Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

GANに基づくスタイル変換による生成データを用いたジェスチャ認識の精度向上

鈴木 乃依瑠[1,a)]　渡辺 祐貴[1,b)]　中澤 篤志[1,c)]

概要：センサデータからのジェスチャ認識は，モバイル・ウェアラブル技術において必須の技術である．機械学習技術の発達に伴い，近年では学習データに基づくジェスチャ認識が主流となっている．一方，ジェスチャに伴う動作データはユーザ依存性が高いため，高い認識性能を発揮するためにはユーザ毎のサンプルデータを事前に得るなどが必要になることが問題だった．本研究では敵対的学習を用い，ユーザの少クラスのジェスチャのセンサデータから未知のジェスチャのセンサデータを生成し，識別器を学習することで高精度な認識を行う手法を提案する．具体的には，GAN に基づくスタイル変換ネットワーク (Style Transformer Network) を用いてあるユーザの少クラスの動作サンプルから多クラスの動作サンプルを生成し，識別器の学習に用いる．実データを用いた実験において，既存手法で 70%の認識精度が提案手法によって 86%に向上することを確認し，提案手法の有用性を示した．

1. Introduction

センサデータからのジェスチャ認識は，モバイル・ウェアラブル技術において必須の技術である．ジェスチャ認識では，より高い認識精度を実現したり，より多くのジェスチャクラスを認識できるようにすることで，システムの信頼性や可用性を高めることが可能になる．

一方，認識の高精度化と多クラス化の両方を同時に実現するのは本質的に困難な問題である．これは図 1 に示すように，同じジェスチャであっても，異なるユーザの動作データは特徴空間中の差異や重なりがあるため，高精度かつ汎化した識別器を作るには，各ジェスチャの特徴空間を大きく取る必要があり，これはジェスチャの多クラス化に排反するからである．逆にジェスチャ種類を増やすと個々の特徴空間中の領域が小さくなるため識別境界を細かく設定する必要があるが，それは個人差を許容しない境界になることを意味する．

これを解決するためには，ユーザ毎の識別境界を設定することが考えられる [1]．図 1 の例では，現在の全ユーザに対応した識別境界 (General boundary) とは若干異なる新たな境界 (User 1 boundary, User 2 boundary) を学習することでより良い分離が可能になることがわかる．一方でこのような境界を得るためには，ユーザの全ジェスチャデータを得，再学習を行う必要（ユーザキャリブレーション）があるが，ユーザには大きな負担となり可用性を低下させる．

これに対し我々は，ユーザの一部のジェスチャデータが欠損した状況を想定し，その欠損データを他のデータから生成することを考える．これが実現できれば，ユーザの全ジェスチャデータを取得する必要がないため，可用性が大幅に向上する．

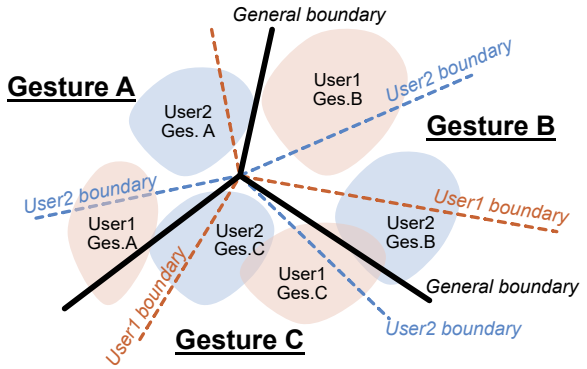

図 1　ジェスチャ認識でのデータの個人差とジェスチャの識別境界．データには個人差があるので，全ユーザに対応した境界 (General boundary：黒線) では区別できないジェスチャ／ユーザが生じる．そのため，ユーザごとのデータを学習することで，より精度の良い識別が実現できる (User 1 boundary, User 2 boundary)．一方で，あるユーザに対応した識別境界を他のユーザに使用すると，一般的に性能は低下する．また同様に，ジェスチャの種類を多くすると，その変動が個人差と識別不可能になり認識精度は一般的に低下する．

[1] 京都大学
Kyoto Uniersity
[a)] n-suzuki@ii.ist.i.kyoto-u.ac.jp
[b)] watanabe@ii.ist.i.kyoto-u.ac.jp
[c)] nakazawa.atsushi@i.kyoto-u.ac.jp

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

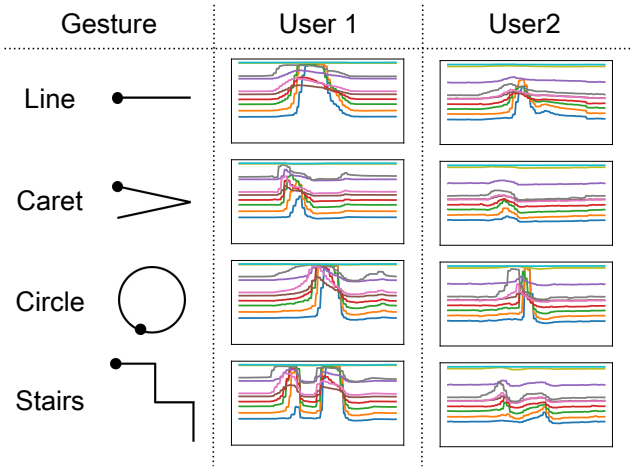

図 2　ジェスチャデータの一部．4 クラスのジェスチャに対する User1 と User2 の動作データ．グラフの異なる色が異なるセンサを表す．ここからジェスチャクラスごとに共通の特徴が見られることや，同じジェスチャでもユーザごとの個性差が出ていることがわかる．

