東京工科大学

博　士　学　位　論　文

覗き見耐性を有し多人数の利用に対応可能な空中筆記による本人確認手法に関する研究

西暦 2022 年 9 月

釜石　智史

目次

図目次

表目次

第 1 章

はじめに

1.1 背景

本人確認は，様々なサービスを安全に利用するのにおいて必要不可欠であり，状況に応じていくつかの種類が用いられている．最も一般的な手法はパスワード，またはPIN によるもので，Web などのオンラインサービス，ATM，クレジットカードなど多くの場所で用いられている．パスワードの他には，指紋や静脈などの生体情報を用いた本人確認や，セキュリティデバイスを用いた本人確認方法もある．また，人と人で直接対面する場合，身分証明書の確認などが通常用いられている．これらの方法にはそれぞれ利点と欠点があり，すべてのサービスにおいて万能に適用できる方法はない．

民間団体が提供するサービスにおける本人確認には，パスワードを用いるものが一般的である．再発行の手間や変更時の利便性，実装と運用のコスト，本人確認に掛かる時間などを考慮すると，最も適しているからである．

指紋や顔などの生体情報による本人確認は，ドアのロックや，スマートフォンのロック解除などに用いられている．生体情報は任意に変更することがほぼ不可能であることが問題であり，仮に変更したいと考えた場合は，整形手術で顔や指紋を変える必要がある．そのため，生体情報が流出しないように厳格に管理する必要がある．研究としては，本物ではない指紋を登録して照合するものがあり [1]，残留指紋から，指紋照合装置に誤認識させることができる，偽の指を作ることも可能であると示されている [2]．また，銀行などで使用されている指静脈による本人確認においても，装置に誤認識させることができると示されている [3].

生体情報による本人確認技術として，日々の行動を，位置情報などのデジタルデータとして記録する，ライフログを用いるものも存在する [4, 5, 6]．ライフログ自体は，絶対的に完全に個人を識別できるものではなく，一定割合の攻撃者を対象から除外することで，セキュリティを高めるために使用される．指紋や静脈よりは簡単にライフ

ログは変更することはできるが，行動を強制的に変更する必要があり，日々の生活に支障や負荷が生じることは否めない．また，ライフログが漏洩した場合に，指紋や静脈よりも攻撃者による複製が容易であることも問題である．更にライフログには日々の位置情報など，プライバシーに関わるものが含まれているため，漏洩によりプライバシーの侵害を受ける危険性もある．

オンラインサービスの中には，セキュリティデバイスを用いて本人確認を行うものがある [7, 8, 9]．この手法は物理的な盗難に対して脆弱である．また，デバイスを紛失したり，デバイスが故障したりすると，再発行までの間サービスが利用できなくなる．なお，再発行の過程においてソーシャルエンジニアリングなどによる攻撃を受けないしくみが必要であり，簡易な手段で再発行を行うことには問題がある．

対面による本人確認の場合，そのセキュリティは対応した人物の能力に大きく依存する．クレジットカードの署名を目視で確認する場合，人間にとって類似の署名を見分けることは困難である．身分証明書を確認する場合にも，それが偽造されたものであるかどうか見抜くにはそれなりの知識や経験が必要である．偽造防止の特殊な IC チップを含む旅券も存在するが，すべての民間団体が容易にその確認を行えるわけではない．一方で，米国のスーパーマーケットや小規模な雑貨店で一般的に用いられている，レジにおけるクレジットカードの電子的な署名は，機械的に手書きの署名を照合する技術があれば，人間の目視に依存しない本人確認が可能である．この署名が機械的に確認されているかどうかは非公表であり，わからないが，手書きの署名を照合する研究 [10, 11, 12] は存在するため，理論的には可能である．

以上より，情報漏洩のリスクや変更・再発行のコスト，人間の確認能力に依存しない性質などを考慮すると，パスワードが最も無難な選択肢であることは間違いない．しかし，パスワードも万能ではない．覚えやすいパスワードは攻撃者に推測されやすいし，複雑で長いパスワードは記憶や入力が困難である．そこで，追加の本人確認手段を用いてセキュリティを強化する手法も採られている．例えば，通常と異なる IP アドレスからログインした場合，登録されたメールアドレスにワンタイムパスワードが送信され，その入力を要求されるなどがある．クレジットカードによっては，海外の IP アドレスから決済しようとすると一時的にロックが掛かる場合もある．前述のセキュリティデバイスも，基本的には，セキュリティデバイス単体での利用より，パスワードと併用による 2 段階の本人確認のために用いられていることが多い．

1.2 目的

本研究の目的は，本人確認を行うためのデバイスの再発行を行うことなく，更に，指紋などの変更できない生体情報を使用せずに本人確認を行う方法として，空中に指で

筆記を行うことである．空中に筆記をすることで筆跡が残らないため，何を書いているのか一目ではわからないようにできる．これにより覗き見に対してある程度の耐性を持たせることができる．

本研究は 3 つの研究で構成されている．1 つ目は Leap Motion による空中筆記に関する研究で，この研究では空中で指で署名を筆記させることにより，覗き見に対する耐性を持たせることを行いたかった．既存のクレジットカード使用時の本人確認で使用される，PIN やサインの問題点は覗き見に弱いという問題点があった．そこで，Leap Motion と呼ばれるデバイスを用いて空中に指で署名を書かせ，その署名を検証することで個人識別を行う手法を提案した．空中に筆記を行うことにより，筆跡を残さないようにした．これにより，一度の覗き見によるコピーを難しくし，覗き見によるコピーに対してある程度の耐性を持たせることで，既存の問題点を改善しようとした．この研究ではクレジットカード利用時の個人の検証を想定しているため，覗き見に対して耐性を持たせることにより，カード利用時の検証動作を覗き見されたとしても，模倣を困難にすることで不正利用を防ぐことができると考えた．この研究では取得する特徴量が少なかったため，精度が悪かった．更に，同じ文字の署名を複数人が登録できない問題もあった．

2 つ目の研究では 1 つ目の研究の改善として，取得する特徴量を増やすことと，増えた特徴量に対応するため，機械学習を使用することで，検証速度と精度の向上を図り，同じ筆記を行った場合でも正しく個人を識別できるようにしたいと考えた．この研究はドアロックへの利用も想定しているため，署名を空中に筆記させるのではなく，一方向に手を移動させるなどの単純な動作で，個人を識別できるようにしたかった．1 つ目の研究は精度が低いという問題があった．ドアの鍵として一般的に使用される IC カードや物理的な鍵は，紛失時に再発行が必要であり，盗難にも弱いという問題点がある．ドアロックとしてパスワードもあるが，パスワードは前の研究でも指摘した通り，覗き見される問題点がある．1 つ目の研究からの改善方法として，Leap Motion を使用し，一方向の線を空中に筆記させ，筆記時の特徴から個人識別を行う．これにより登録者全員が同じものを書いたとしても，正しく個人識別をできることを目的とした．前の研究では人差し指の情報しか取得していなかったが，すべての指の情報を取得することで特徴の数を増やした．特徴量が増えたことにより，前の研究の検証方法では時間がかかると考え，訓練には時間がかかるが，検証は早く行うことができる機械学習を使用した．この研究がうまくいった場合，既存のドアロックの問題点である，再発行の必要性がなくなる．更に，手を一方向に動かすだけで良いため，鍵を持ち歩いたり，パスワードを覚える必要もない．この研究では登録者の数が増減した場合に機械学習を再度行う必要があるうえ，登録者が多くなった場合には機械学習を行うコストが高くなる問題があった．更に 1 つ目の研究と比較して空中に筆記する動作

が単純であり，覗き見に対する耐性が低くなるため，人前で利用することができない問題もあった．

3 つ目の研究では既存の本人確認を強化するための 2 段階目の本人確認を行いたかった．既存の 2 段階目の本人確認には，パスワードや，専用のデバイスを用いて本人確認を行うが，パスワード，専用のデバイスの問題点は前の 2 つの研究で記述した通り，デバイスは紛失時に再発行が必要であり，盗難にも弱いという問題点があり，パスワードは覗き見される問題点がある．2 段階目の本人確認として使用する場合，多人数での利用を想定する必要があり，前の研究の機械学習方法では多人数の利用に対応できないことと，覗き見耐性が低いという問題があった．そこで，この研究では 1 つ目と 2 つ目の研究を組み合わせ，好きな単語を空中に筆記させ，書かせた単語を一方向の線に分解し，分解してできた線を筆記する時の特徴とし，機械学習を使用して筆記時の特徴から本人確認を行う方法を提案する．この研究では 1 つ目の研究と同じように何らかの単語を書くことになるが，単語そのものの比較ではなく，単語を書く時の癖を登録し，それを本人確認を行うときに比較するため，登録時と本人確認時に書く単語を毎回ランダムにすることができる．筆記する単語をランダムにすることで，筆記時の手の動きも毎回変わるため，1 つ目の研究より覗き見に対しての耐性を持たせられると考えた．筆記した単語を線に分解することで 2 つ目の研究と同じように機械学習を利用して本人確認を行うことができる．そこで，2 つ目の研究から機械学習を変更し，更に，1 つのモデルに登録する人数を固定することで，利用者が増減した場合にも機械学習を一からやり直すことなく対応できるようにする．これにより多人数での利用を可能にする．

研究の目的をまとめると，この研究では筆記時の癖を本人確認の特徴として使用するため，パスワードや本人確認デバイスが必要がなく，本人確認デバイスの再発行を行う必要がない．空中に指で筆記をすることで筆跡が残らないため，サインの筆跡やパスワードの入力などにある，覗き見をされてしまう問題点を解決できる．仮に覗き見をされて筆記したものが分かってしまったとしても，全ての指の動きを真似しなければならないため，攻撃者が簡単に真似できないようにした．本人確認のたびに空中に単語を 1 つだけ書かせ，機械学習のモデルに入れて本人確認を行うため，検証に長時間を要しない．機械学習で利用者が増減した場合でも，再度機械学習をする必要をなくすことで，本研究の利便性を向上させる．更に，機械学習を行うときのコストを一定に保たせることで，多人数での利用に対応できるようにする．多人数での利用を想定しているため，機械学習のモデルの中に攻撃者のデータが含まれていなくても正しく本人確認を行えるようにする．

これらの目的を，本研究の最終的なシステムの要件とし，これを特徴とする．以下に要件を記述する．

要件 1　本人確認デバイスの再発行を行わない
要件 2　覗き見によるコピーに対してある程度の耐性を持つ
要件 3　一度の覗き見によるコピーを行わせない
要件 4　入力および本人確認に長時間を要しない
要件 5　利用者の増加による機械学習の再訓練[*1]を行わない
要件 6　利用者が非常に多い場合でも，機械学習による訓練コストを一定に保つ
要件 7　攻撃者のデータが訓練に含まれない場合にも本人確認が行える

要件 1 は利用者ごとに異なるセキュリティデバイスや，IC カードを使用せず，市販されている Leap Motion を用いて空中に筆記を行うときの癖を個人の特徴として本人確認を行う．そのため Leap Motion は誰がどれを使ってもよく，本人確認デバイスの再発行の必要がなくなる．空中に筆記を行うときの癖を利用する場合，この要件は満たすことができるため，3 つの研究全てが要件 1 を満たす．要件 2，3 についても空中に筆記を行うことで満たすことができる．空中に筆記を行うことで筆跡が残らないため，1 回見ただけではどのような文字を書いたかわからなくできる．仮に，攻撃者が録画により何度も見てコピーをしようとしても指の動き，向きをコピーせねばならず，攻撃者に大きな負担を与えることができる．要件 2 については 3 つの研究全てで満たすことができる．要件 3 については 2 つ目の研究は一方向の移動しかしないため，どのような動きをしているのか，わかってしまうため満たすことができない．しかし，手を動かしているときの各指の向きや速さを真似するのは攻撃者に負担がかかる．1 つ目と 3 つ目の研究は署名や単語を書いたとしても，1 度の覗き見では書いた文字がわからないため，満たすことができる．要件 4 については機械学習を使用することで登録時の訓練には時間がかかるが，検証時の本人確認の時間を短縮させた．そのため 2 つ目と 3 つ目の研究が要件を満たすことができる．要件 5 は一度訓練した機械学習のモデルを，登録者が増えた時に再度訓練させず，新たなモデルを作ることで満たすことができる．更に，登録者が減少した場合でも，既存のモデルに利用者がいる場合はそのモデルを使い続けるようにすることで再訓練する必要をなくす．要件 6 は 1 つのモデルに登録する人数を固定することで，人数が増えたとしても，1 つのモデルの訓練コストは変わらないようにすることで満たすことができる．要件 7 は訓練時に攻撃者のデータを用意しなくとも，本人確認を行うときに正しく本人確認をできるようにすることで満たすことができる．要件 5，6，7 は 3 つ目の研究が要件を満たしている．

本研究の技術を用いることで，セキュリティトークンや IC カードのような，持ち

[*1] 本論文における「訓練」は，機械学習における training の訳語である．「学習」が使用される論文もあるが，これは learning の訳語として使用されており，混乱が生じるため本論文では training の訳語としては使用しない．ただし，「過学習」は over fitting / over training の訳語として定着してしまっているため，そのまま使用するものとする．

主ごとで異なるデバイスは，盗難や紛失の際に再発行が迅速に行えないことが問題であったが，覗き見に対する耐性を持たせたうえで，パスワードなどによる本人確認を補助的に強化することができる．これにより，ネットショッピングなどの決済において，追加の本人確認手段として利用でき，e-Tax などのしくみへの置き換えや，将来の選挙における電子投票などにも使用できる．さらに，覗き見に対する問題にも対処しているため，対面による本人確認にも利用できる．

第 2 章

関連研究

本章では，1.2 節で挙げた要件に関連研究を照らし合わせながら紹介する．

2.1 機械学習を利用しない筆記を用いた本人確認手段

Hanyu らは，データセットに登録した，ひらがな，カタカナ，漢字の一部からランダムに数文字を選択し，iPad に指で書かせることで，筆記を行うときの筆跡を特徴として，ユーザ間のユークリッド距離を求めることで，個人識別を行う研究を行っている [13]．精度は誤判定率 3.1% で，全体的な精度は 92.0% であった．更に，アルファベット以外のひらがな，カタカナ，漢字であったとしても，個人の識別をすることが可能であると述べている．この研究は iPad 上に指で筆記を行うことから，タッチパネル上に指の跡が残ることになる．そのため，スマッジアタックをされる可能性があり，要件 2 と要件 3 を満たさない可能性がある．

Kato らは，ペンタブレットを用いて筆記時のペンの筆圧，傾き，筆跡から個人識別を行う研究を行っている [14]．枠やいくつかの「○」が書かれたパターンのシートを用意し，筆記時にそのシートに従い，枠の中で数字を筆記させたり，○から○まで線を引かせたりしたときの筆記時のストロークを特徴とし，DP マッチング (動的計画法) することで筆者の識別をする．FAR と FRR が等しいときの精度は 92.59% となった．未知の筆者を識別するため，全ての登録者との比較を行うと考えられる．そのため，登録者が多い場合に要件 4 を満たせなくなる可能性がある．更に，ペンを動かすときの動作を見られてしまうことから，要件 2，3 を満たさない可能性がある．

Takahashi らは，スマートフォン上に記号を指で書くことで，個人識別を行う研究を行っている [15]．記号を筆記するときの指から特徴を取得し，筆記の比較はユークリッド距離を用いて行っている．個人識別の精度は EER が 10% とあるため，90% となる．テスト時にすべての登録者で比較を行うため，登録者が多い場合に要件 4 を満

たさすことができない可能性がある．更に，スマートフォン上に指で書くため，残った指紋から筆跡を辿られることが考えられ，要件 2 と要件 3 も満たさない可能性がある．

Sae-Bae らは，タブレット端末のような大型のマルチタッチパネルの上に 5 本の指を置かせ，時計回りになぞるや，一方向にスワイプするなどのジェスチャーによって，個人を識別する研究を行っている [16]．タッチパネルでなぞったジェスチャーから，DTW により，個人識別のを行う．その結果，90% の精度で個人識別を行うことができたと述べられている．この研究もパネル上に指の跡が残ることから，要件 2 と要件 3 を満たさない可能性もある．

Shen らは，手首に専用のセンサを取り付け，ペンで紙に筆記を行ったときの動作を DTW によって比較し，個人識別を行う研究を行っている [17]．軸 X，Y，Z の加速度と角速度を特徴として取得している．結果として，false negative rate は 1.78% で false positive rate of 6.7% と記述されている．この研究では登録されているデータでグループを作り，検証を行うデータを登録されているデータのグループで検証を行っているため，登録者が増えるとグループが大きくなり，検証に時間がかかると考えられ，要件 4 を満たすことができない．

坂田らは，Leap Motion による特徴抽出から個人を識別する手法を提案している [18, 19]．坂田らの研究では，3 回同じように空中で「＋」の記号を書かせ，その特徴を比較している．その結果，最初の被験者と同じ特徴を持つ被験者は，X 方向で 11 人，Y 方向で 14 人，Z 方向で 17 人であり，3 方向すべてが同じとなる被験者は 6 人であった．坂田らはこの結果より，Leap Motion による空中署名は個人識別に十分な精度を持つと述べている．

畠中らは，Leap Motion を用いて空中に名前を手書きで書かせ，DP マッチングによって個人識別を行う研究を行っている [20]．閾値を最良の 0.58 としたときの本人拒否率は 6.2% であり，この時の他人受入率は，本人の動作を見てなりすました場合は 12.5%，筆記動作を隠して同じ文字列を入力した場合は 1.5%，異なる文字列を入力した場合は 0.0% となっている．この論文中で言及されているが，他人が署名の動作を知っている，もしくは動作は知らなくても入力した文字列を知っている場合には，なりすまされてしまう可能性がある．更に畠中らは，入力の途中で手の形を変えることで筆記動作を知らないが，文字列を知っている攻撃者に対する耐性を高めた研究を行っている [21]．この手法では，手の形を変える回数，順番，形は登録された通りでなければならない．そのため，覗き見攻撃を受けてしまった場合に対する耐性は変わらず，更に，動作が増えたことと，覚えることが増えたことから要件 2，3，4 を満たすことができない．

Renuka らは，ペンの構造をした入力デバイスを用いて文字を書き，その文字を識別する評価を行っている [22]．文字認識アルゴリズムは言及されていない．また，文字

入力を認証に使用するとしているが，書かれる文字の目視が困難と述べている，入力した文字によるパスワード認証が想定されている．精度としては 98.25% で文字の識別に成功している．彼らは 0 から 9 までの数字を書く評価しか行っていないが，筆記の軌跡を見る限り覗き見による判別は容易であり，筆記の特徴を本人確認に使用しないため要件 2 および要件 3 を満たさない．

Hu らは，テンプレートクラスタリングに基づく 2 段階の署名検証システムによってペンタブレットで書かれた署名が誰のものか分類する研究を行っている [23]．クラスタリングの閾値を超えていた署名のみを，2 段階目の個人識別に使用している．EER は 1 段階目で 2.50%，2 段階目で 0.83% となっている．テスト時には，入力された署名をすべての登録された署名で，類似性スコアの比較を行わなければならないため，登録者が多い場合に要件 4 を満たさない．

Xiao らは，Leap Motion で空中に署名を手書きし，個人識別を行う研究を行っている [24]．空中に書いた署名と手の形からテンプレートマッチングを用いて筆者を検証している．入力する署名はテンプレートとして登録し，比較により検証を行っている．テンプレートマッチングは 3 つの方式で行い，その中で最も精度の高かった方式と，手の形から最終的な評価としている．結果としては，手の形については精度が 65.2% であり，筆記時の動作については 96.2% であった．全ての被験者が自分自身の署名を登録すると記述されていることから，全員が違う形の署名を書いていると考えられる．これにより，本人のテンプレートと他人の署名を比較すれば，まったく違うということになってしまうため，精度は高くなってしまうことが考えられる．また，署名全体をテンプレートマッチングにより比較しているため，署名が長くなると訓練や検証に掛かる時間も増大する．そのことから要件 4 を満たすことができない．

以上より，既存の関連研究で機械学習を使用しない場合は要件 4 を満たすことができないものがある．これは，テストの際の入力に時間が掛かることではなく，入力されたデータを登録済みのデータと比較する際に生じる問題である．

一部要件 4 を満たしている手法があるが，これらのものは特定の署名（もしくは名前の筆記など）を使用しているか，動きそのものに覗き見耐性がない．例えばジェスチャーを登録する Sae-Bae らの手法は，ジェスチャーを真似されることを想定していないが，本研究の手法では同じ単語の入力を続けて要求しなければこれを避けることができる．Renuka らの手法は，そもそもパスワード入力であるため，覗き見耐性が考慮されていない．特定の署名を登録する手法に関しては，本論文の要件 1〜7 を基本的にはすべて満たす．しかし，一度覗き見たものが動画などで，攻撃者が何度も練習可能である場合には要件 3 を満たさなくなる可能性がある．一方，本研究の手法では同じ単語の入力を続けて要求しなければこの攻撃は行えない．

2.2 機械学習による筆記を用いた本人確認手段

Alkaabi らは筆記した画像データを用いて個人識別を行う研究を行っている [25]. 機械学習には CNN(Convolutional Neural Network: 畳み込みニューラルネットワーク) を用いる．本人が書いた署名と，他人が同じ署名を書いたデータを訓練データとして訓練を行う．検証時には 2 つの署名が同一人物が書いたものであるかの識別を行う．精度としては EER が 14% となっている．利用者が増えるたびにそれまでの利用者の署名とも区別するのであれば要件 5，6 を満たさない可能性がある．筆記した画像データを見られた場合に容易にコピーされる可能性があるため，要件 2，3 を満たさない．

Lu らは，特製のウェアラブルデバイスが貼り付けられた手袋をつけ，空中に署名を手書きして個人識別を行う研究を行っている [26]．DTW(Dynamic Time Warping: 動的時間伸縮法) で入力信号の長さを合わせ，SVM(Support Vector Machine) による分類を行っている．SVM での分類は本人のデータと他人のデータで分類を行い，アカウントごとで分類器を構築している．EER は 0.1% となっている．この研究で使用されているデバイスは市販されているものではないため，作る必要があり，入手性に問題がある．更に Lu らの方式は，本人と他人の 2 値分類であり，利用者が増えた場合，再訓練が必要と考えられ，要件 5，要件 6 を満たさない．

Behera らは，Leap Motion の上で利用者に空中に署名を筆記させることで，個人識別を行う研究を行っている [27]．Leap Motion で利用者に空中に筆記させた署名を DTW で入力信号の長さを合わせ，k-NN(K-nearest neighbor algorithm:k 近傍法) を用いて分類する方法と，HMM(Hidden Markov Model: 隠れマルコフモデル) に入力して分類する方法の 2 つを提案している．どちらもどの研究と比較してなのかわからないが，k-NN を用いた場合，認識精度が 6.8%、検証精度が 9.5% 向上したと述べていて，HMM を用いた場合，認識精度が 9.9%、検証精度が 6.5% 向上したと述べられている．この論文では Leap Motion が既存の生体認証システムの代替となることを確信していると述べられていて，Leap Motion で個人識別を行うことの有効性を示している．k-NN で多値分類する場合と HMM で 2 値分類する場合において，利用者が増加する際に要件 5 を満たさない．また，利用者が多い場合に要件 6 も満たさない．

Nohara らは，機械学習を使用し，スマートフォンのフリック入力から，個人識別をする研究を行っている [28]．スマートフォンでフリック入力で文字を入力したときの指の向き，加速度を特徴として，これを自己組織マップと呼ばれる機械学習を使用して，訓練を行い，個人を識別している．分類の結果，識別精度は 92.8% となった．Nohara らはこの結果から，個人の識別において SOM を使用することは有効であると述べている．

Yamamoto らは，Leap Motion で利用者に数字を書かせ，CNN により個人識別を行う研究を行っている [29]．利用者には 0〜9 の数字を書かせ，全員のデータを訓練し，2 クラスに分類する．1 クラスは本人で，もう 1 つのクラスが他人となっている．そのため，利用者ごとにモデルを作ることになる．結果として，FRR は 3.8%，FAR は 5.9% となった．利用者ごとに CNN を用いて 2 値分類を行っているため，利用者が増加する際に本人と他人のモデルを作り直す必要があるため，要件 5 を満たさない．また，利用者が多い場合に要件 6 も満たさない．

Mohammed らは，ペンタブレットで筆記したアラビア語の署名の画像で個人識別を行う研究を行っている [30]．この論文では SVM と k 平均法を両方使用している．ペンタブレットでアラビア語を書いた時に，筆者を k 平均法により識別し，複数人が登録されている SVM で本人に分類されるかで，筆記の検証を行っている．その結果，精度は 96.67% となった．k 平均法と SVM の使用は，利用者が増加する際に再度訓練が必要であり，要件 5 を満たさない．また，利用者が多い場合は訓練コストが上昇するため，要件 6 も満たさない．

Singh らは，紙に書かかれた手書き文字から筆者の個人識別を行う研究を行っている [31]．ペンで紙に文字を書かせ，紙の画像から文字を解析することにより，筆跡から k 平均法により，筆記をした人の識別を行っている．k 平均法による多値分類を使用しているため，利用者が増加した際，再度訓練が必要なため，要件 5 を満たさない．また，利用者の増加に比例して機械学習による訓練コストが増加するため，要件 6 を満たさない．

de Rosa らは，センサーが内蔵されたペンで名前の筆記を行い，取得された傾きや加速などの要素を画像化することで，個人を識別する研究を行っている [32]．ネットワークには CaffeNet，CIFAR10 full，MNIST が使用されている．結果は最良の場合で精度が 61.54% となっている．画像の分類には CNN による多値分類が用いられており，利用者が増加する際に要件 5 を満たさない．また，利用者が多い場合に要件 6 も満たさない．

小南らは，ペンタブレットを用いて署名を筆記させ，個人識別を行う研究を行っている [33]．小南らの手法では，筆圧と筆跡の 2 つが HMM で訓練され，モデルとなる．検証の成功率は 99.8% となっている．ただ，攻撃に対する評価がないため，FAR がいくつであるのかはわからない．この研究では，入力した文字が見えるようになっているため，覗き見に対して弱く，要件 2 を満たさない．また，この HMM は利用者全体で訓練を行う必要があるうえ，検証においても全ての被験者と比較を行うため，利用者が増加する際に要件 5 を満たさない．また，利用者が多い場合に要件 6 も満たさない．

高橋らは，筆跡画像を幾何学的に解析することで，個人識別を行う研究を行ってい

る [34]．ペンで書いた筆跡の画像をスキャナーで取り込み，画像をグレースケール化し，ニューラルネットワークに入力している．筆者の識別には階層型ニューラルネットワークにより多値分類をしている．最良で 78% の識別率が得られたと述べている．多値分類であるため，利用者が増加する際に要件 5 を満たさない．また，利用者が多い場合に要件 6 も満たさない．

以上より，既存の関連研究で機械学習を使用する場合は要件 5 と要件 6 を満たすことができない．2 値分類の場合，本人と他人を分類するために，他人を訓練する必要があり，他人は自分以外の登録者のことになる．そのため，新たに利用者が増えた場合には他人を増やすために再訓練が必要になる．再訓練を行わなかった場合，新たな利用者が本人に部分に入る可能性がある．多値分類は既存の関連研究では利用者の数だけ分類するように訓練を行うようにしているため，1 つのモデルに登録する人数の限界について言及していない．そのため，利用者が増えた場合にはモデルを作り直すことになる．更に，利用者が増えることで訓練のコストも上がることになる．

なお，要件 7 については，これらの関連研究において言及されていなかった．論理的には，テストデータによる本人確認を行う前に，OC-SVM (One Class Support Vector Machine) などのアルゴリズムを本人の訓練データに対して適用することで，解決が可能である．実際に OC-SVM を用いた筆跡による本人確認の研究も行われており，Guerbai らは，OC-SVM は大量のサンプルがある際に精度を上げるのに効果的であるが，サンプルの中に訓練に入れると精度を下げてしまうものも含まれているため，OC-SVM カーネルに改良を加えることで精度を上げる手法を提案している [35]．ただし，OC-SVM の適用は，その計算コストが追加されることに加え，利用者人数分の OC-SVM のモデルをシステム側で保持し続けなければならない問題が生じる．さらに，OC-SVM を前段階のフィルタとして追加すると，最終的な FRR (False Rejection Rate: 本人拒否率) に影響を与えるため，FRR の再評価も必要になる．

なお，要件 5 は転移学習を行うことでも解決可能である．Granet らや Aneja らは，転移学習を用いて筆記を認識する研究を行っている [36, 37]．ただし，彼らの研究は筆記により個人を識別するものではなく，筆記された文字を判別するものである．

第 3 章

要素技術

3.1 Leap Motion

Leap Motion Controller(図 3.1) は Leap Motion 社 (現 Ultra leap 社) により開発されたモーションセンサデバイスである．これは指の動きやスワイプなどのジェスチャーが検出できるので，タッチパネルのない PC などをタブレット PC と同じような感覚で操作することができる．他には， ゲームを自分の手を使って操作することができる．例えばゲーム内のチェスの駒を手でつかんで， 前に進めるなどができる (図 3.2).

