秘密計算を用いた顔認証の構成について

手島 宏貴[1]　山下 恭佑[1]　矢内 直人[1]　岡村 真吾[2]

概要：機械学習を用いた顔認証技術は様々な場面で利用が進んでいる一方，顔画像と機械学習モデルが参照する顔画像データベース双方において，プライバシーの観点から画像を秘匿する必要がある．本稿では，顔画像の特徴量抽出において機械学習を用いた秘匿顔認証を提案し，それに適した機械学習の設定を実験的に明らかにする．具体的には，顔画像の特徴量を実数値・整数値・バイナリ値のいずれかにすること，また，認証時に顔画像を比較する際の統計的距離をコサイン距離・ハミング距離・ユークリッド距離のいずれかにすることで議論する．機械学習として ArcFace，秘密計算は CrypTen を用いて計算時間と認証精度を評価したところ，二つの知見を得た．まず計算時間と精度の観点において最も優れた設定は，顔画像の特徴量を実数値，認証の際の統計的距離をユークリッド距離とした場合であることを確認した．次に，様々なデータ分布に対して安定的な挙動を示す設定は特徴量がバイナリ値，統計的距離がハミング距離の場合であることを確認した．

キーワード：機械学習, 顔認証, 秘密計算, ハミング距離, ユークリッド距離

A Construction of Facial Authentication with Secure Computation

Koki Tejima[1]　Kyosuke Yamashita[1]　Naoto Yanai[1]　Shingo Okamura[2]

Abstract: While a machine learning-based face authentication technology has been used in various situations, it is necessary to protect both face images and a database of face images referenced by the machine learning model for privacy. In this paper, we propose a privacy-preserving face authentication system based on machine learning for feature extraction of face images and then investigate a suitable setting for the system through extensive experiments. Specifically, we discuss real/integer/binary numbers as features of facial images and cosine/Hamming/Euclidean distance as statistical distances for the authentication. We found two key insights when we evaluate the execution time and authentication accuracy using ArcFace for machine learning and CrypTen for secure computation. First, by examining the registration process of face images, we confirm that the execution time can be improved in a database. Second, we demonstrate that the Euclid distance is superior to the Hamming distance, which is more compatible with secure computation, and that the Euclid distance achieves higher authentication accuracy in a comparable execution time.

Keywords: machine learning, face authentication, secure computation, Hamming distance, Euclid distance

1. 序論

1.1 背景

近年, 深層ニューラルネットワークを用いた画像認識性能の向上に伴い, 顔画像を利用した認証の精度が上がり, 多くの場面で活用されるようになってきた. 例えば, アミューズメント施設やイベント会場への入場時の本人確認などに顔認証が利用されており, 非接触な認証として利用が広まっている [1]. しかし, 顔画像は生体情報の一種で個人を特定するのに十分な情報であり, また, 生涯不変な情報であることから, 漏えいしたときに大きな被害をもたらす可能性がある. このため, 情報が漏えいしたとしても利用者への被害を抑えられるように, システム内においても秘匿されることが望ましい [2].

[1] 大阪大学, Osaka University
[2] 奈良工業高等専門学校, National Institute of Technology, Nara College

一般に, ユーザが顔認証システムを利用する際にはあらかじめシステムのデータベースに顔画像を保存しておき, 認証する際に入力された顔画像と比較し認証する. 認証結果として, 入力された顔画像の人物がシステムに登録されているか否かを出力する. 認証結果から誰がシステムに登録されているかがわかってしまうため, 顔画像を含むデータベースと入力に加えて出力も秘匿する必要がある. 従って, ユーザの顔画像をシステムのデータベースと認証の際の入出力双方の観点から秘匿する必要がある. そのような秘匿性をもつ顔認証はデータを秘匿化した状態で任意の関数を評価する秘密計算を用いることで実現できる. 秘密計算を用いると高い安全性を達成しつつ, 精度の劣化が小さい顔認証システムを実現することができる. このような顔認証システムを**秘匿顔認証システム**と呼称する.

1.2 貢献

本稿では, 顔画像の特徴量抽出に機械学習を用いた 1 対 1 秘匿顔認証システムを設計し，その精度と計算時間を実験的に評価する. 特に機械学習の設定がこれらの数値にどう影響するかを明らかにする.

ここで着目する機械学習の設定とは，抽出する顔画像の特徴量の形式（実数値，整数値，バイナリ値）と，特徴量比較の際の統計的距離の形式（コサイン距離，ユークリッド距離，ハミング距離）である. 既存の秘匿顔認証 [3] や一般的な秘匿推論 [4], [5], [6], [7] では，秘密計算に適したアルゴリズムとして，顔画像の特徴量やアルゴリズム内のパラメータを実数値や整数値からバイナリ値に変換することで計算時間の改善を試みている. またコサイン距離は ArcFace [8] をはじめ機械学習による画像認識で広く用いられている [9], [10] 一方で，ハミング距離は秘密計算を用いた顔認証において，計算量削減に用いられている [3]. これらの統計的距離に加えてユークリッド距離を機械学習と秘密計算いずれにもよらないような一般的な統計的距離として用い，秘匿顔認証を評価した.