ジェスチャデータの変換には，近年画像のスタイル変換で注目されている敵対的学習 (Generative Adversarial Network (GAN)) を用いた**スタイル変換ネットワーク (Style Transfer Network)** の一種である StarGAN を用いる．これは，異なるユーザでも同じクラスのデータには共通の特徴がある，あるいは異なるクラスのデータでも個人の個性が反映されていると考えられるからである．図 2 の例では，ユーザに関係なくジェスチャクラス Line は Circle に比べてセンサー値の山が時間的に短く，Stairs には山が二つ見られることがわかる．また，User2 のデータの変化は User1 よりも小さく時間的には短いことが確認できる．つまり，このようなジェスチャクラスごと，ユーザごとの傾向をスタイル変換ネットワークが学習し動作データの変換を行うことで，未知のデータの生成を行うことが可能になる．

敵対的学習ではデータを生成する Generator と，その結果の real/fake を見分ける Discriminator からなるが，StarGAN では，Generator に変換元／対象となるスタイルクラスを与え，Discriminator は real/fake に加えスタイルクラスを出力させて敵対的学習を行うことで，データのスタイル変換を行うことができる．本研究では，スタイル変換において変換すべき要素 (style) と保持すべき要素 (context) として以下の 2 パターンを考える．

Intra-user setup: ジェスチャクラスを style，ユーザを context とする（同一ユーザ内でのジェスチャクラス間の変換）．

Inter-user setup: ユーザを style，ジェスチャクラスを context とする（同じジェスチャクラスのユーザ間の変換）．

スタイル変換ネットワークで変換・生成したデータによりデータセットの欠損を補ってジェスチャ識別器を学習することで，生成データを使用しないときよりも精度の高い認識が行えることを目指す．

2. Related Work

2.1 Deep Learning for Gesture Recognition

近年の深層学習の急速な発展に伴い，センサーデータからのジェスチャ認識に深層学習を適用した研究が数多く行われている．

Walse ら [2] は，PCA を特徴量抽出，DNN を分類器としてジェスチャ認識を行った．Hammerla ら [3] は特徴量抽出・分類をまとめて DNN で行った．

CNN を用いる場合，入力データの形として 2 通りの方法が考えられる．一つはデータをセンサーの各次元をチャネルとした 1D 画像として扱い 1D 畳み込みを適用する手法 [4], [5]，もう一つは何らかの手法でセンサーデータを仮想 2D 画像に変形して 2D 畳み込みを適用する手法 [6], [7] である．本研究では後者を採用し，センサーデータをセンサー数×フレーム数という 2D 画像として扱う．

Wang ら [8] は，stacked autoencoder (SAE) を用いた Greedy Layer-wise Training[9] による事前学習と fine-tuning による認識を行った．Inoue ら [10] は，Deep RNN を用いて高スループットでジェスチャ認識を行うことができるアーキテクチャを提案した．また，Ordóñez ら [11] は CNN と LSTM を組み合わせたモデルを提案し，センサーデータの空間的特徴量と時間的な相関の両方を考慮することにより高精度なジェスチャ認識が可能であることを示した．

このようにセンサーデータからのジェスチャ認識モデルは多数提案されているが，その一方でセンサーデータにスタイル変換を適用してデータを生成する研究は未だ無い．

2.2 Generative Adversarial Networks

Generative Adversarial Networks(GAN) [12] は近年画像生成を始めとするコンピュータビジョンの分野で注目を集めている生成モデルである．real データと fake データを認識するように学習する Discriminator と，Discriminator を騙すような real データに近い fake データを生成するように学習する Generator の二つのネットワークを競い合わせて学習させることで，高精度なデータを生成することができる．Conditional GAN (CGAN) [13] では各ネットワークの入力に画像のラベル情報を加えることで条件付けされた画像の生成を可能とした．また，ACGAN [14] では Discriminator に real/fake の識別だけでなくクラスの識別もさせることで高精度な画像生成を実現した．

GAN は画像のスタイル変換でも顕著な結果を残している．pix2pix [15] は CGAN をベースとしたモデルで，対になる画像間の変換を学習する．また，CycleGAN [16] は，変換した画像を元のドメインに再変換した時に元の画像

