Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

歩容認証モデルによる学習元歩容データの推定について

江原広晃[1]　新妻 弘崇[1]　八木 康史[1]

概要：人の歩容には個人を識別できるだけの情報が含まれおり，人の歩容から個人認証が可能である.もし歩容認証システムに意図した通りの認識結果が得られるような歩容データが作成可能なら，第三者が他人に成り代わって認証を通すことができ，認証システムの信頼性に問題が生じる.このように他人に成り代わって認証できるか検証するため，本研究では歩行情報を集約したデータである歩容エネルギー画像(Gait Energy Image;GEI)から個人を識別できるように機械学習した分類器を使用して分類器の学習データを推定し，意図した分類結果が得られるような分類器への入力を作成する手法を提案する.

キーワード：セキュリティ，Model Inversion 攻撃，歩容認証，学習データの推定

Estimating Training Data From Gait Identification

HIROKI EHARA[†1]　HIROTAKA NIITSUMA[†1]
YASUSHI YAGI[†1]

Abstract: Gait contains enough information to identify the individual. If it is possible to generate gait data that is identified as intended, a third party can be identified as another person, and the reliability of the identification system will be damaged. In order to make sure whether it is possible, we estimate the training data of the classifier that has been machine trained to classify individuals from the Gait Energy Image (GEI), which is data that aggregates gait information. In this paper, we propose a method to estimate the training data of the classifier, and to generate images classified as intended by the classifier.

1. はじめに

近年の深層学習技術の急速な発展により，深層学習を利用したサービスやアプリケーションが増加している．深層学習は機械学習の一種で，訓練データから機械学習モデルを学習させて目的の出力になるように訓練する必要がある．例えば顔画像から個人を識別するアプリケーションでは訓練データとして多くの顔画像を使用して機械学習モデルを学習する必要がある．この機械学習モデルから機械学習に使用された訓練データ情報を推測する逆問題が解けるならば，様々な問題が発生し得る．例えば訓練データの顔画像を機械学習モデルから入手できたなら，顔画像は個人情報であるためプライバシーの侵害になり得る．さらには入手した顔画像を使用すると，別人を顔画像の人物として識別してしまうことができてしまう問題もある．

実際に機械学習モデルから秘匿情報を取得する技術が発見されている．例えば機械学習モデルから学習データを復元することができる Model Inversion 攻撃[1]や，訓練データの中に任意のデータが存在するか推定する Membership Inference 攻撃[2]などが存在する．そういった背景から近年 Differential Privacy という枠組みに注目が集まっている． Differential Privacy とは機械学習モデルの出力から元データが推定されないようにする仕組みである．この仕組みには，例えば機械学習モデルに結果にはほとんど影響を与えないようなノイズを混ぜることで，元データが推定されづらくするといった取り組み[3]などある．このように機械学習モデルから元データが推定できるかは重要な懸案事項となっている．

深層学習技術を利用したアプリケーションの中には個人の生体特徴を個人の識別に使用するサービスが存在する．生体特徴とは，それぞれの人が持つ個人ごとに異なる身体的特徴や個人独特の行動的特徴を指し，静脈や DNA, 指紋, 顔, 歩容など様々な生態特徴が認証システムや DNA 鑑定, 指紋認証といったものなど様々な分野で利用されている．ここで歩容による識別とは，人の歩く姿から個人を特定する歩容認証という技術[3][4]のことである．歩容認証は防犯カメラで撮影された低解像度の映像に対しても認証を行うことができるため，犯罪捜査に使用されている．イギリスで 2008 年に歩容認証による認証結果が証拠として認められ[5]，日本では 2009 年に裁判所にて歩容鑑定結果が証拠として認められたという事例がある[6]．このように歩容は個人を識別するのに十分な生体的特徴として利用されている．もし第三者が生体認証において本人と同じような身体的特徴を持ったデータを取得可能ならば，第三者が本人に成り代わって認証を通してしまう危険性がある．よって認証システムから訓練データを推定できるかどうかは重要な懸念事項である．

本研究では，図 1 のように歩容認証モデルから狙った

1 大阪大学
Osaka University

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

個人の歩容データを推定し，その推定歩容データが歩容認証モデルに狙った通りの人と認識されうるのか検証した．歩容認証モデルとしては後述する Gait Energy Image (GEI) を入力したときに個人を識別しその人の識別子である ID を出力するように学習した分類器を想定する．分類器の出力は入力画像の歩容の持ち主として最も確信度が高い人の ID のみ出力することを想定している．この歩容認証モデルから訓練データに含まれる個人情報を推定したい攻撃者は，訓練データの推定のために分類器に入力を与えたときの出力情報のみ取得できるものとする．図 2　訓練データの概要図のように攻撃者は訓練データの情報は保持していないものとする．

本論文の構成を以下に示す．まず 2 章で本研究の関連研究について紹介する．次に 3 章で本研究で参考にした先行研究について紹介する．4 章にて本研究で分類器への攻撃で使用した提案手法について解説し，5 章では分類器への攻撃実験とその結果を示す．最後に 6 章では本研究のまとめについて述べる．

