Vol.2019-HCI-182 No.29
2019/3/19

3次元 CNN を利用した Wi-Fi CSI によるジェスチャ認識

宮代 理弘[1]　宮下 芳明[1]

概要：家庭や公共施設などで Wi-Fi 電波が普及してきたなか，Wi-Fi 電波から取得できる CSI (Channel State Information; チャネル状態情報) を利用したセンシング技術の研究が進んでいる．近年では， CSI が多次元のデータであることに着眼し，画像識別で使われている CNN (Convolutional Neural Network; 畳込みニューラルネットワーク) を応用した技術が提案されている．一方で，CNN の技術を使った手法では，ジェスチャ認識のような時間経過によって状態が変化するものには対応できていない．本研究では，動画の内容推定に利用されている 3 次元 CNN を応用することによって，時系列情報を含んだ状態で学習させるジェスチャ認識手法を提案する．実験の結果から，正解率 0.932 以上を得ることができ，従来手法と比較しても高い正解率を実現した．また，提案した手法を実際に導入する場合に懸念される点について考察した．

1. はじめに

Wi-Fi 技術の普及により，一般家庭や公共施設など，恒常的に Wi-Fi 電波を取得できる場所が増えてきた．この Wi-Fi 電波から電波の状態を表す CSI[*1] の変位を取得し，ひとの行動をセンシングする技術が盛んに研究されている．この技術によって，デバイスフリーでひとの位置を取得したり [1–3]，行動を認識したり [4,5]，個人を識別したり [6] することが可能になっている．

既存のコンピュータビジョン技術によるジェスチャ認識では，「身体の一部や障害物によるオクルージョン」や「背景の動きや煙・暗闇などによる認識率の低下」，「浴室などに設置するときの不快感」などの問題点が指摘されている [7]．これらの問題を解決できるとして，Wi-Fi 電波を利用したセンシング技術は，注目を集めている．また，すでに普及している Wi-Fi システムに則ることで，新たなデバイスを導入する必要がなく，身体にデバイスを取り付ける必要もないといった利点も挙げられている [8]．

近年では， CSI が多次元のデータであることに着眼し，同様に多次元のデータである画像を識別する CNN[*2] の技術を応用した手法が提案されている．CNN を利用すると他の機械学習と比較して，データへ複雑な前処理をする必要が省け，認識用途が違う場合でも学習プロセスを共通化することができる．CNN を利用した手法では，個人を識別することや，手のサインを識別することには成功している [6] が，ジェスチャ認識などの時間経過によって状態が変わるものへの応用がなされていない．一方で，CNN は 3 次元に拡張して時系列情報を含んだ学習を行う手法 [9]（以下，3 次元 CNN）が提案されており，動画の内容推定などに利用されている．そこで本研究では， 3 次元 CNN の技術を応用して，時系列情報を含んだ学習を行うことで，ジェスチャを認識する手法を提案し，評価を行った．

2. Wi-Fi 電波から取得できる情報

Wi-Fi 電波は伝搬中に減衰していくため，送受信の距離によって状態が変位している．また，室内においては反射や回折などによって，減衰や位相シフトが発生する．この Wi-Fi 電波の変位を分析することで，ひとの行動を推測することが可能になる．Wi-Fi 電波の変位を表す情報には，RSSI[*3] と CSI がある．

2.1 RSSI

RSSI とは，IEEE 802.11 のデータリンク層のメディアアクセス制御副層にて提供される，電波強度を示す値である．室内において， Wi-Fi 電波は反射や回折などによって，複数の経路（以下，マルチパス）から伝搬する．経路ごとに位相シフト，減衰，時間遅延が異なるため，受信し

[1] 明治大学
Meiji University

[*1] Channel State Information; チャネル状態情報

[*2] Convolutional Neural Network; 畳込みニューラルネットワーク

[*3] Received Signal Strength Indicator; 受信信号強度

た電波は経路ごとに変化した波形を組み合わせたものになる．それを踏まえると，信号電圧は式 1 で表せる [10]．ここで，a_i と θ_i は，それぞれ i 番目の経路における振幅と位相であり，N はマルチパスの総数である．

$$V = \sum_{i=1}^{N} \|a_i\| \mathrm{e}^{-j\theta_i} \tag{1}$$

RSSI は，この信号電圧をデシベル（dB）表記したものであり，式 2 で表せる．

$$\mathrm{RSSI} = 10 \log_2(\|V\|^2) \tag{2}$$

2.2 CSI

Wi-Fi 電波は，マルチパスの影響によって，振幅と位相に変位がおきる．CSI は，IEEE 802.11 の物理層で得られる，振幅と位相の変位を複素数の絶対値と偏角で表した値の多次元行列である [11]．近年では，CSI を取得するソフトウェアツールも公開されており，Intel 5300 [12] や Atheros 9390 [13] のような一般的な商用の Wi-Fi NIC から取得できる．