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

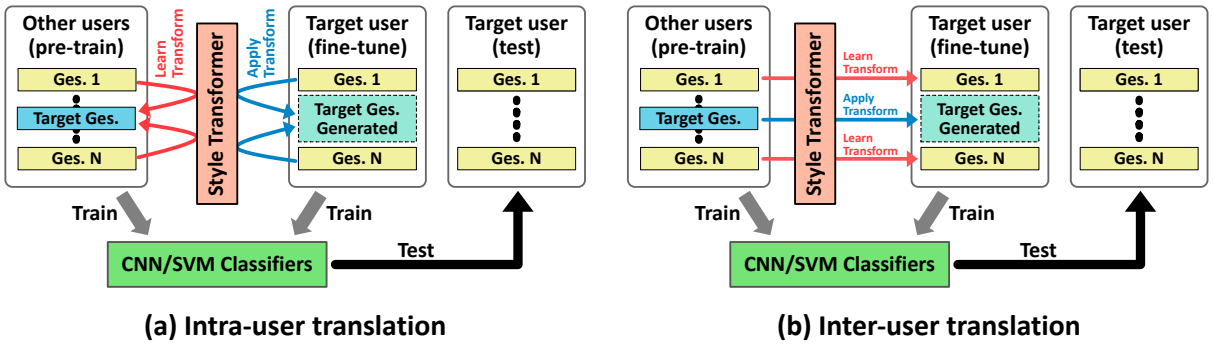

図 3 提案手法の概要．pre-train データは欠損のない複数のユーザのデータ，fine-tune データと test データは target ユーザのデータである．fine-tune データは一部のジェスチャクラスが欠損していると想定する．スタイル変換ネットワークは (a) ジェスチャクラス間の変換 (Intra-user setup)，または (b) ユーザ間の変換 (Inter-user setup) を学習する．学習したスタイル変換ネットワークで生成したデータと pre-tarin データ，fine-tune データを用いてジェスチャ識別器 (CNN/SVM) を学習する．

に戻るようにする cycle consistency loss を損失関数に加えることで，1 対 1 対応していない 2 つの画像群の間のスタイル変換の学習を可能とした．CycleGAN では 3 つ以上のドメイン間の変換を学習するときドメインペアの数だけ Generator を用意する必要があるが，この問題を解決したのが StarGAN [17] である．StarGAN は Generator の入力画像と共に変換先ドメインのラベルを与え，Discriminator に画像のドメイン識別をさせることで 3 つ以上のドメイン間の変換を 1 つの Generator と Discriminator で学習することを可能とした．

3. Method

提案手法はスタイル変換ネットワークとジェスチャ識別器 (CNN/SVM) から成る（図 3）．本研究では，データを以下の三種類に分類する．

pre-train データ　pre-train ユーザ（欠損なし）のデータ
fine-tune データ　target ユーザ（欠損あり）の学習データ
test データ　target ユーザのテストデータ

まず pre-train データと fine-tune データを用いてスタイル変換ネットワークを学習し，データの生成を行う．その後，pre-train データで事前学習した識別器を生成データと fine-tune データを用いてさらに学習する (fine-tuning)．

以下ではまず提案するネットワークの構成について述べ，その後 Loss 関数について述べる．

3.1 Style Transformer Network

提案手法のネットワークは，StarGAN のフレームワークをベースとする Generator および Discriminator の 2 つの CNN から構成される．概要を図 4，各ネットワークの構造を図 5 に示す．

3.1.1 Generator

Generator $G : \mathbf{x}, \mathbf{s}, \mathbf{s}' \rightarrow G(\mathbf{x}, \mathbf{s}, \mathbf{s}')$ は $[0, 1]$ の範囲に正規化された入力データ $\mathbf{x}$ とその style ラベル $\mathbf{s}$，変換先 style ラベル $\mathbf{s}'$ を受け取り，スタイル変換されたデータ $G(\mathbf{x}, \mathbf{s}, \mathbf{s}')$ を出力する．構造としては，encoder と decoder，そしてその間をつなぐ 3 つの Residual block [18] から成る．encoder 部分では，1 層の畳み込み層のあと 2 つの畳み込み block とそれに続く average-pooling 層により downsampling を行う．各 Residual block では，畳み込み block の入力と出力を shortcut connection により足し合わせる．decoder 部分では，2 つの unpooling 層とそれに続く畳み込み block，そして 1 層の 2D 逆畳み込み層により upsampling を行う．これらに含まれる各畳み込み block では、1D 畳み込みと 2D 畳み込みを並列に行う．1D 畳み込みによって各センサーごとの時間方向の相関を，2D 畳み込みによってセンサー間の相関を含めた特徴量を得ることができ，これらを合わせることで各センサーごとの特徴を保持したままセンサー間の相関を考慮することができる．各畳み込み層と逆畳み込み層には Batch Normalization [19] を適用し，畳み込み block の出力層以外の畳み込み層には活性化関数として Leaky ReLU を適用した．

3.1.2 Discriminator

Discriminator $D : \mathbf{x} \rightarrow \{D_{adv}(\mathbf{x}), D_{style}(\mathbf{x}), D_{cont}(\mathbf{x})\}$ は入力データ $\mathbf{x}$ を受け取り，$\mathbf{x}$ の真のサンプル集合に対する尤度 $D_{adv}(\mathbf{x})$，各 style クラスに対する尤度ベクトル $D_{style}(\mathbf{x})$ と，各 context クラスに対する尤度ベクトル $D_{cont}(\mathbf{x})$ を出力する．構造としては 2 層の 2D 畳み込み層と 1 層の全結合層，そのあとに並行する 3 つの全結合層から成る．各畳み込み層では max-pooling 層による downsampling を行う．出力層となる各全結合層はそれぞれ $D_{adv}(\mathbf{x})$，$D_{style}(\mathbf{x})$，$D_{cont}(\mathbf{x})$ を出力する．