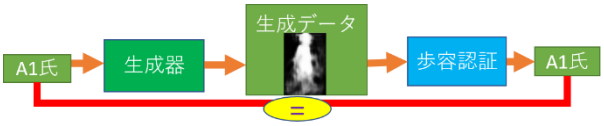

図 1　歩容認証モデルへの攻撃の模式図
Figure 1　Schematic of an attack on a gait identification model

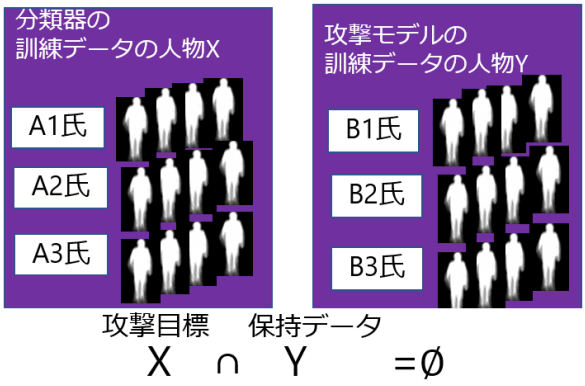

図 2　訓練データの概要図
Figure 2　Schematic of training data

2. 関連研究

2.1 Model Inversion Attack

Model Inversion Attack は分類器の学習元データを推定する攻撃で，特定のラベルに対するその分類器の分類確率が最大になるような入力を生成することで学習元データを推定する．Fredrikson らは遺伝情報から投薬量を推定する線形分類器を利用して，分類器による推定結果である投薬量から遺伝情報が推測できることを示した**[7]**．これを例にすると，特定の投薬量の分類確信度が最大となる入力を探索したとき，その入力は分類器が学習した特定の投薬量に対する教師データの代表であり，特定の遺伝情報を表している．Fredrikson らは決定木に対しても入力の推定が可能であることを示し**[1]**，また Zhang らは深層学習モデルに対しても Generative Adversarial Network(以降 GAN と呼称)を使用することで Model Inversion Attack が可能であることを示した**[8]**.

本研究の想定する攻撃との主な違いは想定している分類モデルの出力の違いである．Model Inversion Attack は攻撃対象の分類モデルの出力として分類ラベルに対する確信度といった数値を使用している．しかし本研究では攻撃対象の分類モデルからは入力に対する一番確信度が高い分類ラベルのみ得られることを想定しており，Model Inversion Attack よりも分類モデルから得られる情報が少ない．

2.2 Membership Inference Attack

攻撃対象の分類器の入力に対する出力の分類結果を観察することによって入力データが分類器の学習データに含まれているかどうかを判別する攻撃を Membership Inference Attack という[2]．　この攻撃では攻撃者は分類器の内部構造を知らないという前提で学習データであるかを判別する．学習データに含まれるデータを分類器に入力した場合に出力である分類結果の確信度は，学習データに含まれないデータを入力にしたときと比較すると，大きくなる傾向がある．これを利用して，任意の入力を与えて分類結果の確信度から任意のデータが攻撃対象の分類器の学習データに含まれていたかを推定することができる．Membership Inference Attack を利用することで秘匿されるはずのテストデータ情報が漏洩する危険性があり，プライバシーが侵害されることが懸念されている．

本研究の目的と Membership Inference Attack とは学習データを推定するという文脈では似ているが，Membership Inference Attack では任意のデータが学習データに含まれているかいないかの判別情報が目的であり，本研究ではこの推定は行わない．

3. 関連研究

ここでは本研究の関連研究について述べる．

3.1 PreImageGAN

草野らは MNIST の手書き数字データを訓練データとした数字分類器から PreImageGAN という学習モデルを使用して,「攻撃者が予測ラベルのサンプルを知識として全く持っていない場合であっても，補助データを活用することにより，知識として持たないラベルのサンプルの生成分布を推定することができること」[9]を示した．PreImageGAN は Wasserstein GAN(WGAN)と conditional GAN(cGAN)モデルを応用して構成されており，WGAN や cGAN は GAN の構造が基盤になっている．ここでは PreImageGAN の構造を GAN と WGAN と cGAN の説明を通じて解説し，本研究の提案手法は Auxiliary Classifier GAN(ACGAN)を参考に作成

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

しているため，ACGAN についても説明をしていく．

3.1.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks(GANs)は敵対的生成ネットワークと訳される，これは Goodfellow らが提唱した機械学習による生成モデルの設計手法[10]である．GANs は Goodfellow らの提案した GAN(Vanilla GAN)のモデルを元に構成され，図 3 のように Generator と Discriminator の二つの学習ニューラルネットワークから構成される．以後図中では Discriminator を D, Generator を G と表記する．Vanilla GAN では Generator は潜在ベクトルと呼ばれる n 次元ベクトルを入力としてデータを生成し，Discriminator は自身に入力されたデータが Generator の生成したデータか訓練データかを判定する． Generator は Discriminator が誤認識するよう訓練データと同じデータが生成出来るように学習し，Discriminator は自身が正確に認識できるように学習するため，敵対的に学習することになり，Generator と Discriminator の精度が学習を重ねる毎に高まる．そうして Vanilla GAN は訓練データに近いデータを生成出来るようになる．Vanilla GAN の目的関数は(1)となり，Discriminator が利益を最大化させると同時に Generator が Discriminator の利益を最小化させようとすることが表されている．