秘匿顔認証の特徴量抽出器を ArcFace [8]，秘密計算を CrypTen [11] で実装し，実験を行った. その結果，以下の二つの知見を得た. (i) 計算時間と認証精度の観点から最も高い性能を発揮したのは，特徴量を実数値，統計的距離をユークリッド距離に設定した場合であった. (ii) 顔画像の幅広いデータ分布に対して最も安定しているのは，特徴量をバイナリ値に設定した場合であった. 実験の詳細については 5 節を参照されたい.

2. 関連研究

2.1 秘匿顔認証

秘匿顔認証に関する既存研究として SciFI [3] を挙げる. SciFI では画像検索にハミング距離を用い，計算量を削減している. 本稿の提案手法は SciFI を機械学習に拡張したものとみなせる.

また, 差分プライバシーを用いた顔認証 [12] が研究されている. これは, 入力とデータベースの顔画像にノイズを加えることでデータを秘匿する. しかし, ノイズが大きすぎると認証の精度が著しく低下してしまう. 一方, ノイズが小さいと人間の目では画像が誰なのかがわかってしまうため, 秘匿性が低くなってしまうという問題点がある.

2.2 秘匿生体認証

生体認証の秘匿化は一般にキャンセラブルバイオメトリクスと暗号理論的手法を用いた認証に大別される [13]. キャンセラブルバイオメトリクスでは生体情報の特徴量に対して不可逆の変換を施すことで，生体情報が秘匿化できる [14]. しかし，不可逆変換ゆえに元となる生体情報が欠落した際に修復不可能であるという課題が指摘されている [15].

暗号理論的手法を用いた秘匿生体認証は多岐にわたるが, 特に秘密計算を用いた手法を紹介する. 準同型暗号 [16] を用いた指紋認証方式が提案されている [17]. また，この方式をさらに改良した方式も提案されている [18]. 秘匿顔認証としては, 準同型暗号方式を用いた顔認証が既に提案されている [3], [19]. また，準同型暗号と Garbled Circuit [20] を組み合わせた秘匿顔認証も提案されている [21], [22].

2.3 秘匿推論

秘匿推論は，あるクライアントが何らかの処理を，あるサーバ上で実行される機械学習に委託するような場合にその入出力などの秘匿性を保証する. クライアントの入力を x，機械学習アーキテクチャを f，サーバの持つパラメタを θ とする. 秘匿推論は以下の三つの要件を満たす. (i) x と $f(x,\theta)$ がサーバに明かされない. (ii) θ がクライアントに明らかにされない. (iii) f の処理中の値はクライアントにもサーバにも明かされない.

様々な暗号技術を用いた秘匿推論が知られている. 例えば加法型秘密分散法 [23]，通信紛失プロトコル [24], Garbled Circuit [20] などが用いられている.

3. 秘匿顔認証

本稿では秘匿顔認証のうち，1 対 1 認証と呼ばれるものを提案する. 以下では 1 対 1 認証を導入したのち，秘匿顔認証のための要件を整理する.

3.1 顔認証

顔認証は大まかに特徴量抽出器とデータベースという 2 種類のエンティティから構成される. 特徴量抽出器とは, 与えられた顔画像からその特徴量を求め出力するエンティティであり，顔認証システムの入力インターフェースとしての役割を果たす. データベースには顔情報の特徴量が登

録されており，特徴量抽出器から受け取った特徴量を登録情報と突合し，認証を行う．データベースはサーバ上に実装されるものであるが，一般にサーバが複数存在することが考えられる．これにはデータベースに冗長性を持たせる，データを分割してセキュリティを高めるなどさまざまな目的が考えられるが，いずれにせよこのような場合サーバ間で通信を行うことが必要となる．

顔認証は通常，登録と認証の 2 段階に分けられる．登録段階では認証したい人物の顔情報の特徴量を抽出し，データベースに登録する．認証段階では，データベースが特徴量抽出器から受け取った特徴量と登録されている特徴量とを比較し，このユーザが登録されているかどうかを判定する．最後に認証結果がユーザに送付され実行完了となる．

本稿ではとくに 1 対 1 顔認証を扱う．1 対 1 顔認証とは，認証にユーザごとに固有の ID を用いる方式である．登録段階では顔情報のみならず該当する人物のユーザ ID を登録し，これらを紐づける．認証の際にはユーザの顔情報とユーザ ID を入力する．与えられたユーザ ID を基に，この ID と紐づけて登録されている顔情報と今入力されている顔情報とを比較し，同一人物であるかの判定を行う．このように登録されている全ての顔情報と比較するのでなく，ユーザ ID によってデータベースの探索範囲が予め限定されているのが 1 対 1 顔認証の特徴である．

3.2 秘匿顔認証の要件

3.2.1 守るべき情報

まず秘匿顔認証で守るべき情報は何かを述べる．1.1 節で述べた通り，秘匿顔認証ではユーザのプライバシ情報である顔情報を秘匿することを目的とする．従って最低限秘匿化する必要があるのは，顔画像を含む入力情報と，データベースに登録されている情報である．さらに, 認証した人物や認証結果の記録が必要な場合は，出力の認証結果も秘匿されるべきである．そのようなアプリケーションの例として入退室管理が挙げられる．入退室管理などでは誰が利用したかを記録する必要があるので, 認証結果には認証の可否に加え，認証された人物の情報, ユーザの ID または顔画像が含まれる．このため, 認証結果も秘匿することを考える．

