0.307	0.073	0.307
	ピッキング	0.000	0.000	0.128	0.002	0.000	0.000	0.000	0.000	0.000	0.000
		0.307	0.307	0.307	0.002	0.307	0.307	0.000	0.307	0.002	0.002

まず，元データ以外の 7 つの代表値が，元データの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．表 7.36 の結果，元データを 0.7 倍および 1.3 倍にした線形補間データについて帰無仮説が採択され，元データの代表値との間に有意差は認められない．元データをスライドさせたデータ，指先から肩までのデータ，指先から肩まで以外のデータ，指先から肘まで以外のデータ，指先から手首まで以外のデータについては帰無仮説が棄却され，元データの代表値に対して有意差があるといえる．この結果，元データと線形補間データの間には，周期を変更しただけにすぎず，各骨格座標の線形補間前と後でほぼ全ての骨格座標で似た値を取ることとなる．そのため検定による違いがみられなかったように考えられた．元データとスライドしたデータの間には，全ての骨格座標の値がスライドにより異なってしまうため，それにより違いがみられたと考えられる．

次に，指先から肩までのデータ以外の 3 つの代表値が，指先から肩までのデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．表 7.36 の結果，指先から肩まで以外のデータ，指先から肘まで以外のデータ，指先から手首まで以外のデータについて帰無仮説が棄却され，指先から肩までのデータの代表値に対して有意差があるといえる．この結果，右腕の骨格座標を取得した場合と右腕以外の骨格座標を取得した場合とでは，真逆の骨格座標を取得しており大きく異なるため，違いがみられたと考えられる．

次に，指先から肩まで以外のデータ以外の 2 つの代表値が，指先から肩まで以外のデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．表 7.36 の結果，指先から肘まで以外のデータ，指先から手首まで以外のデータについて帰無仮説が棄却され，指先から肩まで以外のデータの代表値に対して有意差があるといえる．この結果，肩または肘の骨格座標がある場合とない場合とより，違いがみられたと考えられる．

最後に，指先から肘まで以外のデータ以外の 1 つの代表値が，指先から肘まで以外のデータの代表値と有意差がないことを帰無仮説とし，有意差があることを対立仮説として，有意水準 0.05 で両側仮説検定を行った．表 7.36 の結果，斜め背後から骨格座標を取得した場合において，指先から手首まで以外のデータについては被験者 2 名は帰無仮説が採択され，指先から肘まで以外のデータの代表値との間に有意差は認められない．被験者 8 名は帰無仮説が棄却され，指先から肘まで以外のデータの代表値との間に有意差があるといえる．一方，真横から骨格座標を取得した場合において，指先から手首まで以外のデータについては被験者 5 名は帰無仮説が採択され，指先から肘まで以外のデータの代表値との間に有意差は認められない．被験者 5 名においては帰無仮説が棄却され，指先から肘まで以外のデータの代表値との間に有意差があるといえる．なお，斜め後ろおよび真横から骨格座標を取得とした場合の両方において，有意差がなかった被験者は 1 名であった．両方の場合において有意差がなかった被験者 3 と有意差があった被験者 7 を比較した際に，大きな違いとして肘の値が大きく変化していることが有意差の要因ではないかと思われた．

行動評価と Wilcoxon の符合付順位和検定の結果より，元データの代表値が一部の骨格座標を取得した場合のデータの値との間には，有意差があり，この比較は有効であったことを示している．次に，指先から肩まで取得したデータの代表値とそれ以外の骨格座標を取得した場合との間にも，有意差があり，指先から肩までの骨格座標の行動評価は 73%

と高く，ピッキングおよび鍵開けを行っている右腕と右腕以外の比較は重要であることを示している．次に，指先から肩まで以外のデータの代表値と指から肘および手首まで以外の骨格座標を取得した場合のデータの間にも，有意差があり，指先から手首以外の骨格座標を取得した場合において，81%と 71%と 70%以上の精度を示す被験者がおり，肘は重要になる場合があることを示している．次に，指先から肘以外の骨格座標を取得したデータの代表値と指先から手首以外の骨格座標を取得した場合のデータの間には，有意差がみられず，肘の骨格取得による違いはないと思われた．しかしながら，被験者 2 名においては 70%を以上を示しており，肘に共通するベクトルによる違いがみられたのではないかと考えられた．