3.1.3 Loss Function

損失関数は Adversarial Loss, Style-class Loss, Context-

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

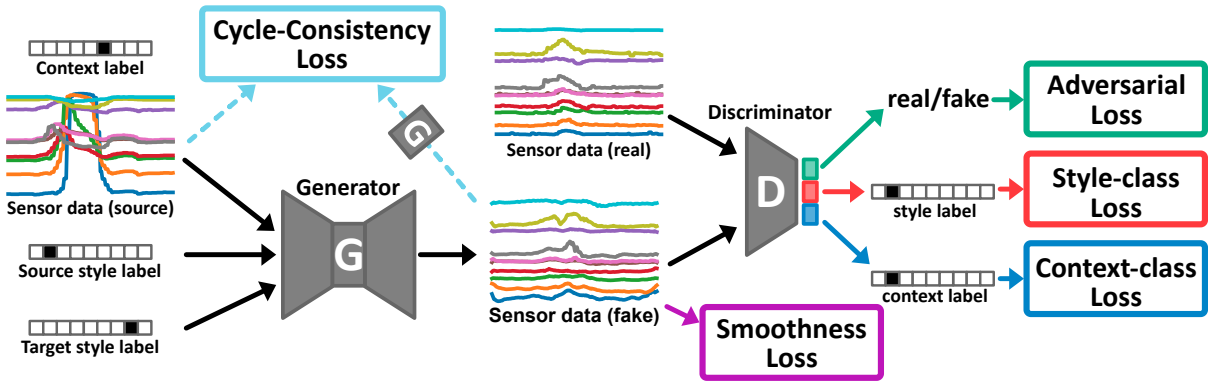

図 4　スタイル変換ネットワークの概要．ネットワークは 2 つの CNN(Generator, Discriminator) から成る．Generator は Input Gesture Data のチャネル方向に変換元 Style Label と変換先 Style Label を結合したものを入力として Fake Gesture Data を生成する．Discriminator にはこの Fake Data あるいは変換先 Style のデータセットからサンプリングした Real Data を入力する．Discriminator は 3 つのヘッドを持ち，それぞれ入力データの Real データセットに対する尤度，各 Style クラスに対する尤度，各 Context クラスに対する尤度を出力する．図中の各矩形は 3.1.3 節に示した Generator の損失関数を構成する各項を表す．

class Loss, Cycle-Consistency Loss, Smoothness Loss の 5 項から成る．

Adversarial Loss. 生成されるデータが real データに近くなるように学習するための項である．本研究では Modified GAN Loss[12] を用いる．

$$\mathcal{L}^G_{adv} = -\mathbb{E}_{\mathbf{x}}[\log D_{adv}(G(\mathbf{x},\mathbf{s},\mathbf{s}'))]$$
$$\mathcal{L}^D_{adv} = \mathbb{E}_{\mathbf{x}}[\log D_{adv}(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}}[\log D_{adv}(G(\mathbf{x},\mathbf{s},\mathbf{s}')))].$$

Style-class Loss. 生成されるデータが変換先として指定したスタイルに変換されるよう学習するための項である．Generator に関しては fake データについて，Discriminator に関しては real データについて Discriminator が出力した style クラス尤度との Binary Cross Entropy をとる．

$$\mathcal{L}^G_{style} = \mathbb{E}_{\mathbf{x},\mathbf{s},\mathbf{s}'}[-\log D_{style}(\mathbf{s}'|G(\mathbf{x},\mathbf{s},\mathbf{s}'))]$$
$$\mathcal{L}^D_{style} = \mathbb{E}_{\mathbf{x},\mathbf{s}}[-\log D_{style}(\mathbf{s}|\mathbf{x})].$$

Context-class Loss. 生成されるデータが元データの context を保持するように学習するための項である．Generator に関しては fake データについて，Discriminator に関しては real データについて Discriminator が出力した context クラス尤度との Binary Cross Entropy をとる．$\mathbf{c}$ は context クラスを表す．

$$\mathcal{L}^G_{cont} = \mathbb{E}_{\mathbf{x},\mathbf{s},\mathbf{s}',\mathbf{c}}[-\log D_{cont}(\mathbf{c}|G(\mathbf{x},\mathbf{s},\mathbf{s}'))]$$
$$\mathcal{L}^D_{cont} = \mathbb{E}_{\mathbf{x},\mathbf{c}}[-\log D_{cont}(\mathbf{c}|\mathbf{x})].$$

Cycle-Consistency Loss. 元の入力データ $\mathbf{x}$ と，$\mathbf{x}$ にスタイル変換を適用したものをさらに元のスタイルへと変換したデータ $G(G(\mathbf{x},\mathbf{s},\mathbf{s}'),\mathbf{s}',\mathbf{s})$ が同じになるように学習するための項であり，両者の二乗誤差を取る．

$$\mathcal{L}_{cycle} = \mathbb{E}_{\mathbf{x},\mathbf{s},\mathbf{s}'}\left[||\mathbf{x} - G(G(\mathbf{x},\mathbf{s},\mathbf{s}'),\mathbf{s}',\mathbf{s})||_2^2\right].$$