3.2.2 攻撃者のモデル

本稿では複数存在するデータベースサーバが結託したとしても上述の情報を秘匿できる秘匿顔認証方式を提案する．ただし全てのサーバが結託した場合情報を秘匿することは原理的に困難であるため，少なくとも 1 台のサーバは正常な状態であるとする．また，本稿では顔認証のうち特に認証段階における攻撃を考える．従って顔情報の登録は正常に行われるものとする．

冗長性を持たせるために複数台のサーバが存在する場合は，全てのサーバでデータが同期されているため 1 台でもサーバが汚染されると情報漏洩の恐れがある．従って本稿で考える攻撃者のモデルは，セキュリティを高めるために複数のサーバが用意されている場合を暗に前提としている．

3.3 本稿の問題設定

本稿では顔情報の特徴量抽出に機械学習を用いた秘匿顔認証アルゴリズムを提案する．特にサーバが汚染された場合でも顔情報が漏洩しないアルゴリズムを目標とし，パブリックチャンネルを流れる入出力情報の保護は考慮しないものとする．なぜならば，これらの情報は TLS などのセキュアチャンネルを用いることで保護することが可能なためである．ただし認証結果を各サーバが知ることは情報漏洩につながる恐れがあるため，各サーバが認証結果を知ることなく認証が完了するアルゴリズムを提案する．

4. 提案手法

4.1 全体像

本稿で提案する秘匿顔認証は秘匿でない顔認証アルゴリズム，秘密分散ベース秘密計算を構成要素とする．これらの構成要素を用いて，顔情報登録，特徴量抽出，秘匿顔認識という機能を実現する．本節ではこれらの構成要素を導入したのち，提案する秘匿顔認証を俯瞰する．

4.1.1 構成要素の説明

以下ではあるアルゴリズム X のサブルーチン A を X.A と表すことにする．秘匿でない顔認証アルゴリズム FR はサブルーチンとして顔登録機能 FR.Reg，特徴量抽出器 FR.FtrExt，顔認証機能 FR.FaceRcg を持つ．一般の顔認証アルゴリズムではこれらの他に顔認識機能が求められるが，本稿の提案アルゴリズムでこれを用いることはないため説明は割愛する．データベースに顔情報を登録する際には，FR.FtrExt で抽出した特徴量が用いられる．顔認証の際は入力された顔情報から特徴量を抽出したのちデータベースと照合し，認証成功/失敗を判定する．

秘密計算はいくつかの種類に大別されるが，本稿では特に秘密分散ベース秘密計算 [23], [25] を扱う．秘密分散ベース秘密計算 SC はサブルーチンとしてシェア生成機能 SC.Share と評価関数 SC.Eval を持つ．シェアとは秘匿したい情報を何らかの形で分割したものである．ある定められた組み合わせを集めると元の情報が復元できる一方，シェア単体からは情報が復元できないという性質を持つ．評価関数は複数のシェアを入力したときに，シェアが定められた条件を満たすならば元の秘匿したい情報を明かすことなく所望する計算を行えるという性質を持つ．加法と乗法に関する秘密計算は既に知られているため，任意の演算に対する秘密計算が可能である．また，本稿では特に秘密計算の性質として秘匿推論を求める．すなわち，秘密計算に参加した各エンティティには計算結果のシェアのみが出力として与えられるとする．

SFR.Reg$(id, Pict)$:
$X^{id} \leftarrow$ FR.FtrExt$(Pict)$;
$(X_1^{id}, \cdots, X_n^{id}) \leftarrow$ SC.Share(X^{id});
For $i = 1, \cdots, n$ **do**
$\mathsf{DB}_i^{id} := (id, X_i^{id})$
return $\{\mathsf{DB}_i^{id}\}_{i\in[n]}$

SFR.FtrExt$(id, Face)$:
$x \leftarrow$ FR.FtrExt$(Face)$;
$(x_1, \cdots, x_n) \leftarrow$ SC.Share(x);
return $\{id, x_i\}_{i\in[n]}$

SFR.FaceRcg$(id, \{\mathsf{DB}_i^{id}\}_{i\in[n]}, \{x_i\}_{i\in[n]})$:
return $\{y_i\}_{i\in[n]} \leftarrow$ SC.Eval$((x_i)_{i\in[n]}, (X_i^{id})_{i\in[n]})$

図 1: 秘匿顔認証アルゴリズム SFR = (Reg, FtrExt, FaceRcg) の構成.