第8章

まとめ

監視カメラは犯罪の抑止や事件後の犯人の特定には役立っているが，家に侵入している犯人をその場で捕らえるためには，常時有人監視しなければならない．また，カメラの映像で監視する場合には，被撮影者のプライバシが問題となる．そこで本論文では，Kinectの骨格座標のみを用いて，家のドアから不正に侵入する行為を検知するシステムを提案した．

そのなかで，従来研究では高精度で分類できるかどうか示されていない，ピッキングと鍵開けという類似の動作に注目し，少ない被験者数で分類精度について評価した．

SVMを用いた評価では，全骨格座標を取得した場合や，右腕の指先から肩までの骨格座標を取得した場合でそれぞれ平均F値は76%や73%となった．なお，右腕の指先から肩までの骨格座標を取得した場合，被験者20名を扱った際には，一部の被験者10名のうち6名が73%，85%，86%，86%，89%，97%と，より高精度で分類され，訓練用サンプル数の増加が精度の向上に繋がることが示された．一方，右腕の指先から肩まで以外の骨格座標を取得した場合や，右腕の指先から肘まで以外の骨格座標を取得した場合では，被験者20名を扱った際には，平均F値はそれぞれ45%，50%となり，精度は五分五分で分類行えていない．よって，ピッキングや鍵開け動作を行っている右腕に分類に必要な特徴量があらわれていることが考えられた．

本システムでは，個人識別に映像を用いないため，画像や映像によるプライバシの侵害なく，訓練用サンプルを利用者から集められる．さらに，映像と比較して，Kinectの骨格座標のデータは小さいため，大量のデータを各家庭からセンタに送信しても，データ転送がトラヒックに与える影響は映像よりも小さい．さらに，全家庭がセンタに映像を送信する必要もなく，センタにデータを提供しない家庭は，センタへの通信設備も不要である．以上より，提案手法およびシステムは，家庭の防犯対策に有用であるといえる．将来の研究では，CNNにおいてチャネル数の増加やレイヤー数の変更により高い精度となる可能性が考えられ，今後の課題である．

謝辞

本研究を進めるにあたり，指導教員である宇田隆哉講師には，研究の方針および作戦の決定から研究発表，論文執筆に至るまで，多くのご指導を頂きましたことを心より感謝申し上げます．また，東京工科大学博士後期課程，宇田研究室所属の釜石智史氏には週報やプログラムの確認，機械学習用サーバの構築等を行って頂きましたことを心より感謝申し上げます．さらに，サンプルデータの取得をお手伝い頂きました宇田研究室の学部 3 年生から修士 2 年生の皆さまに心より感謝申し上げます．

参考文献

[1] 世瀬周一郎. 五輪控え AI で不審者検出 三菱電機，高齢者ら支援も. http://style.nikkei.com/article/DGXMZO10563380S6A211C1000000?channel=DF220420167276, 2017. (2021 年 1 月 13 日閲覧).

[2] Oxford Languages and Google - Japanese — Oxford Languages. https://languages.oup.com/google-dictionary-ja/. (2021 年 1 月 21 日閲覧).

[3] Y. Torigoe, Y. Nakamura, M. Fujimoto, Y. Arakawa, and K. Yasumoto. Strike Activity Detection and Recognition Using Inertial Measurement Unit towards Kendo Skill Improvement Support System. *Journal of the Sensors and Materials*, Vol. 32, No. 2, pp. 651–673, 2020.

[4] 村上優樹, 三浦雅展. ポピュラー音楽で用いられるドラムパターンを対象としたフィルイン自動検出システムの開発. 電子情報通信学会論文誌 D, Vol. 92, No. 9, pp. 1456–1466, 2009.

[5] 小渡悟, 神里志穂子, 星野聖. 類似動作で意味が異なる手話単語の所要時間が弁別に及ぼす影響. 映像情報メディア学会誌, Vol. 56, No. 2, pp. 302–306, 2002.

[6] The Institute of Electronics, Information and Communication Engineers. https://search.ieice.org/bin/search.php?lang=J. (2021 年 1 月 13 日閲覧).

[7] 赤沢史嗣, 佐藤優太, 村松大吾, 松本隆, 中村厚, 宗田孝之. 手のひら可視光分光画像における現状特徴を用いた生体認証の試み. 電子情報通信学会論文誌 A, Vol. J97-A, No. 10, pp. 665–668, 2014.