Smoothness Loss. ノイズを抑えて滑らかなデータを生成するための項であり，時間方向に隣り合う値の二乗誤差の合計を取る．

$$\left[\mathbf{y}_0^\mathsf{T} \quad \dots \quad \mathbf{y}_T^\mathsf{T}\right] = G(\mathbf{x},\mathbf{s},\mathbf{s}')$$
$$\mathcal{L}_{smooth} = \sum_{t=1}^{T} ||\mathbf{y}_t^\mathsf{T} - \mathbf{y}_{t-1}^\mathsf{T}||_2^2.$$

ただし T は，ジェスチャのフレーム数を表す．

Generator，Discriminator の損失関数はこれらの loss とハイパーパラメタを用いてそれぞれ以下のように表される．

$$\mathcal{L}_G = \lambda_{adv} * \mathcal{L}^G_{adv} + \lambda_{style} * \mathcal{L}^G_{style} + \lambda_{cont} * \mathcal{L}^G_{cont} + \lambda_{cycle} * \mathcal{L}_{cycle} + \lambda_{smooth} * \mathcal{L}_{smooth}$$
$$\mathcal{L}_D = \lambda_{adv} * \mathcal{L}^D_{adv} + \lambda_{style} * \mathcal{L}^D_{style} + \lambda_{cont} * \mathcal{L}^D_{cont}$$

実験では，$\lambda_{adv} = 1.0$, $\lambda_{style}= 10.0$, $\lambda_{cont}= 10.0$, $\lambda_{cycle}= 100.0$, $\lambda_{smooth} = 1.0$ とした．

4. Experiments

4.1 Dataset

提案手法の性能を評価するため，二種類の実データを用いて実験を行った．

4.1.1 CheekInput

CheekInput [20] は Yamashita らの提案した，頬をタッチサーフェースとして情報入力を行う Head Mounted Display である．フレームの下部及び側部に複数個配置された反射型の光センサーで頬からフレームまでの距離を複数点で計測することで皮膚形状の変化を取得する．本研究では，

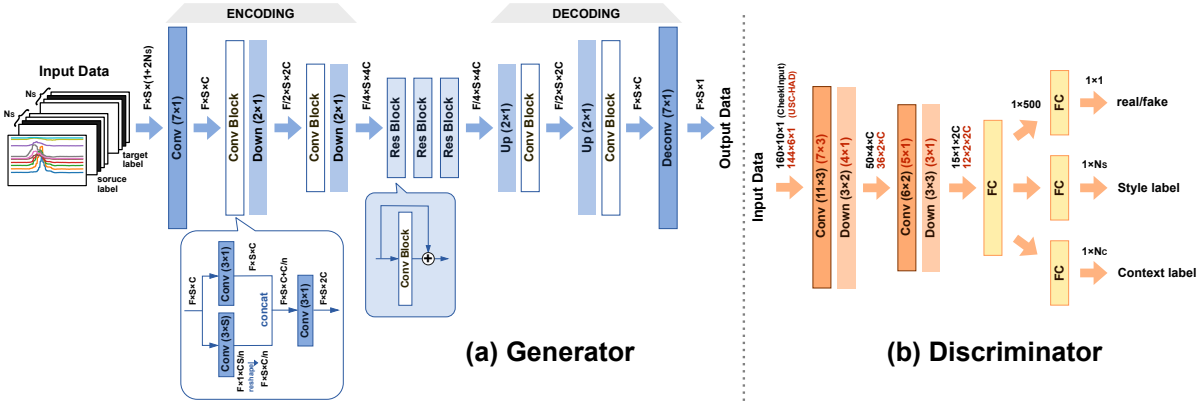

図 5　Generator および Discriminator のネットワーク構造.

CheekInput の左頬側の 10 個のセンサーで計測した 4 種類のジェスチャデータを利用する．具体的には，各ユーザは二回の trial(A, B) を行い，各 trial でランダムに提示される順に各ジェスチャをそれぞれ 30 回ずつ計測する（各ジェスチャの計測時間は 30Hz で 160 フレーム）．したがって，ユーザ一人あたりのデータ数は 2 trials × 4 gestures × 30 instances = 240 である．10 人のユーザのデータを収集し，6 人を pre-train ユーザ（trial A を pre-train データ，trial B を validation データ）とし，4 人を target ユーザ（trial A を fine-tune データ，trial B を test データ）とした．

4.1.2　USC-HAD

USC-HAD [21] はユーザの腰に取り付けた MotionNode sensing platform（二種の三軸センサーによる計測ユニット）で計測された walking，running などの 12 クラスのアクティビティデータである．そのうち 5 クラス（sitting, standing など）には大きな値の変化が見られないため，本研究ではそれ以外の 7 クラスを使用した．100Hz で計測されている元データを 50Hz にダウンサンプリング後，144 フレームごとに分割した．14 人のユーザのうち 9 人を pre-train ユーザ（データの 70%を pre-train データ，30%を validation データ）とし，5 人を target ユーザ（70%を fine-tune データ，30%を test データ）とした．