Leap Motion は指だけでカーソルを操作したり，手を使ったゲームで遊ぶことをしたりできるようにするコントローラである． 内部には赤外線 LED と， 赤外線カメラがあり，LED から照射された赤外線が手や物などで反射し， それを赤外線カメラで受信し， 手と指の動きやペンなどの形状を取得することができる．図 3.3 のように Leap Motion は赤外線カメラの画像を処理し，3D 空間上で手を動かすことができる．

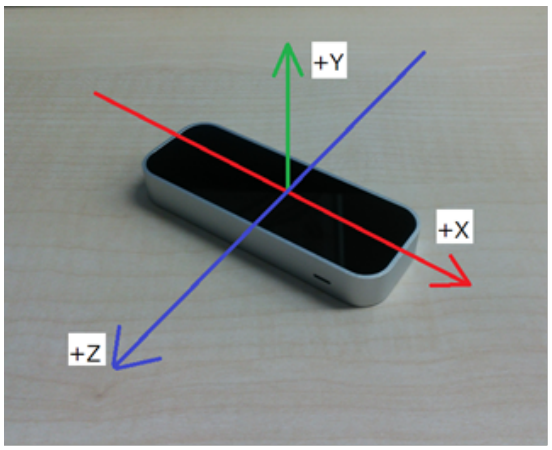

図 3.1　Leap Motion

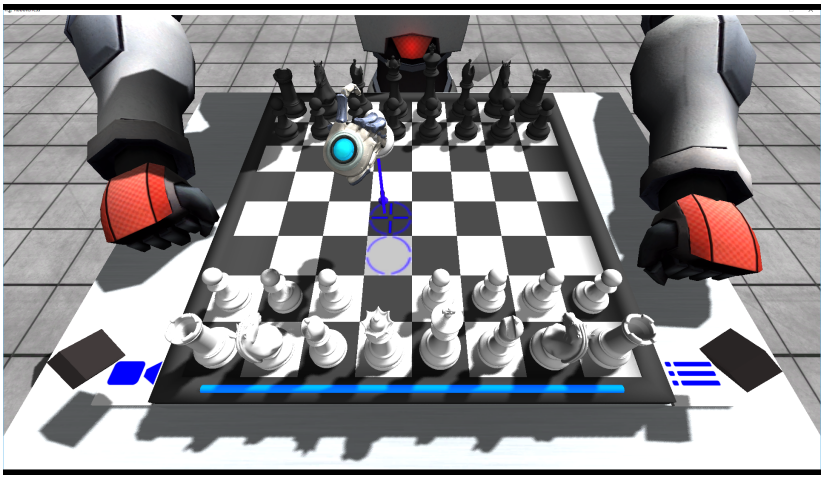

図 3.2　Leap Motion を用いたゲーム

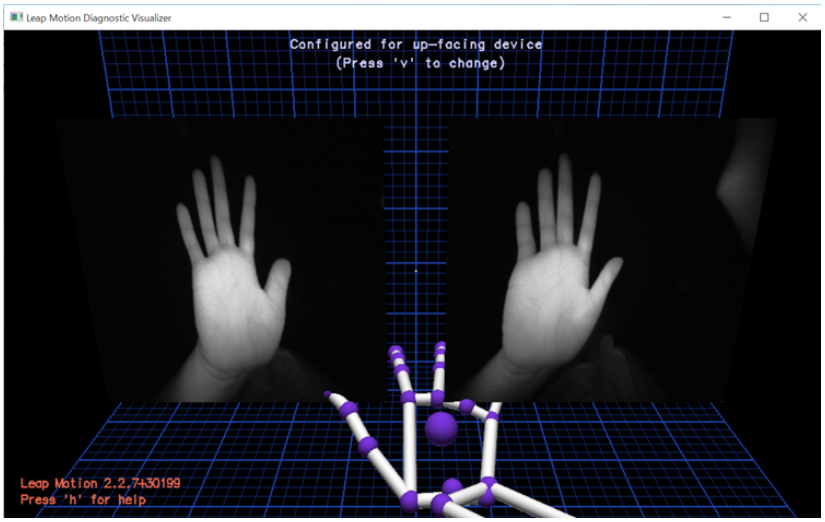

図 3.3　Leap Motion の動作

Leap Motion は 1/100[mm] 単位の精度で座標を取得できるので，小さくて素早い動きであったとしても高い精度で認識することが可能である．実際に「あ」と入力したときの 3 軸方向の移動のグラフが図 3.4，図 3.5，図 3.6 となる．

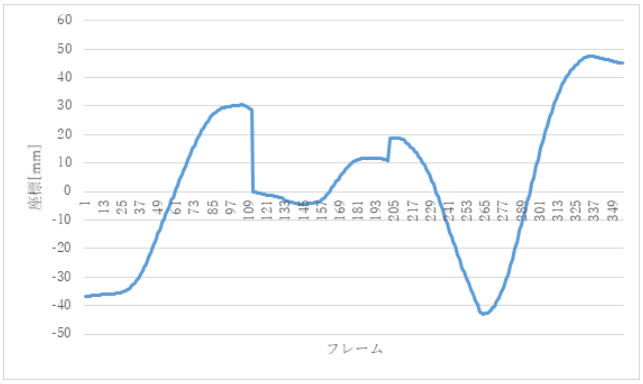

図 3.4　「あ」を書いた時の人差し指の X 軸の座標

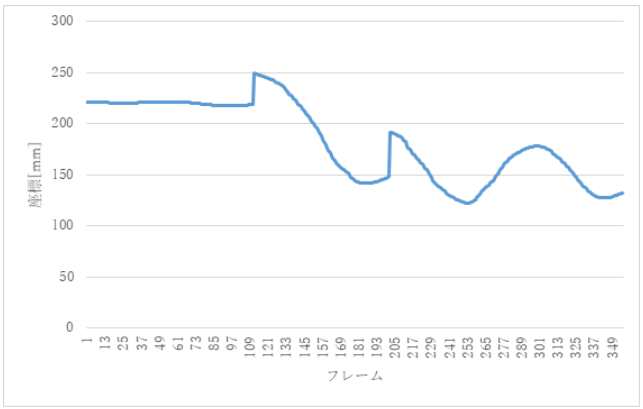

図 3.5　「あ」を書いた時の人差し指の Y 軸の座標

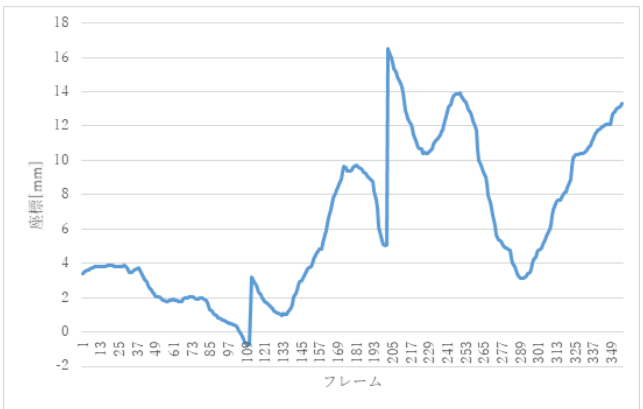

図 3.6　「あ」を書いた時の人差し指の Z 軸の座標

Leap Motion はソフトウェアのバージョンアップにより検出精度が向上している．ソフトウェア V1 と V2 では追跡可能な部位が大きく変わっており，V2 ではとても細

かいところまで追跡が可能になっている．V1 では図 3.7 のように指先の方向と，手のひらの追跡しかできないため， 指の関節の長さや指の細かな曲がり具合を検出できない．しかし V2 では図 3.8 のように関節の追跡が可能になっており，指の折り曲げと，指の種類，右手左手の検出が可能になった．これにより指の種類の判別，指の長さ，太さを測ることができるようになった．

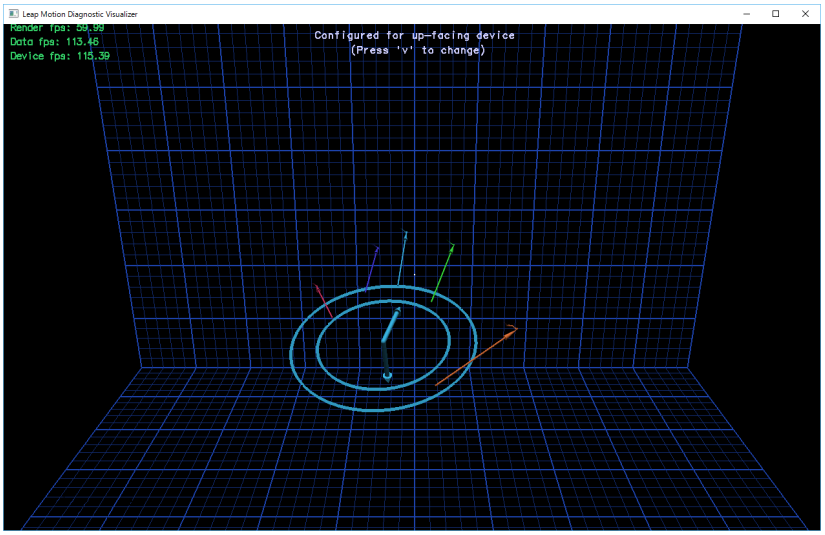

図 3.7　バージョン 1 の Leap Motion の動作

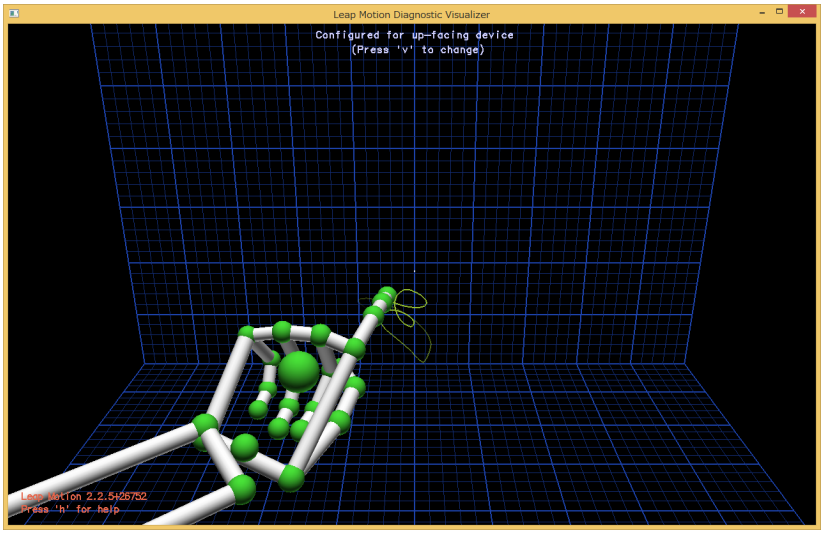

図 3.8　バージョン 2 の Leap Motion の動作

3.2　動的時間伸縮法

動的時間伸縮法 (Dynamic Time Warping:DTW) は，音声認識などで，比較する二つのデータの時間の長さが違うときにデータを伸縮させ，長さを合わせて比較をできるようにするアルゴリズムである．DTW は DP マッチングとも呼ばれる．DTW ではデータ同士の各点の距離が最短となる経路 (Warping Path) を見つけ，Warping Path より，DTW 距離を計算し，2 つのデータの類似度の比較を行う．

イメージとしては図 3.9 のようになる．一つのデータは長さが 4 しかないが，もう一つのデータは長さが 8 となっている．2 つのデータの横軸の長さが違うため，このままでは比較することができない．そこで，DTW を用いる．

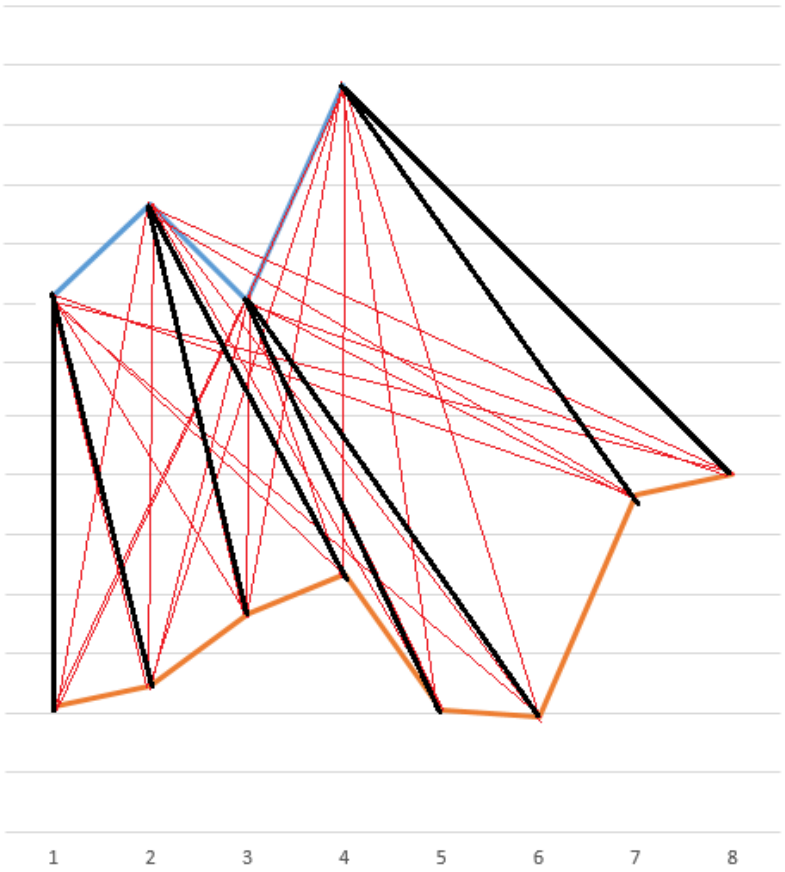

図 3.9　DTW による全ての点の比較

図 3.9 の黒い太線が各点で最も短い経路となっている．経路の長さの計算方法は，ある点とある点の数値の差を求め，最も数値が小さくなった，点と点の経路が最短ということになる．点が 4 つのデータの 1 つ目の点は，点が 8 つのデータの点 8 個全て

と経路の計算を行う．点が 4 つのデータの 2 つ目から 4 つ目までも同じようにすべての点と計算を行う．このようにして Warping Path を求めることができる．

DTW 距離は 2 つのデータがどれだけ類似していないかを数値で表すものである．そのため DTW 距離が大きい (離れている) ほど 2 つのデータは類似していないということになる．逆に DTW 距離が 0 に近いほど 2 つのデータは類似しているということになる．計算方法は Warping Path の全ての点の値を足し，Warping Path の点の数で割ることで求められる．これにより横軸の長さが一致しないデータ同士の比較を行うことができる．

2 つの時系列データ $X(X_1, X_2, X_i)$，$Y(Y_1, Y_2, Y_j)$ を n 個としたとき，2 つの時系列データの要素間の距離は式 3.1 で求める．

$$\delta(i, j) = |X_i - Y_j| \tag{3.1}$$

2 つの要素間の距離から DTW を求めるには式 3.2 を用いる．

$$DTW(X_i, Y_j) = min \sum_{k=1}^{n} \delta(i, j) \tag{3.2}$$

3.3 自己組織化マップ

自己組織化マップ (Self-Organizing Maps:SOM) は Kohonen により，提案された競合学習型ニューラルネットワークの一つである [38]．要素の数が多い高次元のデータを，2 次元などの低次元空間に写像することにより，高次元のデータの関係性を可視化することができる．

自己組織化マップ (以下 SOM) には入力層と競合層の 2 層が存在する．入力層は n 次元のベクトルデータで，競合層は m 次元空間上に，入力層のベクトルと同じ次元の重みベクトルが対応しているノードが配置されている．この重みベクトルを更新することで訓練が行われる．入力層から競合層に入力ベクトルが与えられたとき，競合層の全てのノードで，入力ベクトルと自身の重みベクトルとのユークリッド距離が最小のノードを探索する．そのノードは重みベクトルが入力ベクトルと最も類似していることを示している．このノードを勝利ノードという．勝利ノードが決まったら，近くの領域のノードの重みベクトルを更新する．これにより，勝利ノードを含む近くのノードの重みベクトルは入力ベクトルの値に近づいていく．

マップの更新は以下の式 3.3，3.4，3.5 で行う．入力ベクトルを $\vec{i}$，重みベクトルを $\vec{b}$ とする．勝利ノードの座標を $N_v = (x_v, y_v)$ とし，近傍半径内にあるノード n の座標を $N_n = (x_n, y_n)$ とする．t を訓練の回数，T を総訓練回数，δ は近傍半径の広がりとする．

$$\vec{b}(t+1) = \vec{b}(t) + H_n(t)|\vec{i}(t) - \vec{b}(t)| \tag{3.3}$$

$$H_n(t) = \alpha(t) * \exp(-\frac{|\vec{N}_n - \vec{N}_v|^2}{2\delta^2}) \tag{3.4}$$

$$\alpha(t) = 1 - \frac{t}{T} \tag{3.5}$$

この計算により，近似した重みを持つノード同士は近い位置に配置される．近い位置に配置されたノードはそれらのベクトルの性質が近似していることを表している．逆に，性質が異なるベクトル同士はマップ上に離れて配置される．これにより，高次元の要素を持つデータ間の非線形な関係性を可視化することが可能になる．

3.4　サポートベクトルマシン

サポートベクトルマシン (Support Vector Machine:SVM) は多次元のデータを図 3.10 のように 2 次元のグラフに写像し，似ているデータでグループを作り，そのグループの間に境界線を引くことで，グループを分け，未知のデータが入力された場合，どのグループに分類されるか判定することできる教師あり機械学習である．SVM はデータの次元が大きくなったとしても精度が高くできる．

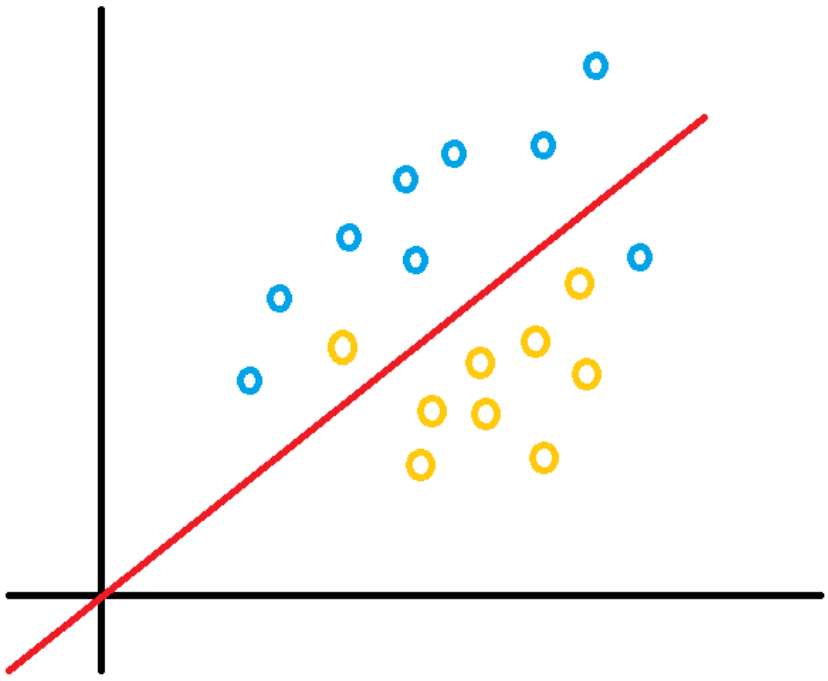

図 3.10　線形サポートベクトルマシン

図 3.10 のように直線で境界線を引くのは線形 SVM と呼ばれる．線形 SVM を式で表すと式 3.6 となる．

$$g(x) = \begin{cases} 1 & f(x) \geq 0 \\ -1 & f(x) < 0 \end{cases} \tag{3.6}$$

$$f(x) = w^T x + b \tag{3.7}$$

$f(x)$ は傾き w、切片 b の超平面 (2 次元なら図 3.10 のような境界線) となっていて，x は特徴ベクトルである．$f(x)$ の結果からどちらの境界になっているかの判断を $g(x)$ で行う，しかし，線形 SVM は直線でしか引けないため，図 3.10 の水色と黄色の 1 つの点が別のグループの中にあるように，誤分類されることがある．そこで境界線を非線形に引くことでグループの中で少し外れているデータも正しく分類できるようにするのが非線形 SVM である．非線形 SVM で最も一般的なのが RBF カーネルとなっている．RBF カーネルは図 3.11 のように境界線が引かれる．更に境界線は囲うように引くこともできるため 2 クラス分類だけでなく，多クラスで分類をすることができる．これにより，線形 SVM より誤分類を減らし，更に多クラスに対応することができる．

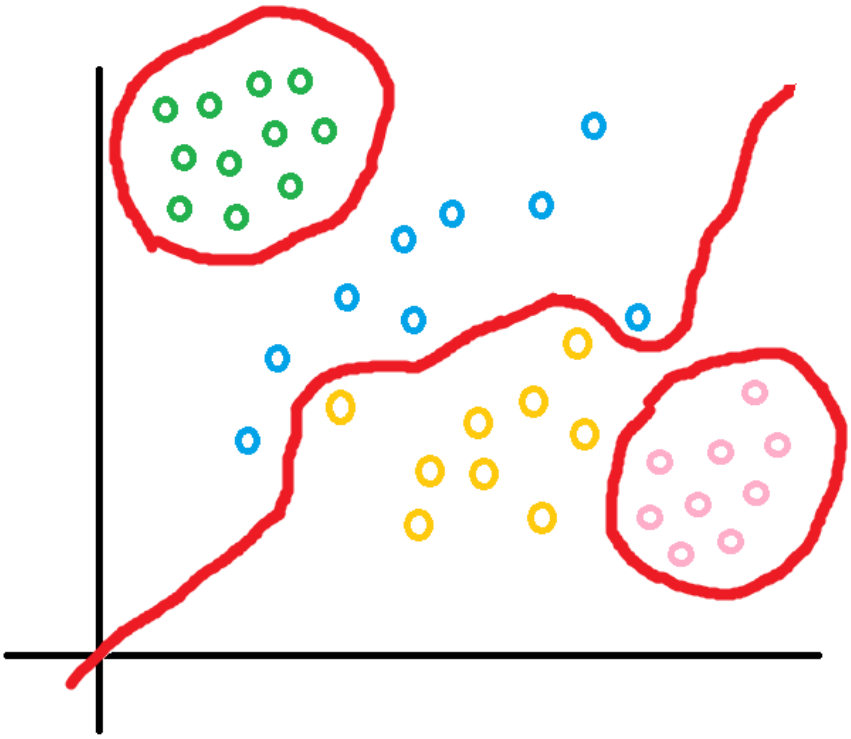

図 3.11　RBF カーネルサポートベクトルマシン

3.5 勾配ブースティングツリー

勾配ブースティングツリー (GBDT) は Gradient(勾配降下法)，Boosting(ブースティング)，Decision Tree(決定木) の 3 つを組み合わせた手法である．

勾配降下法は損失関数を最小に降下させるための手法であり，損失関数の今の誤差を算出し，そこから勾配 (偏微分) を求める．次に，求めた勾配に応じて、決められたルールに従い移動する．損失関数の値がほとんど変化しなくなった場合に終了する．勾配降下法のルールにはいくつかの種類があり，SGD，Momentum SGD，AdaGrad，Adam などがある．

ブースティングはアンサンブル学習の一つであり，アンサンブル学習とは複数のモデルを同時に訓練を行い，それらの訓練結果をほかのモデルと組み合わせ，1 つの大きな訓練モデルを作ることにより，精度の向上を向上させるものである．アンサンブル学習では複数のモデル (決定木) を用意し，モデルごとに別々で訓練を行い，未知のデータで予測を行う．モデルごとの予測結果を多数決や平均などにより 1 つにまとめ，全てのモデルの予測結果として出力する．各モデルの誤判定率を θ としたとき，全体の誤判定率を数式で表すと式 3.8 となる．m はモデルの数で k は誤判定を下したモデルの数となる．

$$P(k) = {}_mC_k\theta^k(1-\theta)^{m-k} = \frac{m!}{k!(m-k)!}\theta^k(1-\theta)^{m-k} \tag{3.8}$$

ブースティングはある決定木を訓練した後に誤判定の結果を加味したうえで，次の決定木の訓練を行う．これにより予測と実際のデータとの誤差の平均を低くすることができる．ただし，ブースティングは時間がかかることと過学習が起こる可能性がある．

3.6 ランダムフォレスト

ランダムフォレストは，2001 年に Breiman が提案した機械学習アルゴリズムである [39]．決定木を複数用いてアンサンブル学習を行うことで，過学習を起こりにくくしたという特徴を持つ．更に，決定木の作成は並列化できるため，高速で作成することができる．

ランダムフォレストは複数の決定木を作る際に，訓練に使用するサンプルは，決定木間で重複することを許したうえでランダムで選択する．このサンプルの抽出方法はブートストラップ法（別名：バギング）と呼ばれる．抽出したサンプルのデータセットから決定木を作るときに，特徴量は決定木ごとでランダムな個数を選ぶ．決定木ごとに特徴量を変えることにより，決定木間の相関が低くなることで汎化性能を高くす

ることができる．この訓練方法はアンサンブル学習の一つである．

訓練用のデータセットを $X(X_1, X_2, X_n)$ とし，木の数だけ X のバギングを行う．バギングでは分割したデータセットごとでサンプルの重複が認められるため，ある分割したデータセットに X_1 が入っているとき，別のデータセットにも X_1 が入っていることがある．バギングの総回数を B とし，今のバギングの回数は b とし，f_b は 1 つの木とし．x_b を予測するとき，ランダムフォレストの訓練は式 3.9 で表すことができる．

$$f = \frac{1}{B}\sum_{b=1}^{B} f_b(x_b) \tag{3.9}$$

最終的にすべての決定木 f_b の結果から多数決をとり，1 つの結果を出す．

3.7　ニューラルネットワーク

本項は岡谷らの「深層学習 (機械学習プロフェッショナルシリーズ)」[40] と，斎藤らの「ゼロから作る Deep Learning」[41] を参考に記述している．

ニューラルネットワークは脳の構造と挙動を人工的に再現したものである．図 3.12 の形のユニットと呼ばれるものを使い，ニューロンを再現している．

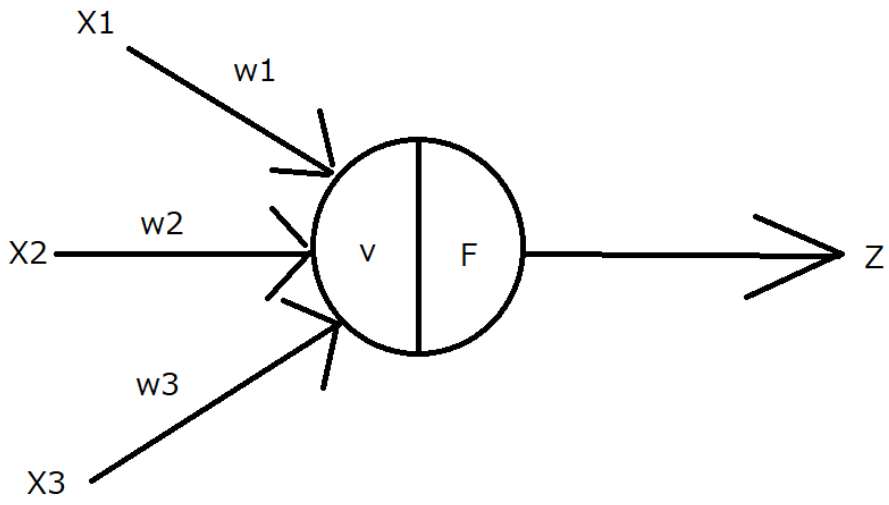

図 3.12　1 ユニット

ユニットは前の層のユニットから入力信号 x を受け取るが，信号には，各信号の重要性を表す「重み」w がかけられている．信号すべてを足し合わせ，そのユニットの

発火のしやすさ（バイアス:b）をかけることで，v となる．この v を活性化関数 F に入れると出力信号 Z になる．

図 3.12 のユニットのとき，式 3.10 で v を求めることができる．

$$v = x_1 w_1 + x_2 w_2 + x_3 w_3, b \tag{3.10}$$

このユニットを連続でつなげていったものがニューラルネットワークであり，図 3.13 のような形になる．図の中の○が各々のユニットを表している．ニューラルネットワークは入力層，中間層（隠れ層），出力層の 3 つから成り立っていて，ある層のユニットが次の層のユニットすべてにつながって信号が伝播していく．中間層は複数あることもある．

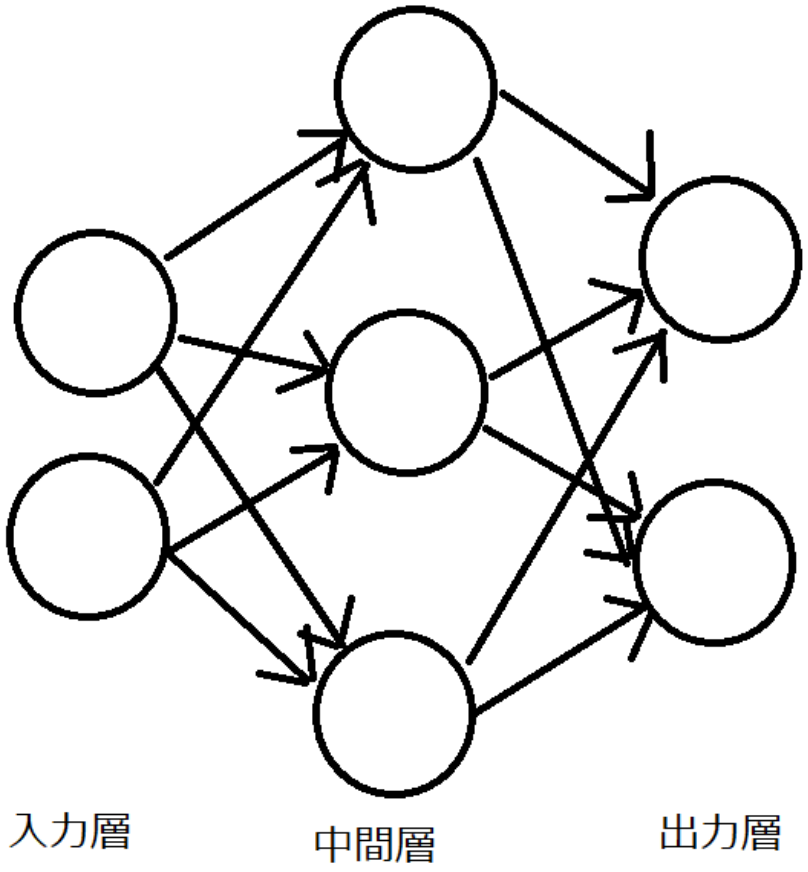

図 3.13　ニューラルネットワークの例

あるユニットが前の層から入力された信号の総和を出力信号に変換し，次のユニットに信号を送るとき，入力された信号を出力信号に変換する関数を，活性化関数 (Activation function) と呼ぶ．活性化関数では入力信号の総和をしきい値を境にして

出力が切り替わる．これによって脳のニューロンの発火を再現している．活性化関数の種類には以下に挙げるものがある．

- ステップ関数
- シグモイド関数
- ReLU 関数

この中でシグモイド関数がよくニューラルネットワークで使用されていたが，最近では ReLU 関数が使われることが多くなっている．

ステップ関数は式 3.11 で表すことができる．v が入力信号の総和である．

$$f(x) = \begin{cases} 0 & (v < 0) \\ 1 & (v \geq 0) \end{cases} \tag{3.11}$$

シグモイド関数は式 3.12 で表すことができる．

$$f(x) = \frac{1}{1 + \exp(-v)} \tag{3.12}$$

ReLU 関数は式 3.13 で表すことができる．

$$f(x) = max(0, v) \tag{3.13}$$

入力層から出力層の入力まではユニットに入力される信号に重みをつけて計算したあとに，活性化関数に入力し，その結果の信号を出力するが，出力層では分類問題であればその結果を人の目でわかる形で出力するために，ソフトマックス関数を使用する．図 3.14 を用いて説明する．図 3.14 は，入力画像が 1 4 のどれであるかの分類を行う．ニューラルネットワークである．4 クラス分類のため，ユニット数は 4 となる．ソフトマックス関数の出力は，ユニットごとで 0 から 1.0 の実数をとり，すべての出力の総和は 1 となる．この出力があるクラスである確立である．1 である確率が 0.25(25%)，2 である確率が 0.2(20%)，3 である確率が 0.15(15%)，4 である確率が 0.4(40%) というように出力され，分類を行うことができる．

ソフトマックスは式 3.14 で求めることができる．y_i が出力で，x が入力値とする．

$$softmax(y_i) = \frac{\exp(x_j)}{\sum_{k=0}^{N} \exp(x_k)} \tag{3.14}$$

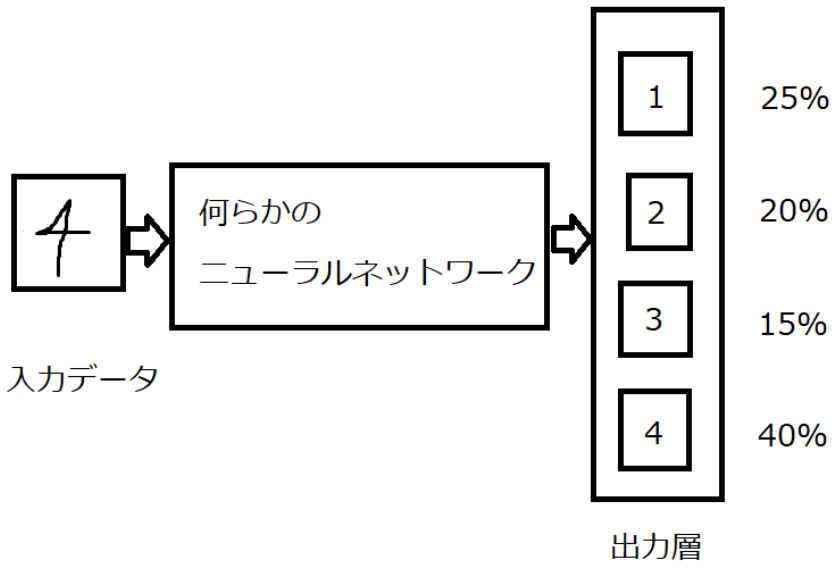

図 3.14　ソフトマックスの例

3.8　畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) は，前述したすべてのユニットが繋がった全結合ニューラルネットワークとは違い，畳み込み層とプーリング層を使用する．この 2 つの層を使用することで，入力されたサンプルから効率的に特徴を抽出することができ，訓練にかかる時間を短くしたり，分類の精度を上げたり，過学習を防ぐことができるのが期待できる．

畳み込み層ニューラルネットワーク (以下 CNN) の概略図を図 3.15 に示す．この図の矢印の方向で入力側 (Input) から出力側 (Output) に向けて，畳込み層 (convolution layer) と，プーリング層 (pooling layer) がペアで並び，畳み込みを行っている．このペアが複数回繰り返させることで深層学習を実現できる．畳み込み層とプーリングは基本的にはペアであるが，場合によっては畳込み層を複数回繰り返された後にプーリング層を 1 層つなげる場合もある．また，畳込み層とプーリング層の後に，画像処理の場合は局所コントラスト正規化 (local contrast normalization： LCN) 層を挿入することもある．畳込み層と，プーリング層の繰り返しの後には，全結合層 (Full connected layer) が配置される．以下に各層での動作を記述する．

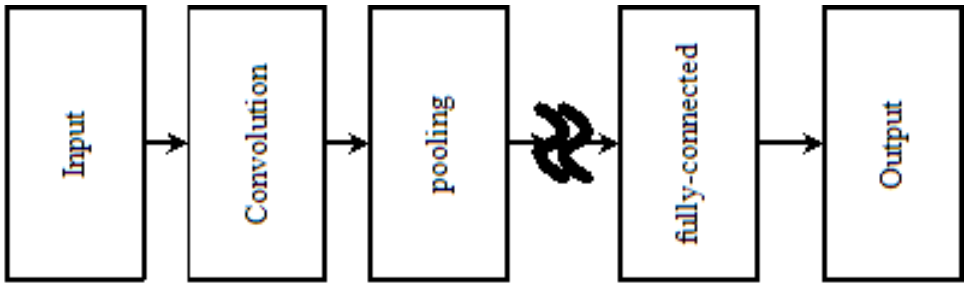

図 3.15　CNN の概略図

3.8.1　畳み込み層

畳み込み層では，画像のなどは濃淡のパターン検出，数値データの場合は特徴抽出をする．エッジ抽出などの特徴抽出を行う．畳み込みの動作を数式で表すと式 3.15 になる．

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq} \tag{3.15}$$

白黒画像を例として数式を説明する．画像サイズを $W * W$ とし，また，画像の 1 ピクセル (画素) を $(i,j)(i = 0, ...,W\text{-}1)$，$(j = 0, ...,W\text{-}1)$ とする．画素 (i,j) を x_{ij} として，画素は負の値を含む実数値をとる．次に，フィルタと呼ばれる小さい画像について考え，そのサイズを $H * H$ とする．フィルタの画素は $(p,q)(p = 0, ...,H\text{-}1)$，$(q = 0, ...,H\text{-}1)$ とし，画素 h_{pq} は任意の実数値をとる．これらを数式に当てはめ計算を行うことで指定した大きさに画像を畳み込むことができる．

3.8.2　プーリング層

プーリング層は畳込み層で得られた特徴の位置による感度を若干低下させることにより，対象とする特徴量の位置が画像内などで若干変化したとしても，出力を変化させない役割がある．これにより，位置に関係なく局所的な特徴を得ることができる．

