

Style Transfer Networkを用いた動作データの個性化による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ー依存性が高いため、高い認識性能を発揮するためには、ユーザー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、ユーザー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 Transfer Network) を用いてあるユーザー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。CNNによる識別器を用いて精度を評価したところ、提案手法は91.2%、既存手法は78.2%となり、提案手法の有用性を確認した。

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、異なるユーザー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、ユーザー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、現在の全ユーザー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

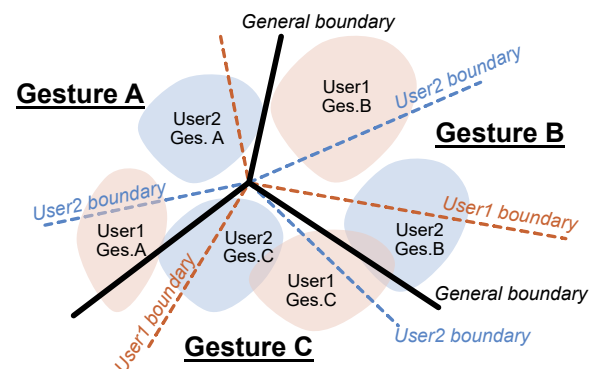


図1 ジェスチャ認識でのデータの個人差とジェスチャの識別境界。データには個人差があるので、全ユーザーに対応した境界 (General boundary: 黒線) では区別できないジェスチャ/ユーザーが生じる。そのため、ユーザーごとのデータを学習することで、より精度の良い識別が実現できる (User 1 boundary, User 2 boundary)。一方で、あるユーザーに対応した識別境界を他のユーザーに使用すると、一般的に性能は低下する。また同様に、ジェスチャの種類を多くすると、その変動が個人差と識別不可能になり認識精度は一般的に低下する。

方でこのような境界を得るためには、ユーザーの全ジェスチャデータを得、再学習を行う必要 (ユーザーキャリブレーション) があるが、ユーザー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、ユーザー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ユーザー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、ユーザーの全ジェスチャデータを取得する必要がない

¹ 京都大学

Kyoto University

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

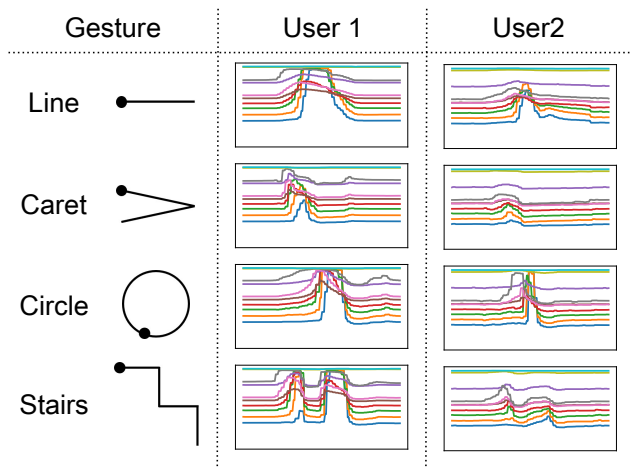


図 2 ジェスチャデータの一部. 4 クラスのジェスチャに対する User1 と User2 の動作データ. グラフの異なる色が異なるセンサを表す. ここから同じジェスチャでもユーザーごとの個性 (style) の差が出ていることがわかる.

ため, 可用性が大幅に向上する.

動作データの変換には, 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, 異なる動作クラスのジェスチャデータでも, 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, 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, このようなユーザー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, スタイルクラスとしてユーザーの ID を想定する (図 3). すなわち, 学習時にはユーザー間 (User 1, 2) で対応のある動作データ (Gesture A, B, C) によりスタイル変換を学習した後, User 1 の欠損データ (Gesture D) を User 2 のデータを変換して生成する. 変換されたデータにより User 1 のジェスチャ識別機を学習することで, 他者のデータを用いて学習したときよりも, 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.