[8] 矢野昌平, 荒川隆行, 越仲孝文, 今岡仁, 入澤英毅. 誤差の周波数拡散と加算平均処理による耳音紋認証の精度向上. 電子情報通信学会論文誌 A, Vol. J100-A, No. 4, pp. 161–168, 2017.

[9] 武村紀子, 白神康平, 槇原靖, 村松大吾, 越後富夫, 八木康史. 畳み込みニューラルネットワークを用いた視点変化に頑健な歩容認証. 電子情報通信学会論文誌 A, Vol. J99-A, No. 12, pp. 440–451, 2016.

[10] 長田礼子, 尾崎哲, 青木輝勝, 安田浩. 手指動からの特徴抽出によるリアルタイム個人認証. 電子情報通信学会論文誌 D, Vol. J84-D2, No. 2, pp. 258–265, 2001.

[11] 橋本侑樹, 村松大吾, 小方博之. 替え玉防止に向けたペン持ち方認証法におけるなりすまし耐性の強化. 電子情報通信学会論文誌 A, Vol. J96-A, No. 12, pp. 769–779, 2013.

[12] 西田貴幸, 福元伸也, 鹿嶋雅之, 佐藤公則, 渡邊睦. タイピング時における FEI を用いた個人認証に関する研究. 電子情報通信学会論文誌 A, Vol. J103-A, No. 12, pp. 303–305, 2020.

[13] 山田猛矢, 福元伸也, 鹿嶋雅之, 佐藤公則, 渡邊睦. キー操作とマウス操作の動的バイオメトリクスを用いた継続認証アルゴリズム DPTM の提案と認証精度. 電子情報通信学会論文誌 A, Vol. J103-A, No. 11, pp. 255–269, 2020.

[14] 矢内浩文, 水野喜夫. 発話時の頭部揺らぎを利用した個人分類. 電子情報通信学会論文誌 D, Vol. J95-D, No. 9, pp. 1686–1687, 2012.

[15] J. M. Pang, V. V. Yap, and C. S. Soh. Human behavioral analytics system for video surveillance. In *Proceedings of the 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, pp. 23–28, 2014.

[16] Y. Horiuchi, Y. Makino, and H. Shinoda. Computational Foresight: Forecasting Human Body Motion in Real-time for reducing Delays in Interactive System. *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS'17)*, pp. 312–317, 2017.

[17] 中原啓太, 山口弘純, 東野輝夫. 移動型センサと kinect を用いた家庭内の行動ロギング手法. 2016 年度 情報処理学会関西支部 支部大会 講演論文集, 第 2016 巻, 2016.

[18] 中島雅貴, 小篠裕子, 斎藤英雄. 時系列情報を考慮した人体骨格追跡と評価. 電子情報通信学会技術研究報告, Vol. 117, No. 391, pp. 267–270, 2018.

[19] 渡邊昭信, 味松康行, 村上俊夫, 中村克行. 上方視点距離画像を用いた人物姿勢推定手法の検討. 情報処理学会研究報告, Vol. 2017-CVIM-209, No. 29, pp. 1–13, 2017.

[20] 森駿文, 菊池浩明. 深度センサによる保養特徴量を用いた個人識別・追跡方式の提案. コンピュータセキュリティシンポジウム 2017 論文集, 第 2017 巻, pp. 972–979, 2017.

[21] B. Dehbandi, A. Barachant, D. Harary, J. D. Long, K. Z. Tsagaris, S. J. Bumanlag, V. He, and D. Putrino. Using Data From the Microsoft Kinect 2 to Quantify Upper Limb Behavior: A Feasibility Study. *IEEE Journal of Biomedical and Health Informatics*, Vol. 21, No. 5, pp. 1386–1392, 2017.

[22] 山口弘純, 安本慶一. エッジコンピューティング環境における知的分散データ処理の実現. 電子情報通信学会論文誌 B, Vol. J101-B, No. 5, pp. 298–309, 2018.

[23] 川村隆浩, ワコラ, 中川博之, 田原康之, 大須賀昭彦. インタラクションシーケンスに着目した商品検索目的抽出エージェントの開発. 電子情報通信学会論文誌 D, Vol. J94-D, No. 11, pp. 1783–1790, 2011.

[24] 川村隆浩, 越川兼地, 中川博之, 清雄一, 田原康之, 大須賀昭彦. メディア情報の Linked Data 化と活用事例の提案. 電子情報通信学会論文誌 D, Vol. J96-D, No. 12, pp. 2987–2999, 2013.