ここでは，サイズ $W * W * K$ の入力画像上で画素 (i,j) を中心とする $H * H$ 正方領域をとり，この中に含まれている画素の集合を P_{ij} で表す．ここで，この P_{ij} 内の画素について，チャネル (k) ごとに，独立で，H^2 個ある画素を使って 1 つの画素 H_{ijk} を求める．画素を求める中で，H^2 個の画素の最大値を選ぶものを最大プーリング (max pooling) といい，式 3.16 で求められる．

$$u_{ijk} = \max_{(p,q) \in P_{ij}} z_{pqk} \tag{3.16}$$

H^2 個の画素の平均値を求めたものを平均プーリング (average pooling) といい，式 3.17 で求めることができる．

$$u_{ijk} = \frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk} \tag{3.17}$$

以下に最大プーリングの式を示す．

3.8.3 全結合層

畳み込み層とプーリング層を通して特徴部分が取り出されたデータを一つのユニットに結合し，活性化関数によって変換された値を出力する．ユニットの数が増えると特徴量空間の分割数が増し，各領域を特徴付ける特徴変数の数が増える．全結合層は前項で述べた，ニューラルネットワークと同じものである．

3.8.4 損失関数

ニューラルネットワークの訓練において，重みのパラメータを探索する必要があるが，探索を行うときに指標として用いられるのが損失関数 (loss function) である．損失関数には任意の関数を使用可能だが，一般には 2 乗和誤差か，交差エントロピー誤差が使用される．

3.8.5 ミニバッチ

損失関数の計算は，すべてのデータの 1 つごとの損失関数の和である．データが 100 個ならば 100 個分の損失関数を求め，そこから総和を求めると訓練データの損失関数が出力できるが，ニューラルネットワークはビッグデータの解析にも使われるため，時にはデータの数が途方もないぐらい多くなることもある．そこで，すべてのデータの損失関数を求めていると時間がかかるため，一部のデータを取り出し，それらの損失関数の総和を求め，その結果を全体の損失関数を求めた近似とするようにする．これにより高速で損失関数を求めることができる．これをミニバッチ学習と呼ぶ．

3.8.6 最適化

ニューラルネットワークにおいて，損失関数の値をできるだけ小さくできるパラメータを見つかることが，訓練の目的である．しかし，パラメータは非常に複雑なため，最適なパラメータは簡単には見つけることはできない．更に，深層学習のようにネットワークが深くなるとパラメータの数は膨大になり，更にパラメータを探すこと

は難しくなる．最適なパラメータを探す単純な方法として，確率的勾配降下法 (SGD) と呼ばれる最適化の方法はあり，単純で実装も簡単ではあるが，場合によっては非効率なこともある．そこで，場合に合わせて最適化手法を変える必要がある．一般的な最適化手法は以下のものがある．

- SGD
- Momentum
- AdaGrad
- Adam

3.8.7 過学習

ニューラルネットワークにおいて，訓練データの損失関数の値を小さくすることは重要ではあるが，訓練の本当の目的はこれから与えられる未知のデータに対して正しい推定を行えることである．

訓練における損失関数の値はパラメータ更新のたびに減少するが，テストの損失関数の値はあるタイミングで増え始めることがある．これは，訓練データにパラメータを合わせすぎたあまりにテストデータの推定がうまくできなくなっている状態，過学習となる．

過学習が起きる原因として主に 2 つの理由が挙げられる．

- パラメータを大量に持つネットワーク
- 訓練データが少ない

過学習はニューラルネットワークの精度に大きくかかわるため，これ避けるために訓練を早めのうちに切り上げるか，以下のような対策を講じることができる．

3.8.7.1 Weight decay

過学習抑制のために昔からよく用いられる手法である．これは訓練の過程で大きな重みをもつことに対してペナルティを課すことで過学習を抑制するものである．

過学習自体が重みが大きくなることで発生することが多くあるため，大きな重みにペナルティを課すことは効果的である．

3.8.7.2 Dropout

ニューラルネットワークが大きく，複雑な場合，Weight decay では対応ができないため，Dropout を用いる．

Dropout はユニットをランダムで消去する方法である．ユニットを消すことで信号の伝播が行われなくなる．これにより過学習を防ぐことができる．

3.8.7.3 Batch Normalization

Batch Normalization により，訓練を早くすることができ，初期値に依存することがなく，過学習を抑制することができる．さらに Dropout をなくすこともできる．

3.9 機械学習の評価方法

機械学習により作られた分類器の予測結果と真の結果に基づいて，適合率 (Precision) と再現率 (Recall) から F 値 (F-score) を算出し，分類器の精度を評価する．ここでは例として正と負の 2 値分類と考え，表 3.1 のように TP (True Positive)，FP (False Positive)，FN (False Negative)，TN (True Negative) に分類する．TP は予測結果が正であり，なおかつ実際の結果も正であった場合になる．この分類から適合率と再現率を求め，F 値を算出する．

表 3.1　予測結果と実際の結果

		実際の結果	
		正	負
予測結果	正	TP	FP
	負	FN	TN

適合率は正と予測したデータのうち実際に正であるものの割合を示す．式 (3.18) を用いて算出する．

再現率は実際に正であるもののうち正であると予測されたものの割合を示す．式 (3.19) を用いて算出できる．

F 値は適合率と再現率の調和平均となっている． F 値は 1(100%) が最大で，この値が高いほど正しく分類ができているといえる．式 (3.20) を用いて算出する．

$$Precision = \frac{TP}{TP + FP} \tag{3.18}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.19}$$

$$F - score = \frac{2Recall * Precision}{Recall + Precision} \tag{3.20}$$

第 4 章

Leap Motion による空中筆記に関する研究

4.1 Leap Motion による空中筆記に関する研究の目的

この研究の目的は，クレジットカード使用時の本人確認で使用することである．既存のクレジットカード使用時の本人確認方法は PIN の入力か，ペンで署名を行い，カード裏に書いてある署名と比較することである．PIN を入力する方法では覗き見に対して脆弱という問題があり，署名の筆跡鑑定の精度は署名を確認する人の能力に依存してしまうという問題がある．更に，署名そのものはカードに書いてあるため，真似をするだけであれば誰にでも真似をすることができてしまうのが問題点である．それだけでなく，筆記を練習することで，更に見分けが付かない署名を書くこともできてしまう．そこで，この研究では覗き見に対して強く，筆跡が残らない方法として，空中に筆記を行う方法を提案した [42].

自分の署名を空中に指で書いた時の速度と動き，指の向きから個人の検証をできるようにすることで，筆跡が残らないため，覗き見をすることを難しくすることを目的とした．仮に筆記情報が流出してしまった場合でも，登録する文字を変更することで入力時の軌跡が変わるため，覗き見した人が，筆記時の癖を真似できるようになったとしても，すでに入力する軌跡は変更されているため，再び覗き見して新しい筆記を練習しなければならなくなるため，悪用するのは困難になる．これにより，PIN の覗き見と，署名のコピーの問題を改善する．クレジットカード使用時の本人確認に本研究を使用することにより，カード使用時の安全性を高めることができると考えた.

4.2 Leap Motion による空中筆記に関する研究の手法

この研究では Leap Motion を使い空中に指で署名を書き，書いた署名と登録してある署名がどれぐらい似ているかで個人の検証を行う．Leap Motion を用いて空中に文字を描くため，筆跡が残らず，紙に書かれた文字のように，その文字をなぞって複写するような入力は行えない．そのため，攻撃者は何度も覗き見をするか，録画した動画で練習し，入力を行う必要がある．要件 2 において，覗き見耐性を「ある程度」としたのは，繰り返し録画されたものを見て模倣された場合には，破られる可能性が十分にあることを考慮したためである．手法の詳細は以下の通りとなる．

4.2.1 Leap Motion で取得するデータ

Leap Motion では人差し指の X，Y，Z 軸の座標と加速度の 6 つの特徴を記録する．これは登録時と検証時どちらも同じとする．

4.2.2 登録フェーズ

登録時は好きな署名を書いてもらい，筆記時の動作を記録し，そのデータを保存する．登録時は何かしらの処理を行うこともなくただデータを保存する．

4.2.3 検証フェーズ

4.2.3.1 署名の入力

検証時は登録した署名と同じものを書く．

4.2.3.2 検証

登録したデータと検証を行うデータの比較には DTW を用いる．DTW を用いることで，長さの違う 2 つの時系列データがどれぐらい似ているかを比較することができる．6 つの特徴の 1 つずつで DTW 距離の計算を行う．DTW による比較の結果，6 つの特徴の DTW 距離が最も小さかった登録者が，検証を行った人と同じであれば検証に成功したということになる．最も小さいのが別の人だった場合は本人の検証は失敗となる．

4.3 Leap Motion による空中筆記に関する研究の評価

5 人の被験者 A，B，C，D，E に 1 つずつ署名を登録してもらい，自分で書いた署名と同じ文字の署名と，他人が書いた署名と同じ文字の署名を 4 つの合計 5 つの署名を 10 回ずつ検証してもらい，各署名の回数ごとの X，Y，Z 軸，X，Y，Z 加速度の DTW 距離を求め，それらを足し合わせた値を出力させた．1 回の検証の中で DTW 距離が最も短い登録者が，選ばれた人となる．各被験者の結果を表 4.1-表 4.5 に示す．表 4.1 は被験者 A が自分の署名 (A) と，他人の署名 (B〜E) を書いた時の回数ごとの DTW 距離である．署名 A とは，文字の「A」ではなく，被験者 A の名前である．表 4.2 以降は被験者 A の結果と同様に被験者 B 以降の結果である．

表 4.1　被験者 A の検証結果

署名 / 回数	A	B	C	D	E	最小
1	1.07	0.29	1.83	3.36	2.45	B
2	1.09	0.47	3.05	3.43	2.44	B
3	6.79	0.74	4.51	4.27	0.86	B
4	0.44	0.62	0.66	3.49	2.03	A
5	0.86	0.35	1.59	3.93	2.27	B
6	0.41	0.78	1.42	3.71	2.94	A
7	0.74	0.56	0.98	3.33	1.82	B
8	3.23	68.67	1.09	3.61	1.93	C
9	0.49	5.72	1.17	2.92	17.11	A
10	0.52	51.76	1.2	3.64	3.51	A
分類数	4	5	1	0	0	

表 4.2　被験者 B の検証結果

署名 / 回数	A	B	C	D	E	最小
1	0.9	0.19	0.83	5.63	1.09	B
2	0.81	0.2	0.79	4.41	0.92	B
3	0.65	0.33	1.3	5.16	1.31	B
4	0.96	0.25	0.53	7.54	35.42	B
5	0.6	0.5	0.46	3.66	1.16	B
6	0.85	0.57	0.79	3.76	1.16	B
7	14.33	0.42	0.46	4.32	1.16	B
8	0.53	0.37	0.51	3.89	0.99	B
9	0.58	0.45	0.75	3.63	1.05	B
10	0.39	0.18	0.67	3.68	1.13	B
分類数	0	10	0	0	0	

表 4.3　被験者 C の検証結果

署名 / 回数	A	B	C	D	E	最小
1	1.04	0.35	0.2	3.93	0.36	C
2	0.24	0.52	0.28	3.86	0.33	A
3	2.43	0.29	0.18	3.94	0.42	C
4	0.22	0.34	0.16	3.56	0.36	C
5	0.24	15.88	2.29	3.18	0.67	A
6	0.21	0.29	0.15	3.75	0.31	C
7	0.2	0.33	0.17	4.68	0.45	C
8	0.26	0.27	0.32	3.56	0.35	A
9	0.2	0.54	0.31	3.41	0.34	A
10	0.17	0.29	0.23	3.43	0.58	A
分類数	5	0	5	0	0	

表 4.4　被験者 D の検証結果

署名 / 回数	A	B	C	D	E	最小
1	7.76	0.52	2.49	3.25	10.13	B
2	5.72	1.02	2.9	2.9	3.07	B
3	2.32	1.29	2.21	3.62	2.76	B
4	3.4	1.24	2.4	3.28	2.24	B
5	1.88	1.3	2.52	3.54	2.74	B
6	3.09	3.32	2.53	4.04	3.35	C
7	1.73	1.94	2.96	3.34	2.63	A
8	1.96	1.6	2.45	3.41	2.87	B
9	2.19	20.07	3.39	3.57	2.33	A
10	3.55	1.45	3.5	3.1	2.14	B
分類数	2	7	1	0	0	

表 4.5　被験者 E の検証結果

署名 / 回数	A	B	C	D	E	最小
1	1.08	0.55	1.53	5.75	1.06	B
2	0.74	0.77	1.49	4.37	0.66	E
3	1.1	1.11	1.49	5.75	0.67	E
4	0.47	1.18	1.43	4.92	0.44	E
5	1.63	1.18	1.15	5.64	3.16	C
6	0.88	1.27	0.98	5.19	0.38	E
7	0.82	2.25	1.55	5.62	0.3	E
8	1.21	1.25	3.76	5.03	0.27	E
9	1.11	1.51	1.45	5.74	0.3	E
10	1.62	2.05	1.65	5.83	0.47	E
分類数	0	1	1	0	8	

表 4.6　全被験者の FAR と FRR

被験者	FRR	FAR
A	0.6	0.175
B	0	0.325
C	0.5	0.075
D	1	0
E	0.2	0
平均	0.46	0.115

表 4.1 より A が自分の署名を検証した結果，10 回のうち 4 回検証に成功した．このため FRR は 60% となった．表 4.2-表 4.5 のうち A が一番低かった数から，FAR を求めると 17.5% となった．これと同じように表 4.2-表 4.5 の FRR と FAR も求めた結果を表 4.6 に示す．表 4.6 より全被験者での平均 FRR は 46%，FAR は 11.5% となった．

4.4　Leap Motion による空中筆記に関する研究の考察

表 4.6 より，FRR が 0% の人がいれば，100% になる人もいて，平均 46% ということは 2 回に 1 回は失敗するということになる．1 回の筆記動作は自分の署名を書くだけであるため，長くはならないが，DTW で比較する時間もあるため失敗するとストレスになると考えられる．

FAR に関しては全体では 11.5% で，人によっては 0% の人もいたが，最悪のケースだと被験者 B の 32.5% がある．これは B 以外の誰かが B の署名を書くと，3 回に 1 回は被験者 B に分類されてしまうということになる．表 4.4 より，被験者 D が書いた署名は半分以上が被験者 B として分類されている．逆に被験者 D の自分の署名が自分として分類された数は 0 であるため，D の署名を書く時の特徴は B に似ていて，D は登録時と検証時で違う動きをしていたのではないかと考えられる．

Leap Motion 側の問題として，プログラムを動かし続けていると Leap Motion が熱を持ち始めるので，熱によって手の動きを読み取るための赤外線カメラに悪影響を与えていた可能性も考えられる．

更に，プログラムを起動してから最初のうちは正しく入力できるが，約 10 分起動させていると座標と加速度が正しく表示されなくなったり，画面に指の示している位置を表示するポインタの動きが悪くなったりしたので，プログラム自体に問題があった可能性があると考えられる．

今回は登録した署名は人によって違う文字だったため，ある程度は正しく分類する

ことはできていたが，仮に同じ文字を登録した場合は，DTW による比較では登録してある同じ文字の署名 2 つと比較したときに，どちらも DTW 距離が同じぐらいに低くなり，分類の精度が悪くなると考えられる．そこで，同じものを書いたとしても正しく本人確認できるようにする必要がある．

この研究の平均 FRR より，1 回で約 46% の確立で失敗するということは，2 回連続で失敗する確率は約 21%，3 回連続で失敗する確率は約 9.7% となる．4 回連続は約 4.4% となる．5 回連続は約 2.1% と回数を重ねるほど失敗する確率は減っていくが，この研究の方法でクレジットカードの本人確認をすることを想定した場合，そこで，本人確認を失敗できる回数を制限しなかった場合，本人が何度も本人確認をやり直していると，覗き見をしている人が署名を記憶できてしまう機会が増えてしまう問題がある．更に攻撃者が何度も本人確認ができてしまうと，いつかは突破できてしまう問題がある．そこで本人確認には連続で失敗できる回数を設定する必要がある．署名をコピーされる可能性と，本人確認を行う人のストレスから考えて，回数は 5 回を限度とするのが良いと考えられる．5 回目で成功する確率は 97.9% となるので，登録者が 10 人いたとした場合，5 回目であればほぼ全員が本人確認に成功できると考えられる．しかし，FAR の最悪のケースが 32.5% であり，3 回に 1 回は突破できてしまう問題と，何度も覗き見見られることでコピーされることを考えると，理想としては連続で失敗できるのは 2 回か多くても 3 回までとするのが望ましい．本人確認に成功する確率を上げる (FRR を下げる) ことで，連続で失敗できる回数は減らすことができるため，この研究では FRR と FAR を十分に低くできなかったといえる．

FRR と FAR が悪かった理由として，人差し指の情報しか取得しなかったことが原因だと考えられる．全ての指の情報を取得することによって，特徴をさらに増やすことができ，個人の差を大きくすることで，FRR と FAR の精度を改善することができると考えられる．更に，個人の差を大きくすることで同じものを書いたとしても正しく本人確認をすることができると考えられる．ただ，DTW は 2 つの時系列データの 1 つの特徴の全てのデータの DTW 距離を総当たりで計算するため，特徴を増やした場合には DTW では検証にさらに時間がかかってしまうことになる．そのため，別の検証方法を用意する必要があると考えられる．

第 5 章

一方向の移動を機械学習で検証する研究

5.1　一方向の移動を機械学習で検証する研究の目的

この研究の目的は 1 つ目の研究の改善を行い，この手法をドアロックへ利用することである。前の章では，Leap Motion を用いて署名を空中に指で筆記しているときの入力時の軌跡，速度から本人確認を行った．しかし，2 回に 1 回は失敗したり，最悪の場合 3 回に 1 回は攻撃者が突破できてしまうと，精度があまり高くなかった．更に，DTW の性質上同じ文字の署名を複数人が登録した場合に，別の人の署名であったとしても，似ていると判定され，精度が悪くなる可能性がある問題も考えられた．そこで，この研究では，取得する特徴量を増やしたうえで，署名を筆記をさせるのではなく，単純に一本の線を入力させることで同じものを書いても正しく分類できるようにし，機械学習を用いることで，使いやすさと精度の向上を目指した [43].

この研究は一方向の入力と単純なもののため，ドアのロック解除で使うことを想定している．そのため，利用者は 10 人程度を最大として想定している．既存のドアロックには最も一般的な物理的な鍵や，指紋，PIN，IC カード，スマートフォンによる遠隔操作が使われている．これらのうち鍵，カード，スマートフォンは盗まれた場合に容易に侵入されてしまう問題と，再発行を即時で行うことができないという問題がある．指紋と PIN はデバイスに残った指紋をコピーされたり，入力した番号を特定される危険性がある．特に PIN は覗き見攻撃者の問題点もある．この研究では既存のドアロックと比べて，デバイス再発行の必要性をなくすことと，覗き見に対する耐性を強化している．更に機械学習を使うことで精度の向上を図った.

前の研究では人差し指の軌跡しか認識させていなかったため，個人の特徴の差を大きくすることができなかった．そこで，この研究では，Leap Motion への入力を，空

中で筆記する動作から，手を単一方向に動かすという簡単な動作にした．その代わりに，入力時に手を広げさせ，すべての指を認識させるようにすることで，手から取得するデータの数を増やした．これにより，個人の差を大きくできると考えた．更に，識別の精度を向上させるために，自己組織マップ (Self-Organizing Maps:SOM) を用いて，しきい値を設定するという方法を提案した．この研究ではドアロックを解除するだけのため，検証を行った人が登録者のうちの誰であるかを判定することより，登録者のうちに入っているかの方が重要であるため，教師なし訓練である自己組織マップを用いた．Nohara らの研究より，SOM は個人識別には有効な手法であると述べられており，Nohara らの研究はフリック入力の話であり，本研究はフリック入力ではないが，同じような一直線の動きであったため，SOM を採用した．これらの改善により，前の研究と比べて Leap Motion での入力は単純にできるうえ，精度を高くできると考えた．更に，5 本の指すべての動きを真似しなくてはならないため，前の研究よりもコピーをするのは難しくできると考えた．

5.2　一方向の移動を機械学習で検証する研究の手法

この研究では前の研究と同じように，Leap Motion を用いて手を動かしたときの速度，指の向きから個人を識別する．

この一方向を入力するために図 5.1 の装置を作った．これは Leap Motion から左右に 10cm ずつ離れた位置に糸を垂らして 20cm の空間を作っている．左右の糸を始点と終点とし，逆の糸まで手を移動させる．これにより全ての人は同じ距離を動かすことになる．同じ距離を移動していても，人によって指の向き，移動速度が異なるためそれを特徴とし，分類のための材料とする．

図 5.1　一方向入力取得用の環境

5.2.1　入力する方向

分類のために単一方向に手を動かすことになるが，入力する方向は 1 つだけでなくいろいろな方向で実験を行う．向きとしては以下の 6 つとする．

- 右から左
- 左から右
- 右下から左上
- 左上から右下
- 右上から左下
- 左下から右上

単なる上下がないのは装置の都合上，上と下に糸が張れなかったため，上下の移動距離が人によってバラバラになってしまうからである．

5.2.2 取得する特徴

この研究では Leap Motion にて以下の特徴を読み取り，個人の特徴量として定義する．

- 指の X 方向の移動距離 [cm]
- 指の Y 方向の移動距離 [cm]
- 指の Z 方向の移動距離 [cm]
- 単位時間当たりの指の X 方向への移動距離 [cm]
- 単位時間当たりの指の Y 方向への移動距離 [cm]
- 単位時間当たりの指の Z 方向への移動距離 [cm]
- 指の先端の X 方向の向き [rad/s]
- 指の先端の Y 方向の向き [rad/s]
- 指の先端の Z 方向の向き [rad/s]
- 指の付け根の手の X 方向の向き [rad/s]
- 指の付け根の手の Y 方向の向き [rad/s]
- 指の付け根の手の Z 方向の向き [rad/s]
- 指の先端の手の X 方向の速度 [mm/s]
- 指の先端の手の Y 方向の速度 [mm/s]
- 指の先端の手の Z 方向の速度 [mm/s]
- 指の付け根の手の X 方向の速度 [mm/s]
- 指の付け根の手の Y 方向の速度 [mm/s]
- 指の付け根の手の Z 方向の速度 [mm/s]

これを各指と手に関して取得する．手の場合は先端ではなく手のひらの中心となり，付け根は腕と手の付け根となる．更に入力していた時間 [msec] も特徴とするため，全部で 109 個の特徴を取得する．

5.2.3 入力

入力時に Leap Motion の上に手を大きく広げ，右か左の糸まで手を移動させる．そうしたら画面に反対側の糸まで手を移動させるように指示が出る．指示に従って反対側の糸まで手を移動させたら一度手を Leap Motion の探知外までどかす．そうしたら再び Leap Motion の上に手をかざすように指示が出るので，入力の最初のように Leap Motion の上に手をかざす．その後画面に指示された回数繰り返す．それが終了したら今度は反対側から開始するように指示が出るのでそれも繰り返す．それも終了

したら別の位置から開始するように指示が出るので指示された通り入力する．それを繰り返すことでいろいろな方向のデータを取得する．

5.2.4　検証

SOM を用いて検証を行う．SOM のパラメータは Nohara らの研究と同一パラメータのトーラス型 SOM を使用する．表 5.1 に SOM のパラメータを示す．

表 5.1　SOM のパラメータ

ノード数	80 × 80 (6400)
学習係数	0.05
初期近傍半径	0.05
総訓練回数	20000

各方向に各個人のデータを入力時間順で並び替え，等間隔でデータを 10 個ずつ取り出す．それを全員分行い，1 つのデータとしてまとめて，正規化を行う．正規化をしたらそれをトレーニング用データとし，SOM に入れ，訓練を行う．訓練が終了したら入力層から 10 個の本人の入力ベクトルを抜き出し，その勝利ノードを求める．その勝利ノードを中心とした円を定め，この円を閾値として定義し，円の内側を SOM での本人領域と定義し，内側にあるデータが本人として分類された数となる．

5.3　一方向の移動を機械学習で検証する研究の評価

SOM の本人領域の中に本人のデータがいくつ入っているかで False Rejection Rate(FRR) を求め，本人領域の中にいくつ他人のデータが入っているかで False Acceptance Rate(FAR) を求める．更に，本人領域の半径を変化させながら，半径ごとに FRR と FAR がいくつかを求めた．その結果を表 5.2 に示す．

表 5.2　本人領域の半径と FRR と FAR

半径	FRR[%]	FAR[%]
3	98	0
4	92	0.29
5	54	1.71
6	34	1.71
7	20	4.29
8	20	6.57
9	14	10.29
10	14	12.86
11	14	19.71
12	12	30.57
13	10	37.71
14	10	49.14

5.4　一方向の移動を機械学習で検証する研究の考察

表 5.2 より，半径 6 から 9 の時が FRR，FAR ともに前の研究よりもよくなっている．FRR と FAR のバランスがとれているのが半径 9 だったため 9 で考察を行う．半径 9 の時の FRR が 14% と FAR が 10.3% だったため，約 10 回に 1 回は失敗または突破されることになるが，FRR に関しては前の研究では FRR は 0.46 だったのに比べると約 30% ほど改善している．FAR に関しては前の研究では FAR は 0.115 だったため約 1% 改善している．更に，この研究のシステムは SOM 半径を増減させることで FRR と FAR を調整することができる．

前の研究では 6 個しかとらなかった特徴量を，この研究では 109 個にしたが，逆に特徴量が多すぎるせいで特徴がつかめなくなって FRR が上がった可能性がある．そのため，特徴量の数についてはもう少し考える必要がある．特に，109 個の中で同じようなデータをとっている可能性もあるため，Leap MotionAPI リファレンスを確認し，Leap Motion がどのようなデータをとれるのかよく確認する必要がある．

この研究では一方向入力という単純な入力だけで，前の研究よりも FAR と FRR を低くすることができた．しかし，一方向しか入力しないのでは覗き見をしなくても，何回も試していれば突破できてしまうのではないかと考えられる．この研究の手法をドアロックとして使用したとき，前の研究と同様に連続で失敗できる回数を設定する必要がある．今回の研究は FRR は半径 9 としたときに 14% となっているため，1 回

目は 14% の確立で失敗し，2 回目では約 2% の確率で失敗する．3 回目となると約 0.3% となり，ほぼ全員が検証に成功するため，2 回までなら失敗してもよいと考えられる．この回数であれば攻撃を行ったとしても，突破される確率が上がることはなくなると考えられる．

このシステムでは SOM を用いて 10 人程度での使用を想定していたが，仮に登録者数がこれより多くなった場合には人数に対してマップが狭くなり，個人の範囲が多くなることで登録していない人が範囲外になることが少なくなり，FAR が上がってしまう可能性がある．そのため，多人数での使用はできないと考えられる．

1 つ目の研究と同じように空中に筆記を行うが，この研究ではすべての指の情報を取得することで特徴量を増やし，入力するものは署名ではなく一方向の線にすることで，覗き見に対する耐性を持たせたうえで，更に機械学習を使用することで精度の向上を図った．しかし，動作が単純であるため，1 つ目の研究に比べて覗き見に対して弱くなってしまっている．ただ，すべての指の動作を真似しなければならないため，一概には覗き見耐性が下がっているわけではない．更に機械学習は大人数になった場合には使用できない問題がある．そこで，単一方向の 1 つのデータだけでなく，1 回の入力で複数の方向を組み合わせて入力させるのが良いのではないかと考えられる．更に使用する機械学習を変更することで多人数での使用ができると考えられる．

第 6 章

提案手法

最後の研究は既存の本人確認を強化するための 2 段階目の本人確認として使用することを想定している．既存の 2 段階目の本人確認では，2 つ目のパスワードや，専用のデバイスを用いて本人確認を行うが，パスワード，専用のデバイスの問題点は，パスワードは覚える必要性や，入力を覗き見られる問題があり，専用のデバイスは紛失時に再発行が必要であり，盗難にも弱いという問題点がある．2 段階目の本人確認として使用する場合，多人数での利用を想定する必要があり，2 つ目の研究の手法では多人数の利用に対応できないことと，一方向しか入力しないことで覗き見耐性が低いという問題があった．そこで，本システムでは 1 つ目と 2 つ目の研究を組み合わせ，空中に筆記した単語を複数の方向ごとに分解し，分解したデータ 1 つ 1 つを機械学習により検証する．これにより，最低でも 1 つ目の研究と同程度の覗き見耐性を確保できるうえ，更に，登録時と検証時で同じ文字を書かなくても本人確認をできるようにすることで，筆記を行うときの特徴を覗き見でわかりにくくすることで，更に耐性を向上させることができる．本システムは，パスワードなどによる本人確認手段を強化するものとして想定しているため，本人確認する人が登録者のうちの誰であるかを検証するため，教師あり訓練を使用する．ラベルを付けることで精度を向上させることができると考えた．更に，1 つのモデルに登録する人数を固定することで，多人数の場合であったとしても対応できるようにする．

6.1　システム概要

本システムは，Leap Motion を使用して利用者の本人確認手段を強化する．1 章の末尾に挙げた要件 1〜7 を全て満たすものであり，これを特徴とする．前の 2 つ研究と同様に Leap Motion を用いて空中に文字を描くため，筆跡が残らない．また，攻撃者は，覗き見を何度も行うか，指の動きを参考にできる位置から撮影を行い，筆記を

行った人の特徴を練習し，実際に手を動かして入力を行う必要がある．

このシステムは，パスワードによる本人確認を強化するものであるため，利用者は最初に従来通りの方法でパスワードなどの本人確認を行う．この本人確認は本研究とは独立しているため方法を問わない．その次に，Leap Motion を用いて画面に表示された単語を空中に指で筆記する．提案システムの構成と利用者の操作手順について以下に示す．

- 登録フェーズ
  1. 画面に表示される単語を空中に筆記する
  2. 筆記した単語を分解する
  3. 分解した中から数千個サンプルを抽出する
  4. (1)〜(3) を登録したい人数分行う
  5. 抽出したサンプルを人ごとに分け機械学習にて訓練する
- 個人識別フェーズ
  1. 画面に表示される単語を空中に筆記する
  2. 筆記した単語を分解する
  3. 分解した中から数千個サンプルを抽出する
  4. 登録フェーズで訓練したモデルを利用して抽出したサンプルを検証する
  5. 検証した結果から本人か他人かを分類する

登録フェーズでは，利用者は，システムから入力を求められる a 個の単語を筆記して入力する．a は利便性を考慮し 10 程度であり，本システムの評価においては 10 としている．

利用者が入力した筆記情報は，筆記を分解するための訓練済みモデルに入力され，筆記時の特徴として分解される．この訓練済みモデルは，この利用者とは関係なく事前に作成されたものであり，いかなる利用者にも依存しないものである．

機械学習における 1 回の訓練に入力される利用者を k 人と仮定すると，1 人の利用者による筆記時の特徴は，$k-1$ 人の筆記時の特徴と合わせられ，機械学習による訓練が行われる．つまり，誰の筆記かを判定する k 値分類の訓練である．この $k-1$ 人の筆記時の特徴は，システム側で任意に集めたものであり，k は利便性を考慮し 10 程度であり，本システムの評価においては 10 としている．訓練されたモデルは，利用者ごとに割り当てられた ID と結びつけられて保存される．

個人識別フェーズでは，利用者は，まず最初に通常の本人確認手段を実行する．これはパスワードなどによる本人確認手段である．続いて，追加の本人確認手段として，利用者は，システムから入力を求められる 1 個の単語を筆記して入力する．このとき，利用者が入力した ID とこの筆記時の特徴が，利用者の本人確認手段を強化するもの

となる．入力された 1 個の単語の筆記は，筆記を分解するための訓練済みモデルに入力され，筆記時の特徴として分解される．システムは，入力された ID に結びつけられている，筆記時の特徴の訓練済みモデルに，1 個の単語の筆記時の特徴を入力し，個人識別の判定を行う．この判定は，2 つの段階に分けられる．まず，入力された筆記時の特徴が，筆記時の特徴の訓練済みモデルの k 値分類により，k 人の誰に一番近いかを判定する．システム全体の利用者が m 人とすると，常識的に m は k より大きく，k は正規ユーザ 1 名とシステム側で任意に集めた $k-1$ 名で構成されるため，この中に攻撃者はいない．よって，攻撃者がいる場合，この k 人に含まれる人間以外の誰かということになる．筆記時の特徴の訓練済みモデルの k 値分類により，筆記時の特徴が一番近いものが正規ユーザ以外の $k-1$ 人のいずれかになれば，その時点で本人確認は失敗する．次に，筆記時の特徴が一番近いものが正規ユーザとなった場合の手順について説明する．本システムにおいては，本人確認の閾値がある．この閾値を上回れば，正規ユーザ本人であると判定され，そうでなければ本人確認に失敗する．

提案システムは最終的なシステムであるため，全ての要件を満たしているか確認を行う．要件 1 については，本システムでは Leap Motion というデバイスを本人確認用に必要とする．しかし，e-Tax の IC カードやハードウェアトークンなどと異なり，利用者ごとに本人確認をして再発行を行う必要はなく，1 つのデバイスを複数人で使い回すことも可能である．よって，再発行という手順は存在せず，要件 1 を満たす．