4.1.2 全体像の説明

本稿で提案する秘匿顔認証では，一人のユーザ（クライアント）と n 台のサーバの間で顔認証を行う．サーバにはあらかじめ顔情報（とユーザ ID）を登録しておく必要があるが，顔情報は n 個のシェアに分割されており，各サーバにはユーザ ID に紐づく一つのシェアが与えられている．顔認証の際には特徴量抽出器にユーザ ID と顔情報を入力として与える．顔情報は n 個のシェアとして分割され，各サーバに分配される．各サーバはあらかじめ登録されたシェアと分配されたシェアを用いてサーバ間で秘密計算を行う．この際の秘密計算は分配されたシェアと登録されたシェアの統計的距離の比較を行い，あるしきい値 d 以上の値が得られた場合認証成功を，そうでなければ認証失敗を出力する．認証結果は各サーバにシェアとして分配され，各サーバからユーザへ出力結果のシェアを送付する．以上のように各サーバに与えられるのは（ユーザ ID と）シェアのみであるため，各サーバが顔情報や認証結果を復元することはできず，結果秘匿顔認証が実現できる．

4.2 アルゴリズムの詳細

ここでは秘匿ではない顔認証アルゴリズム FR = (Reg, FtrExt, FaceRcg) と秘密分散ベース秘密計算 SC = (Share, Eval) を用いた秘匿顔認証アルゴリズム SFR = (Reg, FtrExt, FaceRcg) の構成を提案する．サーバは n 台あるものとする．秘匿顔認証アルゴリズム SFR = (FaceReg, FtrExt, RaceRcg) を図 1 に示す.

顔登録サブルーチン SFR.Reg は識別子 id と顔情報 $Pict$ を入力とし，以下のように動作する．初めに $Pict$ を特徴量抽出器 FR.FtrExt に入力し，特徴量 X^{id} を得る．一般に特徴量はベクトルとして表現される点に留意されたい．さらに X^{id} に対するシェアをシェア生成機能 SC.Share を用いて求め，シェア $X_1^{id}, \cdots, X_n^{id}$ を得る．最後に $i = 1, \cdots, n$ に対して $\mathsf{DB}_i^{id} := (id, X_i^{id})$ とし，これらを出力する．ここで DB_i^{id} はサーバ i に保存される，識別子 id とそれに紐づく顔情報のシェアである．

特徴量抽出器 SFR.FtrExt は SFR.Reg と非常によく似た動作をする．識別子 id と顔情報 $Face$ を与えられると，SFR.FtrExt 同様シェア $x_1, \cdots, x_n$ を求め，id とともに出力する．ここで x_i はサーバ i に分配されるシェアである．

顔認証 SFR.FaceRcg は入力として id, $\{\mathsf{DB}_i^{id}\}_{i\in[n]}$, $\{x_i\}_{i\in[n]}$ が与えられる．これは各サーバがそれぞれの入力を用いて計算を実行するという意味であり，一つのアルゴリズムに全てのシェアが入力として与えられるわけではないことに注意されたい（さもなくばシェアから元の特徴量を求めることが可能になる）．秘密計算 $\{y_i\}_{i\in[n]} \leftarrow$ SC.Eval$((x_i)_{i\in[n]}, (X_i^{id})_{i\in[n]})$ を実行し，その結果を出力する．ただし SC.Eval は 4.1.2 節で導入した動作をするものとする．すなわち，y_i は各サーバに送られる認証結果のシェアであり，各サーバがそれぞれのシェアをクライアントに送付することで最終的な顔認証の結果が得られる．

4.3 実際の構成

5 節では秘匿顔認証を実装するが，あらかじめその内容について簡単に触れる．実験は 1 台の特徴量抽出器と 2 台のデータサーバを用いる．つまり，$n = 2$ の場合で実験を行っている．特徴量抽出器は機械学習（ArcFace [8]）を用いて実装している．また，特徴量として実数値，整数値，バイナリ値（ハッシュ値）をそれぞれを用いた場合について実験を行っている．秘密計算には CrypTen [11] を用いている．秘密計算における特徴量間の統計的距離の計算にはコサイン距離，ハミング距離，あるいはユークリッド距離のいずれかを用いる．

5. 実験

本節では前節で述べた構成について，実験評価の内容を述べる．

5.1 実験目的

本実験では特徴量抽出器として機械学習を用いた秘匿顔認証について，その計算時間と認証精度の観点から，どのような機械学習の設定が相応しいかを明らかにする．特に，実際の利用を考えた際にはモデルの訓練として用いるデータセットに顔画像が含まれていないようなユーザのシステムも考えられることから，訓練データを持たないようなユーザの認証についても確認する必要がある．なお，本

表 1: 顔特徴量抽出器の実装環境

OS	U buntu 20.04.3 LTS
GPU	NVIDIA Quadro GV100 32GB
CPU	Intel Xeon Gold6240 2.6GHz
メモリ	95GB
ストレージ	512GB

表 2: データベースサーバの実装環境

AWS インスタンス名	t2.micro
OS	Ubuntu 20.04
CPU	Intel(R) Xeon(R) CPU E5-2676 v3
メモリ	55GB
平均データ送信速度	2.17Gbit/s

稿では機械学習の設定の違いとして以下に述べる二つの観点，すなわち特徴量の形式とそれらの統計的距離の計算に着目して検討する．この二つの観点は秘匿顔認証において，特徴量の抽出とそれを用いた秘密計算にそれぞれ関わるものである．

まず特徴量の形式について，特徴量には実数値，整数値，バイナリ値を用いる．実数値は一般に精度を得やすく機械学習で良く用いられる一方，整数値は剰余演算を伴う秘密計算との親和性が高い [26], [27]．バイナリ値は値域を極小化した整数値であり，機械学習では次元削減 [9] や計算速度の改善 [28] に用いられるほか，秘密計算への応用も知られている [4], [5], [7].