[25] B. Hoffman and R. Bhattacharya. 機械学習と、深層学習の基礎知識. https://developer.ibm.com/jp/technologies/machine-learning/articles/l-machine-learning-deep-learning-trs/, 2016. (2021 年 1 月 13 日閲覧).

[26] 岡谷貴之. 深層学習. 講談社, 2015.

[27] 竹内一郎, 烏山昌幸. サポートベクトルマシン. 講談社, 2015.

[28] 玄関の鍵をキーレスに交換したい！種類・費用や注意ポイントを紹介 — 鍵屋の鍵猿. https://sls.co.jp/kagizaru/other/post863/, 2020. (2021 年 1 月 23 日閲覧).

[29] 一戸建て･一軒家向けの防犯対策|ホームセキュリティのセコム. https://www.secom.co.jp/homesecurity/plan/kodate/. (2021 年 1 月 13 日閲覧).

[30] 新築一戸建ての防犯ホームセキュリティ. https://www.alsok.co.jp/person/ikkodate/. (2021 年 1 月 13 日閲覧).

[31] 鍵の本：ピッキング完全対策：カギ業界 VS 仁義なき犯罪集団. シーズムック. 満足王国シリーズ, No. 33. シーズ情報出版, 2001.

[32] 平成 30 年の刑法犯に関する統計資料. https://www.npa.go.jp/toukei/seianki/H30/h30keihouhantoukeisiryou.pdf, 2019. (2021 年 1 月 13 日閲覧).

[33] 平成 30 年住宅・土地統計調査 住宅及び世帯に関する基本集計 結果の概要. https://www.stat.go.jp/data/jyutaku/2018/pdf/kihon_gaiyou.pdf, 2019. (2021 年 1 月 13 日閲覧).

[34] 全国の都市における人の動きとその変化ー平成 27 年全国都市交通特性調査　集計結果よりー. https://www.mlit.go.jp/common/001213314.pdf. (2021 年 1 月 13 日閲覧).

[35] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. P. de la Blanca. Multimodal feature fusion for CNN-based gait recognition: an empirical comparison. *Journal of the Neural Comput & Applic (2020)*, pp. 14173–14193, 2020.

[36] Y. Yang, Z. Cai, Y. Yu, Y. Wu, and L. Lin. Human Action Recognition Based on Skeleton and Convolutional Neural Network. In *Proceedings of the 2019 Photonics Electromagnetics Research Symposium - Fall (PIERS - Fall)*, pp. 1109–1112, 2019.

[37] W. Nie, W. Wang, and X. Huang. SRNet: Structured Relevance Feature Learning Network From Skeleton Data for Human Action Recognition. *Journal of the IEEE Access*, Vol. 7, pp. 132161–132172, 2019.

[38] 岡留有哉, 魏文鵬, 相薗敏子. 複数の再帰型ニューラルネットワークを用いた需要予測アーキテクチャの開発. 電子情報通信学会論文誌 D, Vol. J103-D, No. 1, pp. 24–33, 2020.

[39] Q. Ye, X. Yang, C. Chen, and J. Wang. River Water Quality Parameters Prediction Method Based on LSTM-RNN Model. In *Proceedings of the 2019 Chinese Control And Decision Conference (CCDC)*, pp. 3024–3028, 2019.

[40] Y. Dong, R. Wen, Z. Li, K. Zhang, and L. Zhang. Clu-RNN: A New RNN Based Approach to Diabetic Blood Glucose Prediction. In *Proceedings of the 2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB)*, pp. 50–55, 2019.

[41] 上田修功, 斉藤和巳. 多重トピックテキストの確率モデル—パラメトリック混合モデル—. 電子情報通信学会論文誌 D, Vol. J87-D2, No. 3, pp. 872–883, 2004.

[42] 岩井秀成, 池田郁, 土方嘉徳, 西田正吾. レビュー文を対象としたあらすじ分類手法の提案. 電子情報通信学会論文誌 D, Vol. 96-D, No. 5, pp. 1222–1234, 2013.

[43] M. Shirakawa, K. Nakayama, T. Hara, and S. Nishio. Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes. *Journal of the IEEE Transactions on Emerging Topics in Computing*, Vol. 3, No. 2, pp. 205–219, 2015.

[44] Y. Choubik and A. Mahmoudi. Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data. In *Proceedings of the 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pp. 307–311, 2016.