要件 2 ついては，Leap Motion による空中筆記が，ある程度の覗き見耐性を持っているため満たす．また，個人識別フェーズでシステムから入力を求められる 1 個の単語は毎回変わるため，一度もしくは数回覗き見たものをそのまま模倣して入力することはできない．これは要件 3 についても当てはまる．なお，要件 2 の定義が厳密でないのは，提案手法に絶対的な覗き見耐性はないからである．

要件 4 については，個人識別フェーズでシステムから入力を求められるものは 1 個の単語であり，クレジットカードの利用時に署名する作業，つまり，1 つ目の研究とほぼ同等の時間的負荷と考えられるため満たす．

要件 5 については，機械学習を行うのは利用者を含めた k 人であり，システムの利用者数 m には依存していないため満たす．また，登録フェーズ後に m に増減があったとしても，登録フェーズをやり直すことはない．これは要件 6 についても当てはまる．

要件 7 については，機械学習を行うのは利用者を含めた k 人である一方，筆記時の特徴が一番近いものが正規ユーザとなったとしても，その際の閾値によって本人か本人でないかを判別可能であるため満たす．

6.2 登録フェーズ

6.2.1 Leap Motion によるデータの取得

登録フェーズにおいて，利用者は，システムから入力を求められる単語を，Leap Motion を使用して筆記して入力する．

Leap Motion で取得するのは利き手の 5 指の指先と手のひら中央の合計 6 箇所の情報である．各箇所の情報は，以下のものを取得する．

- 手のひら X 軸の移動距離 [mm]
- 手のひら X 軸の回転角度 [rad]
- 手のひら X 軸の速度 [mm/s]
- 手のひら Y 軸の移動距離 [mm]
- 手のひら Y 軸の回転角度 [rad]
- 手のひら Y 軸の速度 [mm/s]
- 手のひら Z 軸の移動距離 [mm]
- 手のひら Z 軸の回転角度 [rad]
- 手のひら Z 軸の速度 [mm/s]
- 親指 X 軸の移動距離 [mm]
- 親指 X 軸の回転角度 [rad]
- 親指 X 軸の速度 [mm/s]
- 親指 Y 軸の移動距離 [mm]
- 親指 Y 軸の回転角度 [rad]
- 親指 Y 軸の速度 [mm/s]
- 親指 Z 軸の移動距離 [mm]
- 親指 Z 軸の回転角度 [rad]
- 親指 Z 軸の速度 [mm/s]
- 人差し指 X 軸の移動距離 [mm]
- 人差し指 X 軸の回転角度 [rad]
- 人差し指 X 軸の速度 [mm/s]
- 人差し指 Y 軸の移動距離 [mm]
- 人差し指 Y 軸の回転角度 [rad]
- 人差し指 Y 軸の速度 [mm/s]
- 人差し指 Z 軸の移動距離 [mm]
- 人差し指 Z 軸の回転角度 [rad]

- 人差し指 Z 軸の速度 [mm/s]
- 中指 X 軸の移動距離 [mm]
- 中指 X 軸の回転角度 [rad]
- 中指 X 軸の速度 [mm/s]
- 中指 Y 軸の移動距離 [mm]
- 中指 Y 軸の回転角度 [rad]
- 中指 Y 軸の速度 [mm/s]
- 中指 Z 軸の移動距離 [mm]
- 中指 Z 軸の回転角度 [rad]
- 中指 Z 軸の速度 [mm/s]
- 薬指 X 軸の移動距離 [mm]
- 薬指 X 軸の回転角度 [rad]
- 薬指 X 軸の速度 [mm/s]
- 薬指 Y 軸の移動距離 [mm]
- 薬指 Y 軸の回転角度 [rad]
- 薬指 Y 軸の速度 [mm/s]
- 薬指 Z 軸の移動距離 [mm]
- 薬指 Z 軸の回転角度 [rad]
- 薬指 Z 軸の速度 [mm/s]
- 小指 X 軸の移動距離 [mm]
- 小指 X 軸の回転角度 [rad]
- 小指 X 軸の速度 [mm/s]
- 小指 Y 軸の移動距離 [mm]
- 小指 Y 軸の回転角度 [rad]
- 小指 Y 軸の速度 [mm/s]
- 小指 Z 軸の移動距離 [mm]
- 小指 Z 軸の回転角度 [rad]
- 小指 Z 軸の速度 [mm/s]

これらを合計すると，1 回あたり 6 箇所× 3 軸× 3 項目の 54 項目の情報が取得される．本システムでは，この 1 回を 1 フレームと呼び，200 フレーム/秒でこれらの情報の取得を行う．なお，移動距離とは前フレームからの移動距離であり，回転角度とは指や手のひらの向きである．回転角度は軸に向かって右回転がプラスとなる．また，速度とは Speed ではなく Velocity であり，プラスとマイナスの向きを持つ．Leap Motion の API Overview には，Distance，Time，Speed，Angle の 4 つの項目が取

得できると記載されているが，Time は Distance と Speed から算出可能であるため 3 つの項目を使用した．

Time については，フレームごとに経過時間を記録し，前のフレームとの時間の差から 1 フレームの時間を算出した．Speed は次に示す式 (6.1) で算出している．

$$\frac{Distance(mm)}{Time(ms)} * 1000 = Speed(mm/s) \tag{6.1}$$

6.2.2 筆記の分解

利用者が入力した筆記情報は，筆記を分解するための訓練済みモデルに入力され，筆記時の特徴として分解される．本項では筆記を分解する方法について説明する．

本システムで使用する，分解された筆記とは，「左から右」，「右から左」，「上から下」，「下から上」の 4 つである．以後，本論文ではこれらを「4 つの向き」と総称する．筆記中にこれ以外の要素が現れても無視する．例えば，曲線の場合にはどの程度曲がっているかを考慮する必要があるし，斜め方向の線の場合にはどの程度傾いているかを考慮する必要がある．これらを除去して単純化するため，前述の 4 つの向きのみを用いた．なお，曲線の一部を拡大した場合，もしくは角度が浅い斜め方向の線であった場合には，直線に近似可能な要素が含まれる場合がある．これらは 4 つの向きとして分解される．

筆記を分解するためには筆記の情報が必要となる．これらは，筆記の分解専用の被験者を用いて集められる．筆記の分解専用の被験者とは，システムの利用者とは別に集められた人物を指す．筆記の分解専用の被験者には，位置が決められた空間が用意され，その空間に前述の 4 つの向きの直線が筆記される．これら，すべての被験者の筆記は，4 つの向きのそれぞれの向きごとにラベルが付与され，すべてまとめて機械学習に入力される．なお，機械学習においては，入力される 1 つのデータの塊を 1 サンプルと呼ぶため，本システムでも機械学習に入力されるデータの単位をサンプルと呼称する．本研究で用いる機械学習においては，入力時のサンプルサイズを統一する必要がある．サンプルサイズは前述の 54 項目×フレーム数に依存し，フレーム数は筆記にかかる時間によって変化する．本システムでは，フレーム間での線形補間を用いて，被験者の平均フレーム数になるようサンプルサイズを統一する．例えば，平均フレーム数が 20 である場合，18 フレームのサンプルは 20 フレームになるように線形補間が行われる．こうして，統一されたサンプルサイズに線形補間したすべてのサンプルを使い，訓練および検証が行われる．

6.2.3 筆記の特徴抽出

システムは利用者に単語を提示する．この単語は辞書などに掲載されているものでよく，本システムの評価ではパブリックドメインの辞書からランダムに選択している．具体的には「apple」などである．単語の長さの規定はないが，長いほうがより多くの特徴を抽出できる．本システムの評価では 3 文字以上の単語としている．利用者は，システムから入力を求められる a 個の単語を筆記して入力する．a は 10 程度である．9 や 11 では不適切というわけではないが，本システムの評価においては切りのよい数字である 10 としている．入力された a 個の筆記データは，それぞれ元のフレーム数が 0.2 倍から 3.0 倍になるように，0.2 きざみでフレーム間での線形補間が行われる．この線形補間後のすべてのデータから，6.2.2 項の訓練済みモデル作成時に使用された，平均フレーム数の長さとなるサンプルが切り出される．

具体的には，線形補間後のある筆記データが 1〜n フレームで構成され，6.2.2 項の訓練済みモデルにおける平均フレーム数が ave とすると，i 番目のサンプルは i フレームから $ave+i-1$ フレームまでが切り出され，合計 $n-ave+1$ 個のサンプルが切り出されることになる．

利用者が入力した a 個の単語の筆記すべてに対して，線形補間してから切り出されたすべてのサンプルは，6.2.2 項の訓練済みモデルに入力される．この訓練済みモデルは，入力されたサンプルを 4 つの向きのいずれかに分類するが，その際にソフトマックス値も出力する．1 個の単語の筆記から作成された大量のサンプルに対して，このソフトマックス値が高かった上位 b 個のみを，次項で述べる筆記の特徴の訓練にて使用する．本システムでは，b は数十〜数千である．

線形補間のイメージを図 6.1 に示す．

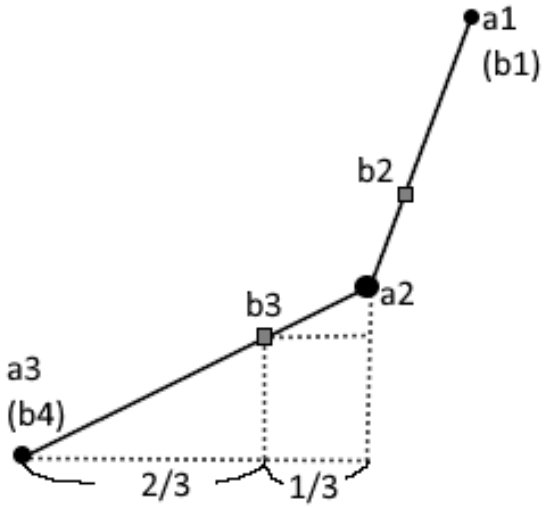

図 6.1　線形補間のイメージ

図のように，例えば a_1，a_2，a_3 と 3 つの点があり，これを 4 つの点 b_1，b_2，b_3，b_4 にしたいとする．a_1 と b_1 は同一であり a_3 と b_4 は同一であるため，b_2 と b_3 を線形補間によって導出すればよい．位置が均等になるようにするため，a_2 から見て a_1 や a_3 から 1/3 の距離に b_2 と b_3 を持ってくる．前フレームからの移動距離や回転角度の場合，a_3 の持つ値の 1/3 が a_2 から b_3 までの変化量となり，a_2 の持つ値の 1/3 が b_2 から a_2 までの変化量となる．速度の場合には，a_3 と a_2 の差の 1/3 を a_2 に足したものが b_3 の値となる．このように，線形補間によってフレーム数を増やしたり減らしたりして調整する．

次に，サンプルの切り出し方について図 6.2 を用いて説明する．

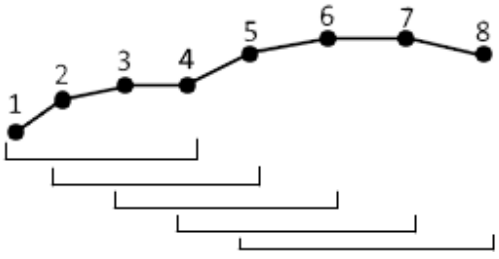

図 6.2　サンプル切り出しの例

実際には 1 サンプルは 20 フレームであるが，ここでは簡略化のために 8 フレームの筆記データ ($n = 8$) から 4 フレームずつのサンプル ($ave = 4$) を切り出すものとする．図 6.2 内の番号が筆記データのフレーム番号に対応する．まず，1〜4 フレーム目が切り出されて 1 つ目のサンプルとなる．次に，2〜5 フレーム目が切り出されて 2 つ目の

サンプルとなる．3 つ目のサンプルは 3〜6 フレーム，4 つ目のサンプルは 4〜7 フレーム，5 つ目のサンプルは 5〜8 フレームである．よって，$n - ave + 1 = 8 - 4 + 1 = 5$ 個のサンプルが切り出される．

切り出されたサンプルは，機械学習により「左から右」,「右から左」,「下から上」,「上から下」の 4 つのいずれかに分類される．このとき，どの程度の信頼を持ってそこに分類されたかの基準となるのがソフトマックス値であり，例えば「左から右」に分類されたサンプルのソフトマックス値が高ければ，それは「左から右」である可能性が高いが，ソフトマックス値が低かった場合，他の 3 つよりは「左から右」である可能性が高いだけということになる．

6.2.4　筆記の特徴の訓練

6.2.3 項で述べた，1 利用者あたり a 個の筆記× b 個のサンプルを，この利用者のサンプルとして機械学習に入力し，訓練および検証を行う．この利用者による筆記時の特徴は，システム側で用意した $k - 1$ 人の筆記時の特徴と合わせられ，k 値分類が行われる．k は 10 程度である．訓練後は，この利用者に ID が割り当てられ，その ID に基づく利用者の筆記の特徴の訓練済モデルが保存される．

6.3　個人識別フェーズ

個人識別フェーズでは，利用者は，まず通常の本人確認手段を実行する．本システムではパスワードなどと併用することを想定しているが，それに制限されるものではない．本システムの技術とは関係がないためそれらの記述は省略する．

このフェーズを図示すると，図 6.3 のようになる．例として，この時のサンプル数は 1500 としている．

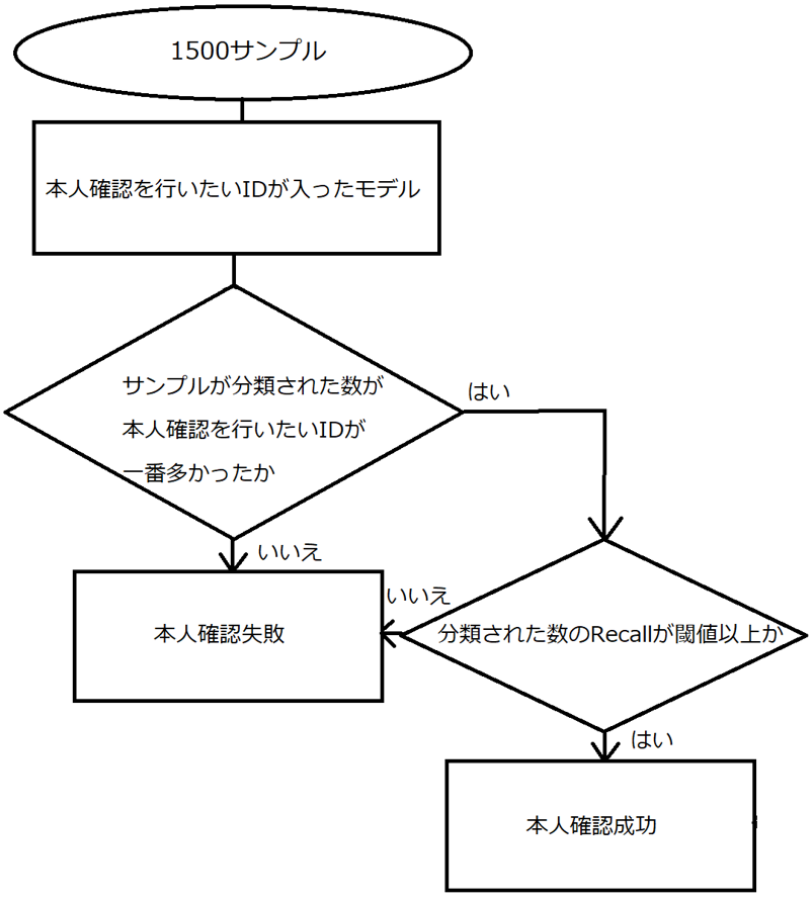

図 6.3　個人の識別識別フェーズのフローチャート

本人確認の後，追加の本人確認手段として，利用者は，システムから入力を求められる 1 個の単語を筆記して入力する．この単語の条件は，6.2.3 項で述べたものと同様である．入力された筆記は，6.2.3 項で述べたものと同様の手順で，筆記の特徴抽出が行われる．利用者が入力した 1 個の単語の筆記に対して，線形補間してから切り出されたすべてのサンプルは，6.2.2 項の訓練済みモデルに入力される．この訓練済みモデルを使った分類により，1 個の単語の筆記から作成された大量のサンプルに対して，ソフトマックス値が高かった上位 b 個のサンプルが個人識別のテストに使用される．

前述の b 個のサンプルは，6.2.4 項で述べた，その ID に基づく利用者の筆記の特徴の訓練済モデルに入力される．k 値分類の結果，最も多くのサンプルがその ID に分類されなかった場合には，個人識別に失敗となる．最も多くのサンプルがその ID に分類された場合には，さらに Recall による閾値の判定が行われる．この Recall が事前に設定された閾値を上回れば個人識別に成功とし，閾値以下であれば失敗となる．

Recall の算出方法は次の通りである．まず，b 個のサンプルの分類結果を，TP (True Positive)，FP (False Positive)，TN (True Negative)，FN (False Negative) に

分ける．そして，これらから Recall を求める．この閾値は，FAR (False Acceptance Rate) と FRR (False Rejection Rate) を考慮し，システム側もしくは利用者側で任意に設定可能である．この閾値は利用者ごとに異なってよい．

第 7 章

実装

7.1 実装環境

本節では，実装に使用した言語やドライバとそのバージョンについて記述する．Leap Motion のデータは Leap Motion API[44](version 2.3) を使用して取得した．プログラミング言語に関しては，データの取得プログラムには C# 7.0，機械学習プログラムには Python3.8.10 を使用した．また，Python の機械学習用ライブラリとして，Scikit-learn(1.1.0)，Chainer(version 7.8.1) を使用した．データの取得を行った PC の環境を表 7.1 に示す．機械学習を実行した PC の環境を表 7.2 に示す．

表 7.1　開発，データ記録プログラム実行環境

OS	Windows 10 Education
CPU	Intel(R) Core(TM) i7-3820
GPU	NVIDIA GeForce GTX 960 SLI
Memory	64GB

表 7.2　機械学習実行環境

OS	Ubuntu 20.04.4 LTS
CPU	Intel(R) Core(TM) i5-12400
GPU	NVIDIA GeForce RTX 3080 Ti
Memory	128GB

7.2　筆記のための実装

7.2.1　筆記記録環境の構築

6.2.2 項で述べた筆記の分解，6.2.3 項で述べた筆記の特徴抽出，6.3 節で述べた個人識別フェーズのために，図 7.1 に示す通り物理的な環境を構築した．使用したディスプレイは 20 インチの Dell 2007FPb (UXGA) であり，モニタの上下および左右の中心から 1.5cm の位置に竹串 (直径 2mm) をセロハンテープで固定し，画面から 2cm の距離に，垂直または水平になるよう，竹串同士をつなぐ糸を張った．つまり，画面の中心に，糸で囲まれた 3cm 四方の四角形ができる．この四角形の真下に Leap Motion の中心が来るように，Leap Motion を配置した．Leap Motion は，アルミ製のヒートシンクを用いて机のテーブル面から 2cm 浮かせて固定しており，机のテーブル面から，水平に張られた一番下の糸までの距離は 17cm である．2cm 浮かせた理由はディスプレイ下に取り付けた竹串が Leap Motion のカメラに映らないようにするためである．机の高さは，一般的な事務机の 70cm である．

図 7.1　筆記の分解のためのデータ取得環境

Leap Motion は画面から 5cm 離れた位置に固定した．これは，Z 軸 (奥行き) 方向にも指を動かせるようにするためと，近すぎるとディスプレイから発せられる熱により赤外線の受光が妨害されるためである．ディスプレイを用いたのは，筆記時の特徴を抽出するための実験時に，指がきちんと Leap Motion に認識されているかどうか，

被験者がその場で確認できたほうがよいと考えたためである．Leap Motion の有効範囲は X，Y，Z 方向すべてにおいて 2.5cm から 60cm となっており，本実装の環境はこの有効範囲内に収まっている．

6.2.2 項で述べた，筆記の分解のための訓練用に被験者が筆記を行う際，糸で囲まれた 3cm 四方の四角形の中で筆記を行う．これは，各被験者が任意の位置に任意の距離の直線を自由に描いてしまい，直線の特徴がうまく取れなくなってしまうことを防ぐためである．例えば，左から右の直線を筆記する場合には，最初に画面中央を人差し指が指すようにして手を Leap Motion の上にかざして指を認識させ，四角形の左辺の左側に人差し指がくるように手を移動する．そこから，四角形の右辺の右側に人差し指がくるまで指を移動させ，手を Leap Motion の範囲外に移動して 1 回の筆記を入力させる．このとき，機械学習に使用されるデータは，四角形の範囲内に人差し指の座標があるものだけとなる．同様に，右から左，上から下，下から上の直線も四角形の範囲内に人差し指の座標があるものだけが，機械学習に使用されるデータとなる．

6.2.3 項で述べた筆記の特徴抽出，6.3 節で述べた個人識別フェーズのために被験者が単語を入力する際には，糸で作った範囲に関係なく，ディスプレイの手前を広く使って筆記する．

7.2.2 一方向記録プログラムの実装

このプログラムは提案手法 6.2.1 項の署名を分解するための一方向の移動を入力をさせるものである．このプログラムで各方向の移動を記録し，分解用の訓練，検証データにする．Leap Motion に手をかざすと現在の状態 (記録しているかしてないか) と，入力する方向の指示が表示される．ここで入力した一方向の移動の長さの平均を出し，筆記を検証するときの分割するフレームの長さの基準とする．

6.2.1 項で述べた 54 の特徴の取得方法は，移動距離である Distance は DistanceTo，回転角度である Angle は AngleTo という関数を使用して取得した．これらは「Vector - Leap Motion C# SDK v2.3 documentation」に掲載されており，どちらも Vector を入力すると float の値を返す．Vector クラスには x，y，z メンバがあり，3 つの軸を合わせて 1 つの移動距離と回転角度を求められる．それぞれの軸で別個に距離や角度を求めるため，x の場合は例えば DistanceTo(Vector.x,0,0) のように 1 つの軸だけにしてから計算を行っている．速度は移動距離と時間から計算した．これら 3 つの項目を親指，人差し指，中指，薬指，小指と手のひらの 6 箇所で取得し，X，Y，Z の 3 軸を使い 54 項目となる．入力されたデータは CSV ファイルとなって出力される．一行 1 フレームであるため，フレーム数分の行*54 列である．記録した方向ごとのデータのフレームの長さの平均を求め，その長さに合わせ線形補間を行う．長さを合わせ

たサンプルを作り，機械学習で訓練を行い，その結果を用いて線形補間と，フレーム数で分割された単語から良いサンプルを抽出する．

7.2.3 筆記記録プログラムの実装

このプログラムは提案手法 6.2.2 節を実現するためのプログラムで，見た目は一方向記録プログラムと似ているが，内部の処理が大きく異なっている．一方向記録の時は，糸のところに手を移動させると記録を開始し，反対側の糸まで移動すると記録を終了するという仕組みで動いているが，この筆記記録用のプログラムは Leap Motion の上で手のひらを広げた状態から，拳を握ると，画面上にランダムな単語が出現する．それを確認し，手を広げると記録開始となる．そこからは一筆書きで筆記を行う．入力が終わったら拳を握り，記録が終了する．取得するデータは一方向記録プログラムと同じだが，フレームの長さが文字の長さだけ長くなっている．このプログラムでは実験用に数百回入力させるようにしているが，実際に登録や，検証を行う場合は，それほど多く筆記を行わせることはない．記録した単語のデータ線形補間で伸縮させ，分解用の機械学習に入れることで検証用の一方向の入力のサンプルを適当な個数抽出する．

7.2.4 線形補間プログラム

線形補間プログラムは上記の記録プログラムで記録したデータの線形補間を行う．一方向のみの場合はフレームを伸縮するだけだが，筆記の場合はフレームを伸縮したうえ，そこから決められた長さにフレームを切り取る．フレームの伸縮方法，切り取り方は提案手法 6.2.3 項の通りである．

7.2.5 なりすまし実験環境

被験者の書いた筆記を真似するために，筆記動作の録画を行う必要がある．そこで，録画の環境の構築を行った．

まず，被験者は座った状態で筆記を行う．そこにカメラとして，2 台のスマートフォン (Nexus 5X Android8) を，三脚に取り付け，被験者の左右に 1 台ずつ置き，肩の上にカメラが来るぐらいまで高さを上げる．カメラの位置は両肩の端に合わせる．カメラの前後位置は，被験者が正面にあるディスプレイ中央を見た時に，視界に映らないぐらいの位置にカメラを置く．カメラの向きは，中央にディスプレイの中央が写る位置にし，カメラの画面下に Leap Motion が写る位置に上下の向きを合わせる．これにより本人からはカメラが見えないが，カメラからは筆記動作と筆記している単語を見ることができる状況を作る．実際にカメラで録画した映像の画像が図 7.2 となる．

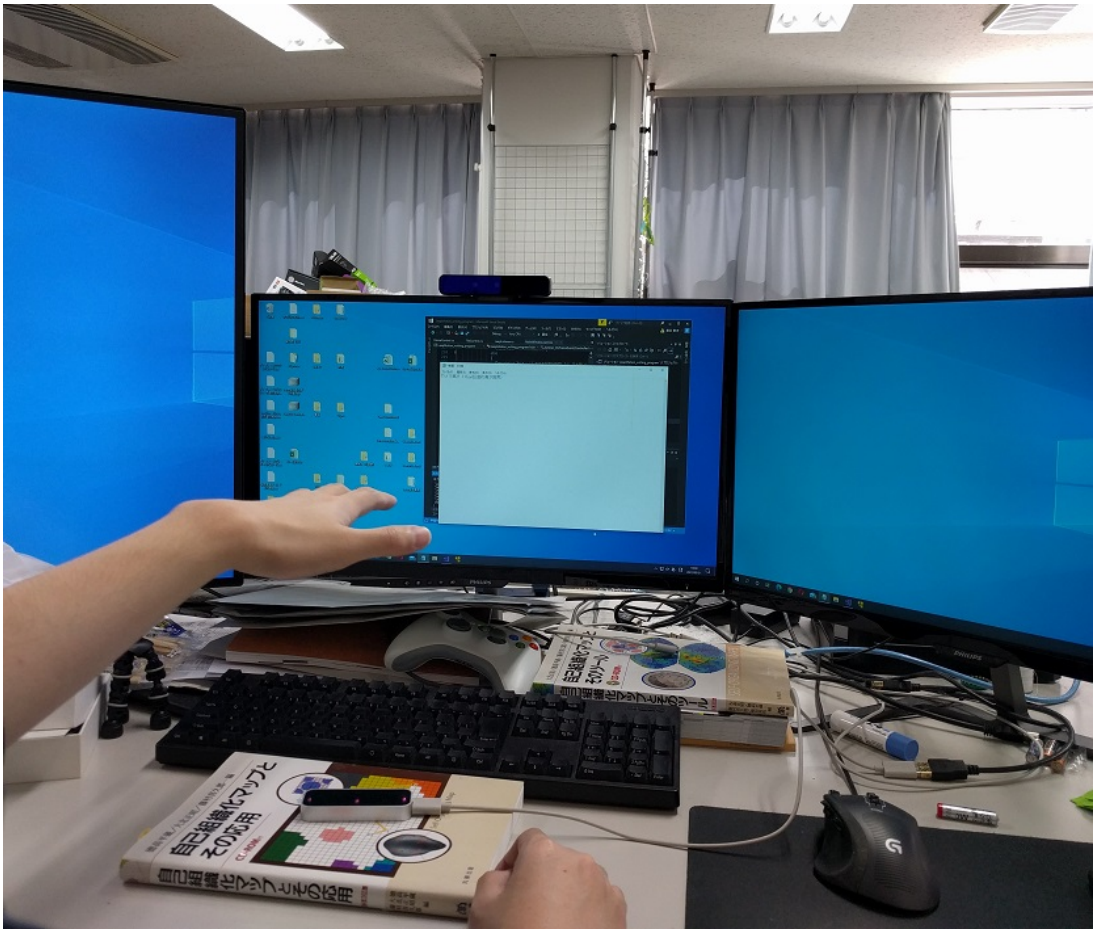

図 7.2　カメラによる筆記の録画

7.3　機械学習の実装

機械学習の実装は Python のパッケージの 1 つである，Scikit-learn を使用して実装をした．使用する機械学習については，機械学習アルゴリズム選択ガイド [45] に基づいて選択した．この研究では教師ありの分類問題のため，以下の機械学習が選ばれた．

- カーネルサポートベクトルマシン
- 勾配ブースティングツリー
- ランダムフォレスト
- ニューラルネットワーク

サポートベクトルマシンのカーネルは最も一般的である RBF カーネルを使用した．これらの機械学習を用いて方向の分類の精度を評価し，良かったものを筆記の分解で利用する．

Python での実装方法は Scikit-learn をインポートしてから，訓練データと検証データと，それぞれのラベルを用意し，model.fit(訓練データ，訓練ラベル) で学習できる．model には各機械学習のクラスが入る．ただし，ニューラルネットワークの実装については，別のフレームワークを使用するため，次項に記述する．クラス名は以下の通りである．

- カーネルサポートベクトルマシン：SVC()
- 勾配ブースティングツリー：GradientBoostingClassifier()
- ランダムフォレスト：RandomForestClassifier()

これらのインスタンス生成時に () の中にハイパーパラメータを記述することができる．何も書かなかった場合はデフォルトの値が使用される．評価ではハイパーパラメータはデフォルト値で行う．ただし，並列処理を行い高速化するために CPU の数は 4 つとするパラメータを記述した．

次に，model.predict(検証データ) を使用して予測したラベルのリストを取得できる．最後に予測したラベルと検証ラベルのリストから TP，TN，FP，FN を算出し，Precision，Recall，F-Score を求め，評価を行う．

7.4 ニューラルネットワークの実装

Yamamoto らの研究 [29] を参考にし，畳み込みニューラルネットワーク (CNN: Convolutional Neural Network) を Chainer を用いて実装を行った．ネットワークの構造については，Yamamoto らの研究をそのまま模倣したものと，Yamamoto らの研究のものに Ioffe らの理論 [46] を取り入れて改善したものの 2 つを用いた．図 7.3 に Yamamoto らのものを模倣したネットワークを，図 7.4 に Ioffe らの理論を取り入れて改善したネットワークを示す．