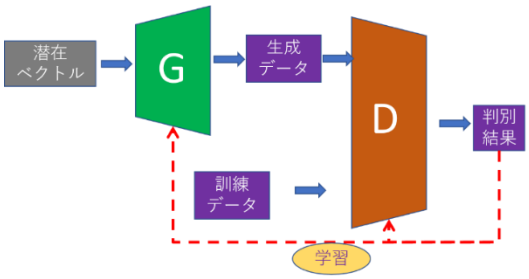

図 3 Vanilla GAN のモデル構造

Figure 3 Model structure of Vanilla GAN

$$\min_G \max_D V(D,\ G) = E_{x\sim d_x}\left[\log\left(D(x)\right)\right] + E_{z\sim d_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{1}$$

3.1.2 Wasserstein GAN(WGAN)

Wasserstein GAN(WGAN)の目的関数は(2)で表される．Vanilla GAN では mode collapse という生成データが類似するという問題が起こりやすい[11]．WGAN は Wasserstein 距離を用いた目的関数を使用することで mode collapse 問題を緩和している．目的関数中の$||D||_L \leq 1$は D が 1-Lipschitz を満たすことを意味している[12].

さらに WGAN の損失関数に Gradient Penalty の項を追加して改良したモデルとして WGAN-gp が提唱されており[13]，このモデルでは生成データと訓練データから合成した中間データを Discriminator に入力することで Gradient Penalty を計算する．

$$\min_G \max_{||D||_L \leq 1} V(D,\ G) = E_{x\sim d_x}[D(x)] - E_{z\sim d_z}\left[D\left(G(z)\right)\right] \tag{2}$$

3.1.3 conditional GAN

cGAN では Generator に潜在ベクトルの他に生成条件となる条件ベクトルとなるラベルを追加し，Discriminator への入力に生成データ又は訓練データの他に入力データに対応するラベルを入力する．これにより Generator はラベルに応じたデータを生成し，Discriminator は入力されたデータがラベルに対応する訓練データかを判別できるようにしている[14]．cGAN の目的関数は (3)で表され，Vanilla GAN の目的関数の G と D への入力に条件ベクトルとなる c を追加している．これにより Generator の生成画像をラベルで制御することが出来る．

$$\min_G \max_D V(D,\ G) = E_{(x,\ c)\sim d_{x,\ c}}\left[\log\left(D(x,\ c)\right)\right] + E_{z\sim d_z,\ c\sim d_c}\left[\log\left(1 - D\left(G(z,\ c),\ c\right)\right)\right] \tag{3}$$

3.1.4 PreImageGAN

図 4 PreImageGAN のモデル構造を図 4 に示す．この図のように PreImageGAN は，WGAN-gp と cGAN とを合体して，Discriminator への入力にラベルを除外し，Generator に入力するラベル情報として訓練データを攻撃目標の分類器(Classifier 又は C と表記)に入力したときの出力の分布を使用している．この出力の分布を条件としてラベル代わりに使用することで，訓練データに対応するラベルが不明な時，又は分類器が分類するラベル情報が不明な時でも学習することを可能にしている．さらに，Generator に入力するラベルと，その入力ラベルに応じて Generator によって生成されたデータを Classifier に入力して得られた出力の分布とを比較し差を算出する項が Generator の損失関数に追加され，この二つの分布が一致するように Generator は学習する．この構造により，Generator は GAN 特有の敵対的学習により訓練データによく似たデータを生成出来るようになると同時に，Generator は Generator への入力ラベルとその結果生成されたデータを Classifier によって分類された結果が等しくなるように学習することが出来る．Generator は Classifier の入力と出力を逆転させた入出力になり，よって Generator は Classifier の出力ラベルから Classifier の入力を生成することができる．これはつまり Generator は Classifier の学習データを生成することができるという原理であり，この原理に基づいて PreImageGAN モデルは構成されている．なお図中の Classifier は攻撃対象の分類器であるため，PreImageGAN の学習においては Classifier が学習することはない．

PreImageGAN の目的関数は (4)で表される．式中の C は Classifier に相当し，入力データ x を引数にして Classifier による分類結果のラベルを出力する関数 C(x)である．右辺の第一と第二項は cGAN の項で，cGAN の条件ベクトルとし

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

て Classifier による分類結果 C(x)を使用している. 式の第三項は Generator への入力の条件ベクトルとその結果生成されたデータを Classifier に入力して得られた分類結果 C(G(x))とが近くなるようにするための項で lambda はこの項の強さを調整するためのパラメータである[9].

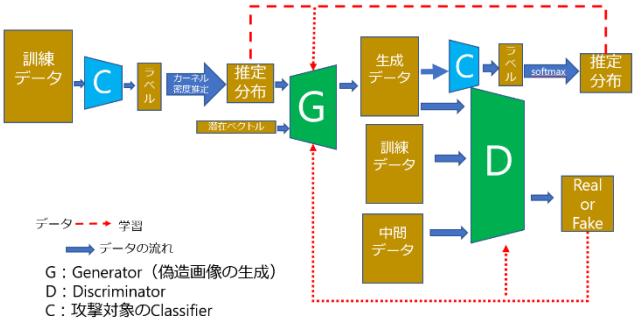

図 4 PreImageGAN のモデル構造

Figure 4 Model structure of PreImageGAN

$$\min_{G} \max_{\|D\|_L \le 1} V(D,\ G) = E_{(x,\ c)\sim d_{x,\ c}}[D(x,\ c)] - E_{z\sim d_z,\ y\sim d_y}\left[D\left(G(z,\ y),\ C(G(z,\ y))\right)\right] + \lambda E_{z\sim d_z,\ y\sim d_y}\left[l\left(C(D(G(z,\ y)),\ y)\right)\right] \tag{4}$$