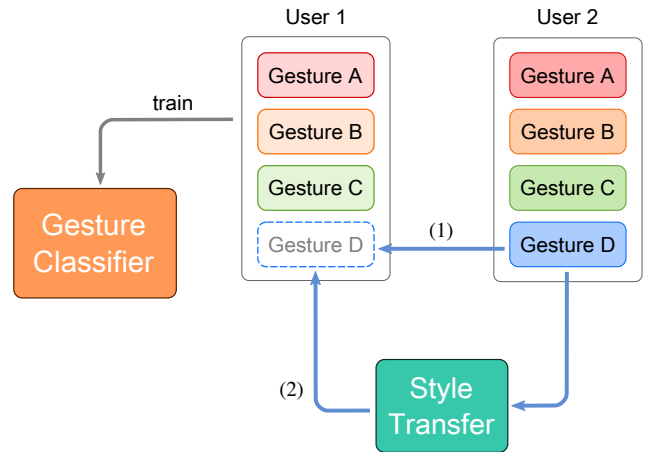


図 3 User 1 の学習データに欠損があった場合, (1) User 2 の対応するデータを直接補完するよりも (2) User 2 のデータを Style transfer によって User 1 の style に変換した上で補完する方が User 1 のデータに対する識別精度が向上すると考えられる.

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ンネルとした 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