マルチパスによる変位は，CIR[*4] として時間線形フィルタにモデル化される．CIR $h(\tau)$ は，各マルチパスにおけるインパルス応答の総和であり，時不変系を仮定した場合，式 3 で表せる [14]．ここで，a_i，θ_i，τ_i は，それぞれ i 番目の経路における振幅，位相，時間遅延である．また，N はマルチパスの総数，$\delta(\tau)$ はディラックのデルタ関数である．

$$h(\tau) = \sum_{i=1}^{N} a_i \mathrm{e}^{-j\theta_i} \delta(\tau - \tau_i) \tag{3}$$

マルチパスによる周波数変動は，周波数選択性フェージングによる位相の重畳として考えられる．よって，マルチパスによる変化は，CIR をフーリエ変換した CFR[*5] によって，特徴づけることが可能になる．CFR $H(f)$ は，式 4 として表せる．ここで，$\|H(f)\|$ は CFR の振幅であり，$\angle H(f)$ は CFR の位相である．

$$H(f) = \|H(f)\| \mathrm{e}^{j\angle H(f)} \tag{4}$$

IEEE 802.11 で採用されている OFDM[*6] では，複数の周波数帯域（以下，サブキャリア）でデータを伝送する．また， IEEE 802.11 n/ac などの Wi-Fi 規格においては，MIMO[*7] が採用されており，送信側・受信側ともに複数のアンテナを保持している．このことより，ひとつのパケット通信から複数の CFR を算出できる．

送信側のアンテナ数を N_{Tx}，受信側のアンテナ数を N_{Rx} とする．$\mathbf{X_k}$ と $\mathbf{Y_k}$ をそれぞれ k 番目のサブキャリアで得られる N_{Tx} 次元の送信ベクトル，N_{Rx} 次元の受信ベクトルとしたとき，式 5 が成り立つ．ここで $\mathbf{N_k}$ は N_{Rx} 次元のノイズベクトルであり，$\mathbf{H_k}$ は， k 番目のサブキャリアにおける $N_{\mathrm{Tx}} \times N_{\mathrm{Rx}}$ 次元の CFR の行列である．2.4 GHz 帯においては，30 サブキャリアからデータを算出するため，$\mathbf{H_k}$ が 30 個取得でき，$30 \times N_{\mathrm{Tx}} \times N_{\mathrm{Rx}}$ の多次元行列として CSI を得る．

$$\mathbf{Y_k} = \mathbf{H_k X_k} + \mathbf{N_k} \tag{5}$$

RSSI による値は，OFDM におけるサブキャリアごとの強度の総和であるが，CSI はサブキャリアごとのデータを多次元行列の形で保持しているため，より高度な分析が可能となっている．CSI はとくに，時間分解能と周波数分解能，安定性について優れている [10]．

3. 関連研究

3.1 ニューラルネットワークによる画像・動画認識

3.1.1 CNN による画像認識

CNN とは，Fukushima らによって提唱された Neocognitron [15] をベースとした，ニューラルネットワークの一種である．Neocognitron は，特徴抽出を行う S 細胞層と，特徴量の位置ずれを許容する働きを持つ C 細胞層（通称，Pooling 層）とを，交互に階層配置した多層神経回路としている [16]．CNN では，S 細胞層の代わりに畳込み層を用意し，フィルタの畳込み処理によって特徴抽出を試みている．LeNet [17] によって，誤差逆伝播法による学習と max-pooling 層を使った学習が確立され，昨今の CNN の基礎となっている．

LeNet 以降，様々な CNN アーキテクチャが考案されてきたが，その中でも画像認識で大きな精度向上を成功させたものが ResNet [18] である．ResNet では，152 層もの超多層アーキテクチャを採用している．また，残差学習という，途中の特徴マップを数層先にバイパスする方法を提案している．