3.1.5 Auxiliary Classifier GAN(ACGAN)

cGAN の Discriminator の出力を入力データが訓練データか生成データかの判別だけでなく入力データがどのラベルに分類されるかの情報も出力するようにして, Discriminator がラベル分類結果が正しくなるように学習するモデルが Auxiliary Classifier GAN(ACGAN)である. つまり ACGAN では Discriminator は訓練データの真偽判別のみならず分類器の機能も果たす. ACGAN ではさらに Generator は自身への入力ラベルとその結果 Generator によって生成されたデータが Discriminator によって分類された結果とが同じになるように学習する. ACGAN の Discriminator と Generator の目的関数はそれぞれ(7), (8)で表される. (5)は cGAN から来る項であり, (6)はラベル分類の項である. この構造にすることによって Generator へのラベル入力による生成データの制御が容易になり, 以前の GANs より高精度な画像を生成出来る[15].

$$L_S = E_{x\sim d_x}[\log(D(x))] + E_{z\sim d_z,\ y\sim d_y}\left[\log\left(1 - D\left(G(z,\ y)\right)\right)\right] \tag{5}$$

$$L_C = E_{x\sim d_x,\ y\sim d_y}[similarity(D_C(x),\ y)] + E_{z\sim d_z,\ y\sim d_y}\left[similarity\left(D_C\left(G(z,\ y)\right),\ y\right)\right] \tag{6}$$

$$V_D(D,\ G,\ D_C,\ C) = \max(L_S + L_C) \tag{7}$$

$$V_G(D,\ G,\ D_C,\ C) = \max(-L_S + L_C) \tag{8}$$

3.2 Gait Energy Image (GEI)

歩行映像をデータとして機械学習を行うとき, 映像から歩容特徴を抽出したデータを使用する方法がよく採用されている. 歩行映像から歩容特徴を抽出した表現方法として視覚的特徴から抽出する方法があり, その抽出方法の一つに Gait Energy Image(GEI)と呼ばれる表現方法がある. GEI は歩行者のシルエット 1 周期分の画像列を平均化して得られる[16].

歩容特徴を抽出した表現は, 歩容からの年齢推定[17]や性別の推定[18]など, あらゆる歩行関連の研究に使用され, 歩容による生体認証や犯罪捜査における個人特定のための歩容認証などにも使用されている[5] [19].

4. 提案手法

本研究の目的は, 歩容認証モデルの学習データを推定することができるかの検証である. 歩容認証モデルとは歩容特徴を抽出した画像である GEI を入力に与えたときに, 出力として GEI の歩容の主を示す識別子を出力するよう深層学習した分類器(Classifier)のことを指す. 学習データを推定できたとは分類器に狙ったラベルで分類されるようなデータの作成に成功したこととする.

本章では本研究で用いた手法について説明する.

4.1 Simplified PreImageGAN (SPIGAN)

分類器の学習データを推定する手法として PreImageGAN を簡略化して作成した Simplified PreImageGAN (SPIGAN)を提案する. SPIGAN の構造を図 5 に示す. この図で示すように SPIGAN は前節で述べた PreImageGAN の WGAN-gp モデルといくつかの関数を外して, ACGAN モデルを追加したモデルである. 本研究の想定では Classifier の出力から得られる情報が入力に対するラベル分類の確信度ではなく確信度が最も高いラベルのみであり, 情報量が少ない. このため ACGAN モデルを追加することで, Discriminator が Classifier と同等の分類器になるように学習し, 結果 Discriminator が Classifier の代わりに分類確信度を出力することで, Classifier で得られなかった確信度情報を補う. WGAN-gp を外した分, Classifier への攻撃モデルの学習が不安定になることを避けるため, Generator や Discriminator の損失関数として Hinge 関数を使用し学習の安定化を図った.

4.2 SPIGAN for MNIST

MNIST の手書き数字データセットの数字分類器に対する推定攻撃を行うために使用した手法である Simplified PreImageGAN for MNIST(SPIGAN for MNIST)について説明する.

SPIGAN for Mnist の Classifier には MNIST の手書き数字データセットの画像を分類できるように学習済みの Classifier を用いる. 手書き数字データセットには 0 から 9

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

までの 10 個の整数が手書きされた画像が多数含まれ，Classifier はそれらの画像を入力されたときに入力画像が 0 から 9 までのどの数字であるかを判定し出力する．今回は訓練データを用意して SPIGAN for Mnist を学習させた後，学習済みの SPIGAN for MNIST の Generator へ ID を入力して画像を生成することで，目的の Classifier に狙ったラベルで識別される画像を得る．