4.2　Training and Data Generation

スタイル変換ネットワークの学習には Optimizer として Adam[22] を使用，β_1=0.5，β_2=0.999 とした．学習率は G が 0.0002，D が 0.0001 とした．CheekInput はバッチサイズ 24 で 80000iteration，USC-HAD はバッチサイズ 50 で 70000iteration の学習を行った．

生成されたデータの一部を図 6 に示す．変換によって Ground-truth に類似した特徴を持つデータが生成されていることがわかる．このことから，スタイル変換はジェスチャデータに対しても有効であるといえる．

4.3　Recognition

ジェスチャ識別器として CNN と SVM を使用し，1 クラス欠損における以下の手法の認識精度を比較した．

CNN_PT　pre-train データのみで学習

CNN_FT1　CNN_PT を fine-tune データ（欠損補完なし）で fine-tuning

CNN_FT2　CNN_PT を fine-tune データ（欠損補完あり）で fine-tuning（提案手法）

SVM_PT　pre-train データのみで学習

SVM_FT1　pre-train データと fine-tune データ（欠損補完なし）で学習

SVM_FT2　fine-tune データ（欠損補完あり）で学習（提案手法）

また，提案手法の欠損クラス数の増加に対する耐久性を調べるため，Intra-user setup で CNN を用いたストレステストを行った．

4.3.1　Missing One Gesture Class

1 クラス欠損における各手法の比較を図 7 に示す．CNN を用いた提案手法 (CNN_FT2) は，どちらのデータセットにおいても生成データを使用しない手法 (CNN_PT, CNN_FT1, SVM_PT, SVM_FT1) に比べ高い精度を出している．一方 SVM においては，USC-HAD では一部のユーザで提案手法 (SVM_FT2) が SVM_PT や SVM_FT1 に勝っているが，CheekInput では精度が低くなった．ただし，SVM_FT1 は SVM_FT2 に比べ学習データ量が多いため，学習に約 500 倍の時間を要した（FT1 が約 1.5 秒，FT2 が約 0.03 秒）．これらの結果から，提案手法によるスタイル変換は特に CNN による認識においてデータに欠損があるときの認識精度向上に有効であることがわかる．SVM における提案手法の結果が優れなかった原因として，スタイル変換ネットワークの Discriminator が CNN ベースであるため，生成データが CNN にとって認識しやすい特徴を持つデータである可能性があることが考えられる．

4.3.2　Stress Test

Intra-user setup で欠損クラス数を増加させたときの認識

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

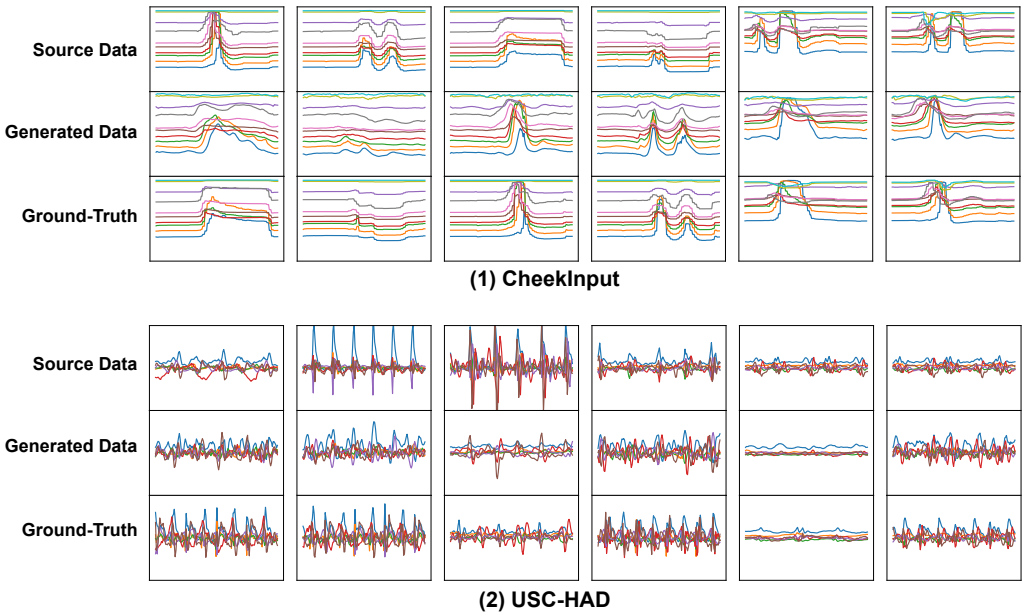

図 6 スタイル変換ネットワークによる生成結果．上段が生成元データ，中段が生成データ，下段が Ground-truth.