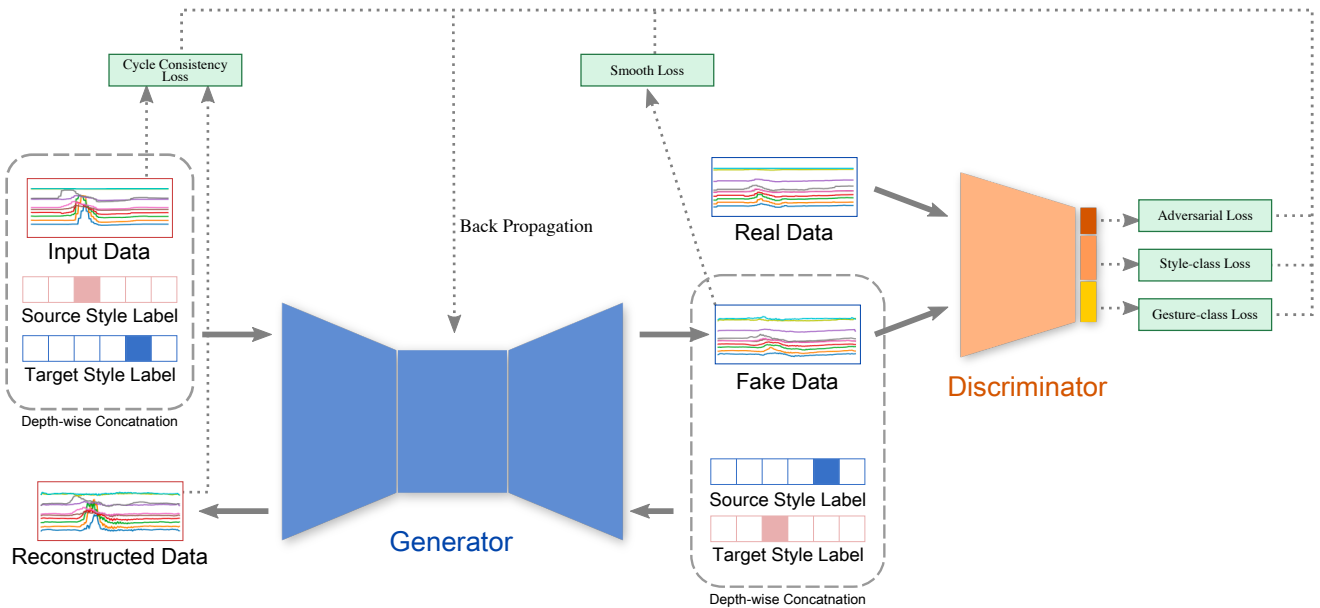


図 4 提案手法の概要図. ネットワークは2つの CNN(Generator, Discriminator) から成る. Generator は Input Gesture Data のチャンネル方向に変換元 Style Label と変換先 Style Label を結合したものを入力として Fake Gesture Data を生成する. Discriminator にはこの Fake Data あるいは変換先 Style のデータセットからサンプリングした Real Data を入力する. Discriminator は3つのヘッドを持ち, それぞれ入力データの Real データセットに対する尤度, 各 Style クラスに対する尤度, 各ジェスチャクラスに対する尤度を出力する. 図中の緑の各矩形は 3.2 節に示した Generator の損失関数を構成する各項を表す.

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

提案手法の概要を図 4 に示す. 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 関数および学習手法について述べる.

3.1 Networks

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. 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 $\mathbf{x}$ とそのスタイルラベル $\mathbf{s}$, 変換先スタイルラベル $\mathbf{s}'$ 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, 入力に対し時間方向のみの畳み込み (local な畳み込み) と時間方向とセンサー種方向の両方の畳み込み (global な畳み込み) を並列に行った後, これらの出力をチャンネル方向に結合させたものに対してさらに畳み込みを行う. local な畳み込みによって各センサーごとの時間方向の相関を, global な畳み込みによってセンサー間の相関を含めた

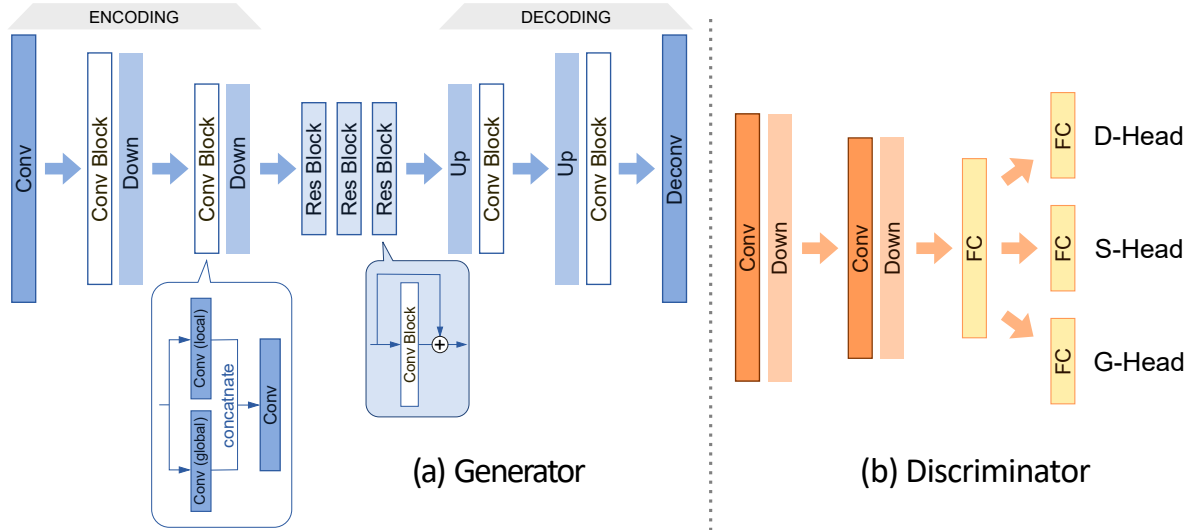


図 5 Generator および Discriminator のネットワーク構造.