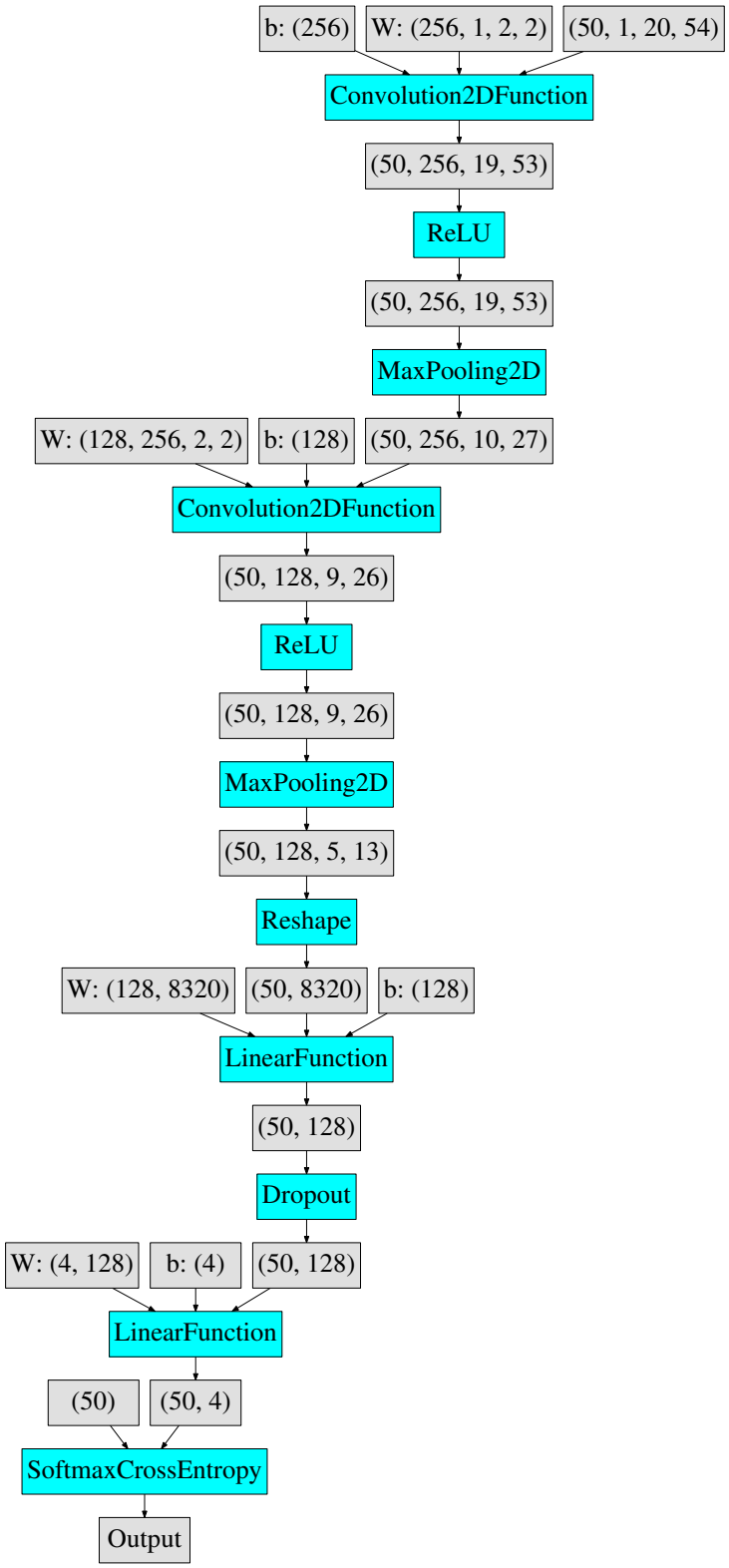

図 7.3　畳み込みニューラルネットワークにおける層構造

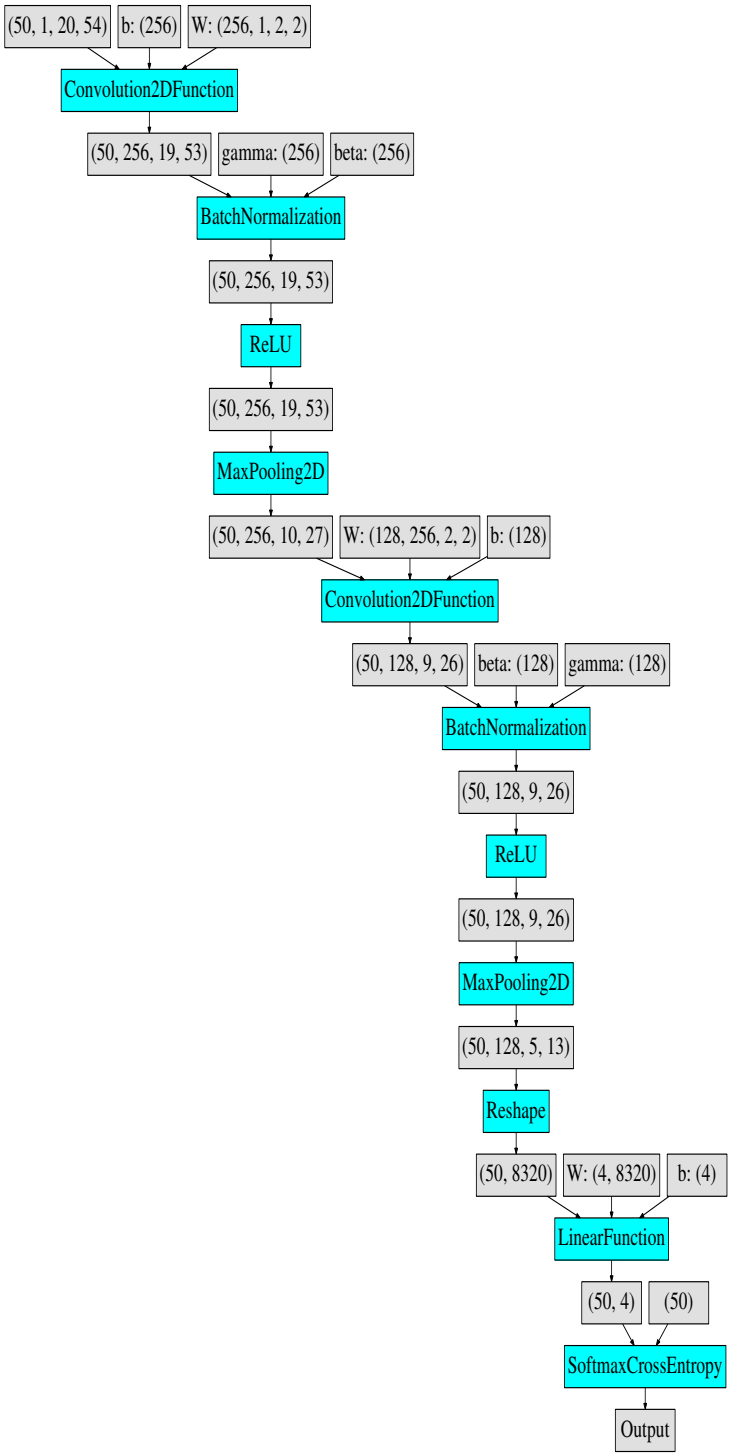

図 7.4　畳み込みニューラルネットワークにおける層構造 2

まず，図 7.3 に示したネットワークから説明する．機械学習に用いたパラメータは，Yamamoto らのものと同様とした．ただし，ネットワークの入力ユニット数と出力ユニット数は，私が取得したサンプル数および，私の研究の出力による都合があるため，この限りではない．k-size は 2，padding は 0，入力するサンプルは 2 次元配列であ

り，チャネル数は 1 である．活性化関数は ReLU を，最適化関数は Adam を用いた．プーリング層には最大プーリングを用い，ウィンドウサイズは 1 である．全結合層の入力ユニット数は，直前のプーリング層の出力ユニット数から Chainer が自動的に決定し，中間層の場合には 128 である．Dropout の確率は 50% である．

一方，Yamamoto らのものと異なり，出力層のユニット数は，筆記分解ならば 4 方向の 4，個人識別の場合は訓練に使用するサンプルの被験者数である．

エポック数については上限を 100 とし，Early stopping を用いて訓練中に validation/loss が 5 エポックの間下がらなかった場合に訓練を終了するようにした．これは，過学習防止のためと，訓練時間短縮のためである．

バッチサイズに関しては Yamamoto らの論文に記述がなかったため，Kritsis らの，Leap Motion の座標データを CNN に入力してジェスチャーを判別する研究 [47] を参考にし，これと同様の 50 とした．

次に，Ioffe らの理論を加えて改善したネットワークについて説明する．Yamamoto らのネットワークでは，全結合層の間で Dropout を利用しているが，Ioffe らによると，そこから全結合層を 1 つ減らし，さらに Dropout を削除し，代わりに畳み込み層の終わりに Batch Normalization を追加することで Dropout のみで訓練したときより過学習を抑えることができると述べているため，私もそれに従った．なお，Dropout は Dropout 率のパラメータを設定する必要があるが，Batch Normalization は手動で設定する必要のあるパラメータが存在しないため，パラメータ設定の手間が減るという利点もある．さらに，全結合層間の Dropout が存在しなくなったことにより，訓練速度の向上も見込める．

評価においては，これらのネットワークを Accuracy と Loss で比較し，良いものを用いた．

上記の 2 つのネットワークとの比較用に，全結合のみのニューラルネットワークも実装し，これの評価も行う．図 7.5 に全結合のみのニューラルネットワークを示す．ネットワークの構造としては，畳み込み層とプーリング層を全結合層に変えただけである．パラメータは畳み込みニューラルネットワークのものと同じとし，Batch Normalization を使用する．

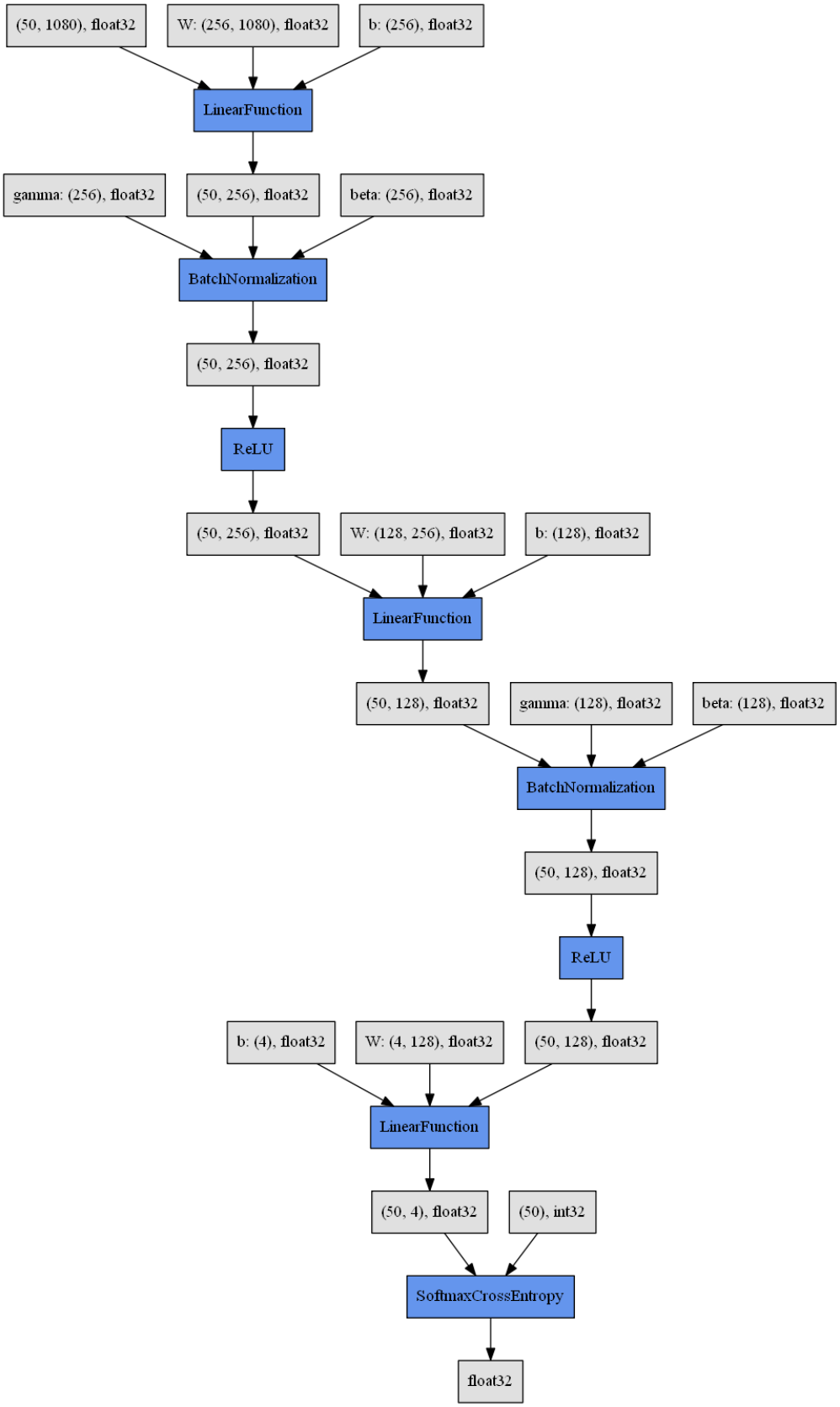

図 7.5　全結合のみのニューラルネットワーク

訓練およびテストに使用するサンプルの割合に関しては，Yamamoto らの研究のとおりとした．サンプル数が少ない場合には，k-分割交差検証を行うこともあるが，本研究のサンプル数は評価に十分であると判断し，Yamamoto らの研究に合わせた．

第 8 章

評価

本章では，6 章で述べた提案システムの妥当性について評価を行う．

スペースの都合により，本章の表中では，方向を Dir，右から左を R-L，上から下を U-D，左から右を L-R，下から上を D-U，平均を Ave，標準偏差を Std と表記し，平均値の項目においては，平均値の右に括弧書きで標準偏差の値を記すものとする．

8.1　筆記の分解についての評価

本節では，6.2.2 項で述べた筆記の分解についての評価を行う．筆記を分解するための筆記の情報は，システムの利用者とは別に集められた筆記の分解専用の被験者によって入力されることを説明した．よって，評価の手順としては，筆記の分解専用の被験者によって入力された筆記の情報が，機械学習により適切に訓練および検証が行われているのかの評価を行い，適切に訓練済みモデルが作成された場合には，そのモデルによって利用者の筆記が適切に分解されるのかテストの評価を行うことになる．

利用者の筆記が適切に分解されるのかのテストについては，6.2.3 項で述べたように，テスト結果の値が良いか悪いかに関係なく，ソフトマックス値が高かった上位 b 個のサンプルが筆記の特徴の訓練に使用されることとなる．つまり，最終的に利用者が個人識別される精度を評価するまで，利用者の筆記が適切に分解されたかどうかは判断できないため，この部分の評価は本節ではなく次節以降で行うものとする．よって，本節では，筆記の分解専用の被験者によって入力された筆記の情報が，機械学習により適切に訓練および検証が行われているかのみ評価を行う．

本節で行う筆記の分解専用の被験者として，10 名の被験者を用意した．これら 10 名の被験者は，東京工科大学宇田研究室所属の学生である．これら 10 名の被験者が，7.2 節の環境で 4 つの向きの直線をそれぞれ 150 回入力した．

なお，実験を行う中で，センサの不具合により方向を入力し終えても記録が止まら

ず記録し続けることがあった．実験を観察していると，すべての被験者による 1 回の入力は，長く見積もっても 1 秒も掛かっていなかった．そこで，フレーム数が 200 を超えたデータは，センサの不具合と見なして破棄した．これは，Leap Motion が 1 秒間に 200 フレームの取得を行うためである．7.3 節で述べた Yamamoto らの研究に合わせ，各被験者における 1/3 を検証用サンプル，1/3 を訓練用サンプル，残りをテスト用サンプルとした．前述の破棄があるため，その分サンプルが減っている．

被験者から筆記を取得した結果，6.2.2 項で述べた平均フレーム数は 20.206 となった．小数点を四捨五入し，本システムで筆記の分解に利用するサンプルの平均フレーム数は 20 とした．

訓練および検証は以下の通りに行われた．訓練用サンプル全体からランダムで 180 個を選択し訓練を行い，すべての検証用サンプルで検証を行うという手順を 10 回繰り返した．

最初に BatchNormalization と Dropout どちらを使った場合のほうが精度が良いか比較をする．訓練および検証を，7.4 節で述べた図 7.3，7.4 の 2 つのネットワークを用いてそれぞれ行った．ハイパーパラメータは 7.4 節の通りである．出力層のユニット数は 4 つの向きのため 4 である．入力層のユニット数 180 は，Chainer が自動決定したものである．

Validation accuracy および loss の平均値と標準偏差を表 8.1 に示す．

表 8.1　Yamamoto らのネットワークと Ioffe らの理論に基づき改良されたネットワークの比較

ネットワーク	Accuracy	Loss
Yamamoto らの手法版	0.962(0.059)	0.256(0.126)
Ioffe らによる改良版	0.996(0.003)	0.031(0.023)

表 8.1 より，Ioffe らの理論による改良を加えたもののほうが，精度が高く標準偏差も小さかった．よって，本論文の評価において CNN と全結合ニューラルネットワークは，これ以降，すべて BatchNormalization を用いるネットワークにて評価を行っている．

筆記分解のための方向の訓練は以下の機械学習 (ニューラルネットワーク) を使用し，これらの精度を評価する．

- カーネルサポートベクトルマシン (RBF)
- 勾配ブースティングツリー
- ランダムフォレスト

- 全結合ニューラルネットワーク
- 畳み込みニューラルネットワーク

カーネルサポートベクトルマシンにおける Precision，Recall，F 値の平均値および標準偏差を表 8.2 に示す．カーネルサポートベクトルマシンを使用して方向の分類を訓練した結果の F 値は 94.5% となった．勾配ブースティングツリーにおける Precision，Recall，F 値の平均値および標準偏差を表 8.3 に示す．勾配ブースティングツリーを使用して方向の分類を訓練した結果の F 値は 99.4% となった．ランダムフォレストにおける Precision，Recall，F 値の平均値および標準偏差を表 8.4 に示す．ランダムフォレストを使用して方向の分類を訓練した結果の F 値は 99.6% となった．全結合ニューラルネットワークにおける Precision，Recall，F 値の平均値および標準偏差を表 8.5 に示す．全結合ニューラルネットワークを使用して方向の分類を訓練した結果の F 値は 99.5% となった．畳み込みニューラルネットワークにおける Precision，Recall，F 値の平均値および標準偏差を表 8.6 に示す．畳み込みニューラルネットワークを使用して方向の分類を訓練した結果の F 値は 99.7% となった．

上記の結果より，最も精度が高かったのは畳み込みニューラルネットワークとなった．そのため，畳み込みニューラルネットワークを用いて筆記分解用のモデルを作り，以降の評価に使用した．モデルは 10 回の訓練のうち一番結果が良かったものを使用する．

表 8.2　筆記を分解するための訓練の評価結果 (カーネルサポートベクトルマシン)

方向	Precision	Recall	F-score
右から左	0.955(0.011)	0.951(0.018)	0.953(0.01)
上から下	0.943(0.007)	0.998(0.002)	0.969(0.003)
左から右	0.995(0.006)	0.825(0.024)	0.902(0.013)
下から上	0.955(0.015)	0.955(0.013)	0.955(0.009)
平均	0.962(0.004)	0.932(0.007)	0.945(0.006)

表 8.3　筆記を分解するための訓練の評価結果 (勾配ブースティングツリー)

方向	Precision	Recall	F-score
右から左	0.985(0.008)	0.992(0.004)	0.989(0.004)
上から下	0.996(0.002)	0.998(0.001)	0.997(0.001)
左から右	0.998(0.002)	0.992(0.006)	0.995(0.003)
下から上	0.998(0.002)	0.991(0.006)	0.994(0.003)
平均	0.994(0.002)	0.993(0.002)	0.994(0.002)

表 8.4　筆記を分解するための訓練の評価結果 (ランダムフォレスト)

方向	Precision	Recall	F-score
右から左	0.992(0.002)	0.993(0.004)	0.992(0.002)
上から下	0.997(0.002)	0.998(0.001)	0.998(0.001)
左から右	0.999(0.001)	0.994(0.003)	0.996(0.001)
下から上	0.997(0.002)	0.999(0.001)	0.998(0.001)
平均	0.996(0.0)	0.996(0.001)	0.996(0.001)

表 8.5　筆記を分解するための訓練の評価結果 (全結合ニューラルネットワーク)

方向	Precision	Recall	F-score
右から左	0.987(0.006)	0.996(0.004)	0.991(0.002)
上から下	0.998(0.002)	0.997(0.002)	0.998(0.001)
左から右	0.999(0.001)	0.993(0.005)	0.996(0.002)
下から上	0.996(0.004)	0.997(0.003)	0.997(0.002)
平均	0.995(0.001)	0.996(0.001)	0.995(0.001)

表 8.6　筆記を分解するための訓練の評価結果 (畳み込みニューラルネットワーク)

方向	Precision	Recall	F-score
右から左	0.996(0.002)	0.996(0.004)	0.996(0.002)
上から下	0.994(0.004)	0.996(0.002)	0.995(0.003)
左から右	0.998(0.001)	0.998(0.002)	0.998(0.001)
下から上	0.998(0.002)	0.998(0.002)	0.998(0.001)
平均	0.997(0.002)	0.997(0.001)	0.997(0.001)

8.2 筆記の特徴抽出についての評価

6.2.3 項で述べた，利用者からの筆記の特徴抽出について評価を行った．利用者として，東京工科大学宇田研究室所属の学生から 11 名の被験者を用意した．なお，これらの被験者は，8.1 節における筆記の分解専用の被験者とは完全に別人である．

本節の評価における環境には図 7.1 と同じものを使用したが，被験者は糸を無視して筆記を行っている．筆記開始位置は画面の中央とした．入力する単語は，インターネット上のパブリックドメインの辞書 [48] に掲載されている 65600 件の英単語から，ランダムで選択した．なお，データ量が少なくなることを危惧し，3 文字以上の単語のみが選択されるようにした．被験者が筆記する文字は，被験者全員が十分に慣れているブロック体である．各被験者が異なる 100 単語をそれぞれ入力した．なお，被験者間で同一単語が選択される可能性はある．

まず，6.2.3 項で，ソフトマックス値が高かった上位 b 個のみを筆記の特徴の訓練に使用すると述べた点について評価を行った．本システムでは，b は数十〜数千と想定している．そこで，b を 90，150，300，600，900，1500，3000，6000 として評価を行った．これは，1/3 を検証用サンプル，残りを訓練用サンプルとする都合上，割り切れる値で切りのよい値を選んだことによる．

11 人の被験者の一人 1 単語を学習したモデルに登録者の学習していない単語 99 個を検証し，1 単語のサンプル b 個が TP として分類された数が一番多い場合を本人とするとし，本人確認に失敗した確率 (FRR) からサンプル数別の FRR を求めた．その平均値と標準偏差を表 8.7 に示す．

表 8.7　サンプル数別の全被験者の FRR の平均と標準偏差

数	90	150	300	600	900	1500	3000	6000
Ave	0.63	0.62	0.6	0.56	0.49	0.43	0.41	0.39
Std	0.13	0.18	0.21	0.2	0.21	0.2	0.23	0.24
計	0.76	0.8	0.81	0.76	0.7	0.63	0.64	0.63

表 8.7 より，単純に FRR の平均値が最も低いものは，サンプル数が 6000 のときのものである．しかし，この実験の被験者は 10 名のため，サービスの提供を受けるユーザ数が非常に多くなる場合には，そのばらつきも考慮する必要がある．そこで，平均値に標準偏差を足した値を，最悪のケースにおける平均値と考え，その値がもっとも小さくなる 1500 を以降の実験で使用することにした．なお，平均値に標準偏差を足

した値は，サンプル数 1500 と 6000 との差がないが，サンプル数が少ないほうが訓練時間が短くなり，より適切であるといえる．

8.3 筆記の特徴の訓練についての評価

6.2.4 項で述べた，筆記の特徴の訓練における評価を行った．利用者としての被験者は，8.2 節における評価と同一人物の 11 名である．6.2.4 項の通りとすると，1 利用者あたり a 個の筆記× b 個のサンプルを，k 値分類して評価を行うこととなる．まず，本システムとして a は 10 程度を想定しているため 10 とした．b については，8.2 節の評価結果から，1500 が最適であるため 1500 とした．そして，k であるが，被験者 11 名に対して，次節で行う評価の都合上 10 としている．11 名から 10 名を選択する方法については以下に記す．

訓練および検証を次の通り行った．訓練用サンプル作成のための単語は，1 回の訓練ごとに 8.2 節の評価で被験者が入力した単語の中から，被験者ごとに 10 個ずつ選択している．被験者ごとに入力した 100 単語の内，訓練用サンプル作成に使用されなかった残りの 90 単語が検証用サンプル作成のために使用される．10 名の被験者による評価は，次のように被験者を選択して行った．被験者に A〜K と仮の識別名を付ける．被験者 A の評価は，被験者 B〜K のうちいずれか 1 名を除いた 9 名と被験者 A で行う．10 回の評価の内，被験者 B〜K は均等に 1 回ずつ除かれるようにした．被験者 B の評価は，被験者 A および C〜K のうちいずれか 1 名を除いた 9 名と被験者 B で行い，以下は同様である．被験者 C〜K も同様である．この訓練の検証を 8.1 節で F 値が高かった 3 つの機械学習，ランダムフォレスト，全結合ニューラルネットワーク，畳み込みニューラルネットワークで行った．勾配ブースティングツリーについては，F 値は上記の 3 つに近い結果であったが，訓練の時間が他の機械学習より遅く，2 番目に遅かった CNN が 97 秒なのに対して，勾配ブースティングツリーは 2909 秒と極端に遅かったため，8.1 節よりもサンプルが多くなりさらに遅くなると考え，本項の評価を行わなかった．

ランダムフォレストの結果を表 8.8 に，全結合ニューラルネットワークの結果を表 8.9 に，畳み込みニューラルネットワークの結果を表 8.10 に示す．

表 8.8 より，ランダムフォレストは，すべての被験者において F 値の平均が 0.994 以上となり，その標準偏差は 0.002 であった．表 8.9 より，全結合ニューラルネットワークは，F 値の平均が 0.982 以上となり，その標準偏差は 0.006 であった．表 8.10 より，畳み込みニューラルネットワークは，すべての被験者において F 値の平均が 0.98 以上となり，その標準偏差は 0.001 であった．

表 8.8　筆記時の特徴の訓練の検証結果 (ランダムフォレスト)

被験者	Precision	Recall	F-score
A	0.999(0.002)	0.998(0.002)	0.999(0.001)
B	0.996(0.003)	0.998(0.002)	0.997(0.002)
C	0.997(0.003)	0.996(0.004)	0.996(0.003)
D	0.994(0.004)	0.995(0.003)	0.994(0.002)
E	0.999(0.0)	0.999(0.001)	0.999(0.0)
F	0.999(0.001)	1.0(0.0)	1.0(0.001)
G	0.995(0.003)	0.994(0.004)	0.994(0.002)
H	0.999(0.003)	0.999(0.001)	0.999(0.001)
I	0.998(0.001)	0.998(0.002)	0.998(0.001)
J	0.997(0.002)	0.993(0.007)	0.995(0.003)
K	0.995(0.005)	0.997(0.002)	0.996(0.002)
平均	0.997(0.001)	0.997(0.001)	0.997(0.001)

表 8.9　筆記時の特徴の訓練の検証結果 (全結合ニューラルネットワーク)

被験者	Precision	Recall	F-score
A	0.991(0.005)	0.998(0.002)	0.995(0.003)
B	0.981(0.012)	0.995(0.004)	0.988(0.006)
C	0.993(0.005)	0.992(0.004)	0.992(0.004)
D	0.99(0.006)	0.987(0.013)	0.989(0.008)
E	0.997(0.004)	0.997(0.002)	0.997(0.002)
F	0.998(0.002)	1.0(0.0)	0.999(0.001)
G	0.986(0.006)	0.979(0.009)	0.982(0.006)
H	0.997(0.003)	0.995(0.004)	0.996(0.002)
I	0.995(0.002)	0.994(0.005)	0.995(0.003)
J	0.993(0.005)	0.981(0.004)	0.987(0.003)
K	0.991(0.004)	0.993(0.006)	0.992(0.004)
平均	0.992(0.002)	0.992(0.002)	0.992(0.002)

表 8.10　筆記時の特徴の訓練の検証結果 (畳み込みニューラルネットワーク)

被験者	Precision	Recall	F-score
A	0.989(0.005)	0.995(0.007)	0.992(0.004)
B	0.973(0.026)	0.991(0.006)	0.982(0.012)
C	0.997(0.003)	0.993(0.005)	0.995(0.003)
D	0.988(0.008)	0.98(0.014)	0.984(0.007)
E	0.994(0.005)	0.999(0.001)	0.996(0.002)
F	0.994(0.004)	0.999(0.001)	0.997(0.003)
G	0.986(0.007)	0.975(0.016)	0.98(0.01)
H	0.994(0.005)	0.993(0.005)	0.993(0.004)
I	0.993(0.003)	0.993(0.004)	0.993(0.002)
J	0.991(0.012)	0.974(0.012)	0.982(0.006)
K	0.988(0.005)	0.992(0.006)	0.99(0.004)
平均	0.99(0.004)	0.99(0.004)	0.99(0.004)

8.4　個人識別の評価

8.4.1　本人確認の評価

6.3 節で述べた，個人識別の評価を行った．利用者としての被験者は，8.2 節における評価と同一人物の 11 名である．

まず，利用者本人がその利用者として分類されるかどうかの評価を行った．仕様が 6.3 節の通りとすると，k 値分類における k 名の内の 1 名が利用者本人，残りの 9 名がシステム側で用意した被験者ということになるが，訓練および検証は 8.3 節のものをそのまま利用し，例えば被験者 A を本人とした場合，被験者 B〜K の内の 9 名がシステム側で用意した被験者となるように評価した．被験者 B を本人とすると，その他の 9 名がシステム側で用意した被験者となり，被験者 C 以降も同様である．1 被験者 100 単語のうち 10 単語を訓練，検証用にし，残りの 90 単語をテスト用とする．結果を混同行列としてランダムフォレストの結果を表 8.11 に，全結合ニューラルネットワークの結果を表 8.12 に，畳み込みニューラルネットワークの結果を表 8.13 に示す．縦軸の A〜K が単語の入力を行った利用者であり，横軸がその単語が誰に分類されたかを表す．値は，単語数の 10 回の平均値と標準偏差である．

表 8.11　利用者本人における個人識別の混同行列 (ランダムフォレスト)

	A	B	C	D	E	F	G	H	I	J	K
A	81.0 (5.8)	2.6 (1.7)	0.1 (0.3)	2.9 (3.6)	0.0 (0.0)	0.0 (0.0)	3.2 (4.1)	0.0 (0.0)	0.2 (0.6)	0.0 (0.0)	0.0 (0.0)
B	1.5 (1.6)	79.0 (6.2)	0.8 (0.7)	4.2 (3.4)	0.3 (0.5)	0.0 (0.0)	0.5 (0.9)	0.0 (0.0)	1.1 (3.3)	2.2 (2.1)	0.4 (0.5)
C	0.0 (0.0)	0.0 (0.0)	85.0 (5.6)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.9 (2.7)	0.6 (1.2)	0.0 (0.0)	0.0 (0.0)	3.5 (4.6)
D	0.6 (0.5)	3.6 (2.2)	0.1 (0.3)	79.6 (4.6)	0.7 (0.5)	0.0 (0.0)	3.3 (4.0)	0.1 (0.3)	0.0 (0.0)	2.0 (2.4)	0.0 (0.0)
E	0.0 (0.0)	0.0 (0.0)	0.1 (0.3)	0.0 (0.0)	89.0 (1.8)	0.0 (0.0)	0.0 (0.0)	0.8 (1.5)	0.0 (0.0)	0.0 (0.0)	0.1 (0.3)
F	0.0 (0.0)	0.0 (0.0)	0.2 (0.4)	0.0 (0.0)	1.5 (3.1)	86.4 (4.4)	0.0 (0.0)	0.1 (0.3)	0.5 (0.5)	0.0 (0.0)	1.3 (1.6)
G	0.2 (0.6)	0.5 (0.9)	5.5 (2.7)	5.2 (5.8)	0.0 (0.0)	0.0 (0.0)	70.3 (4.7)	0.0 (0.0)	0.0 (0.0)	6.4 (4.1)	1.9 (2.3)
H	0.0 (0.0)	0.4 (0.5)	0.1 (0.3)	0.2 (0.4)	1.9 (3.6)	0.0 (0.0)	0.1 (0.3)	85.9 (3.9)	0.6 (0.8)	0.5 (0.7)	0.3 (0.6)
I	0.9 (0.7)	0.5 (0.7)	0.1 (0.3)	1.2 (1.5)	0.5 (0.7)	0.0 (0.0)	1.2 (0.7)	2.2 (1.7)	78.7 (2.9)	4.1 (2.9)	0.6 (1.0)
J	0.4 (0.9)	0.7 (1.2)	0.1 (0.3)	1.5 (2.6)	0.0 (0.0)	0.0 (0.0)	3.9 (3.3)	0.1 (0.3)	0.1 (0.3)	82.8 (3.6)	0.4 (0.7)
K	0.2 (0.6)	0.2 (0.4)	7.3 (5.5)	0.6 (1.0)	0.2 (0.6)	0.0 (0.0)	0.7 (0.6)	0.5 (0.8)	1.3 (1.7)	0.6 (0.7)	78.4 (4.9)

表 8.12　利用者本人における個人識別の混同行列 (全結合ニューラルネットワーク)

	A	B	C	D	E	F	G	H	I	J	K
A	84.6 (4.9)	0.1 (0.3)	0.9 (2.7)	1.3 (3.9)	1.5 (4.5)	0.2 (0.6)	0.5 (1.5)	0.2 (0.6)	0.1 (0.3)	0.2 (0.6)	0.4 (1.2)
B	0.4 (1.2)	86.7 (1.1)	0.3 (0.9)	0.3 (0.9)	0.2 (0.6)	0.2 (0.6)	0.3 (0.9)	0.6 (1.8)	0.3 (0.9)	0.4 (1.2)	0.3 (0.9)
C	0.2 (0.6)	0.0 (0.0)	88.8 (1.4)	0.1 (0.3)	0.0 (0.0)	0.1 (0.3)	0.1 (0.3)	0.5 (1.5)	0.1 (0.3)	0.1 (0.3)	0.0 (0.0)
D	0.1 (0.3)	0.2 (0.6)	0.8 (2.4)	86.3 (2.1)	0.4 (1.2)	0.6 (1.8)	0.4 (1.2)	0.2 (0.6)	0.3 (0.9)	0.2 (0.6)	0.5 (1.5)
E	0.0 (0.0)	0.0 (0.0)	0.1 (0.3)	0.0 (0.0)	89.7 (0.6)	0.0 (0.0)	0.0 (0.0)	0.2 (0.6)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
F	0.0 (0.0)	0.0 (0.0)	0.1 (0.3)	0.0 (0.0)	0.0 (0.0)	89.9 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
G	1.3 (3.9)	1.1 (3.3)	1.1 (3.3)	1.3 (3.9)	1.1 (3.3)	2.0 (6.0)	75.1 (3.8)	2.3 (6.9)	1.5 (4.5)	1.6 (4.8)	1.6 (4.8)
H	0.4 (1.2)	0.0 (0.0)	0.3 (0.9)	0.2 (0.6)	0.4 (1.2)	0.8 (2.4)	0.8 (2.4)	86.1 (2.7)	0.1 (0.3)	0.2 (0.6)	0.7 (2.1)
I	0.7 (2.1)	0.9 (2.7)	0.5 (1.5)	0.7 (2.1)	0.5 (1.5)	1.0 (3.0)	0.7 (2.1)	1.1 (3.3)	82.2 (1.9)	0.8 (2.4)	0.9 (2.7)
J	0.4 (1.2)	0.8 (2.4)	1.8 (5.4)	0.7 (2.1)	1.1 (3.3)	1.6 (4.8)	0.1 (0.3)	1.2 (3.6)	1.2 (3.6)	80.0 (4.9)	1.1 (3.3)
K	0.7 (2.1)	0.7 (2.1)	0.4 (1.2)	0.2 (0.6)	0.8 (2.4)	0.3 (0.9)	0.8 (2.4)	1.1 (3.3)	0.6 (1.8)	0.5 (1.5)	83.9 (2.5)

表 8.13　利用者本人における個人識別の混同行列 (畳み込みニューラルネットワーク)