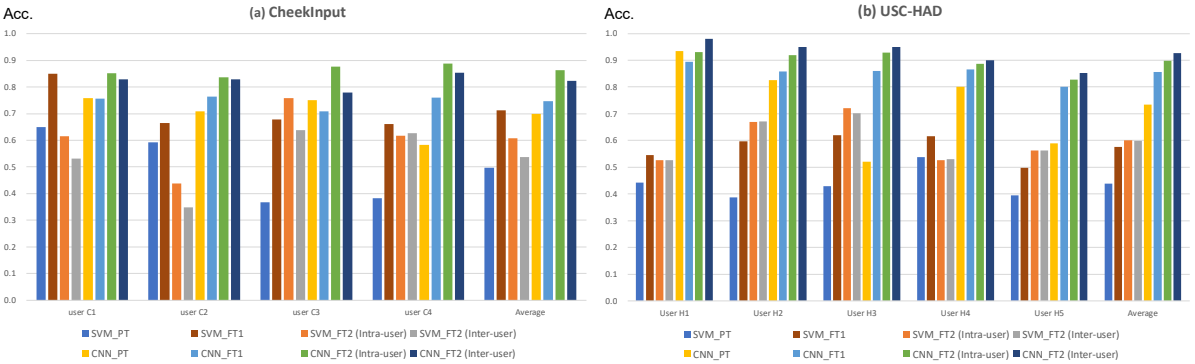

図 7 1 クラス欠損における実験結果．CNN_FT2 と SVM_FT2 が提案手法.

精度の変化を図 8 (CheekInput) と図 9 (USC-HAD) に示す．いずれのパターンにおいても欠損クラス数が増えるに従って認識精度が低下する傾向がある．それは特に CNN_FT1 において顕著であるが，一方で提案手法 (CNN_FT2) では精度の低下幅が小さく抑えられており，ほとんどのケースで CNN_FT1 の精度を上回っている．また，提案手法が CNN_PT の精度を上回るのに必要なデータ量（クラス数）は 25% (=1/4，CheekInput) から 43% (=3/7，USC-HAD) となっており，これは提案手法がユーザのデータ収集コストの削減に非常に有効であることを示している．

5. Conclusion

本研究では，ユーザの未知のジェスチャに対して認識精度を向上させるための手法として，ジェスチャクラス間あるいはユーザ間のデータのスタイル変換を用いる手法を提案した．また，この変換を行うためのスタイル変換ネットワークおよび学習のフレームワークを提案した．実験では二つの実データセットに提案手法によるスタイル変換を適用し，生成したデータを用いて識別器の学習を行うことで生成データを使用しない手法に比べて認識精度が向上することを確認した．また，提案手法が欠損クラス数の増加による認識精度の低下を緩和することを確認し，ユーザのデータ収集コストの削減に対する提案手法の有用性を示した．

本研究の実験では，スタイル変換における変換元のデータとしてあらゆるジェスチャクラスあるいはユーザのパターンを試し，その選び方によって認識精度が変化することがわかった．したがって，今後の展望として認識精度の向上により有効となるデータを生成するための変換元データの選び方を調べるため，データの解析を進めていく予定である．

謝辞 ご討論頂いた山田誠氏，実験に協力頂いた伊藤

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

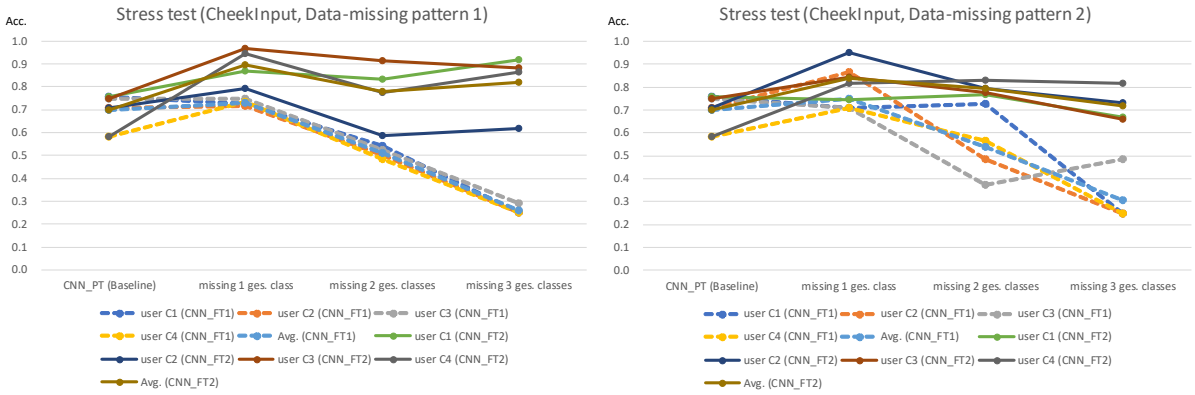

図 8 ストレステスト (CheekInput) の結果．実線と破線がそれぞれ CNN_FT2（提案手法）と CNN_FT1 を表す．欠損パターン 1（左）はジェスチャクラス 1 から 3 の順に欠損を増やすパターン，パターン 2（右）はクラス 4 から 2 の順に欠損を増やすパターンである．