SPIGAN for MNIST の Discriminator と Generator の目的関数はそれぞれ(12)，(13)で表される．式中の C は Classifier に相当し，入力データ x を引数にして Classifier による分類結果のラベルを出力する関数 C(x)である．式中の D_C は Discriminator の分類機能に相当し，入力データ x とデータに対応するラベル C(x)を入力として Discriminator による分類結果のラベルを出力する関数である．(9)は cGAN の目的関数のうち，条件ベクトルとして訓練データを Classifier に分類された結果を使用し，Hinge 関数を適用した項である．この項により与えられた訓練データに近いデータを生成するように学習する．SPIGAN はこの学習によって目標データに近いデータを生成することが狙いであるため，訓練データには推定目標データに近いものを使用する必要がある．(11)の第二項は PreImageGAN の Generator が Classifier に Generator への入力と同じラベルで判別されるようにする項である．(11)の第一項と(10)は ACGAN の Discriminator の分類結果が Classifier の分類結果に近づくよう，また Generator が Discriminator に Generator への入力と同じラベルで判別されるように学習する項である．(10)と(11)の λ, μ, νはそれぞれ学習を調節するパラメータで，similarity の算出には全て Hinge 関数を用いた．

$$L_S = E_{x\sim d_x}[\min(0,-1+D(x))] - E_{x\sim d_x, z\sim d_z}\left[\min\left(0,-1+D\left(G(z,C(x))\right)\right)\right] \tag{9}$$

$$L_{C_G} = \lambda E_{x\sim d_x, z\sim d_z}\left[similarity\left(D_C\left(G(z,C(x))\right),C(x)\right)\right] \tag{10}$$

$$L_C = \mu E_{x\sim d_x, z\sim d_z}[similarity(D_C(x),C(x))] + \nu E_{x\sim d_x, z\sim d_z}\left[similarity\left(C\left(G(z,C(x))\right),C(x)\right)\right] \tag{11}$$

$$V_D(D,\ G,\ D_C,\ C) = \max(L_S + L_C) \tag{12}$$

$$V_G(D,\ G,\ D_C,\ C) = \max(-L_S + L_C + L_{C_G}) \tag{13}$$

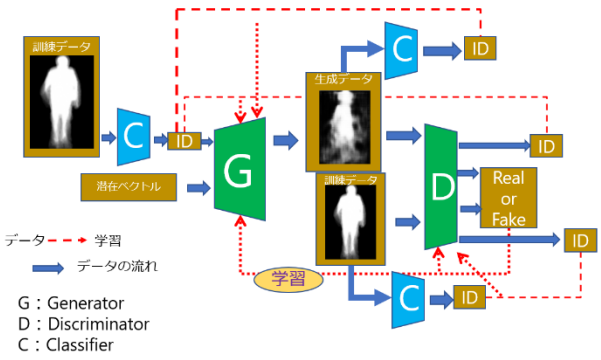

図 5　SPIGAN for Mnist のモデル構造

Figure 5　Model structure of SPIGAN for MNIST

4.3 SPIGAN for GEI

入力された GEI が誰のものかを分類する分類器に対する推定攻撃を行う際に本研究で使用した手法である Simplified PreImageGAN for GEI(SPIGAN for GEI)について説明する．誰の歩容であるかという個人の識別符号としてそれぞれの被験者には ID が割り振られており，SPIGAN for GEI の Classifier には GEI を分類し ID を出力できるように学習済みの Classifier を用いる．SPIGAN for MNIST では学習済みの SPIGAN for MNIST の Generator へ ID を入力して画像を生成することで目的の Classifier に狙ったラベルで識別される画像を得ていた．しかし同じ方法では GEI 分類器への推定攻撃では学習速度が遅かったため，Generator による生成画像が Classifier によって Generator への入力ラベルと同じラベルと判別された画像を保存するモデルである Storage を用意することで対処した．Storage は図 6 のように ID 毎の攻撃に成功した画像を保管し，ID が Storage に入力されると，保存された ID に対応する画像を出力するモデルである．SPIGAN の学習時，epoch 毎に Generator へのラベル入力として 0，1，2…と ID を入力していき，その生成画像の Classifier による分類結果が入力 ID と等しくなった画像を全て Storage に格納する．これにより，各ラベルにおいて一度でも攻撃に成功したらよくなり，Generator が一度に全ての入力ラベルに対し攻撃が成功する画像を生成できるように学習する必要がなくなるため，より高い確率での攻撃を可能にした．

SPIGAN for GEI の Generator の目的関数は(12)，Discriminator の目的関数は(13)である．SPIGAN for MNIST との GAN 内部での大きな違いは図 7 のように Discriminator への入力として条件ベクトルが増えたという点である．この入力は Discriminator の分類器としての精度を上昇させ，Classifier では秘匿されているラベル分類の確信度情報をより得やすくするためのものである．この変更点により(9)，(10)，(11)はそれぞれ(14)，(15)，(16)に置き換わる．

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

$$L\_S = E_{x\sim d_x}[\min(0, -1 + D(x))] - E_{x\sim d_x, z\sim d_z}\left[\min\left(0, -1 + D\left(G(z, C(x)), C\left(G(z, C(x))\right)\right)\right)\right] \quad (14)$$

$$L_{C_G} = \lambda E_{x\sim d_x, z\sim d_z}\left[similarity\left(D_C\left(G(z, C(x)), C\left(G(z, C(x))\right)\right), C(x)\right)\right] \quad (15)$$

$$L_C = \mu E_{x\sim d_x, z\sim d_z}\left[similarity\left(D_C(x, C(x)), C(x)\right)\right] + \nu E_{x\sim d_x, z\sim d_z}\left[similarity\left(C\left(G(z, C(x))\right), C(x)\right)\right] \quad (16)$$