次に，特徴量間の統計的距離の計算にはコサイン距離，ハミング距離，あるいはユークリッド距離のいずれかを用いる．それぞれの計算方法は 5.2.3 節で述べるが，コサイン距離は ArcFace [8] をはじめ機械学習による画像認識で広く用いられている [9], [10]．一方，ハミング距離は秘密計算を用いた顔認証において，計算量削減に用いられている [3]．ユークリッド距離は機械学習と秘密計算いずれにもよらないような一般的な統計距離として用いる．上述した特徴量と統計的距離の計算から，秘匿顔認証における計算時間と認証精度を議論する．

5.2 実験設定

5.2.1 実装環境

特徴量抽出器と秘密計算の実装はそれぞれ ArcFace [8] と CrypTen [11] を用いて行った．CrypTen が PyTorch [29] ベースであることから，PyTorch ベースの環境を構築した．特徴量抽出器のネットワークには ResNet50 [30] を使用している．特徴量抽出器とデータベースサーバそれぞれの実装環境を表 1 と表 2 に記す．4.3 節で述べた通りデータベースサーバ 2 台の構成を考えており，これらは同じ種類の AWS インスタンスを二つ用いることで実装されている．

表 3: データセット

訓練用データセット	人物数 [人]	画像数 [枚]
Faces_Emore	85,742	5,822,653
評価用データセット	**人物数 [人]**	**画像数 [枚]**
CFP_dataset	500	5,000
FaceScrub	530	10,600

5.2.2 データセットとベースライン

本実験では顔特徴量抽出器の訓練とモデルの評価にそれぞれ異なるデータセットを利用する．これは訓練データが十分にないようなユーザの顔画像も正確に認証できるか評価するためである．

表 3 に訓練用データセットと評価用データセットを記す．顔特徴量抽出器の訓練には，Faces_Emore データセットを利用した．評価用データセットには，CFP_dataset [31] と FaceScrub [32], [33] を用いている．具体的には，各人物の画像が CFP_dataset には 10 枚ずつ，FaceScrub には 20 枚ずつあることから，それらのうち半分の枚数をしきい値の計算，残り半分を精度と計算時間の評価にそれぞれ用いた．なお，ベースラインには秘匿化していない ArcFace [8] を用いる．その目的関数は以下である．

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

5.2.3 特徴量と統計的距離の計算

本稿の実験では，特徴量は 512 次元のベクトルで表される．このとき，特徴量を実数値，整数値，バイナリ値とした場合それぞれについて検討する．ベースラインでは特徴量を実数値とするモデルとバイナリ値とするモデルを用いる一方，秘匿顔認証では特徴量を実数値から整数値あるいはバイナリ値へ変換したものをモデルとする．この整数値あるいはバイナリ値への変換は秘密計算の機械学習の応用によく用いられる [4], [6]．なお，CrypTen のシェアには整数シェアとバイナリシェアの 2 種類があるため，それぞれに合わせて特徴量を整数値にしたモデルとバイナリ値にしたモデルを作成している．

前述した通り，認証の際の統計的距離の計算ではコサイン距離，ハミング距離，ユークリッド距離を比較する．コサイン距離は二つのベクトルがどれほど似ているかを表す尺度であり，二つのベクトルがなす角のコサイン値で表現される．ハミング距離は，二つのベクトル間で値が異なる要素の数を示す値である．ユークリッド距離は，二つのベクトルを点と捉えたときの直線距離を表す尺度であり，ベクトル間の差の二乗和で表現される．

特徴量の抽出と統計的距離の計算において，ハミング距離は特徴量がバイナリ値の場合のみ計算できる．一方ユークリッド距離をバイナリ値で計算すると，その計算方法から結果がハミング距離に一致する．つまり，実数値と整数値ではコサイン距離とユークリッド距離を，バイナリ値で

表 4: 顔画像における特徴量の抽出時間

特徴量の形式	特徴量抽出時間 (s)
実数値	6.68e-3
整数値	6.73e-3
バイナリ値	6.53e-3

はコサイン距離とハミング距離をそれぞれ計算する．

計算時間と認証精度は 5,000 回の施行の平均を測定した．なお，計算時間については，特徴量の抽出と認証における統計的距離の計算をそれぞれ計測する．

5.3 実験結果

実装した秘匿顔認証において，特徴量の抽出にかかる時間を表 4 に，認証における統計的距離の計算の時間と精度を表 5 と表 6 にそれぞれ示す．以降では各結果についてそれぞれ説明する．なお，コサイン距離についてはバイナリ値を除き，CrypTen 内でのまるめ誤差が大きく，精度を正確に計算できなかった．このため，コサイン距離は実数値と整数値では精度を記載していない．

5.3.1 顔画像の特徴量の抽出

まず顔特徴量抽出について，表 4 に示すように，各特徴量の形式において抽出時間にほとんど差は見受けられなかった．実験の計算機環境を考えると，これらの差は誤差程度であると考えられる．すなわち，どの特徴量の形式を用いるかは，認証における統計的距離の計算に従って決めることが望ましいと考えられる．