	A	B	C	D	E	F	G	H	I	J	K
A	83.6 (5.0)	0.4 (1.2)	0.1 (0.3)	0.5 (1.5)	0.4 (1.2)	0.4 (1.2)	1.2 (3.6)	0.7 (2.1)	0.5 (1.5)	0.3 (0.9)	1.9 (5.7)
B	0.2 (0.6)	78.5 (9.4)	0.7 (2.1)	1.7 (5.1)	0.9 (2.7)	1.7 (5.1)	0.4 (1.2)	0.0 (0.0)	2.2 (6.6)	3.1 (9.3)	0.6 (1.8)
C	0.5 (1.5)	0.2 (0.6)	87.4 (2.8)	0.2 (0.6)	0.2 (0.6)	0.0 (0.0)	0.1 (0.3)	0.1 (0.3)	0.3 (0.9)	1.0 (3.0)	0.0 (0.0)
D	0.5 (1.5)	0.2 (0.6)	0.3 (0.9)	82.3 (5.3)	0.4 (1.2)	0.5 (1.5)	1.5 (4.5)	1.7 (5.1)	0.6 (1.8)	0.5 (1.5)	1.5 (4.5)
E	0.1 (0.3)	0.0 (0.0)	0.0 (0.0)	0.1 (0.3)	89.6 (0.7)	0.0 (0.0)	0.0 (0.0)	0.2 (0.6)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
F	0.0 (0.0)	0.1 (0.3)	0.0 (0.0)	0.3 (0.9)	0.2 (0.6)	88.4 (2.1)	0.2 (0.6)	0.0 (0.0)	0.1 (0.3)	0.0 (0.0)	0.7 (2.1)
G	1.8 (5.4)	1.4 (4.2)	2.4 (7.2)	1.8 (5.4)	2.4 (7.2)	2.6 (7.8)	68.7 (5.6)	3.4 (10.2)	2.1 (6.3)	1.8 (5.4)	1.6 (4.8)
H	0.5 (1.5)	0.9 (2.7)	0.5 (1.5)	0.3 (0.9)	0.6 (1.8)	0.4 (1.2)	0.6 (1.8)	85.0 (1.8)	0.4 (1.2)	0.6 (1.8)	0.2 (0.6)
I	0.9 (2.7)	0.9 (2.7)	0.9 (2.7)	0.7 (2.1)	0.7 (2.1)	0.4 (1.2)	1.1 (3.3)	0.6 (1.8)	82.2 (1.9)	0.9 (2.7)	0.7 (2.1)
J	1.3 (3.9)	2.0 (6.0)	3.2 (9.6)	1.2 (3.6)	2.7 (8.1)	1.7 (5.1)	0.5 (1.5)	1.2 (3.6)	1.2 (3.6)	73.0 (7.6)	2.0 (6.0)
K	0.8 (2.4)	0.3 (0.9)	0.9 (2.7)	0.5 (1.5)	0.3 (0.9)	0.6 (1.8)	0.5 (1.5)	0.8 (2.4)	0.3 (0.9)	0.4 (1.2)	84.6 (2.2)

表 8.11，8.12，8.13 において，例えば被験者 A が A として分類されても，単語の Recall が閾値を超えない限り個人識別は成功しない．閾値を 0.1 から 0.9 の間とした際に，利用者本人に分類された単語の内，閾値を超えた単語数の平均値と標準偏差を表 8.14，8.15，8.16 に示す．90 単語の内，この閾値を超えた単語数の割合が 1 - FRR の値となる．FRR の平均値と標準偏差も表 8.14，8.15，8.16 に示す．

表 8.14　閾値ごとの利用者本人が本人確認に成功した単語数と FRR(ランダムフォレスト)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	81.0 (6.1)	81.0 (6.1)	81.0 (6.1)	81.0 (6.1)	81.0 (6.1)	80.7 (5.9)	80.1 (6.2)	78.7 (6.7)	72.9 (7.3)
B	79.0 (6.6)	79.0 (6.6)	79.0 (6.6)	79.0 (6.6)	79.0 (6.6)	77.8 (6.6)	74.8 (5.9)	68.0 (8.2)	56.2 (8.8)
C	85.0 (5.9)	85.0 (5.9)	85.0 (5.9)	85.0 (5.9)	85.0 (5.9)	84.9 (6.0)	84.0 (6.5)	78.9 (10.5)	68.7 (13.7)
D	79.6 (4.8)	79.6 (4.8)	79.6 (4.8)	79.6 (4.8)	79.6 (4.8)	78.9 (5.0)	75.0 (5.5)	63.8 (8.5)	44.7 (11.9)
E	89.0 (1.9)	89.0 (1.9)	89.0 (1.9)	89.0 (1.9)	89.0 (1.9)	89.0 (1.9)	88.8 (2.0)	87.0 (3.0)	80.8 (4.8)
F	86.4 (4.6)	86.4 (4.6)	86.4 (4.6)	86.4 (4.6)	86.3 (4.6)	86.3 (4.6)	85.4 (5.3)	82.9 (5.6)	77.0 (8.3)
G	70.3 (4.9)	70.3 (4.9)	70.3 (4.9)	70.3 (4.9)	70.2 (5.0)	69.2 (4.8)	65.3 (7.3)	55.9 (10.2)	40.9 (12.4)
H	85.9 (4.1)	85.9 (4.1)	85.9 (4.1)	85.9 (4.1)	85.8 (4.1)	85.1 (4.2)	83.5 (4.6)	80.5 (5.3)	72.1 (8.1)
I	78.7 (3.0)	78.7 (3.0)	78.7 (3.0)	78.7 (3.0)	78.6 (3.1)	77.9 (3.4)	76.7 (3.6)	73.6 (4.1)	65.3 (5.3)
J	82.8 (3.8)	82.8 (3.8)	82.8 (3.8)	82.8 (3.8)	82.7 (3.7)	81.6 (4.5)	78.8 (3.8)	72.6 (4.5)	58.1 (6.8)
K	78.4 (5.1)	78.4 (5.1)	78.4 (5.1)	78.4 (5.1)	78.3 (5.1)	77.6 (5.9)	74.5 (6.7)	63.7 (10.0)	44.1 (11.9)
平均	81.5 (4.6)	81.5 (4.6)	81.5 (4.6)	81.5 (4.6)	81.4 (4.6)	80.8 (4.8)	78.8 (5.2)	73.2 (7.0)	61.9 (9.0)
FRR	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.102 (0.06)	0.124 (0.07)	0.186 (0.1)	0.312 (0.15)

表 8.15　閾値ごとの利用者本人が本人確認に成功した単語数と FRR(全結合ニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	84.6 (5.1)	84.6 (5.1)	84.6 (5.1)	84.6 (5.1)	84.6 (5.1)	84.6 (5.1)	83.9 (5.3)	81.5 (5.1)	77.8 (4.7)
B	86.7 (1.2)	86.7 (1.2)	86.7 (1.2)	86.7 (1.2)	86.7 (1.2)	86.7 (1.2)	86.2 (1.4)	81.0 (2.6)	61.0 (4.9)
C	88.8 (1.5)	88.8 (1.5)	88.8 (1.5)	88.8 (1.5)	88.8 (1.5)	88.8 (1.5)	88.3 (2.1)	86.0 (4.3)	76.7 (8.1)
D	86.3 (2.2)	86.3 (2.2)	86.3 (2.2)	86.3 (2.2)	86.3 (2.2)	86.3 (2.2)	85.2 (3.6)	76.6 (7.8)	55.4 (12.0)
E	89.7 (0.7)	89.7 (0.7)	89.7 (0.7)	89.7 (0.7)	89.7 (0.7)	89.7 (0.7)	89.7 (0.7)	88.7 (1.8)	83.9 (4.0)
F	89.9 (0.3)	89.9 (0.3)	89.9 (0.3)	89.9 (0.3)	89.9 (0.3)	89.9 (0.3)	89.9 (0.3)	89.8 (0.6)	89.2 (1.0)
G	75.1 (4.0)	75.1 (4.0)	75.1 (4.0)	75.1 (4.0)	75.1 (4.0)	75.1 (4.0)	71.9 (4.6)	57.1 (8.2)	31.0 (7.1)
H	86.1 (2.9)	86.1 (2.9)	86.1 (2.9)	86.1 (2.9)	86.1 (2.9)	86.1 (2.9)	85.0 (3.1)	79.3 (4.0)	69.4 (6.0)
I	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.1 (2.0)	79.7 (2.7)	73.2 (3.3)
J	80.0 (5.2)	80.0 (5.2)	80.0 (5.2)	80.0 (5.2)	80.0 (5.2)	80.0 (5.2)	77.1 (6.5)	60.2 (14.1)	28.0 (10.9)
K	83.9 (2.7)	83.9 (2.7)	83.9 (2.7)	83.9 (2.7)	83.9 (2.7)	83.9 (2.7)	82.8 (2.6)	73.1 (5.2)	54.2 (9.7)
平均	84.8 (2.5)	84.8 (2.5)	84.8 (2.5)	84.8 (2.5)	84.8 (2.5)	84.8 (2.5)	83.8 (2.9)	77.5 (5.1)	63.6 (6.5)
FRR	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.069 (0.06)	0.138 (0.11)	0.293 (0.21)

表 8.16　閾値ごとの利用者本人が本人確認に成功した単語数と FRR(畳み込みニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	82.8 (5.0)	78.7 (5.1)	69.1 (5.8)
B	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	75.2 (12.5)	63.5 (16.0)	39.0 (17.0)
C	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	86.8 (3.7)	83.6 (5.8)	70.6 (12.3)
D	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	80.2 (7.2)	68.9 (13.9)	44.8 (17.2)
E	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	88.5 (2.5)	82.3 (5.7)
F	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.1 (2.2)	86.8 (2.5)	82.9 (5.0)
G	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	64.4 (7.7)	48.7 (12.8)	22.5 (9.9)
H	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	83.6 (2.8)	76.6 (4.7)	61.2 (7.6)
I	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	81.6 (1.9)	78.5 (1.4)	67.6 (5.8)
J	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	69.0 (9.1)	49.5 (12.7)	22.7 (12.1)
K	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	83.2 (2.5)	73.3 (5.1)	49.3 (9.1)
平均	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	80.4 (5.0)	72.4 (7.5)	55.6 (9.8)
FRR	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.107 (0.08)	0.195 (0.14)	0.382 (0.23)

8.4.2　システムへの攻撃に対する評価

8.4.2.1　未登録者の攻撃に対する評価

実際にこのシステムを利用する場合，登録者ではないのにも関わらず登録者のふりをして検証を行ったり，登録者の筆記を真似し，なりすまして検証を突破される可能性がある．そこで，攻撃者が利用者として分類されるかどうかの評価を行った．仕様が 6.3 節の通りとすると，k 値分類における k 名の内の 1 名が利用者本人，残りの 9 名がシステム側で用意した被験者ということになり，攻撃者は当然これ以外の人物となる．内部犯がおり，システム側で用意した被験者が攻撃者となる可能性もあるが，通常は信頼の置ける人物を選別するため，本評価においてはこれを考慮しないものとする．訓練および検証は 8.3 節のものをそのまま利用し，例えば被験者 A を本人とし

た場合，被験者 B〜K の内の 9 名がシステム側で用意した被験者となるように評価した．そして，被験者 B〜K の内の残り 1 名が攻撃者となるようにした．このようにすることで，被験者 11 名を使って，攻撃者が 11 名いる場合の評価を行える．攻撃者の単語は当然システムに訓練していないため，攻撃者の 100 単語すべてをテストとして使用する．なお攻撃者は本人ではないため攻撃に成功したものは FP となるが，ここでは TP として計算する．3 つの機械学習で検証した結果を混同行列として表 8.17, 8.18，8.19 に示す．縦軸の A〜K が単語の入力を行った攻撃者であり，横軸がその単語が誰に分類されたかを表す．値は，単語数の 10 回の平均値と標準偏差である．

表 8.17　攻撃者における個人識別の混同行列 (ランダムフォレスト)

	A	B	C	D	E	F	G	H	I	J	K
A	1.0 (1.8)	8.0 (17.6)	9.1 (23.4)	13.6 (23.7)	13.3 (27.7)	0.5 (1.2)	16.1 (30.1)	10.7 (29.8)	3.5 (7.6)	7.6 (10.2)	16.6 (30.5)
B	0.9 (1.8)	8.4 (17.5)	9.1 (23.4)	10.1 (19.1)	13.3 (27.7)	0.4 (1.2)	23.3 (34.0)	10.6 (29.8)	1.0 (1.2)	7.2 (10.4)	15.7 (30.9)
C	1.0 (1.8)	8.3 (17.5)	9.1 (23.4)	15.8 (23.3)	13.0 (27.9)	0.5 (1.2)	23.4 (33.9)	10.7 (29.8)	3.4 (7.6)	7.6 (10.2)	7.2 (16.5)
D	1.0 (1.8)	6.8 (17.5)	9.1 (23.4)	15.8 (23.3)	13.3 (27.7)	0.5 (1.2)	16.8 (31.3)	10.7 (29.8)	3.6 (7.6)	5.8 (9.8)	16.6 (30.5)
E	1.0 (1.8)	8.4 (17.5)	9.1 (23.4)	15.8 (23.3)	13.3 (27.7)	0.5 (1.2)	23.4 (33.9)	0.7 (1.0)	3.6 (7.6)	7.6 (10.2)	16.6 (30.5)
F	1.0 (1.8)	8.4 (17.5)	8.7 (23.6)	15.8 (23.3)	9.9 (27.1)	0.5 (1.2)	23.4 (33.9)	10.4 (29.9)	3.3 (7.6)	7.6 (10.2)	11.0 (27.8)
G	0.5 (1.2)	8.4 (17.5)	8.3 (23.6)	9.5 (17.5)	13.3 (27.7)	0.5 (1.2)	23.4 (33.9)	10.7 (29.8)	3.6 (7.6)	5.3 (9.0)	16.5 (30.6)
H	1.0 (1.8)	8.3 (17.5)	9.1 (23.4)	15.8 (23.3)	4.2 (10.0)	0.5 (1.2)	23.4 (33.9)	10.7 (29.8)	3.5 (7.6)	7.2 (10.4)	16.3 (30.7)
I	0.6 (1.5)	2.5 (4.7)	9.1 (23.4)	15.8 (23.3)	12.9 (27.9)	0.5 (1.2)	23.3 (34.0)	10.5 (29.8)	3.6 (7.6)	4.9 (8.0)	16.3 (30.7)
J	1.0 (1.8)	8.4 (17.5)	9.1 (23.4)	14.3 (23.8)	13.3 (27.7)	0.5 (1.2)	14.9 (27.4)	10.7 (29.8)	3.6 (7.6)	7.6 (10.2)	16.6 (30.5)
K	1.0 (1.8)	8.1 (17.6)	1.2 (2.6)	15.7 (23.4)	13.2 (27.8)	0.1 (0.3)	22.6 (34.4)	10.6 (29.8)	3.3 (7.6)	7.6 (10.2)	16.6 (30.5)

表 8.18　攻撃者における個人識別の混同行列 (全結合ニューラルネットワーク)

	A	B	C	D	E	F	G	H	I	J	K
A	2.9 (3.6)	9.4 (28.2)	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	9.2 (27.6)	9.6 (28.8)	10.0 (30.0)	9.0 (27.0)	10.0 (30.0)	10.0 (30.0)
B	8.9 (26.7)	6.6 (8.9)	10.0 (30.0)	7.2 (21.6)	10.0 (30.0)	9.9 (29.7)	9.9 (29.7)	10.0 (30.0)	9.0 (27.0)	10.0 (30.0)	8.5 (25.5)
C	10.0 (30.0)	10.0 (30.0)	6.8 (18.5)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	9.5 (28.5)	10.0 (30.0)	10.0 (30.0)	9.9 (29.7)	3.8 (11.4)
D	9.2 (27.6)	7.6 (22.8)	10.0 (30.0)	3.5 (7.2)	10.0 (30.0)	10.0 (30.0)	9.8 (29.4)	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)
E	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	11.4 (22.8)	8.2 (24.6)	10.0 (30.0)	2.4 (7.2)	8.0 (24.0)	10.0 (30.0)	10.0 (30.0)
F	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	9.4 (28.2)	0.7 (1.8)	10.0 (30.0)	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)
G	9.3 (27.9)	9.9 (29.7)	10.0 (30.0)	9.2 (27.6)	10.0 (30.0)	9.6 (28.8)	10.7 (25.3)	9.9 (29.7)	10.0 (30.0)	1.4 (4.2)	10.0 (30.0)
H	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	2.9 (8.7)	10.0 (30.0)	10.0 (30.0)	7.3 (21.2)	9.8 (29.4)	10.0 (30.0)	10.0 (30.0)
I	9.1 (27.3)	9.5 (28.5)	10.0 (30.0)	10.0 (30.0)	9.5 (28.5)	10.0 (30.0)	10.0 (30.0)	9.9 (29.7)	2.0 (3.0)	10.0 (30.0)	10.0 (30.0)
J	9.5 (28.5)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	5.0 (15.0)	10.0 (30.0)	9.8 (29.4)	5.7 (14.8)	10.0 (30.0)
K	10.0 (30.0)	10.0 (30.0)	0.0 (0.0)	10.0 (30.0)	10.0 (30.0)	6.5 (19.5)	9.7 (29.1)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	13.8 (30.5)

表 8.19　攻撃者における個人識別の混同行列 (畳み込みニューラルネットワーク)

	A	B	C	D	E	F	G	H	I	J	K
A	6.5 (14.2)	9.5 (28.5)	10.0 (30.0)	9.0 (27.0)	10.0 (30.0)	10.0 (30.0)	9.8 (29.4)	10.0 (30.0)	5.2 (15.6)	10.0 (30.0)	10.0 (30.0)
B	7.7 (23.1)	6.4 (7.6)	10.0 (30.0)	9.4 (28.2)	10.0 (30.0)	8.9 (26.7)	8.6 (25.8)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	9.0 (27.0)
C	10.0 (30.0)	10.0 (30.0)	6.4 (17.6)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	9.5 (28.5)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	4.1 (12.3)
D	9.4 (28.2)	8.3 (24.9)	10.0 (30.0)	2.6 (5.1)	10.0 (30.0)	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)	9.8 (29.4)	10.0 (30.0)
E	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)	5.7 (11.4)	6.4 (19.2)	10.0 (30.0)	8.2 (24.6)	9.9 (29.7)	10.0 (30.0)	9.9 (29.7)
F	10.0 (30.0)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	0.1 (0.3)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)
G	9.9 (29.7)	9.9 (29.7)	7.3 (21.9)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.4 (22.6)	9.9 (29.7)	10.0 (30.0)	2.6 (7.8)	10.0 (30.0)
H	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	0.4 (1.2)	10.0 (30.0)	10.0 (30.0)	10.3 (28.6)	9.4 (28.2)	10.0 (30.0)	9.9 (29.7)
I	8.3 (24.9)	9.9 (29.7)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	9.3 (27.9)	2.5 (5.3)	10.0 (30.0)	10.0 (30.0)
J	10.0 (30.0)	10.0 (30.0)	10.0 (30.0)	7.9 (23.7)	10.0 (30.0)	10.0 (30.0)	5.9 (17.7)	10.0 (30.0)	10.0 (30.0)	6.2 (13.2)	10.0 (30.0)
K	10.0 (30.0)	10.0 (30.0)	6.4 (19.2)	10.0 (30.0)	10.0 (30.0)	8.9 (26.7)	10.0 (30.0)	8.6 (25.8)	9.9 (29.7)	10.0 (30.0)	6.2 (11.1)

表 8.17，8.18，8.19 において，例えば攻撃者が A に分類されても，Recall が閾値を超えない限り個人識別は成功しない．閾値を 0.1 から 0.9 の間とした際に，攻撃者が誰かに分類された単語の内，閾値を超えた単語数の平均値と標準偏差と FAR の平均値と標準偏差を表 8.20，表 8.21，表 8.22 に示す．縦軸の A〜K が単語の入力を行った攻撃者であり，横軸が閾値を表す．

100 単語の内，ある利用者に分類され，さらにこの閾値を超えた単語数の割合が FAR の値となる．

表 8.20　閾値ごとの攻撃者が本人確認に成功した単語数と FAR(ランダムフォレスト)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	1.0 (1.9)	1.0 (1.9)	1.0 (1.9)	1.0 (1.9)	0.5 (1.0)	0.3 (0.5)	0.2 (0.4)	0.2 (0.4)	0.0 (0.0)
B	8.4 (18.4)	8.4 (18.4)	8.4 (18.4)	7.5 (17.3)	6.5 (15.2)	5.0 (11.0)	3.7 (7.8)	1.9 (3.5)	1.2 (2.1)
C	9.1 (24.7)	9.1 (24.7)	9.1 (24.7)	8.8 (24.1)	8.1 (22.9)	7.3 (21.1)	6.2 (18.6)	5.5 (16.4)	4.2 (12.9)
D	15.8 (24.6)	15.8 (24.6)	15.8 (24.6)	15.6 (24.5)	14.2 (23.4)	10.4 (18.0)	6.9 (12.7)	3.9 (8.0)	1.5 (3.2)
E	13.3 (29.2)	13.3 (29.2)	13.3 (29.2)	12.8 (28.8)	11.9 (27.9)	10.4 (26.3)	9.4 (25.5)	8.7 (23.6)	6.5 (18.2)
F	0.5 (1.3)	0.5 (1.3)	0.4 (1.0)	0.1 (0.3)	0.1 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
G	23.4 (35.7)	23.4 (35.7)	23.3 (35.8)	22.8 (35.7)	20.9 (33.3)	17.6 (28.0)	13.7 (22.2)	9.0 (14.4)	4.1 (7.0)
H	10.7 (31.4)	10.7 (31.4)	10.6 (31.4)	10.5 (31.1)	10.4 (31.1)	10.1 (30.9)	9.4 (29.0)	8.9 (27.4)	7.1 (22.1)
I	3.6 (8.0)	3.6 (8.0)	3.5 (7.6)	3.2 (7.4)	2.2 (5.2)	1.6 (4.1)	0.5 (1.3)	0.1 (0.3)	0.0 (0.0)
J	7.6 (10.7)	7.6 (10.7)	7.6 (10.7)	7.1 (10.5)	6.6 (9.9)	5.7 (8.5)	4.1 (6.0)	3.0 (4.5)	1.6 (2.4)
K	16.6 (32.2)	16.6 (32.2)	16.6 (32.2)	15.8 (31.7)	13.6 (29.6)	12.2 (28.4)	11.5 (27.8)	10.1 (26.4)	8.7 (24.5)
平均	10.0 (19.8)	10.0 (19.8)	10.0 (19.8)	9.6 (19.4)	8.6 (18.2)	7.3 (16.1)	6.0 (13.8)	4.7 (11.4)	3.2 (8.4)
FAR	0.111 (0.07)	0.111 (0.07)	0.111 (0.07)	0.106 (0.07)	0.096 (0.07)	0.081 (0.06)	0.066 (0.05)	0.052 (0.04)	0.035 (0.03)

表 8.21　閾値ごとの攻撃者が本人確認に成功した単語数と FAR(全結合ニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	2.9 (3.8)	2.9 (3.8)	2.9 (3.8)	2.9 (3.8)	2.9 (3.8)	1.6 (2.1)	0.5 (0.8)	0.3 (0.7)	0.1 (0.3)
B	6.6 (9.4)	6.6 (9.4)	6.6 (9.4)	6.6 (9.4)	6.6 (9.4)	4.4 (6.7)	2.4 (4.1)	1.0 (1.9)	0.5 (0.7)
C	6.8 (19.5)	6.8 (19.5)	6.8 (19.5)	6.8 (19.5)	6.8 (19.5)	5.1 (15.1)	4.2 (12.9)	2.5 (7.6)	1.1 (3.5)
D	3.5 (7.6)	3.5 (7.6)	3.5 (7.6)	3.5 (7.6)	3.5 (7.6)	1.9 (4.5)	0.7 (1.9)	0.2 (0.6)	0.1 (0.3)
E	11.4 (24.0)	11.4 (24.0)	11.4 (24.0)	11.4 (24.0)	11.4 (24.0)	8.1 (19.3)	5.7 (15.3)	3.6 (10.4)	1.5 (4.4)
F	0.7 (1.9)	0.7 (1.9)	0.7 (1.9)	0.7 (1.9)	0.7 (1.9)	0.3 (0.9)	0.1 (0.3)	0.1 (0.3)	0.0 (0.0)
G	10.7 (26.6)	10.7 (26.6)	10.7 (26.6)	10.7 (26.6)	10.7 (26.6)	7.8 (21.9)	4.7 (14.9)	2.2 (7.0)	0.3 (0.9)
H	7.3 (22.4)	7.3 (22.4)	7.3 (22.4)	7.3 (22.4)	7.3 (22.4)	6.1 (18.9)	4.7 (14.5)	3.0 (9.1)	2.0 (6.0)
I	2.0 (3.2)	2.0 (3.2)	2.0 (3.2)	2.0 (3.2)	2.0 (3.2)	1.1 (1.9)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)
J	5.7 (15.6)	5.7 (15.6)	5.7 (15.6)	5.7 (15.6)	5.7 (15.6)	3.8 (11.0)	2.0 (6.0)	0.8 (2.5)	0.1 (0.3)
K	13.8 (32.2)	13.8 (32.2)	13.8 (32.2)	13.8 (32.2)	13.8 (32.2)	12.9 (31.7)	11.7 (30.8)	9.5 (27.6)	6.7 (20.8)
平均	6.5 (15.1)	6.5 (15.1)	6.5 (15.1)	6.5 (15.1)	6.5 (15.1)	4.8 (12.2)	3.4 (9.3)	2.1 (6.2)	1.1 (3.4)
FAR	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.054 (0.04)	0.037 (0.04)	0.023 (0.03)	0.013 (0.02)

表 8.22　閾値ごとの攻撃者が本人確認に成功した単語数と FAR(畳み込みニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	3.9 (9.7)	1.6 (4.1)	0.5 (1.3)	0.1 (0.3)
B	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	2.5 (3.3)	1.4 (2.4)	0.7 (1.3)	0.4 (0.7)
C	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	5.5 (16.7)	4.4 (13.6)	2.6 (8.2)	0.9 (2.8)
D	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	0.7 (1.9)	0.2 (0.6)	0.0 (0.0)	0.0 (0.0)
E	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	2.9 (6.7)	1.2 (3.1)	0.6 (1.3)	0.5 (1.0)
F	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
G	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	7.7 (19.0)	4.3 (11.6)	1.8 (5.0)	0.5 (1.6)
H	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	9.9 (29.9)	8.9 (28.1)	7.8 (24.7)	4.6 (14.5)
I	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	0.9 (2.0)	0.5 (1.3)	0.1 (0.3)	0.1 (0.3)
J	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	3.7 (9.0)	2.4 (6.6)	0.9 (2.5)	0.3 (0.9)
K	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	2.7 (5.5)	0.9 (2.0)	0.3 (0.9)	0.1 (0.3)
平均	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	3.7 (9.4)	2.3 (6.7)	1.4 (4.1)	0.7 (2.0)
FAR	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.041 (0.03)	0.026 (0.03)	0.015 (0.02)	0.008 (0.01)

8.4.2.2　録画攻撃に対する評価

新たに被験者を 4 人用意し，7.2.5 項の環境にて録画を行い，一人につき 100 単語を筆記させたときの動作を 1 単語ずつ録画した．更に，筆記した単語の名前も記録した．そこから 5 人の被験者を新たに用意し，4 人の動画をすべて見せ，全ての単語を模倣させた．模倣した人は 400 単語を書いたことになる．動画は何度も見直して良いとし，右肩からの視点，左肩からの視点好きな方の動画を見てかわまないようにした．

評価での登録者数は 10 人としているため，録画した 4 人と，8.2 節の 11 人のうち 6 人を入れた 10 人でモデルを作る．本項も前項までと同じように 3 つの機械学習を比較する．

モデルに登録されている録画された 4 人に対して，録画を見た 5 人が，検証を行う．録画した 5 人を V，W，X，Y，Z とし，登録者 A，B，C，D として検証した結果，

単語が本人として分類された数を表 8.23，8.24，8.25 に示す．この評価では 10 人のモデルのため，表に書かれていない登録者 E〜J に攻撃者 V〜Z の単語が分類されることがある．登録をしていない人が検証を行った結果は前項で行っているため，本項では省略する．

次に Recall を閾値とし，0.1 から 0.9 の間とした際に，攻撃者が誰かに分類された単語の内，閾値を超えた単語数の平均値と標準偏差と FAR の平均値と標準偏差を表 8.26，表 8.27，表 8.28 に示す．

表 8.23　覗き見攻撃者における個人識別の混同行列 (ランダムフォレスト)

	V	W	X	Y	Z
A	0 (0)	1.4 (2.3)	1.6 (3.2)	0.4 (0.5)	0 (0)
B	7.2 (11)	10.4 (12.9)	4 (2.8)	2.6 (3.2)	0 (0)
C	18.6 (37.2)	2.8 (5.6)	17.6 (35.2)	18.2 (24.6)	0 (0)
D	4.8 (7.7)	1 (0.9)	10.6 (19.3)	38.4 (47)	0 (0)

表 8.24　覗き見攻撃者における個人識別の混同行列 (全結合ニューラルネットワーク)

	V	W	X	Y	Z
A	0.6 (1.2)	0.4 (0.8)	0 (0)	0 (0)	0 (0)
B	2.4 (4.8)	14.6 (13.7)	0 (0)	0 (0)	3 (3.8)
C	5.2 (10.4)	1.2 (2.4)	0 (0)	6.6 (13.2)	0.8 (1.6)
D	12.2 (9)	1.2 (1.5)	0 (0)	6.4 (10.9)	0.4 (0.5)

表 8.25　覗き見攻撃者における個人識別の混同行列 (畳み込みニューラルネットワーク)

	V	W	X	Y	Z
A	1.4 (2)	6.6 (4.8)	21.2 (25.6)	17.4 (16.3)	0 (0)
B	0.4 (0.8)	10.6 (14.4)	0 (0)	0 (0)	0 (0)
C	0.2 (0.4)	0 (0)	1.4 (2.8)	0 (0)	0 (0)
D	0 (0)	0 (0)	2.8 (4.7)	0.4 (0.8)	0 (0)

表 8.26　閾値ごとの覗き見攻撃により本人確認を突破された単語数と FAR(ランダムフォレスト)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
B	10.4 (14.4)	10.4 (14.4)	10.4 (14.4)	7.0 (9.6)	4.0 (5.7)	2.2 (3.2)	1.2 (2.2)	0.0 (0.0)	0.0 (0.0)
C	17.6 (39.4)	17.6 (39.4)	17.6 (39.4)	17.4 (38.9)	16.2 (36.2)	15.8 (35.3)	15.0 (33.5)	13.4 (30.0)	8.4 (18.8)
D	38.4 (52.6)	38.4 (52.6)	38.4 (52.6)	38.4 (52.6)	37.2 (50.9)	32.0 (43.8)	28.8 (39.4)	22.8 (31.2)	16.8 (23.0)
平均	16.6 (26.6)	16.6 (26.6)	16.6 (26.6)	15.7 (25.3)	14.4 (23.2)	12.5 (20.6)	11.2 (18.8)	9.0 (15.3)	6.3 (10.4)
FAR	0.166 (0.14)	0.166 (0.14)	0.166 (0.14)	0.157 (0.14)	0.144 (0.14)	0.125 (0.13)	0.113 (0.12)	0.09 (0.1)	0.063 (0.07)

表 8.27　閾値ごとの覗き見攻撃により本人確認を突破された単語数と FAR(全結合ニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	0.6 (1.3)	0.6 (1.3)	0.6 (1.3)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
B	14.6 (15.4)	14.6 (15.4)	14.0 (15.5)	10.4 (12.7)	5.6 (7.8)	2.2 (3.0)	0.4 (0.5)	0.0 (0.0)	0.0 (0.0)
C	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
D	6.4 (12.2)	6.4 (12.2)	6.4 (12.2)	6.0 (11.3)	4.2 (8.8)	2.4 (4.8)	1.0 (2.2)	0.6 (1.3)	0.0 (0.0)
平均	5.4 (7.2)	5.4 (7.2)	5.2 (7.3)	4.2 (6.1)	2.4 (4.2)	1.2 (2.0)	0.4 (0.7)	0.2 (0.3)	0.0 (0.0)
FAR	0.054 (0.06)	0.054 (0.06)	0.053 (0.06)	0.042 (0.04)	0.024 (0.02)	0.012 (0.01)	0.004 (0.0)	0.002 (0.0)	0.0 (0.0)

表 8.28　閾値ごとの覗き見攻撃により本人確認を突破された単語数と FAR(畳み込みニューラルネットワーク)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	1.4 (2.2)	1.4 (2.2)	1.2 (1.8)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
B	10.6 (16.1)	10.6 (16.1)	9.4 (16.6)	7.8 (15.3)	5.2 (11.1)	2.6 (5.8)	0.8 (1.8)	0.0 (0.0)	0.0 (0.0)
C	1.4 (3.1)	1.4 (3.1)	1.4 (3.1)	0.4 (0.9)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
D	0.4 (0.9)	0.4 (0.9)	0.4 (0.9)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
平均	3.4 (5.6)	3.4 (5.6)	3.1 (5.6)	2.2 (4.3)	1.3 (2.8)	0.6 (1.5)	0.2 (0.4)	0.0 (0.0)	0.0 (0.0)
FAR	0.034 (0.04)	0.034 (0.04)	0.031 (0.04)	0.022 (0.03)	0.013 (0.02)	0.007 (0.01)	0.002 (0.0)	0.0 (0.0)	0.0 (0.0)