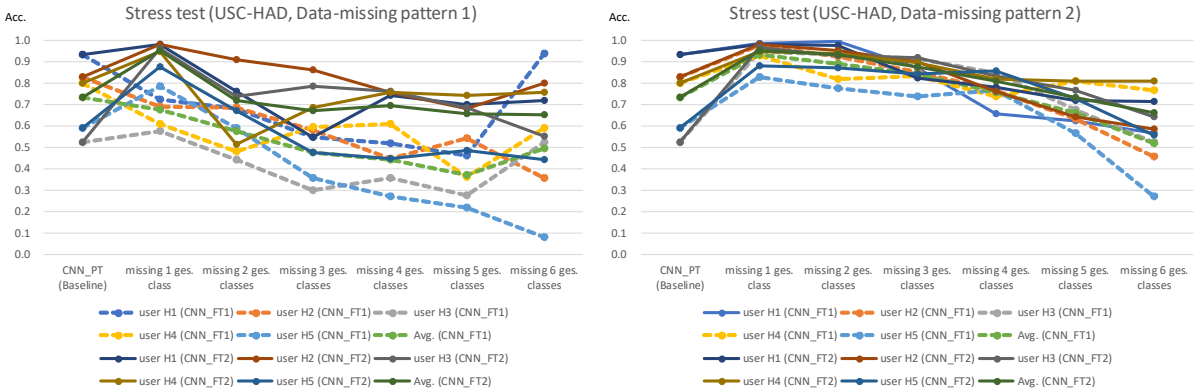

図 9 ストレステスト (USC-HAD) の結果．実線と破線がそれぞれ CNN_FT2（提案手法）と CNN_FT1 を表す．欠損パターン 1（左）はジェスチャクラス 1 から 6 の順に欠損を増やすパターン，パターン 2（右）はクラス 7 から 2 の順に欠損を増やすパターンである．

勇太氏と菊井浩祐氏に感謝する．本研究は JST CREST JPMJCR17A5 の支援を受けたものである．

参考文献

[1] Weiss, G. M. and Lockhart, J.: The impact of personalization on smartphone-based activity recognition, *Workshops at AAAI* (2012).
[2] Walse, K. H., Dharaskar, R. V. and Thakare, V. M.: PCA Based Optimal ANN Classifiers for Human Activity Recognition Using Mobile Sensors Data, *ICTIS'16*, pp. 429–436 (2016).
[3] Hammerla, N. Y., Halloran, S. and Ploetz, T.: Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables, *CoRR*, Vol. abs/1604.08880 (2016).
[4] Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P. and Zhang, J.: Convolutional Neural Networks for human activity recognition using mobile sensors, *ICMAS*, pp. 197–205 (2014).
[5] Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L. and Krishnaswamy, S.: Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition, *IJCAI*, pp. 3995–4001 (2015).
[6] Ha, S., Yun, J. and Choi, S.: Multi-modal Convolutional Neural Networks for Activity Recognition, *IEEE SMC*, pp. 3017–3022 (2015).
[7] Jiang, W. and Yin, Z.: Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks, *ACM Multimedia*, pp. 1307–1310 (2015).
[8] Wang, A., Chen, G., Shang, C., Zhang, M. and Liu, L.: Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders, *Web-Age Information Management* (2016).
[9] Hinton, G. E., Osindero, S. and Teh, Y.-W.: A Fast Learning Algorithm for Deep Belief Nets, *Neural Computation*, Vol. 18, No. 7, pp. 1527–1554 (2006).
[10] Inoue, M., Inoue, S. and Nishida, T.: Deep Recurrent Neural Network for Mobile Human Activity Recognition with High Throughput, *CoRR*, Vol. abs/1611.03607 (2016).
[11] Ordóñez, F. J. and Roggen, D.: Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition, *Sensors*, Vol. 16, No. 1 (2016).
[12] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative adversarial nets, *NeurIPS*, pp. 2672–2680 (2014).
[13] Mirza, M. and Osindero, S.: Conditional Generative Adversarial Nets, *CoRR*, Vol. abs/1411.1784 (2014).
[14] Odena, A., Olah, C. and Shlens, J.: Conditional Image Synthesis with Auxiliary Classifier GANs, *ICML*,

Vol.2020-MBL-96 No.12
Vol.2020-UBI-67 No.12
Vol.2020-CDS-28 No.12
Vol.2020-ASD-19 No.12
2020/9/29

pp. 2642–2651 (2017).
[15] Isola, P., Zhu, J., Zhou, T. and Efros, A. A.: Image-to-Image Translation with Conditional Adversarial Networks, *CoRR*, Vol. abs/1611.07004 (2016).
[16] Zhu, J., Park, T., Isola, P. and Efros, A. A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *CoRR*, Vol. abs/1703.10593 (2017).
[17] Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S. and Choo, J.: StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, *CoRR*, Vol. abs/1711.09020 (2017).
[18] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *CoRR*, Vol. abs/1512.03385 (2015).
[19] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *CoRR*, Vol. abs/1502.03167 (2015).
[20] Yamashita, K., Kikuchi, T., Masai, K., Sugimoto, M., Thomas, B. H. and Sugiura, Y.: CheekInput: Turning Your Cheek into an Input Surface by Embedded Optical Sensors on a Head-mounted Display, *VRST*, pp. 19:1–19:8 (2017).
[21] Zhang, M. and Sawchuk, A. A.: USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors, *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, Pittsburgh, Pennsylvania, USA (2012).
[22] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).