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 $D: \mathbf{x} \rightarrow \{D_{adv}(\mathbf{x}), D_{style}(\mathbf{x}), D_{ges}(\mathbf{x})\}$ 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 $D_{adv}(\mathbf{x})$ 、各スタイルクラスに対する尤度ベクトル $D_{style}(\mathbf{x})$ と、各ジェスチャクラスに対する尤度ベクトル $D_{ges}(\mathbf{x})$ 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 $D_{adv}(\mathbf{x})$ 、 $D_{style}(\mathbf{x})$ 、 $D_{ges}(\mathbf{x})$ を出力する。

3.2 Loss Function

損失関数は Adversarial Loss, Style-class Loss, Gesture-class Loss, Cycle Consistency Loss, Smooth 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。

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

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

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

スタイルクラス尤度との Binary Cross Entropy をとる。

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

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

Gesture-class Loss. 生成されるデータが元データと同じジェスチャラベルにな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 が出力したジェスチャクラス尤度との Binary Cross Entropy をとる。 $\mathbf{g}$ はジェスチャクラスを表す。

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

$$\mathcal{L}_{Ges}^D = \mathbb{E}_{\mathbf{x}, \mathbf{g}}[-\log D_{ges}(\mathbf{g}|\mathbf{x})].$$

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。

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

Smooth 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。

$$\begin{bmatrix} \mathbf{y}_0^T & \dots & \mathbf{y}_T^T \end{bmatrix} = G(\mathbf{x}, \mathbf{s}, \mathbf{s}')$$

$$\mathcal{L}_{Sm} = \sum_{t=1}^T \|\mathbf{y}_t^T - \mathbf{y}_{t-1}^T\|_2^2.$$

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。

$$\begin{aligned}\mathcal{L}_G &= \lambda_{Adv} * \mathcal{L}_{Adv}^G + \lambda_{Style} * \mathcal{L}_{Style}^G + \lambda_{Ges} * \mathcal{L}_{Ges}^G \\ &\quad + \lambda_{Cycle} * \mathcal{L}_{Cycle} + \lambda_{Sm} * \mathcal{L}_{Sm} \\ \mathcal{L}_D &= \lambda_{Adv} * \mathcal{L}_{Adv}^D + \lambda_{Style} * \mathcal{L}_{Style}^D + \lambda_{Ges} * \mathcal{L}_{Ges}^D\end{aligned}$$

実験では, $\lambda_{Adv} = 1.0$, $\lambda_{Style} = 10.0$, $\lambda_{Ges} = 10.0$, $\lambda_{Cycle} = 100.0$, $\lambda_{Sm} = 1.0$ とした.

3.3 Optimization

Optimizer には Adam[20] を使用, $\beta_1=0.5$, $\beta_2=0.999$ とした. 学習率は G が 0.0001, D が 0.00005 とした. 各 iteration につき G・D ともに 1 度ずつ update し, 27,000 iteration で学習を打ち切った.

4. Experiments

提案手法の性能を評価するため, ウェアラブルデバイスで計測したジェスチャの実データを用いてモデルの学習・スタイル変換を行った. また, CNN によるジェスチャ認識を行い, あるユーザーの欠損した学習データを他のユーザーの実データで補った場合と, 他のユーザーの実データからスタイル変換によって生成したデータで補った場合とでテストデータに対する識別精度の比較を行った.

4.1 Dataset

実験では, ウェアラブルデバイスとして Yamashita らの提案した CheekInput [21] を利用した. このデバイスは頬をタッチサーフェスとして情報入力を行う Head Mounted Display で, 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, 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, 各ユーザーはランダムに提示される順に各ジェスチャをそれぞれ 30 回ずつ行い (trial A), 3 分のインターバルを挟んだ後デバイスを再装着して同じように計測 (trial B) を行う. したがって, ユーザー一人あたりのデータ数は 2 trials $\times$ 4 gestures $\times$ 30 instances = 240 で, 本研究では 2 人のユーザーのデータを使用した. 各ジェスチャの計測時間は 30Hz で 160 フレームである.