5.3.2 認証における統計的距離の計算

5.3.2.1 計算時間

認証における統計的距離の計算時間について，表 5 に示す．表において “平文” の列が秘密計算を用いない場合，“秘匿化” の列が秘密計算を用いた場合にそれぞれ該当する．なお，ハミング距離の計算時のみバイナリシェアでの計算時間も測定しており，表 5 では (B) として表す．秘匿顔認証に要する計算時間は，秘密計算をしない場合と比較して，コサイン距離ではおよそ 10,000 倍，ハミング距離とユークリッド距離ではおよそ 1,000 倍になっている．

一方，各特徴量の形式において，コサイン距離の計算はユークリッド距離の計算に比べ，秘密計算をしない場合ではおよそ 2.3 倍程度である．これに対し，秘密計算では実数値と整数値ではおよそ 31 倍，バイナリ値ではおよそ 28 倍程度の時間を要しており，バイナリ値と比べると計算時間が増加している．コサイン距離の計算は実数値と整数値では CrypTen で正確な値が計算できない時もあり，計算時間にばらつきが生じている．

シェアで計算するとユークリッド距離とハミング距離の計算時間に大きな差がないことがわかる．ハミング距離は，バイナリシェアで計算するよりも整数シェアで計算する方が早く計算できる．

表 5: 1 回あたりの認証計算時間

特徴量の形式	統計的距離の計算	計算時間 (s)	
		平文	秘匿化
実数値	コサイン距離	5.93e-5	6.50e-1
	ユークリッド距離	2.84e-5	2.07e-2
整数値	コサイン距離	6.04e-5	7.23e-1
	ユークリッド距離	2.62e-5	2.35e-2
バイナリ値	コサイン距離	6.26e-5	5.94e-1
	ハミング距離	2.88e-5	2.12e-2 2.78e-2 (B)

表 6: 顔認証精度

特徴量の形式	統計的距離の計算	データセット	精度	
			平文	秘匿化
実数値	コサイン距離	FaceScrub	0.8783	-
		CFP_dataset	0.8433	-
	ユークリッド距離	FaceScrub	0.8351	0.8330
		CFP_dataset	0.7172	0.7165
整数値	コサイン距離	FaceScrub	0.8781	-
		CFP_dataset	0.8436	-
	ユークリッド距離	FaceScrub	0.8336	0.8333
		CFP_dataset	0.7159	0.7153
バイナリ値	コサイン距離	FaceScrub	0.7763	0.7748
		CFP_dataset	0.7900	0.7887
	ハミング距離	FaceScrub	0.7939	0.7939
		CFP_dataset	0.7892	0.7892

5.3.2.2 認証精度

秘匿顔認証の精度を表 6 に示す．秘密計算時に誤差が生じるためユークリッド距離とコサイン距離での精度が少し落ちているが，秘密計算による誤差が小さいことが確認できる．実数値を整数値に変換したモデルの精度は，平文の状態では認証精度に大きな誤差が生じないことが知られている [4]．特徴量をバイナリ値にするモデルでは実数値のモデルと比べて，情報量の削減により精度が落ちている一方，ハミング距離の計算では誤差が生じることがなく，精度劣化は見受けられなかった．

6. 考察

本節では，5 節で得た実験結果のうち，特に秘匿化した場合の計算時間や精度に着目し考察を行う．

6.1 計算時間

表 5 より最も高速なのは，特徴量と統計的距離がそれぞれ実数値とユークリッド距離の場合である．最も単純な距離計算を行なっているハミング距離が高速であると考えられたが，このような結果となった理由を考察する．

ユークリッド距離の計算において，通常であれば二つのベクトルの差の二乗和の平方根を計算する．しかし秘密計算において計算量を削減するために平方根は取らず，差の二乗和をユークリッド距離として扱っている．このような

手法は秘匿認証に関する既存研究 [34] でも用いられているものである．また，バイナリシェアを用いてハミング距離を計算する際，XOR は早く計算することができるが sum をとるために整数シェアに変換する必要がある．つまりバイナリシェアから整数シェアに変換しているため，単純な距離計算以上の動作が要求されている．これらのような事由により，上記の結果が得られたと考えられる．

6.2 精度

表 6 より最も高い精度が出たのは，特徴量・統計的距離・データセットがそれぞれ整数値・ユークリッド距離・FaceScrub の場合である．しかし，実数値・ユークリッド距離・FaceScrub の場合と比べて認証率がわずかに 0.03% 上回っているに過ぎず，その精度が 83.33%であることを考慮すると，これらは同程度の精度を達成していると言える．

特徴量をバイナリ値とした場合は，統計的距離やデータセットの種類によらず安定的な精度が実現できた．実数値や整数値の場合と比べてやや精度が劣るのは特徴量をバイナリ化した際に情報が落ちたことが原因だと考えられるが，広い分布に対して安定的な精度が保てるのは特徴量をバイナリ値とした場合であると考えられる．

一方で特徴量が実数値や整数値の場合，CFP_dataset ではコサイン距離とユークリッド距離との精度差が FaceScrub データセットのそれぞれの距離の差より大きくなっている．データセットのデータ分布の差によるものだとすると，ユークリッド距離がデータ分布の影響を受けやすいと考えられる．コサイン距離はデータ分布によらないと考えられ，また特徴量をバイナリ値に変換するとデータ分布による影響を受けにくいと考えられる．今後，原因を究明することで精度向上に努める．