[45] D. Xu, X. Xiao, X. Wang, and J. Wang. Human action recognition based on Kinect and PSO-SVM by representing 3D skeletons as points in lie group. In *Proceedings of the 2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 568–573, 2016.

[46] M. Seifallahi, H. Soltanizadeh, A. H. Mehraban, and F. Khamseh. Alzheimer ' s disease detection using skeleton data recorded with Kinect camera. *Cluster Computing* on Springer, The Journal of Networks, Software Tools and Applications (2020), Vol. 23, pp. 1469–1481, 2020.

[47] S. Raschka. When Does Deep Learning Work Better Than SVMs or Random Forests? `https://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html`. (2021 年 1 月 13 日閲覧).

[48] M. Martinez-Alanis, E. Bojorges-Valdez, N. Wessel, and C. Lerma. Prediction of Sudden Cardiac Death Risk with a Support Vector Machine Based on Heart Rate Variability and Heartprint Indices. *Journal of the Sensors (Basel)*, Vol. 20, No. 19, p. 5483, 2020.

[49] 財津亘, 金明哲. テキストマイニングを用いた著者の年齢層推定—サポートベクターマシーンとランダムフォレストの精度比較—. 日本心理学会大会発表論文集, Vol. 81, pp. 1A–042–1A–042, 2017.

[50] Y. Tang. Deep Learning using Support Vector Machines. *CoRR*, Vol. abs/1306.0239, , 2013.

[51] 杉浦司. Kinect for Windows v2 入門-C++プログラマー向け連載. `https://www.buildinsider.net/small/kinectv2cpp`, 2014. (2021 年 1 月 13 日閲覧).

[52] D. Cournapeau. Scikit-learn: Choosing the right estimator. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html. (2021 年 1 月 13 日閲覧).

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Vol. 12, No. 85, pp. 2825–2830, 2011.

[54] 加茂川誠. Smart Life Net. `http://kamomako.hatenablog.jp/entry/kenkoutisiki/taijyuu-heikin-nenreibetu`. (2021 年 1 月 13 日閲覧).

[55] PHYSIQUE OF YOUTHS BY AGE (2000–14). `https://www.stat.go.jp/data/nenkan/65nenkan/zuhyou/y652402000.xls`. (2021 年 1 月 13 日閲覧).

[56] ピッキングって本当にできるの?`https://keyworldsos.com/2018/04/21/picking-3/`. (2021 年 1 月 13 日閲覧).

[57] 戸田哲郎, Alessandro Moro, 梅田和昇. 単眼カメラから得られる骨格情報を用いた人物識別. 精密工学会学術講演会講演論文集, Vol. 2018S, pp. 267–268, 2018.

[58] A. Sakhapara, D. Pawade, S. Dedhia, T. Bhanushali, and V. Doshi. Machine Learning Based Approach for Person Identification in Group Photos. In *Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, 2018.

[59] K. S. N. Ripon, L. E. Ali, N. Siddique, and J. Ma. Convolutional Neural Network based Eye Recognition from Distantly Acquired Face Images for Human Identification. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[60] 清水仁, 長谷川美紀. SVM を用いた足跡からの男女識別の実現. 電子情報通信学会論文誌 D, Vol. J93-D, No. 5, pp. 642–646, 2010.

[61] X. Wu and J. Zhao. Curvelet feature extraction for face recognition and facial expression recognition. In *Proceedings of the 2010 Sixth International Conference on Natural Computation*, Vol. 3, pp. 1212–1216, 2010.

[62] K. Li, Z. Liu, L. Liang, and Y. Song. Human action recognition using associated depth and skeleton information. In *Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 418–422, 2016.

[63] 菅谷充. Python によるウィルコクソンの符合付順位検定. `https://www.m-sugaya.jp/python/signed_rank_test.html`. (2021 年 1 月 13 日閲覧).

業績

[1] M. Shiraishi and R. Uda. Detection of Suspicious Person with Kinect by Action Coordinate. In *proceedings of the 2019 13th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, vol. 935, pp. 456-472, 2019（Peer reviewed publications）.

[2] 白石将貴，宇田隆哉，藤川真樹．Kinect を用いた行動座標によるピッキング行為の検知．情報処理学会論文誌，Vol. 61，No. 2，pp.486-499，2020.

[3] M. Shiraishi and R. Uda. Comparison of Algorithms and Action Coordinates Sets in Detection of Slight Differences in Motions like Lock-Picking. In *proceedings of the 2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1-8, 2020（Peer reviewed publications）.