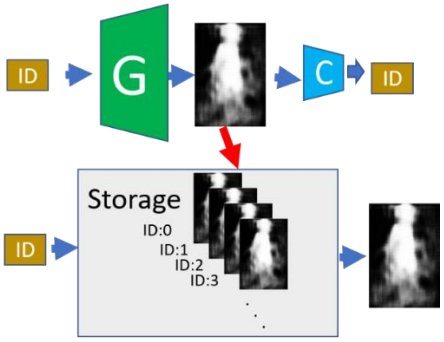

図 6　SPIGAN for GEI の Storage の概略図

Figure 6　Schematic diagram of SPIGAN for GEI's Storage Model

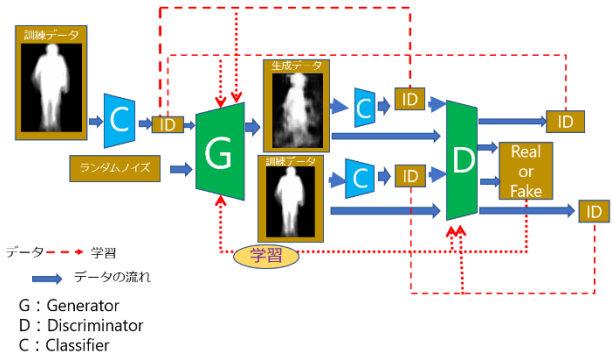

図 7　SPIGAN for GEI のモデル構造

Figure 7　Model structure of SPIGAN for GEI

5. 実験

深層学習で得られた分類器について，どの程度分類器の学習データを推定できるかを本研究の提案手法を用いて実験で確かめる．ここでは学習データを推定できたとは分類器に狙ったラベルで分類されるようなデータを作成することとする．学習データが推定できている度合いは，分類器が分類するクラス数のうちどれほどの割合のラベルを分類器に狙って分類させるか(偽造成功率)で計測する．

5.1 実験設定

本研究の提案手法の推定精度の検証を行うため，まず MNIST の手書き数字データセットを Classifier の訓練データとして使用して学習させ，その後 SPIGAN の訓練データに Classifier の訓練データとは異なる MNIST の画像や EMNIST の手書き英字データセットを使用して SPIGAN の学習を行うことで，MNIST データセットを分類する分類器に対する推定精度を検証する．ただし EMNIST の手書き数字データセットは使用しない．

次に，GEI を Classifier の訓練データとして学習させ，その後 SPIGAN の訓練データに，Classifier 訓練データに出現しない人物の GEI を使用して SPIGAN の学習を行うことで，GEI を分類する分類器への推定精度を検証する．

なお，MNIST の Classifier の MNIST データセットの手書き数字画像の分類成功確率は 98.9%で，GEI の Classifier の分類成功率は，10 クラス分類で 98.8%，110 クラス分類で 95.5%であった．いずれの場合も攻撃者が Classifier の出力で得られる情報は入力画像に対して最も分類の確信度が高いラベルのみである．GEI は OU-MVLP データセット[1]の GEI を使用している．

5.2 実験結果

MNIST データセット分類器を使用した場合の結果は図 9，図 10 で，それぞれ順に SPIGAN の訓練データとして EMNIST データセット，MNIST データセットを使用した．

GEI の分類器を使用した場合の結果は，図 8，図 11，図 12，図 13 であり，そのうち 10 クラス分類器を使用していた結果が図 8 で，110 クラス分類器を使用して SPIGAN のデータ数を 36 枚にした結果が図 11 と図 12，データ数を 364 枚使用した結果が図 13 である．

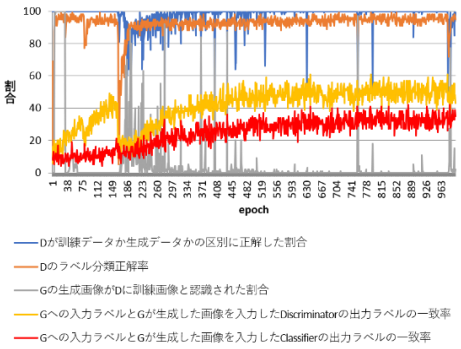

図 8　訓練データを GEI にした時の 10 クラス分類における SPIGAN for MNIST の学習進行度

Figure 8　Learning progress of SPIGAN for MNIST in 10-class classification when the training data is GEI.

1 http://www.am.sanken.osaka-u.ac.jp/BiometricDB/GaitMVLP.html

Vol.2021-NL-248 No.16
Vol.2021-CVIM-226 No.16
2021/5/21

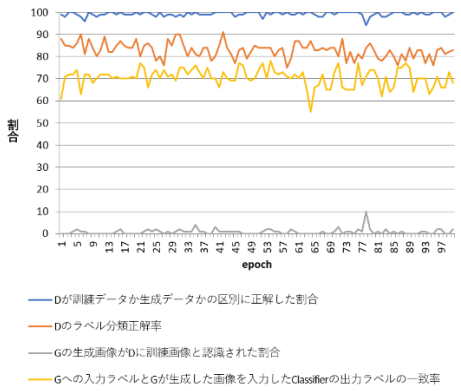

図 9　訓練データを EMNIST にした時の 10 クラス分類における SPIGAN for MNIST の学習進行度

Figure 9　Learning progress of SPIGAN for MNIST in 10-class classification when the training data is EMNIST.