第 9 章

考察

9.1　機械学習の選択に関する考察

評価では，ランダムフォレスト，全結合ニューラルネットワーク，畳み込みニューラルネットワークの 3 つの機械学習を用いて個人識別の検証を行った．ここでは最終的にどの機械学習が優れていたのかの考察を行う．まず，3 つの機械学習の訓練結果を比較する．表 8.8，8.9，8.10 より，訓練の F 値はランダムフォレストの 0.997 が一番高かった．このため，ランダムフォレストが一番精度が良くなると考えられたが，実際に個人識別を行った結果，3 つの機械学習で閾値ごとの FRR をまとめたものを表 9.1 に，FAR をまとめたものを表 9.2 に示す．これ以降表のスペース削減のため，表内のランダムフォレストは RF，全結合ニューラルネットワークは FC，畳み込みニューラルネットワークは CNN と表す．

表 9.1　閾値ごとの利用者本人の FRR

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RF	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.095 (0.05)	0.102 (0.06)	0.124 (0.07)	0.186 (0.1)	0.312 (0.15)
FC	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.057 (0.05)	0.069 (0.06)	0.138 (0.11)	0.293 (0.21)
CNN	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.107 (0.08)	0.195 (0.14)	0.382 (0.23)

表 9.2　閾値ごとの攻撃者による FAR

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RF	0.111 (0.07)	0.111 (0.07)	0.111 (0.07)	0.106 (0.07)	0.096 (0.07)	0.081 (0.06)	0.066 (0.05)	0.052 (0.04)	0.035 (0.03)
FC	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.072 (0.04)	0.054 (0.04)	0.037 (0.04)	0.023 (0.03)	0.013 (0.02)
CNN	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.041 (0.03)	0.026 (0.03)	0.015 (0.02)	0.008 (0.01)

表 9.1 より，FRR は全結合ニューラルネットワークが一番低くなった．表 9.2 より，FAR は畳み込みニューラルネットワークが一番低くなった．ランダムフォレストは FRR，FAR ともにニューラルネットワークより悪い結果となった．

FAR を低くしたのであれば，畳み込みニューラルネットワーク，FRR を低くしたいのであれば，全結合ニューラルネットワークを使用するのが良いと考えられる．ランダムフォレストは 2 つのニューラルネットワークより精度が悪いが，CPU のみで計算ができるため，GPU の性能に期待できない環境であれば役に立つこともあると考えられる．

9.2　録画攻撃に対する考察

8.4.2.2 項では録画攻撃に対する評価を行ったが，本節では録画攻撃に対しての本システムの性能について考察する．録画攻撃の結果の表 8.26，8.27，8.28 の FAR をまとめたものを表 9.3 に示す．未登録の人を検証した FAR である表 9.2 と表 9.3 を比較した結果，録画攻撃であったとしても，FAR が大きく悪化しているわけではないため，筆記を真似したからといってもなりすましに成功する確率は上昇するわけではないということになる．

このことから本システムは覗き見攻撃に対して耐性があるといえる．

表 9.3　閾値ごとの覗き見攻撃による FAR

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RF	0.166 (0.14)	0.166 (0.14)	0.166 (0.14)	0.157 (0.14)	0.144 (0.14)	0.125 (0.13)	0.113 (0.12)	0.09 (0.1)	0.063 (0.07)
FC	0.054 (0.06)	0.054 (0.06)	0.053 (0.06)	0.042 (0.04)	0.024 (0.02)	0.012 (0.01)	0.004 (0.0)	0.002 (0.0)	0.0 (0.0)
CNN	0.034 (0.04)	0.034 (0.04)	0.031 (0.04)	0.022 (0.03)	0.013 (0.02)	0.007 (0.01)	0.002 (0.0)	0.0 (0.0)	0.0 (0.0)

9.3　安全性に関する考察

本節より，畳み込みニューラルネットワークと全結合ニューラルネットワークで考察を行う．提案システムは，パスワードによる個人識別を補助的に強化するために用いるため，1 回の個人識別フェーズにおいて 1 単語のみを入力する仕様としているが，入力に掛かる時間を考慮しなければ，筆記を比較して個人識別を行う研究には畠中らのもの [20]，Xiao らのもの [24]，Behera のもの [27]，Yamamoto らのもの [29] が存在する．

関連研究の FAR と FRR を見ると，畠中らのものが FAR が 0 の時 FRR が 6.2% と関連研究の中で一番良い結果となる．一方，8 章で行った提案システムの評価では FAR を 0 にすることはできなかったが，閾値を変更することにより FAR と FRR のバランスを変えることができる．この閾値は登録者ごとに設定可能であるため，表 9.1 および表 9.2 より，全結合ニューラルネットワークであれば，例えば，閾値を 0.5 にすれば FRR が 0.057 で FAR が 0.072 となるし，閾値を 0.7 にすれば FRR が 0.069 であるが FAR を 0.037 にできる．畳み込みニューラルネットワークの場合は，閾値を 0.5 にすれば FRR が 0.088 で FAR が 0.064 となるし，閾値を 0.7 にすれば FRR が 0.107 になるが，FAR を 0.026 にできる．

提案システムは，パスワードによる個人識別を補助的に強化するために使用することを目的としており，いかなる覗き見も可能であるとすると，そもそも入力しているパスワード自体も覗き見られていることになり危険である．利用環境としては，自宅などで背後から覗き見られないものを想定しており，公共の場においても，クレジットカードの PIN を入力したり，クレジットカードの署名をしたりする場合と同等程度の安全な環境であることを想定している．クレジットカードの PIN の場合，一度覗き見られてしまうと，同じ PIN を攻撃者に入力されてしまうが，提案システムにおいては，録画した映像の動きを真似る必要がある．クレジットカードの署名のように，書かれたものをそのままなぞって再現することもできない．さらに，入力を要求される単語が毎回異なるため，1 回から数回程度の覗き見には耐性がある．前節の録画攻撃に対する耐性から，仮に動きを真似したところで，攻撃を行うことは成功しないといえる．ただし，本論文の録画攻撃の実験では攻撃者は普通の大学生であったため，なりすまして筆記を行うことのプロの場合は突破されてしまう可能性があるが，実験に協力できそうななりすましのプロを探すことができないため，これは評価することができない．

9.4　筆記する単語の固定化に関する考察

本システムでは，好きな単語を書かせたときの，空中筆記時の特徴から本人確認を行っているが，利用者が入力する単語がシステムが要求したものでなくても，特徴が一致した場合，個人識別に成功してしまう．そのため，録画した 1 つの単語を長時間練習し続けた場合，いつかは突破できてしまうと考えられる．

そこで，Leap Motion で入力された数字を識別する Yamamoto らの研究や，タッチパネルの筆跡から文字を読み取る Hanyu[13] らの研究があるため，Leap Motion で書かれた文字の軌跡を平面化することによって，入力した文字を識別し，要求された単語と異なる単語が入力された場合に，システム側で自動的に拒否することは可能であると考えられる．

9.5　1 段階目の本人確認に関する考察

本システムは，本人確認を強化するものとして提案した．1 段階目の本人確認は既存のパスワードなどになるが，1 段階目の本人確認の後に Leap Motion を使って 2 段階となると多少手間がある．

そこで，私の 1 つ目の研究や，DTW を用いた関連研究のように，テンプレートマッチングを使用することで入力した単語と，登録した単語の比較を行うことにより，これを 1 段階目の本人確認として利用することができる．これにより，Leap Motion のみで 2 段階の本人確認を構築することができる．しかし，テンプレートマッチングや文字の識別によるパスワードは，テンプレートとなる単語を登録する必要がある．更に，毎回同じ単語を書くことになるため，多少の覗き見への耐性の低下が考えられる．しかし，登録する単語は必要に応じて変更できるため，定期的に単語を変更することで，覗き見耐性の低下を抑えることができると考えられる．登録する単語を変更するときに 2 段階目の本人確認つまり本システムの再訓練は必要ない．

9.6　本人拒否率に関する考察

9.6.1　連続で失敗できる回数に関する考察

本システムは，パスワードによる本人確認を補助的に強化することを目的としているため，FRR が 0 でない限り，パスワードも正しく筆記も適切に行えたにも関わらず，個人識別に失敗することが起きてしまう．8.4 節の評価では，FRR は 0 とはならなかった．全結合ニューラルネットワークの場合，閾値が 0.7 の時，FRR が 6.9% で

あった．そのため 2 回連続で失敗する確率は 0.48% となる．3 回連続では 0.03% となり，2 回目の時点でほぼ全ての人が本人確認に成功できると言える．そのため，FRR が 0 でなくとも，この FRR の低さであれば，1 回やり直すだけで良いため問題はないと考えられる．特に失敗できる回数を制限しない場合，長時間覗き見を受ける危険性があるため，やり直し回数は 1 回か 2 回に制限した方が良い．

9.6.2 筆記した単語の FRR への関係性に関する考察

被験者が入力した単語について調べたところ，極端にフレーム数が多いものと，極端に少ないものが存在した．極端にフレーム数が多い単語というのは，各被験者の一番文字数が多い単語のフレーム数の平均に標準偏差を足した値を超えたフレーム数のものを示す．全被験者の一番長い単語のフレーム数を平均すると 1106.818 となり，標準偏差は 395.695 となった．足し合わせて少数点を四捨五入した 1503 を超えるフレーム数がある単語は極端にフレーム数が多いということになる．その結果極端にフレーム数が多い単語は 19 個となった．

次に極端に少ないフレーム数を求めるため全被験者の 3 文字の単語のフレームの長さの平均を取り，標準偏差を平均から引いた値より少ないフレーム長の単語の数を求めた．全被験者の 3 文字の単語を書いた時のフレームの長さは 329.339 となった．この時の標準偏差は 140.979 となった．そのため平均から標準偏差を引いた値は 188.36 となる．小数点以下を四捨五入し，188 となる．長さがこれ以下のフレームの単語が短すぎるとして数えた結果，56 単語が極端にフレームが少ないということになった．

これは，筆記が完了した際に終了したと認識されていなかったり，筆記途中であるのに終了したと認識されてしまったりしたものであると推測される．この単語数の割合が 0.75% であり，これらが適切に除去されていれば，0.75%FRR を改善できる可能性がある．

9.7 検証に必要なデータサイズに関する考察

8.4 節における評価において，被験者 10 名分の訓練済みモデル（最適化情報含む）が約 3MB であった．単純に計算すると，100 名の利用者がいれば約 30MB，1 億人だとしても 30TB であり，一般に販売されている PC のハードディスク容量が 4〜8TB ほどであることを考慮すると，1 億人分の利用者情報がサーバに保存されていることに，実用上の不都合はないと思われる．

9.8　サンプル数に関する考察

8.2 節にて 1 つの筆記から 90，150，300，600，900，1500，3000，6000 のサンプルを抽出した．その結果，FAR，FRR の最良が 1500 となったが，ほかのサンプル数の場合に 1500 より悪かった理由について考察する．

90〜900 は単純にサンプル数が 1500 と比べて少ないので，訓練にて 1500 の時より個人の特徴を抽出できなかったのだと考えられる．FAR，FRR 自体は 90，150，300，600，900 の順で改善していっている．

3000 と 6000 の FAR，FRR が 1500 より悪い理由は，6.2 節にて筆記の分解を行ったときに，ソフトマックスの高い，直線になっているサンプルを抽出しているが，1500 ぐらいまでは直線になってるサンプルを集められるが，3000，6000 だと本来は捨てられるべき直線になっていないサンプルも集めてしまって，訓練時にノイズになってしまっていると考えられる．更に 3000 と 6000 は訓練と検証に 1500 サンプルの倍以上の時間がかかるため，実用性的にも 1500 が最適であるといえる．

9.9　閾値に関する考察

8 章の評価では閾値を決定しなかったが，実際にシステムとして利用する場合は Recall をどれぐらいの閾値にするのが最適か考察する．表 9.4 に攻撃者の Recall の平均値と標準偏差を示す．表 9.4 より全被験者の Recall の平均は 0.099 となる．そこに全被験者の Recall の標準偏差 0.078 を足すと 0.177 となる．攻撃者の Recall と標準偏差を考慮して 0.177 を閾値にした場合，FRR は 0.088 となるが，FAR が 0.064 となる．こうなる理由は，表 9.4 より攻撃者の Recall の平均は低いが，場所によっては Recall が非常に高くなる場所があり，そこが FAR を上げていると考えられる．例えば攻撃者 E の登録者 H の部分は 0.85 で，標準偏差を考慮すると 1.0 に到達してしまうため，すべての攻撃者 E の単語は登録者 H として分類されているということになる．これによりどれだけ閾値を上げてもすべての攻撃者をはじくことはできないといえる．

攻撃者の平均や攻撃者の最大に合わせて閾値を設定すると FAR と FRR はどちらかは強固になり，その反対は非常に悪化する．そのため，実際のシステムで利用する場合は環境や利用者に合わせて閾値を設定するのが良いと考えられる．このシステム自体はパスワードなどを補助するものなので，強固にするか手軽に使えるようにするかは個人の自由となる．

表 9.4　攻撃者の Recall の平均と標準偏差

登録者 / 攻撃者	A	B	C	D	E	F	G	H	I	J	K
A		0.35 (0.2)	0.0 (0.0)	0.2 (0.16)	0.0 (0.0)	0.0 (0.0)	0.1 (0.1)	0.0 (0.01)	0.25 (0.22)	0.07 (0.07)	0.02 (0.07)
B	0.19 (0.15)		0.01 (0.05)	0.33 (0.18)	0.02 (0.07)	0.08 (0.11)	0.12 (0.11)	0.01 (0.03)	0.1 (0.1)	0.03 (0.06)	0.11 (0.12)
C	0.0 (0.0)	0.01 (0.02)		0.1 (0.09)	0.0 (0.0)	0.0 (0.01)	0.41 (0.18)	0.02 (0.05)	0.0 (0.01)	0.01 (0.03)	0.45 (0.17)
D	0.29 (0.16)	0.19 (0.17)	0.0 (0.01)		0.0 (0.0)	0.0 (0.01)	0.13 (0.09)	0.03 (0.06)	0.01 (0.02)	0.33 (0.2)	0.02 (0.04)
E	0.0 (0.01)	0.0 (0.02)	0.0 (0.0)	0.0 (0.0)		0.04 (0.07)	0.0 (0.0)	0.85 (0.15)	0.07 (0.07)	0.0 (0.0)	0.03 (0.06)
F	0.02 (0.04)	0.29 (0.2)	0.01 (0.03)	0.0 (0.0)	0.43 (0.22)		0.0 (0.01)	0.0 (0.01)	0.01 (0.04)	0.0 (0.0)	0.23 (0.19)
G	0.11 (0.12)	0.26 (0.17)	0.05 (0.15)	0.1 (0.1)	0.0 (0.0)	0.01 (0.04)		0.01 (0.02)	0.01 (0.02)	0.43 (0.27)	0.02 (0.05)
H	0.0 (0.0)	0.03 (0.05)	0.01 (0.03)	0.03 (0.04)	0.3 (0.19)	0.06 (0.06)	0.04 (0.07)		0.25 (0.15)	0.01 (0.04)	0.28 (0.19)
I	0.46 (0.22)	0.07 (0.07)	0.0 (0.0)	0.02 (0.06)	0.07 (0.09)	0.04 (0.08)	0.02 (0.06)	0.19 (0.15)		0.04 (0.09)	0.1 (0.12)
J	0.06 (0.08)	0.01 (0.03)	0.02 (0.04)	0.22 (0.13)	0.0 (0.0)	0.0 (0.01)	0.63 (0.18)	0.02 (0.04)	0.03 (0.06)		0.02 (0.05)
K	0.0 (0.0)	0.21 (0.17)	0.58 (0.28)	0.01 (0.02)	0.02 (0.1)	0.08 (0.09)	0.04 (0.07)	0.06 (0.09)	0.01 (0.01)	0.01 (0.02)	
平均	0.11 (0.08)	0.14 (0.11)	0.07 (0.06)	0.1 (0.08)	0.08 (0.07)	0.03 (0.05)	0.15 (0.09)	0.12 (0.06)	0.07 (0.07)	0.09 (0.08)	0.13 (0.11)

9.10　登録単語数に関する考察

8 章の評価においては，筆記の特徴の訓練のために利用者が入力した単語数を仮に 10 と固定していた．本節では，何単語の入力が妥当であるかについて考察する．閾値は表 9.1, 表 9.2 より FAR と FRR が最も近い閾値 0.5 とする．このとき FAR:0.064, FRR:0.088 となる．この閾値にて単語数を変え，FAR と FRR がどれぐらい変化するのか考察する．入力する単語数と FRR と FAR を表 9.5 に示す．

表 9.5　登録単語数と本人拒否率と他人受け入れ率の関係

単語数	FAR	FRR
1	0.065	0.456
2	0.062	0.285
3	0.063	0.19
4	0.067	0.166
5	0.066	0.146
6	0.068	0.129
7	0.064	0.106
8	0.067	0.099
9	0.063	0.090
10	0.064	0.088

表 9.5 より，FAR は単語数 2 の時が一番小さいが，FRR は単語数 10 の時が一番小さい．よって，FAR，FRR 両方とも低い点から単語数 10 程度が妥当であるといえる．

単語数が増えていくほど FRR は改善していくが，FAR は変わっていない．訓練する単語数を増やしているので本人の特徴は抽出しやすくなるが，存在しない人の特徴は訓練できないので FAR はあまり下がらないと考えられる．

9.11　登録者数に関する考察

評価では 11 名の被験者しかいないが，サービスとして使用する場合には，利用者は 1 千万人や 1 億人になる場合もある．そこで，8.4 節の結果から分布を求め，被験者の数が増えた場合に FAR と FRR がどの程度になるか推測する．

図 9.1 に登録者を検証したときの Recall の分布のグラフを示す．このグラフは，横軸に Recall の範囲をとった場合に，その範囲にどれだけの単語数が含まれているかを示したものである．

なお，図 9.1 の横軸は Recall の範囲であり，例えば Recall が 0 のものは，実際には Recall が 0 以上 0.05 未満という意味である．Recall が 1 の場合を別扱いしたくなかったので，一番右の範囲のみ 0.95 以上 1 以下となるようにした．縦軸の右側は累積率である．

図 9.1 より，登録者を検証した結果の累積度分布は，なめらかな曲線を描いて上昇している．表 9.6 に 0.2 刻みの閾値ごとの累積率を示す．

表 9.6 より閾値 0.2 の場合，評価で行った 11 人以上の多人数，例えば 1 億人であれ

ば，240 万人が平均 1 回は本人確認をやり直すことになる．閾値 0.8 で 1 億人の場合は，3817 万人が平均 1 回は本人確認をやり直すことになり，その分検証するサーバに負荷がかかる．

図 9.2 に攻撃者を検証したときの Recall の分布のグラフを示す．表 9.7 に 0.2 刻みの閾値ごとの累積率を示す．

表 9.7 の累積率は，攻撃者にとっての攻撃失敗率に相当する．閾値が 0.2 の場合には 82.5% の確率で失敗し，閾値が 0.8 であれば 98.7% の確率で失敗することになる．

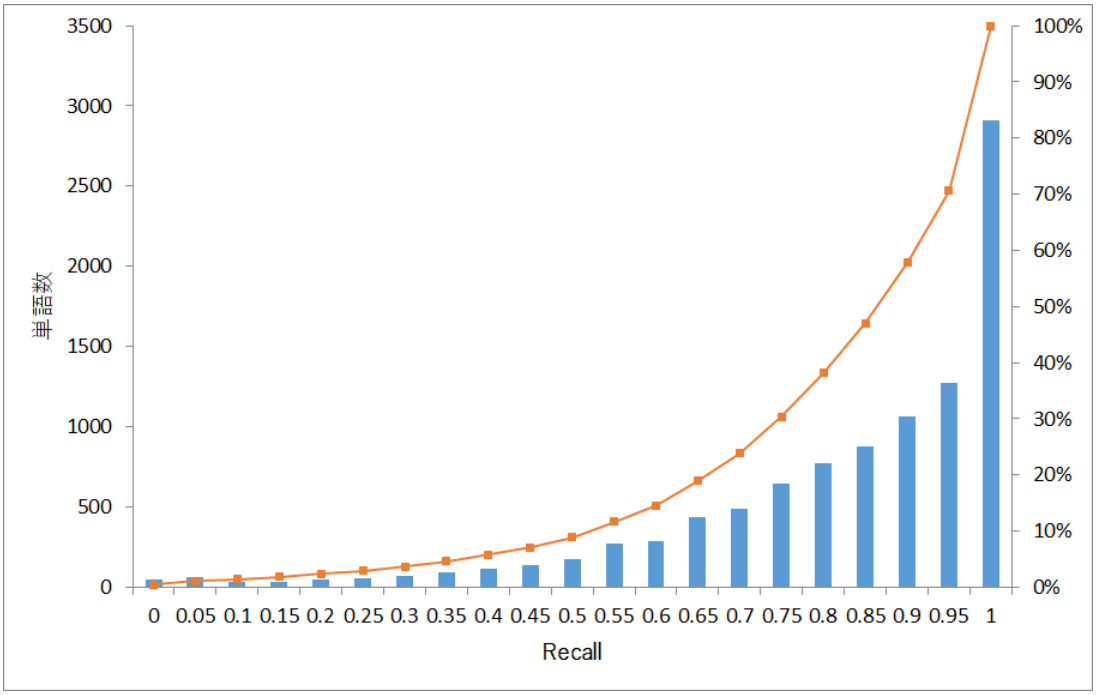

図 9.1　登録者を検証したときの Recall ごとの単語数の分布

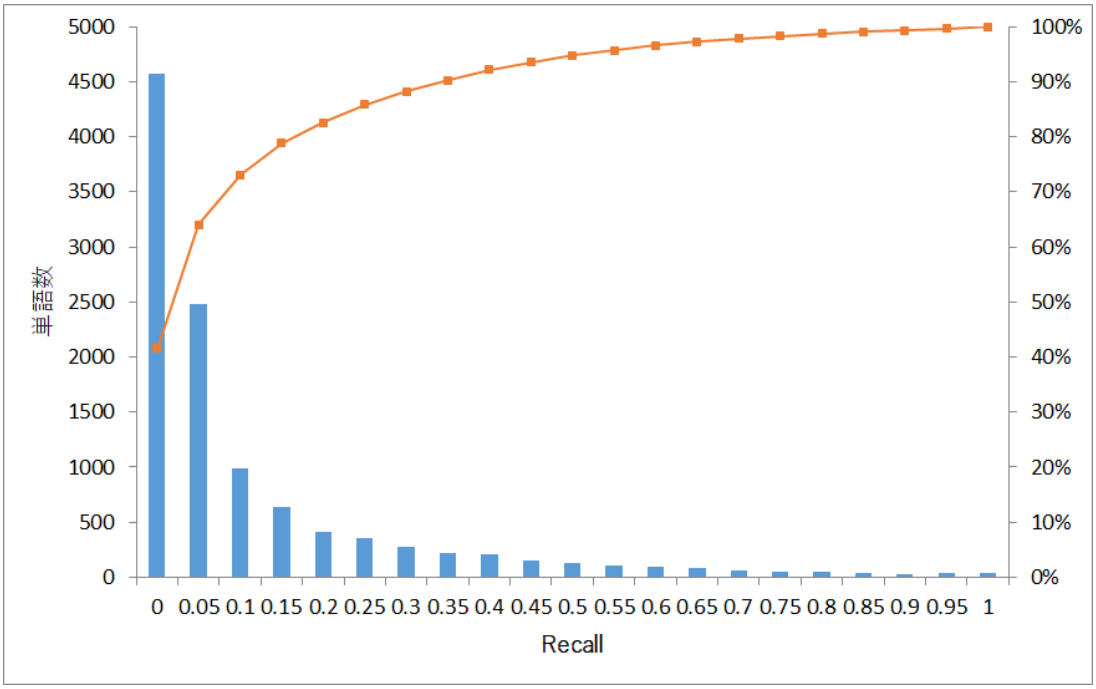

図 9.2　攻撃者を検証したときの Recall ごとの単語数の分布

表 9.6　登録者を検証した時の閾値ごとの累積率

閾値	累積率 [%]
0	0.505
0.2	2.364
0.4	5.758
0.6	14.535
0.8	38.172
1	100

表 9.7　攻撃者を検証した時の閾値ごとの累積率

閾値	累積率 [%]
0	41.527
0.2	82.518
0.4	92.145
0.6	96.527
0.8	98.682
1	100

もちろん，前述の考察は被験者 11 名のデータに基づいて行われたものであり，利用者が増えた場合にはこの 11 名のデータとは異なる特性を持つ者が現れる可能性は否定できない．例えば，FRR，FAR ともに低くできない利用者が現れるかもしれない．本システムは，パスワードによる個人識別を強化するために補助的に使用されるものであるため，利用者の中で適用が困難な者は，利用を推奨しないという選択もある．

本評価においては，FRR が妥当な範囲で FAR が 0 にならなかったが，低い FAR は攻撃を防止するだけでなく，攻撃抑止にもつながると考えられる．例えば，ホテルやレストランでの支払いのような公共の場所で利用する場合，暗証番号やパスワードを入手していたとしても，強引に本手法を突破するには相当な数の再筆記が必要となる．その間，係員や周囲の視線があることを考えれば，精神的なストレスはかなり大きいと推測される．

9.12 Recall に関する考察

8.4 節にて Recall を閾値にした場合の本人，攻撃者の検証から FRR と FAR を求めた．評価では Recall を使用したが，しかし，F 値は Precision と Recall の調和平均である．検証を行う人は本人だとしても，攻撃者だとしても，一人しかいないため TN と FP は存在しない．つまり式 (3.18) より Precision は常に 1 となる．そのため Recall より F 値を閾値にした方が，出力される数値が少し増えるため，FAR と FRR が改善する可能性がある．そこで表 8.16 と表 8.22 の結果を Recall ではなく，F 値を閾値に用いて求める．さらに表 9.4 についても F 値を用いて求める．その結果を表 9.8，表 9.9，表 9.10 に示す．

まず，FRR である表 8.16 と表 9.8 を比較すると，F 値，Recall の差がなく，まったく同じであった．前節の表 9.6 より，累積率が 0.8 から 1 の間が 61.828% のため，本人の検証では Recall，F 値は 0.8 を超えることが多く，0.1 から 0.9 までの間は結果が同じであったのではないかと考えられる．

次に FAR を比較する．表 8.22 と表 9.9 を比較すると，F 値，Recall ともに閾値 0.5 までは FAR が同じだが，閾値 0.6 から F 値，Recall の FAR に差が出てくるようになり，閾値に比例してその差も大きくなる．FAR は Recall のほうが小さくなる．

最後に表 9.4 と表 9.10 を比較する．F 値で数値を出した方が全員の平均で 0.144 となり，Recall で数値を出した時の 0.099 より値が大きくなる．F 値を閾値にすることで全体的に数値が大きくなったため，攻撃者や本人が本人として検証が通りやすくなると考えられる．

このような結果になった理由は Precision が関係していると考えられる．Precision は常に 1 で，F 値を計算するときに分母に 1 足されるため，F 値は Recall より少し高

くなる．高くなったことにより閾値を超えているかで本人か判定するときの判定が少し緩くなるため，FRR が下がると考えられる．逆に FAR が悪化するのも同様の理由だと考えられる．

このことより, FRR を下げたい場合は F 値を使用し, FAR を下げたい場合は Recall を使用するのが良いと考えられる．

表 9.8　閾値ごとの利用者本人が本人確認に成功した単語数と FRR(F 値)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	83.6 (5.3)	82.8 (5.0)	78.7 (5.1)	69.1 (5.8)
B	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	78.5 (9.9)	75.2 (12.5)	63.5 (16.0)	39.0 (17.0)
C	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	87.4 (3.0)	86.8 (3.7)	83.6 (5.8)	70.6 (12.3)
D	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	82.3 (5.6)	80.2 (7.2)	68.9 (13.9)	44.8 (17.2)
E	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	89.6 (0.7)	88.5 (2.5)	82.3 (5.7)
F	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.4 (2.2)	88.1 (2.2)	86.8 (2.5)	82.9 (5.0)
G	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	68.7 (5.9)	64.4 (7.7)	48.7 (12.8)	22.5 (9.9)
H	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	85.0 (1.9)	83.6 (2.8)	76.6 (4.7)	61.2 (7.6)
I	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	82.2 (2.0)	81.6 (1.9)	78.5 (1.4)	67.6 (5.8)
J	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	73.0 (8.0)	69.0 (9.1)	49.5 (12.7)	22.7 (12.1)
K	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	84.6 (2.3)	83.2 (2.5)	73.3 (5.1)	49.3 (9.1)
平均	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	82.1 (4.3)	80.4 (5.0)	72.4 (7.5)	55.6 (9.8)
FRR	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.088 (0.07)	0.107 (0.08)	0.195 (0.14)	0.382 (0.23)

表 9.9　閾値ごとの攻撃者が本人確認に成功した単語数と FAR(F 値)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
A	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	6.5 (14.9)	5.6 (13.4)	2.2 (5.6)	0.2 (0.4)
B	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	6.4 (8.0)	4.4 (5.7)	1.8 (2.6)	0.7 (1.3)
C	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.4 (18.5)	6.0 (17.3)	4.8 (14.8)	2.5 (7.9)
D	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	2.6 (5.4)	1.8 (3.9)	0.7 (1.9)	0.0 (0.0)
E	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	5.7 (12.0)	4.5 (9.5)	1.5 (3.5)	0.6 (1.3)
F	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.1 (0.3)	0.0 (0.0)	0.0 (0.0)
G	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	10.4 (23.9)	9.3 (21.8)	5.2 (13.3)	1.2 (3.5)
H	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.3 (30.2)	10.1 (30.2)	9.3 (29.1)	7.2 (22.8)
I	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	2.5 (5.5)	1.5 (3.5)	0.5 (1.3)	0.1 (0.3)
J	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	6.2 (13.9)	5.0 (12.2)	2.9 (7.6)	0.7 (1.9)
K	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	6.2 (11.7)	5.0 (10.2)	1.7 (4.1)	0.2 (0.6)
平均	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	5.8 (13.1)	4.8 (11.6)	2.8 (7.6)	1.2 (3.6)
FAR	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.064 (0.03)	0.054 (0.03)	0.031 (0.03)	0.014 (0.02)

表 9.10　攻撃者の F 値の平均と標準偏差

攻撃者＼登録者	A	B	C	D	E	F	G	H	I	J	K
A		0.49 (0.22)	0.0 (0.0)	0.3 (0.21)	0.0 (0.01)	0.0 (0.0)	0.18 (0.15)	0.0 (0.02)	0.36 (0.27)	0.12 (0.12)	0.03 (0.1)
B	0.29 (0.2)		0.02 (0.07)	0.47 (0.2)	0.03 (0.1)	0.13 (0.16)	0.21 (0.16)	0.02 (0.05)	0.17 (0.15)	0.05 (0.09)	0.18 (0.18)
C	0.0 (0.01)	0.02 (0.04)		0.16 (0.13)	0.0 (0.0)	0.0 (0.01)	0.56 (0.19)	0.04 (0.07)	0.0 (0.01)	0.01 (0.05)	0.6 (0.18)
D	0.43 (0.19)	0.29 (0.21)	0.01 (0.03)		0.0 (0.0)	0.0 (0.02)	0.22 (0.14)	0.05 (0.09)	0.01 (0.03)	0.47 (0.22)	0.03 (0.07)
E	0.01 (0.02)	0.01 (0.03)	0.0 (0.0)	0.0 (0.0)		0.08 (0.11)	0.0 (0.01)	0.91 (0.11)	0.12 (0.12)	0.0 (0.0)	0.05 (0.1)
F	0.04 (0.07)	0.42 (0.22)	0.02 (0.06)	0.0 (0.01)	0.57 (0.22)		0.0 (0.02)	0.01 (0.02)	0.02 (0.06)	0.0 (0.0)	0.33 (0.25)
G	0.18 (0.17)	0.39 (0.21)	0.08 (0.19)	0.17 (0.16)	0.0 (0.01)	0.01 (0.06)		0.02 (0.04)	0.02 (0.04)	0.55 (0.28)	0.04 (0.09)
H	0.0 (0.0)	0.05 (0.08)	0.01 (0.05)	0.06 (0.07)	0.43 (0.23)	0.1 (0.1)	0.06 (0.11)		0.37 (0.18)	0.02 (0.07)	0.4 (0.23)
I	0.6 (0.23)	0.12 (0.11)	0.0 (0.01)	0.03 (0.09)	0.12 (0.14)	0.07 (0.12)	0.03 (0.09)	0.29 (0.19)		0.06 (0.13)	0.16 (0.17)
J	0.11 (0.12)	0.02 (0.05)	0.03 (0.07)	0.34 (0.17)	0.0 (0.01)	0.0 (0.02)	0.75 (0.15)	0.03 (0.06)	0.04 (0.09)		0.04 (0.08)
K	0.0 (0.01)	0.31 (0.22)	0.69 (0.26)	0.02 (0.04)	0.03 (0.11)	0.14 (0.14)	0.07 (0.11)	0.09 (0.14)	0.01 (0.03)	0.01 (0.04)	
平均	0.17 (0.1)	0.21 (0.14)	0.09 (0.07)	0.15 (0.11)	0.12 (0.08)	0.05 (0.07)	0.21 (0.11)	0.15 (0.08)	0.11 (0.1)	0.13 (0.1)	0.19 (0.14)