4.2 Training and Data Generation

Style Transfer Network の学習・生成には, 2 人のユーザーそれぞれの trial A のデータを使用した. 片方のユーザーの 4 種類のジェスチャのうち 1 種類を欠損データとして, 残りの 3 種類のデータでモデルを学習することで 2 人のユーザー間のスタイル変換を獲得, 学習済モデルを用いて欠損データに対応する他方のユーザーのデータからスタイル変換を行って補完データを生成した. なお, Style Transfer Network は一度の学習で双方向の変換を学習する

	欠損なし	スタイル変換なし	スタイル変換あり
Line	-	0.767	1.000
Caret	-	0.850	0.933
Circle	-	0.725	0.917
Stairs	-	0.725	0.867
average	0.950	0.767	0.929

表 1 テストデータに対する Accuracy (User 1)

	欠損なし	スタイル変換なし	スタイル変換あり
Line	-	0.758	0.725
Caret	-	0.733	0.950
Circle	-	0.908	0.925
Stairs	-	0.783	0.925
average	0.992	0.796	0.894

表 2 テストデータに対する Accuracy (User 2)

ので, 例えば User 1 のジェスチャ 1 が欠損している場合と User 2 のジェスチャ 1 が欠損している場合のモデルは共通である. 生成されたデータの一部を図 6 に示す. 実データと比べて局所的なノイズは多いものの, 波形は変換先ユーザーのデータに近い形に変換できている.

4.3 Recognition

提案手法による生成データを定量的に評価するため, 各ユーザーについて CNN によるジェスチャ認識を行い以下の 3 パターンでテストデータに対する識別精度を比較した.

- (1) 学習データに欠損がない場合.
- (2) 学習データの欠損を他のユーザーの対応するデータで補完する場合.
- (3) 学習データの欠損を他のユーザーの対応するデータに提案手法のスタイル変換を適用したデータで補完する場合.

識別に用いた CNN の構造は Style Transfer Network の Discriminator の 3 つの出力層をジェスチャクラスに対する尤度を出力する層のみとしたネットワークで, 学習時の最適化手法には Adam を用いた. 学習データには trial A のデータ及び trial A のデータからスタイル変換によって生成したデータ, テストデータには trial B のデータを用いた. 各ユーザーについて, 3 パターンの学習データによる学習済みモデルのテストデータに対する識別精度 (Accuracy) を表 1, 表 2 に示す. User 1, User 2 共に, 欠損データを他方のユーザーのデータで補完したときに平均識別精度が 80%を下回ってしまうのに対して, スタイル変換を適用したデータで補完することで 90%前後まで向上させることができている. このことから, 他のユーザーのデータに提案手法のスタイル変換を適用することで欠損データがあっても高い識別精度を達成することが可能であることが示された.