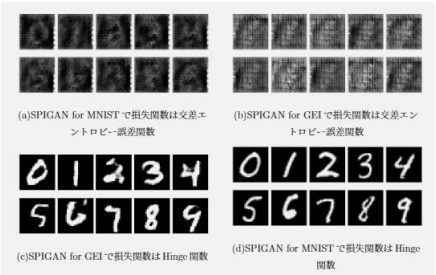

図 10　訓練データを MNIST データセットにした時のそれぞれの場合について 100 epoch 時での生成画像

Figure 10　Generated images at 100 epochs for each case when the training data is the MNIST dataset.

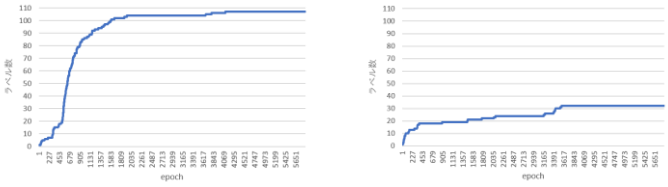

図 11　訓練データとして GEI36 枚使用した時の 110 クラス分類における SPIGAN が偽造成功した総ラベル数(損失関数は左が Hinge 関数，右が交差エントロピー誤差関数)

Figure 11　Total number of labels successfully generated by SPIGAN in 110 classifications when 36 GEIs were used as training data (left: Hinge function, right: cross-entropy error function is used as loss function).

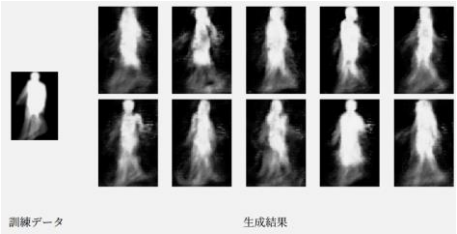

図 12　訓練データを GEI36 枚にした時の 110 クラス分類における SPIGAN for GEI の画像生成結果と訓練データの GEI

Figure 12　GEIs of training data and generated images by SPIGAN for GEI　when there are 36 training GEIs and 110 class classifier.

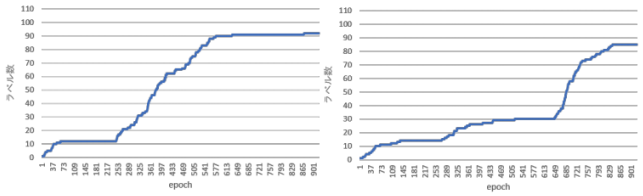

図 13　訓練データとして GEI364 枚使用した時の 110 クラス分類における SPIGAN が偽造成功した総ラベル数
(左：SPIGAN for GEI，右：SPIGAN for MNIST)

Figure 13　Total number of labels successfully generated by SPIGAN in 110 classifications when 364 GEIs were used as training data.(left: SPIGAN for GEI, right: SPIGAN for MNIST)

5.3 考察

10 クラス分類では，Generator が Generator への入力ラベルと同じラベルと Classifier に分類された割合である図 8 の赤線は 40%弱で推移しており，図 9 の EMNIST を訓練データに使用した分類における同確率と比較すると低い．EMNIST の画像は異なるラベルの画像間の見た目の差が顕著であるのに対し，GEI は異なる ID の人の GEI の違いを人の目で見てもわかりづらい．これゆえに EMNIST と比較して Generator による推定精度が悪い可能性が考えられる．

EMNIST を SPIGAN の訓練データに用いた場合，図 10 のように SPIGAN for GEI を使用するよりも，SPIGAN for MNIST を使用した方が，Generator の損失関数に交差エントロピー誤差関数を用いるよりも，Hinge 関数を用いた方が，MNIST 分類器への推定攻撃の結果生成された画像としてはきれいな画像を生成出来ている．よって MNIST 分類器への推定攻撃には Hinge 関数を使用した SPIGAN for MNIST が実験結果中では最適であると言える．

実験結果には記述しなかったが，SPIGAN の訓練データとして 364 枚の GEI を使用し，110 クラス分類器を使用した場合では平均推定成功率 2.2%と低い．このように Generator による生成のみではあまり良い精度が得られなかった．しかし Storage を使用した偽造成功率は SPIGAN for GEI を使用した時 110 ラベル中 92 ラベル推定成功で約 83%となった．目的の分類器の出力は Storage モデルによってほとんどコントロールすることができると言える．

訓練データの GEI を 36 枚まで減らした場合では図 11 の Hinge 関数使用時の偽造成功した総ラベル数は最終的に 110 ラベル中の 107 ラベルにおいて偽造成功しており，偽造成功率は約 97%となった．このことから訓練データを Classifier に入力した時に得られるクラス数が Classifier の分類クラス数と大きく離れている，つまり Generator への入力ラベルとして存在しないラベルの GEI を Generator は生成出来ていることになる．訓練データ 364 枚の時と比較して訓練データ数が少ないため学習時 1epoch 当たりの所