9.13　筆記の特徴の訓練における被験者選択に関する考察

利用者の筆記の特徴を他の 9 名のものとともに訓練する際，この 9 名の特徴がどのようなものであるかによって FRR や FAR に差が出る．利用者と特徴が類似した者が加わると FRR が上がり，利用者と特徴が類似した者がいないと攻撃者による模倣が容易になり FAR が上がる．この特徴を利用し，この 9 名の候補に選ばれる可能性のある被験者のデータを用いて，提案手法の訓練済みモデルを予めいくつか作成しておき，新規の利用者を攻撃者と仮定してこれらのモデルに情報を入力すれば，あるモデルの中で誰に分類されるかの偏りがわかる．この偏りを利用すれば，FRR と FAR のバランスを Recall（もしくは F 値）による閾値以外でも調整可能となる．つまり，利便性と安全性のどちらをより優先したいかを利用者が選択可能なパラメータが増える．

9.14 複数のモデルを使用したアンサンブルな検証方法の考察

ランダムフォレストの特徴にアンサンブル学習と呼ばれるものがあり，アンサンブル学習では複数の決定木で各々の結果を出力し，これらをまとめて多数決を取り，最も多かった結果が，最終的なランダムフォレストの出力となるものである．そこで，この手法を本研究の検証で利用することを考察する．

例として，システムの利用者が A〜Z の 26 人だったする．ここからモデルの登録者を 10 人で分けるが，この場合 2 つのモデルは 10 人登録できるが，3 つ目のモデルは 6 人しかいないためモデルを作ることができない．そのため，提案手法の方法ではこの 3 つ目のモデルを作ろうとした場合，新たに 4 人の利用者が来るまで待つことになる．そこで，この考察では 1 つ，2 つ目のモデルに登録した人を 3 つ目のモデルにも登録することについて考える．同じ利用者を複数のモデルに登録する利点としては 2 つある．

1 つ目は登録する人数が足りないモデルの人数をすぐに満たすことができることである．例えば，1 週間に 1 回利用者をまとめて新しいモデルを作る場合に，利用者が足りなかったモデルに割り当てられた利用者は，モデルを作るのに十分な人数になるまで 1 週間，また 1 週間と待たされる可能性がある．そこでこの方法を利用することで利用者は長く待たされることなくシステムを利用開始することができる．

2 つ目の利点としてはアンサンブル学習と同じように複数のモデルを本人確認に使えることである．本人確認に複数のモデルを使うことで本人確認の結果を 1 つのモデルに依存することがなくなるため，システムの汎化性能を高めることができる．例えば，3 つモデルがあった場合に 1 つのモデルが他人を示したとしても，2 つのモデルが本人を示していれば本人確認に成功することができる．これにより FAR と FRR を下げることができると考えられる．

しかし，モデルをアンサンブルにする方法にはモデルの数が提案手法と比べて多くなる欠点がある．提案手法では 10 人を 1 つのモデルに登録するため 1 モデルで，利用者の 1/10 の数となる．考察の方法では 10 人の利用者がいた場合に，アンサンブル数を 10 とした場合，10 モデル必要になる．こうなった場合，2 値分類や OC-SVM のような一人 1 モデルの方法と同じ数になってしまう．そのため，一人を複数のモデルに登録できる最大数は 10 モデルまでにするべきだと考えられる．これより多くなってしまうと 2 値分類などより保存するモデル数が多くなってしまう．ただ，この考察の方法ではアンサンブルで本人確認が行えるため，一人 1 モデルしかない 2 値分類などと比べると，システムの汎化性能は高くなる．2 値分類などではモデルをアンサン

ブルさせようとした場合，アンサンブルの数が 2 つだとしても登録者の 2 倍の数モデルが必要になってしまう．

更に，2 値分類などと比較すると，この研究の提案手法の要件で示しているように利用者が増えても再訓練の必要がないという利点がある．例として，最初に利用者 10 人で，その後 90 人増え 100 人になるとする．2 値分類などの一人 1 モデルの方法では最初に利用者が 10 人しかいないときは本人 (A とする) のサンプルと，他人のサンプル 9 人分合わせて 10 人分訓練を行う．このモデルで A の本人確認しようとしたときは A のモデルに入れて A(本人) と A 以外の 9 人 (他人) の 2 値分類となる．OC-SVM も 1 クラスと呼ばれているが，本人か，本人以外の 2 値分類である．新たに 90 人が増えた時，新たな 90 人は既存の 10 人分を足して 1 モデルで 1 本人 99 他人のモデルを 90 個作ることになる．しかし，A のモデルの場合，90 人がモデルに訓練されていない状態で本人確認を行うか，90 人を足して本人一人，他人 99 人のモデルを作り直す必要がある．もちろん訓練し直していない状態でも本人確認を行うことはできるが，必然的に精度は低下すると考えられる．モデル数は 100 人いるため 100 となる．

3 つ目の研究の提案手法であれば 90 人が増えても最初の 10 人分のモデルは作り直す必要がなく，90 人を 10 人に分けて 9 モデル，最初のモデルと足して 10 モデルになる．利用者が 100 人になったとしても利用者 A は初期の 10 人分のモデルで本人確認を行うし，利用者 Z は Z が登録されている 10 人分のモデルで本人確認を行う．これにより，新たに登録者が増えても再訓練の必要がない．この状態でアンサンブルにした場合，アンサンブル数 10 で 2 値分類と同じ 100 モデルになる．

本研究の本人確認をアンサンブルで行うことでシステムの汎化性能を向上させ，ひいては精度の向上につながると考えられる．アンサンブルの数を増やすとモデルの数が多くなってしまうが，10 を超えなければ一人 1 モデルの 2 値分類方法よりモデルが多くなることはない．

9.15 実際のシステムに提案手法を組み合わせた場合の考察

本研究を実際に使用することを想定した考察を行う．ここでは現在使われているクレジットカード使用時の本人確認，更に，将来必要になると考えられる，オンライン投票システムでの利用を想定する．

モデルの訓練については 2 つのユースケース共通で以下の方法で行う．

利用者は店頭や役所などでカードを作る時に何度か空中筆記を行い，その筆記データを機械学習で訓練させる．一つのモデルに登録する利用者の数は全てのモデルで一定とする．モデルに登録者が足りない場合には，既に他のモデルで使用している利用

者のデータを利用し，訓練を行う．仮にクレジットカードの利用者を想定した場合，利用者は数十億人になると考えられるが，1 つのモデルに 10 人登録する場合はモデルの数は利用者数の 1/10 となる．ただし，9.14 節のアンサンブルをしている場合は利用者数の 1/10 にはならない．1 つ 1 つのモデルの大きさは全て同じであり，利用者が増減した場合，既に作られたモデルの再訓練を行うことはない．利用者が増加した場合は増加した数を決められたモデルの大きさに合わせ，訓練する．利用者が減少した場合は普通は何もしないが，そのモデルの利用者が一人もいなくなったモデルは削除する．この方法で作られたモデルを利用し，2 つのユースケースを想定する．

9.15.1 クレジットカード決済

今現在のクレジットカード決済時に使われる本人確認方法は，署名を書くことと，PIN を入力することであるが，ここでは PIN と組み合わせて使用することを想定する．

利用者はカードを使った時に最初は PIN を入力し，次に画面に表示された単語を空中に筆記する．どちらも正しければ本人確認は成功となる．どちらかが間違っていた場合にはやり直しとなる．PIN は覚える必要があるが，空中筆記は毎回書く単語が異なるため，特段覚えるものがなく，利用者は空中で筆記を行う数秒程度の負荷が増えるだけである．これだけの負荷でも攻撃者側からすると，手の動き，指の向きを真似をしなくてはいけないうえ，毎回画面には違う単語が表示されるため，大きな負荷となり，攻撃を躊躇させることができると考えられる．仮にカードを使用するところを録画や覗き見されたうえ，カード自体を盗まれたとき，PIN しかない場合はカードを停止させる前に悪用されてしまうことも考えられるが，本研究を組み合わせることで，突破を困難にすることで悪用されるまでの時間を稼ぎ，使われる前にカードを停止させることができると考えられる．抑止に関しては攻撃者の気持ちに依存するため，この手法がどの程度の抑止力があるかは計ることはできない．そのため，ここではカードを使われた場合の悪用防止能力についてのみ考察する．PIN は覗き見により既に突破されているとして，8.4 節の評価より，全結合ニューラルネットワークの場合，閾値が 0.7 の時，FRR が 6.9% であった．そのため 2 回連続で失敗する確率は 0.48% となる．3 回連続では 0.03% となり，2 回目の時点でほぼ全ての人が本人確認に成功できると言える．FAR は 3.7% であったため，攻撃者が練習していた場合はこの確率になるが，「1 回だけカードを使ってるのを見てからカードを盗んだ」などの突発的犯行であれば，PIN での失敗などを含め FAR は更に下がると考えられる．FRR の観点から本人確認のやり直し回数を 2 回か 3 回に制限することで，何度も攻撃を行い無理やり突破することを防ぐことができると考えられる．以上のことより本手法をクレジット

カード決済に組み合わせることで，クレジットカード悪用に対する防止能力を向上させることができる．

クレジットカード決済に本研究を組み合わせることで攻撃者に対する抑止力となり，更に，攻撃を受けたとしても PIN だけの方法より攻撃に耐えることができると考えられる．

9.15.2 オンライン投票

日本においてどこからでも投票ができる，公的なオンライン投票システムはまだ存在していないが，投票率向上のため，将来的には検討することになると考えられる．オンライン投票があることで投票所に行かなくとも，家から投票を行えるようになる．更に，下宿をしている学生などの，住民票と住んでいる場所が違うため，投票に行くために帰省するか，不在者投票を行わなければならない人も，不在者投票の申請をしなくともすぐに投票を行えるようになる．オンライン投票システムが実現できない問題点の一つとして，投票者が本当に本人であるか証明できないことも原因であると考えられる．オンライン投票は仮にマイナンバーカードの電子証明書を使用してオンライン投票した場合，投票を行った人が誰であるかということは分かるが，カードを使った人と，カードに登録されている人が違うという可能性も考えられる．投票所であればマイナンバーカードと持っている人を見比べて本人確認ができるが，直接確認する必要があるため，オンラインでは行えない．マイナンバーカードの電子証明書を使用するには 4 桁の数字が必要になる．そこで，オンライン投票には 4 桁の PIN が使われると想定する．マイナンバーカードで利用者証明用電子証明書を利用するときは，PIN の入力を行うが，3 回連続で失敗した場合，電子証明書の利用がロックされる仕様になっている．このロックを解除するには役所などのマイナンバーに関する窓口にいく必要がある．そのため，攻撃者がカードを入力できたとしても，PIN を総当たりや，ランダムな数字を入れるなどの攻撃をした場合，マイナンバーカードがロックされ，攻撃を続けることができなくなる．しかし，不正投票を行う人が何かしらの方法でカードと PIN の両方を入手できた場合や，入手したカードの PIN を適当に入力したらたまたま成功した場合，投票を行えてしまう．例えば，PIN の盗み方として，不正投票を行う人が誰かがコンビニなどの証明書発行機で PIN を入力しているところを覗き見して，PIN を把握した後に，スリなどでカードを盗み，自宅や，犯罪の拠点などに持ち帰り，オンライン投票システムにログインして勝手に投票をするということができてしまう．他には，高齢者施設など集団で生活している場所で，職員がマイナンバーカードを管理のためなどと適当な方法で取り上げ，PIN を聞き出すか，書いてある紙を探して把握することで，数十，数百人規模の不正投票を一人で行うことがで

きてしまうことも考えられる．仮にマイナンバーカードの盗難などによってオンラインで勝手に投票されたと証明できた場合，無効票にすることも考えられる．投票前であればカードを失効させることで不正投票を防げるが，既にオンライン投票された票を無効にするには難しいと考えられる．その票が誰が入れた票なのかわかるようにしてあれば無効票にするのは難しくはないが，誰が入れた票なのかわかってしまうと秘密投票にならないため，日本国憲法第 15 条第 4 項「すべて選挙における投票の秘密は，これを侵してはならない．選挙人は，その選択に関し公的にも私的にも責任を問はれない.」に違反した投票システムになってしまう．そこで，日本国憲法に違反しないオンライン投票システムは，「誰が」，「どの選挙区に投票権があり」，「投票を行ったか」しかわからないようになると考えられる．投票した時刻がわかってしまうとそれも個人の特定につながってしまうため，投票時刻の記録も行えないと考えられる．これにより，オンラインでの不正投票が行われてしまった場合，その票を見つけることができないと考えられる．そのため，オンライン投票システムは投票前の本人確認が重要となる．

そこで，本人確認において本研究を組み合わせることで，安全なオンライン投票システムを実現することが可能になると考えられる．本研究を組み合わせた投票システムは，パソコンとマイナンバーカードと Leap Motion が必要になる．これらを組み合わせてオンラインでの投票を行う．オンライン投票のために，マイナンバーカードの電子証明書を使用したときに，PIN 入力だけでなく，本研究の提案手法の空中筆記を行わせる．検証の結果どちらも正しければ本人確認は成功となる．どちらかが間違っていた場合にはやり直しとなる．PIN は覚えておく必要があるが，空中筆記は毎回書く単語が異なるため，特段覚えるものはなく，利用者は空中で筆記を行う数秒程度の負荷が増えるだけである．本研究を組み合わせることで仮に攻撃者に PIN が漏洩していたとしても，空中筆記の癖を真似できないと本人確認を突破することができなくなり，不正投票をされる危険性を低くすることができると考えられる．

ここでは不正投票を本人確認で防止する観点から，FAR の低さを重視し，8.4 節の評価より，畳み込みニューラルネットワークの閾値 0.8 の FRR19.5%，FAR:1.5% を使用し，考察を行う．もちろん CNN で閾値 0.9 のほうが FAR を低くできるが，FRR が約 38% もあり，約 5.5% の確率で 3 回連続で失敗することになる．3 回連続で失敗した場合マイナンバーカードはロックされる仕様になっているため，空中筆記を本人確認に組み合わせた場合，3 回連続の失敗には PIN の本人確認だけでなく，空中筆記の本人確認も含まれることになる．そのため，空中筆記の FRR が高いと，本人であるにもかかわらず本人確認に失敗してロックされることが多くなってしまう．閾値 0.8 であれば空中筆記で 3 回連続で失敗する回数は約 0.74% となるため，PIN を間違えていなければ，3 回目には 99% 本人確認に成功できる．しかし，100 人に一人は本人確

認に 3 回連続で失敗して証明書がロックされてしまうこともあると考えられる．利用者証明用電子証明書がロックされた場合投票だけでなく，証明書を必要とするサービス全てが利用できなくなる．その場合は役所の窓口で解除してもらうことになる．もちろん攻撃者も同じ方法でロックを解除することが考えられる．しかし，ロックを解除するにはマイナンバーカード以外の本人確認書類 (免許証など) が必要かつ，マイナンバーカードには顔写真がついているため，顔が違うなどの理由でロック解除を受け付けないと考えられる．オンライン投票はどこでもできる利点があるため，役所にはオンライン投票環境がない人向けに，オンライン投票に対応した投票所（オンライン投票所）が設置されると考えられる．オンライン投票所は主に住民票が今住んでいる場所にない人が投票を行うために利用されると考えられる．マイナンバーカードがなく，オンライン投票を行えない人は従来の投票券での投票になる．オンライン投票で空中筆記が必要な場合はマイナンバーカードの利用者が本人であるかほかの人が確認できないときである．役所や投票所では選挙の関係者などがマイナンバーカードを持っている人の顔と，カードを見比べることができるため，これだけで本人確認を行えることになる．そのため空中筆記による本人確認は省略できる．空中筆記による本人確認は 3 回目で成功する確率は 99% だが，3 回連続で失敗した後にまた 3 回連続で失敗する可能性もある．そのため，カードの持ち主とカードに書かれている人の確認ができるのであれば，空中筆記は省略した方が．投票に失敗する人を少なくできる．オンライン投票所では空中筆記を使う本人確認と，目視による本人確認は混在させられるため，空中筆記に自信がある人は空中筆記方式，自信がない人はマイナンバーカード目視確認方式を選べる．空中筆記方式は操作方法がわかれば，常につく必要がある係員が必要ないため，設置台数も多くでき，投票の行列を短くできる．もちろん自前のオンライン投票環境があれば，並ばずに自宅から投票ができる．役所でマイナンバーカードのロックを解除した人も同様に役所のオンライン投票所で空中筆記方式と目視方式の好きな方で投票を行うか，並ぶのが嫌であれば，家に帰って投票を行うことができる．

FAR は 1.5% であるため，平均約 66 回試行した場合突破することができるが，試行できる回数は 3 回しかなく，突破することは困難であると考えられる．攻撃者が練習していた場合はこの確率になるが，例として，攻撃者がコンビニで被害者のマイナンバーカード利用を覗き見た後，カードを盗んだ場合，攻撃者は一度の覗き見で PIN を把握できたとしても，空中筆記も全て覚えて再現しなければならず，8.4 節の評価の値より，FAR は更に下がると考えられる．一人の攻撃者が何らかの方法 (高齢者施設で職員が不正に入居者のマイナンバーカードを没収するなど) で，複数人からマイナンバーカードと PIN を入手し，不正投票を行おうと思った場合，複数人全ての空中筆記の特徴をまねる必要がある．仮に複数人の空中筆記の動作を見ることができたとして

も，二人分覚えるだけでも一人分と比べて 2 倍の労力がかかり，覚える数の多さから，一人だけの特徴を覚えるのより難しくなると考えられる．そのため，上記したコンビニでの覗き見の例より更に FAR は下がると考えられる．しかし，FAR は 1.5% であるため，仮に攻撃者が 100 枚のマイナンバーカードを入手し，そのうえ，PIN がわかっていた場合，1〜2 枚は突破できて不正投票が行われてしまう．しかし，FAR1.5% というのは盗まれたカードの中での話であるため，全てのマイナンバーカードが盗まれなければ，全体の投票数の 1.5% が不正投票ということにはならない．更に，オンライン投票が認められてたとしても，従来の投票所で行う投票も残っていると考えられるため，投票数の中の不正投票数は更に少なくなると考えられる．カードを盗まれて突破される確率が 1.5% だということは安全であるとは言い切れないが，PIN のみの場合は，カードを盗み，PIN がわかった時点で突破される確率は 100% となり，PIN のみの方法と比べて，不正投票の数を約 98% 減らすことができる．

オンライン投票システム実現のためには投票手段だけでなく，通信の安全，コスト，情報格差などの障壁が多いが，本研究の手法を組み合わせることによって，オンライン投票時の本人確認を PIN のみの方法と比べて，確実に行うことができようになると考えられる．

第 10 章

まとめ

本論文で提案した空中筆記による本人確認手法は，指紋などの身体的特徴を利用した生体認証と異なり，登録内容を変更可能な行動的特徴と本人の癖である身体的特徴の組み合わせを利用したものである．センサーの構造上，筆記動作そのものは人の目から隠すことができないため，動きそのものは見られてしまうが，空中に筆記を行うことで筆跡が残らないようにしたため，1 回見ただけでは動作がわからず，覗き見の耐性を高めることができる．

最初の研究では Leap Motion を使って空中に署名を書く方法を提案した．FRR の高さの問題と，同じ文字の署名を登録した場合精度が悪くなるという問題があったため，次の研究では同じ一方向を機械学習を使って検証する方法を提案した．精度は 1 つ目の研究より良い結果となったが，一方向の入力であるため覗き見に対して 1 つ目の研究と比べて弱くなるという問題があった．更に多人数での使用に問題もあった．そこで，最終的なシステムでは 2 つの研究を組み合わせた方法を提案した．本システムでは空中に好きな単語を筆記させ，それを複数の一方向に分解し，機械学習による本人確認を行った．結果は，最良の機械学習を使用した場合，登録者の本人拒否率を 5.7% とする閾値 0.5 で未登録者による他人受入率を 7.2% に，登録者の本人拒否率を 10.7% とする閾値 0.7 で未登録者による他人受入率を 2.6% となった．考察より，最終的なシステムは録画攻撃をされても FAR が上昇することがないため，ある程度の覗き見耐性を持ち，更に，多人数での利用を可能にした．本システムは筆記時の手の動きそのものを機械学習で分類するのではなく，手の動きを分解して利用することで，再登録することなく入力する単語を変更できる点が特徴である．また，既存研究と異なり，多人数の利用者がいる場合や，利用者の増加がある場合でも，システムとして実用可能な設計となっている．なお，利用者数が極端に多い場合に，適切な精度で利用できない利用者が現れる可能性はある．しかし，本手法を実装したシステムは，既存のパスワードによる個人識別を強化するために補助的に使用されるものであり，利

用者は本システムの使用を選択可能である．本研究は，今後の電子決済やオンライン投票における不正防止技術のひとつとして有効であると考える．

最後に，本研究の最終目的として 1.2 節にて 7 つの要件を示していたが，実際に満たされたか確認を行う．要件 1「本人確認デバイスの再発行を行わない」は，本研究では人ごとで異なる筆記時の癖を使用するため，カードキーのような人ごとに異なる本人確認用のデバイスを発行する必要がない．そのため要件 1 を満たす．2 つ目の要件「覗き見によるコピーに対してある程度の耐性を持つ」は，Leap Motion を用いて空中に筆記をすることにより，筆跡を残さないようにし，覗き見に対して耐性を持たせた．攻撃の評価 8.4.2 項では，他人の単語を本人として検証させたときの FAR と，本人の単語を模倣して書いた FAR に差がなかったことから，覗き見を行うことは FAR を上昇させることがないと言え，本研究は覗き見耐性があるという結果となった．そのため，要件 2 を満たす．3 つ目の要件「一度の覗き見によるコピーを行わせない」は，パスワードや PIN では，キーボードやキーパッドの配置は基本的にはどの製品も同じであり，覗き見を一度行うだけで入力していたパスワード，PIN が分かってしまう可能性があるのに対し，3 つ目の研究では毎回ランダムな単語を書かせる．何て単語を書いたかは画面に表示されるためわかってしまうが，一度の覗き見で単語を書いていた時の全ての指の速さ，向きなどの特徴を覚えるのは困難であり，更にそれを再現することはほぼ不可能であると考えられる．8.4.2 項より，録画攻撃した場合でも FAR が上昇しないため，それよりも見られる時間が少ない一度の覗き見ではコピーは行うことはできないと言える．そのため，要件 3 を満たす．要件 4「入力および本人確認に長時間を要しない」は機械学習を使用することで，本人確認を行うための検証を行う時に，2 つ目の研究では一本の線を空中に書き，3 つ目の研究では単語を空中に 1 つだけ書き，それを分解したサンプルを訓練済みのモデルに入れて検証を行うだけであるため，パスワードを思い出すなどの考える時間が必要なく，機械学習の検証も訓練ほど時間がかからないため，どれだけ 1 回の本人確認が長引いても，筆記動作，機械学習での検証を合わせて 1 分掛かることはない．そのため，入力および本人確認に長時間を要することがなく，要件 4 を満たす．要件 5「利用者の増加による機械学習の再訓練を行わない」，要件 6「利用者が非常に多い場合でも，機械学習による訓練コストを一定に保つ」は，1 つのモデルの登録者数を一定にすることで，利用者の増加があった場合には新たなモデルを作るため，既存のモデルは再訓練を行う必要がない．1 つのモデルの登録者数は一定のため，利用者が非常に多くなったとしても 1 つ 1 つのモデルの訓練コストは変化しない．これにより要件 5，6 を満たす．要件 7「攻撃者のデータが訓練に含まれない場合にも本人確認が行える」は，訓練時に攻撃者のデータを訓練させなくとも，閾値 0.7 で未登録者による攻撃の FAR を 2.6% まで低くすることができた．これにより，攻撃者のデータを訓練させなくとも，正しく本人確認を行

うことができると言え，要件 7 を満たす．以上のことから本研究は 1.2 節にて示した 7 つの要件全てを満たすことができた．

謝辞

本研究を行うにあたり，研究方針，実験方法，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宇田隆哉准教授に深く感謝申し上げます．さらに，本研究の実験の際に被験者を快く引き受けてくださった宇田研究室の皆様に心より感謝申し上げます．

参考文献

[1] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial "gummy" fingers on fingerprint systems. In Rudolf L. van Renesse, editor, *Optical Security and Counterfeit Deterrence Techniques IV*, Vol. 4677, pp. 275–289. International Society for Optics and Photonics, SPIE, 2002.

[2] T. Matsumoto. Gummy and conductive silicone rubber fingers: Importance of vulnerability analysis. In *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '02, pp. 574–576, Berlin, Heidelberg, 2002. Springer-Verlag.

[3] 松本勉. 金融取引における生体認証について. https://www.fsa.go.jp/singi/singi_fccsg/gaiyou/f-20050415-singi_fccsg/02.pdf.

[4] L. Fridman, S. Weber, R. Greenstadt, and M. Kam. Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. *IEEE Systems Journal*, Vol. 11, No. 2, pp. 513–521, 2017.

[5] R. Kobayashi, H. Susuki, N. Saji, and R. S. Yamaguchi. Lifestyle authentication and mithra project. In *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, pp. 464–467, 2018.

[6] R. Kobayashi and R. S. Yamaguchi. One hour term authentication for wi-fi information captured by smartphone sensors. In *2016 International Symposium on Information Theory and Its Applications (ISITA)*, pp. 330–334, 2016.

[7] RSA. Securid トークン. https://www.techmatrix.co.jp/product/securid/token/index.html.

[8] おくとパス Business8. Ic カード windows 認証ソフト. https://www.cca-co.jp/service/octpass-8/.

[9] e ー Tax 国税電子申告・納税システム (イータックス). マイナンバーカード方式について. https://www.e-tax.nta.go.jp/kojin/mycd_login.htm.

[10] 林大介, 赤倉貴子. e-testing におけるタブレット pc とオンライン筆記情報を用

いた筆記認証法の提案. 日本教育工学会論文誌, Vol. 42, pp. 101–104, 2018.

[11] 片桐雅二, 杉村利明. ビデオカメラを用いた空中署名による個人認証の試み. 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 101, No. 125, pp. 9–16, June 2001.

[12] 崎田隆行, 鹿嶋雅之, 佐藤公則, 渡邊睦. 指先トラッキングとその軌跡抽出を用いた個人認証に関する研究. 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 107, No. 384, pp. 59–64, December 2007.

[13] R. Hanyu, Q. Zhao, and Y. Kaneda. A new protocol for on-line user identification based on hand-writing characters. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, 2016.

[14] Y. Kato, T. Hamamoto, and S. Hangai. A proposal of writer verification of hand written objects. In *Proceedings. IEEE International Conference on Multimedia and Expo*, Vol. 2, 2002.

[15] A. Takahashi and I. Nakanishi. Authentication based on finger-writing of a simple symbol on a smartphone. In *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 411–414, 2018.

[16] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. *Conference on Human Factors in Computing Systems - Proceedings*, May 2012.

[17] C. Shen, Q. Lv, Z. Wang, Y. Chen, and X. Guan. Hand-interactive behavior analysis for user authentication systems with wrist-worn devices. In *2018 5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, pp. 90–95, 2018.

[18] 坂田健輔, 高橋大介, 岡本教佳. Leap motion による筆者照合のための特徴抽出の検討 (若葉研究者の集い 2, サマーセミナー 2014〜未来を拓くビジョン技術〜). 映像情報メディア学会技術報告, Vol. 38.32, pp. 19–22, 2014.

[19] 坂田健輔, 高橋大介, 岡本教佳. 三次元空間における指での筆記を利用した筆者照合に関する一検討 (セッション 2, 学生研究発表会). 映像情報メディア学会技術報告, Vol. 39.8, pp. 37–40, 2015.

[20] 畠中一成, 鹿嶋雅之, 佐藤公則, 渡邊睦. Leap motion を用いた空中署名での個人認識システムに関する研究 (バイオメトリクス). 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, Vol. 114, No. 212, pp. 33–38, September 2014.

[21] 畠中一成, 鹿嶋雅之, 佐藤公則, 渡邊睦. 指識別情報を用いたフレキシブル空中署

名個人認証システムに関する研究. 映像情報メディア学会誌, Vol. 70, No. 6, pp. J125–J132, 2016.

[22] R. Renuka, V. Suganya, and B.K. Arun. Online hand written character recognition using digital pen for static authentication. In *2014 International Conference on Computer Communication and Informatics*, pp. 1–5, 2014.

[23] H. Hu, D. Chen, and J. Zheng. Online handwriting signature verification based on template clustering. EBDIT 2019, p. 129–135, New York, NY, USA, 2019. Association for Computing Machinery.

[24] G. Xiao, M. Milanova, and M. Xie. Secure behavioral biometric authentication with leap motion. In *2016 4th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 112–118, 2016.

[25] S. Alkaabi, S. Yussof, S. Almulla, H. Al-Khateeb, and A.A. AlAbdulsalam. A novel architecture to verify offline hand-written signatures using convolutional neural network. In *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1–4, 2019.

[26] D. Lu, K. Xu, and D. Huang. A data driven in-air-handwriting biometric authentication system. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 531–537, 2017.

[27] S.K. Behera, D.P. Dogra, and P.P. Roy. Analysis of 3d signatures recorded using leap motion sensor. *Multimedia Tools and Applications*, Vol. 77, No. 11, pp. 14029–14054, June 2018.

[28] T. Nohara and R. Uda. Personal identification by flick input using self-organizing maps with acceleration sensor and gyroscope. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication(IMCOM '16)*, Vol. 6, January 2016.

[29] S. Yamamoto, S. Ito, M. Ito, and M. Fukumi. Authentication of aerial input numerals by leap motion and cnn. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pp. 189–193, 2018.

[30] A.A. Mohammed, A.K. Abdul-Hassan, and B.S. Mahdi. Authentication system based on hand writing recognition. In *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, pp. 138–142, 2019.

[31] T. Singh and S. Mishra. Image vector classification algorithm for hand-writing verification. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2239–2243, 2014.

[32] G. de Rosa, J. Papa, and W. Scheirer. *Person Identification Using Handwriting Dynamics and Convolutional Neural Networks*, pp. 227–244. March 2018.

[33] 小南嘉史, 西村広光, 富川武彦. 筆跡情報と筆圧情報の hmm を用いたサイン認証. 神奈川工科大学研究報告 B 理工学編, No. 30, pp. 73–78, March 2006.

[34] 高橋真奈茄, 小出洋. 機械学習を用いたパターン認識による筆者識別. 第 57 回プログラミング・シンポジウム予稿集, 第 2016 巻, pp. 133–142, January 2016.

[35] Y. Guerbai, Y. Chibani, and B. Hadjadji. The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*, Vol. 48, No. 1, pp. 103 – 113, 2015.

[36] A. Granet, E. Morin, H. Mouchère, S. Quiniou, and C. Viard-Gaudin. Transfer learning for a letter-ngrams to word decoder in the context of historical handwriting recognition with scarce resources. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1474–1484, August 2018.

[37] N. Aneja and S. Aneja. Transfer learning using cnn for handwritten devanagari character recognition. *2019 1st International Conference on Advances in Information Technology (ICAIT)*, July 2019.

[38] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, Vol. 43, No. 1, pp. 59–69, January 1982.

[39] L. Breiman. Random forests. *Machine Learning*, Vol. 45, No. 1, pp. 5–32, October 2001.

[40] 岡谷貴之. 深層学習. 講談社, 2015.

[41] 斎藤康毅. ゼロから作る Deep Learning–Python で学ぶディープラーニングの理論と実装. オーム社, オライリージャパン, 2016.

[42] S. Kamaishi and R. Uda. Biometric authentication by handwriting using leap motion. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication(IMCOM '16)*, January 2016.

[43] S. Kamaishi and R. Uda. Biometric authentication by handwriting with single direction using self-organizing maps. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication(IMCOM '17)*, January 2017.

[44] Leap Motion. C# sdk documentation - leap motion c# sdk v2 3 documentation. https://developer-archive.leapmotion.com/documentation/v2/csharp.

[45] SAS Japan. 機械学習アルゴリズム選択ガイド. https://blogs.sas.com/content/sasjapan/2017/11/21/machine-learning-algorithm-use/.

[46] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv e-prints*, p. arXiv:1502.03167, February 2015.

[47] K. Kritsis, M. Kaliakatsos-Papakostas, V. Katsouros, and A. Pikrakis. Deep convolutional and lstm neural network architectures on leap motion hand tracking data sequences. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, September 2019.

[48] クジラ飛行机. 無料 英和辞書データ ダウンロード - ブラウザで使える web 便利ツール. `https://kujirahand.com/web-tools///EJDictFreeDL.php`.

業績

[1] S. Kamaishi, and R. Uda, Biometric authentication by handwriting using leap motion. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication(IMCOM’16)*, No.36, p.1-5, January 2016.

[2] S. Kamaishi, and R. Uda, Biometric authentication by handwriting with single direction using self-organizing maps. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication(IMCOM’17)*, No.106, p.1-6, January 2017.

[3] 釜石智史, 宇田隆哉. 特徴の再訓練を必要としない変更可能な筆記. 情報処理学会論文誌, Vol.63, No.4, p.1094-1114, April, 2022.