また特徴量の形式が実数値と整数値の場合，CrypTen を用いて整数シェアにしたときの計算時間と認証精度への影響に特徴量の形式による差がない．そのため CrypTen を用いると特徴量が実数値のままでも良い精度を保てる．

6.3 既存研究との比較

既存研究 [34] では, 特徴量に乱数を加算することにより秘匿化し, 内積を取ることで認証しており, 計算量が秘密計算に比べて小さい．この手法では認証時にユークリッド距離を用いており，高速に比較を行っている．本稿で得られたユークリッド距離を用いた場合でも計算時間が高速であるという結果は，既存研究の結果を補強するものである．

6.4 今後の課題

今後の課題として，表 6 において平文で精度の高いコサイン距離を秘密計算で実装し，その精度を確かめることが挙げられる．本実験では，特徴量の形式が実数値と整数値の場合，コサイン距離は平方根や割り算により，正確に計算することができなかった．一方バイナリ値に変換した場合にはデータが小さいことから，平方根や割り算による誤差が小さく計算することができた．

特徴量の次元数を 512 次元としているが，これを 256 次元や 128 次元と小さくすると実数値や整数値の場合でも正確に計算できる可能性がある．一方，次元数削減による精度の低下が考えられる．また，特徴量を正規化したのちのユークリッド距離の計算がコサイン距離と同等であることから，あらかじめ特徴量を正規化した値でユークリッド距離を計算することで表すことができる可能性がある．ただし，ユークリッド距離がコサイン距離よりも速く計算できることから，精度と計算時間のトレードオフの関係が考えられる．この点を明らかにすることも今後の課題とする．

7. 結論

本稿では特徴量抽出器に機械学習を用いた秘匿顔認証を提案し，特徴量の形式と距離計算の組み合わせによる計算時間と精度への影響を評価した．その結果，精度と実行時間に優れている設定は特徴量が実数値，認証時に用いる統計的距離がユークリッド距離の場合であることがわかった．またこれらの設定ではデータセットごとに精度にばらつきがある一方，特徴量がバイナリ値である場合にはデータ分布によらず安定的な挙動を示すことがわかった．今後の課題としては特に統計的距離がコサイン距離の場合の秘匿顔認証を実装することを挙げる．

謝辞 本研究は JST CREST（課題番号：JPMJCR21M5）の支援を受けたものである．

参考文献

[1] Bschoff, P. and moody, G.: Facial recognition technology (FRT): 100 countries analyzed (2021). https://www.comparitech.com/blog/vpn-privacy/facial-recognition-statistics/.

[2] 27, I. J. S.: ISO/IEC 24745:2011, information technology - security techniques - biometric information protection (2011).

[3] Osadchy, M., Pinkas, B., Jarrous, A. and Moskovich, B.: SCiFI - A System for Secure Face Identification, *Proc. of IEEE S&P 2010*, IEEE, pp. 239–254 (2010).

[4] Kitai, H., Cruz, J. P., Yanai, N., Nishida, N., Oba, T., Unagami, Y., Teruya, T., Attrapadung, N., Matsuda, T. and Hanaoka, G.: MOBIUS: Model-Oblivious Binarized Neural Networks, *IEEE Access*, Vol. 7, pp. 139021–139034 (2019).

[5] Riazi, M. S., Samragh, M., Chen, H., Laine, K., Lauter, K. E. and Koushanfar, F.: XONN: XNOR-based Oblivious Deep Neural Network Inference, *Proc. of USENIX Security 2019*, USENIX Association, pp. 1501–1518 (2019).

[6] Byali, M., Chaudhari, H., Patra, A. and Suresh, A.: FLASH: Fast and Robust Framework for Privacy-preserving Machine Learning, *Proceedings on Privacy Enhancing Technologies*, Vol. 2, pp. 459–480 (2020).

[7] Samragh, M., Hussain, S., Zhang, X., Huang, K. and

Koushanfar, F.: On the Application of Binary Neural Networks in Oblivious Inference, *Proc. of CVPR 2021*, pp. 4630–4639 (2021).

[8] Deng, J., Guo, J., Xue, N. and Zafeiriou, S.: ArcFace: Additive Angular Margin Loss for Deep Face Recognition, *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, pp. 4690–4699 (online), DOI: 10.1109/CVPR.2019.00482 (2019).

[9] Cao, Z., Long, M., Wang, J. and Yu, P. S.: HashNet: Deep Learning to Hash by Continuation, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5609–5618 (online), DOI: 10.1109/ICCV.2017.598 (2017).

[10] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z. and Liu, W.: CosFace: Large Margin Cosine Loss for Deep Face Recognition, *Proc. of CVPR 2018* (2018).

[11] Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M. and van der Maaten, L.: Crypten: Secure multi-party computation meets machine learning, *Advances in Neural Information Processing Systems*, Vol. 34 (2021).

[12] Chamikara, M. A. P., Bertok, P., Khalil, I., Liu, D. and Camtepe, S.: Privacy Preserving Face Recognition Utilizing Differential Privacy, *Computers & Security*, Vol. 97, p. 101951 (2020).