要時間は短くなり，偽造成功率が 80%を超えるまでに学習にかかる時間は訓練データ 364 枚使用しているときは 1 時間 42 分であったのに対し，訓練データが 36 枚の時には 1 時間 2 分と大きく短縮された．この生成された GEI が図 12 の右の画像で，生成画像が訓練データに似ており，画像によっては生成されたものか訓練データかは人の目では見分けられないものもある．このように GEI の生成には見ため成功していると言える．

図 11 から Discriminator と Classifier の分類結果が Generator への入力と同じになるように Generator が学習する項の損失関数を交差エントロピー誤差関数にした場合は Storage を使用しても偽造成功率が低くなっており，SPIGAN for GEI と Storage 使用時でも損失関数は Hinge 関数を使用する方が良いことがわかる．図 13 によると SPIGAN for MNIST より SPIGAN for GEI の方が偽造成功率が 80%を超えるまでにかかる epoch 数が小さい．これは Discriminator への入力としてラベル情報が付加されたことで，Discriminator のラベル分類精度が少ない epoch 数でもよくなり，結果偽造成功率の上昇も速くなったからであると考えられる．

6. おわりに

本論文では，Simplified PreImageGAN(SPIGAN) for GEI を使用して Gait Energy Image(GEI)による歩容認証モデルとして GEI の分類器への推定攻撃を行い，攻撃精度の検証を行った．SPIGAN の Generator をこの生成器として使用した場合では攻撃はほとんど成功しなかったが，Storage モデルを生成器として使用することで，110 クラス分類する分類器に対して，攻撃者が保持する訓練データに分類器の学習データの歩容の主の GEI が含まれていない，かつ訓練データが少ない場合でも，ほぼ 100%の偽造成功率を出すことができた．これにより，GEI の分類器の分類結果をコントロールできるような入力を生成出来ることがわかった．今回は歩容認証システムとして歩容データの分類器を使用したが，今後は個人を認証するシステムとしてより現実的なシステムに対する攻撃で狙った人と認識されるようなデータを作成できるのか検証を行う予定である．

謝辞

本研究に際し,研究やその他様々な場面で御世話になりました大阪大学産業科学研究所八木研究室の皆様方に感謝いたします．

参考文献

[1] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart: Model inversion attacks that exploit confidence information and basic countermeasures, pp. 1322–1333, 2015.
[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov: Membership inference attacks against machine learning models, IEEE Symposium on Security and Privacy (SP), pp. 3–18, 2017.
[3] 五十嵐大,高橋克巳：注目のプライバシーdifferential privacy, コンピュータ ソフトウェア, Vol. 29, No. 4, 2012.
[4] Mark S. Nixon, Tieniu N. Tan, and Rama Chellappa: Human Identification Based on Gait (The Kluwer International Series on Biometrics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
[5] Yasushi Makihara, Darko S Matovski, Mark S Nixon, John N Carter, and Yasushi Yagi: Gait recognition: Databases, representations, and applications, Wiley Encyclopedia of Electrical and Electronics Engineering, pp. 1–15, 1999.
[6] I. Bouchrika, M. Goffredo, J. Carter, and M. Nixon: On using gait in forensic biometrics, Journal of Forensic Sciences, Vol. 56, No. 4, pp. 882–889, 2011
[7] 黒沢健至：防犯カメラ映像の解析技術，セイフティ エンジニアリング, Vol. 185, pp.21‐25, Dec. 2016.
[8] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart: Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing, In23rd USENIX Security Symposium (USENIX Security 14), pp.17–32, San Diego, CA, August 2014.
[9] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song: The secret revealer: Generative model-inversion attacks against deep neural networks, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[10] 光亮草野,淳佐久間：Generative adversarial networks を用いた深層学習モデルに対する concept extraction 攻撃，コンピュータセキュリティシンポジウム 2017 論文集,第 2017 巻, 2017.
[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio: Generative adversarial networks, arXiv preprint arXiv:1406.2661, 2014.
[12] Martin Arjovsky and Léon Bottou: Towards principled methods for training generative adversarial networks, 2017.
[13] Martin Arjovsky and Léon Bottou: Wasserstein gan, 2017.
[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville: Improved training of wasserstein gans, CoRR, Vol. abs/1704.00028, 2017.
[15] Mehdi Mirza and Simon Osindero: Conditional generative adversarial nets. 2014.
[16] Augustus Odena, Christopher Olah, and Jonathon Shlens: Conditional image synthesis with auxiliary classifier gans, In International conference on machine learning, pp. 2642–2651. PMLR, 2017.
[17] Ju Man and Bir Bhanu: Individual recognition using gait energy image, IEEE Trans-actions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 2, pp. 316–322, 2006.
[18] Yasushi Makihara, Mayu Okumura, Haruyuki Iwama, and Yasushi Yagi: Gait-based age estimation using a whole-generation gait database, pp. 1–6, 2011.
[19] 万波秀年,槇原靖,八木康史：歩容における性別・年齢の分類と特徴解析.電子情報通信学会論文誌 D, Vol. 92, No. 8, pp. 1373‐1382, 2009.
[20] Niels Lynnerup and Peter Kastmand Larsen: Gait as evidence, Biometrics, Vol. 3, No. 2, pp. 47–54, 2014