5. Conclusion

本研究では, ユーザーの未知のジェスチャに対して識別

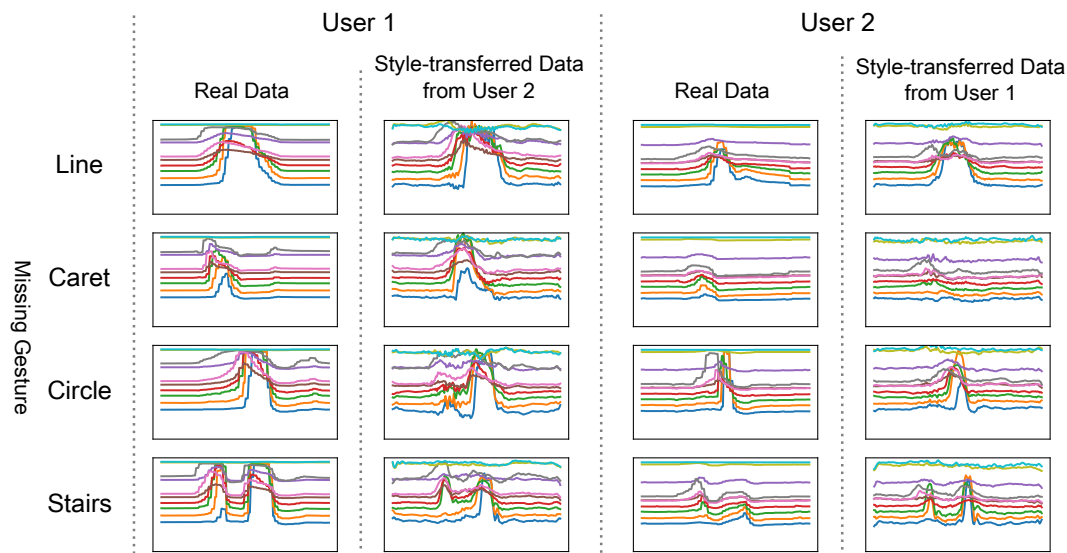


図 6 Style Transfer Network による生成結果。各行は、一番左に示したジェスチャを欠損データとして残りの3種類のジェスチャで User 1, 2 間のスタイル変換の学習を行い、各ユーザーについて欠損データに対応する他方のユーザーのデータからスタイル変換によって生成したデータと Ground truth となる real データを並べたものである。生成されたデータは局所的なノイズはあるものの、概ねの波形は real データに近い形になっていることがわかる。

率を向上させるための手法として、ユーザー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ニューラル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、ウェアラブルデバイスで計測した実データ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ユーザーごとのジェスチャ認識を行うことで認識精度が向上することを確認した。

本研究の実験で行ったのは2人のユーザー間のスタイル変換であるが、提案手法のベースである StarGAN は1つの Generator で複数ドメイン間の変換が可能なモデルである。したがって、今後ユーザーの数を増やしてスタイル変換を学習して提案手法の性能を確認する必要がある。

謝辞 ご討論頂いた山田誠氏、実験に協力頂いた伊藤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] Gary Mitchell Weiss and Jeffrey Lockhart. The impact of personalization on smartphone-based activity recognition. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [2] Kishor H. Walse, Rajiv V. Dharaskar, and Vilas M. Thakare. Pca based optimal ann classifiers for human activity recognition using mobile sensors data. In Suresh Chandra Satapathy and Swagatam Das, editors, *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*, pp. 429–436, Cham, 2016. Springer International Publishing.
- [3] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz.

- Deep, convolutional, and recurrent models for human activity recognition using wearables. *CoRR*, Vol. abs/1604.08880, , 2016.
- [4] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205, Nov 2014.
- [5] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, pp. 3995–4001. AAAI Press, 2015.
- [6] S. Ha, J. Yun, and S. Choi. Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3017–3022, Oct 2015.
- [7] Wenchao Jiang and Zhaozheng Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM International Conference on Multimedia, MM '15*, pp. 1307–1310, New York, NY, USA, 2015. ACM.
- [8] Aiguo Wang, Guilin Chen, Cuijuan Shang, Miaofei Zhang, and Li Liu. Human activity recognition in a smart home environment with stacked denoising autoencoders. In Shaoxu Song and Yongxin Tong, editors, *Web-Age Information Management*, pp. 29–40, Cham, 2016. Springer International Publishing.
- [9] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, Vol. 18, No. 7, pp. 1527–1554, 2006. PMID: 16764513.
- [10] Masaya Inoue, Sozo Inoue, and Takeshi Nishida. Deep recurrent neural network for mobile human activity recognition with high throughput. *CoRR*, Vol. abs/1611.03607, , 2016.

- [11] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors (Basel, Switzerland)*, Vol. 16, No. 1, 2016.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, Vol. abs/1411.1784, , 2014.
- [14] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pp. 2642–2651. JMLR.org, 2017.
- [15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, Vol. abs/1611.07004, , 2016.
- [16] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, Vol. abs/1703.10593, , 2017.
- [17] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *CoRR*, Vol. abs/1711.09020, , 2017.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, Vol. abs/1512.03385, , 2015.
- [19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, Vol. abs/1502.03167, , 2015.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Koki Yamashita, Takashi Kikuchi, Katsutoshi Masai, Maki Sugimoto, Bruce H. Thomas, and Yuta Sugiura. Cheekinput: Turning your cheek into an input surface by embedded optical sensors on a head-mounted display. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST ’17, pp. 19:1–19:8. ACM, 2017.