[13] Gomez-Barrero, M., Maiorana, E., Galbally, J., Campisi, P. and Fierrez, J.: Multi-biometric template protection based on Homomorphic Encryption, *Pattern Recognition*, Vol. 67, pp. 149–163 (2017).

[14] Murakami, T., Fujita, R., Ohki, T., Kaga, Y., Fujio, M. and Takahashi, K.: Cancelable Permutation-Based Indexing for Secure and Efficient Biometric Identification, *IEEE Access*, Vol. 7, pp. 45563–45582 (online), DOI: 10.1109/ACCESS.2019.2908456 (2019).

[15] Rathgeb, C. and Uhl, A.: A Survey on Biometric Cryptosystems and Cancelable Biometrics, *Eurasip Journal on Information Security*, Vol. 2011, No. 1, pp. 1–25 (2011).

[16] Gentry, C.: Fully homomorphic encryption using ideal lattices, *In Proc. STOC*, pp. 169–178 (2009).

[17] Barni, M., Bianchi, T., Catalano, D., Raimondo, M. D., Labati, R. D., Failla, P., Fiore, D., Lazzeretti, R., Piuri, V., Piva, A. and Scotti, F.: A privacy-compliant fingerprint recognition system based on homomorphic encryption and Fingercode templates, pp. 1–7 (2010).

[18] Bianchi, T., Turchi, S., Piva, A., Donida Labati, R., Piuri, V. and Scotti, F.: Implementing FingerCode-based identity matching in the encrypted domain, *Proc. of BIMS 2010*, pp. 15–21 (online), DOI: 10.1109/BIOMS.2010.5610445 (2010).

[19] Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I. and Toft, T.: Privacy-Preserving Face Recognition, *Privacy Enhancing Technologies, 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5-7, 2009. Proceedings* (Goldberg, I. and Atallah, M. J., eds.), Lecture Notes in Computer Science, Vol. 5672, Springer, pp. 235–253 (online), DOI: 10.1007/978-3-642-03168-7_14 (2009).

[20] Yao, A. C.: How to Generate and Exchange Secrets (Extended Abstract), *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, IEEE Computer Society, pp. 162–167 (online), DOI: 10.1109/SFCS.1986.25 (1986).

[21] Sadeghi, A.-R., Schneider, T. and Wehrenberg, I.: Efficient Privacy-Preserving Face Recognition, pp. 229–244 (2010).

[22] Blanton, M. and Gasti, P.: Secure and Efficient Protocols for Iris and Fingerprint Identification, *Proc. of ESORICS 2011* (Atluri, V. and Díaz, C., eds.), Lecture Notes in Computer Science, Vol. 6879, Springer, pp. 190–209 (2011).

[23] Goldreich, O., Micali, S. and Wigderson, A.: How to Play ANY Mental Game, *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, New York, NY, USA, Association for Computing Machinery, p. 218–229 (online), DOI: 10.1145/28395.28420 (1987).

[24] Naor, M. and Pinkas, B.: Computationally Secure Oblivious Transfer, *J. Cryptol.*, Vol. 18, No. 1, p. 1–35 (online), DOI: 10.1007/s00145-004-0102-6 (2005).

[25] Ben-Or, M., Goldwasser, S. and Wigderson, A.: Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation, *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, New York, NY, USA, Association for Computing Machinery, p. 1–10 (online), DOI: 10.1145/62212.62213 (1988).

[26] Bourse, F., Minelli, M., Minihold, M. and Paillier, P.: Fast Homomorphic Evaluation of Deep Discretized Neural Networks, *Proc. of CRYPTO 2018*, LNCS, Vol. 10993, Springer, pp. 483–512 (2018).

[27] Dalskov, A. P. K., Escudero, D. and Keller, M.: Secure Evaluation of Quantized Neural Networks (2019). arXiv preprint, `http://arxiv.org/abs/1910.12435`.

[28] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R. and Bengio, Y.: Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1 (2016). arXiv preprint, `https://arxiv.org/abs/1602.02830`.

[29] Paszke, A., Gross, S., Chintala, S. and Chanan, G.: Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, Vol. 6, No. 3, p. 67 (2017).

[30] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. and Zisserman, A.: VGGFace2: A Dataset for Recognising Faces across Pose and Age, *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pp. 67–74 (online), DOI: 10.1109/FG.2018.00020 (2018).

[31] Sengupta, S., Chen, J., Castillo, C. D., Patel, V. M., Chellappa, R. and Jacobs, D. W.: Frontal to profile face verification in the wild, *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7-10, 2016*, IEEE Computer Society, pp. 1–9 (online), DOI: 10.1109/WACV.2016.7477558 (2016).

[32] FaceScrub: The FaceScrub dataset, (online), available from ⟨http://vintage.winklerbros.net/facescrub.html⟩ (2020).

[33] Ng, H.-W. and Winkler, S.: A data-driven approach to cleaning large face datasets, *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 343–347 (online), DOI: 10.1109/ICIP.2014.7025068 (2014).

[34] 肥後春菜，一色寿幸， 森健吾， 尾花賢：特徴量間のユークリッド距離を類似度とするキャンセラブルバイオメトリクス，SCIS (2022).