3.1.2 3 次元 CNN による動画認識

動画に映っている動作を識別する技術として，CNN を 3 次元に拡張する手法が提案されている．Tran らの C3D では，2 次元での畳込み処理を 3 次元に拡張する（図 1）ことで，時系列のデータを含んだ学習を可能にした [9]．また，C3D の考え方と ResNet を組み合わせて，さらに精度を向上させた事例も Hara らによって報告されている [19]．

3.2 CSI を利用した認識

CSI を利用したデバイスフリーな位置測位としての初めての試みは，2013 年の Xiao らによる Pilot [1] である．あらかじめ，ある地点にいるときの CSI を Fingerprints として保存しておき，相関が高い地点を実際の位置と推定し

*4 Channel Impulse Response; チャネルインパルス応答
*5 Channel Frequency Response; チャネル周波数特性
*6 Orthogonal Frequency Division Multiplexing; 直交周波数分割多重方式
*7 Multiple Input and Multiple Output

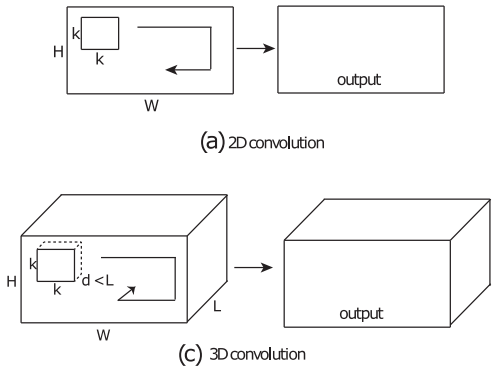

図 1 時間軸方向にも畳込みを行う [9]

ている．また，CSI は時間経過による変動は少ないため，自己相関を求めることによって，室内環境に何かしらの変化が起きたことを検知できる．

ひとの動きを CSI から識別する試みは，ジェスチャ操作の認識に応用されている．Tan らの WiFinger [4] は，指先の動作によるジェスチャの認識を可能にしている．このシステムは指の開閉などの小さな動きを検知できる．CSI に対して主成分分析を行うことで，ジェスチャの個人差へ耐性を持たせている．

Ali らによる WiKey [5] では，CSI の変位からキーボードでタイプした文字を推測できる．タイピングのような小さな動作を検出するために，ローパスフィルターなどのフィルタリング処理へ注力している．この研究から，適切なフィルタリング処理を施せば，小さな動作でも検出できる情報量が CSI には含まれるといえる．

ここまでの CSI を利用した認識技術では，それぞれの目的に合わせて異なる手法を用いている．また，手法ごと予め手動で設定するパラメタ（以下，ハイパーパラメタ）を調整する必要があった．そこで， CNN を利用することで，目的が異なっていても似た手法を使えるようにし，ハイパーパラメタを減らした仕組みも検討されている．

Chen らの ConFi [2] は，CNN を利用した CSI による位置測位技術の最初の事例である．3 アンテナからの CSI の振幅をそれぞれ RGB の値に割り当てて画像を生成し，画像学習の CNN を適用している（図 2）．Wang らの CiFi [3] は，CSI から求めた AoA[*8] を画像として扱い，CNN を応用することで位置測位を行っている．既存の AoA を使った位置測位では，伝播時間の計測が必要だったが，この手法では伝搬時間を使用しない学習を実現した．

Wang らの CSI-Net [6] は，個人識別やサイン識別を CNN で可能にするためのフレームワークである．これは，画像識別の CNN の一種である ResNet をベースに，用途ごと適切なパラメタを適用することで，多用途に使えるようにしている．一方で，時系列データを含めた学習はできないため，ジェスチャ認識などの時間経過で状態が変化するようなものには適用できないことを問題として挙げている．

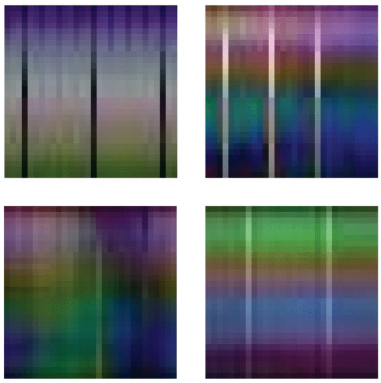

図 2 CSI を RGB に変換した画像 [2]

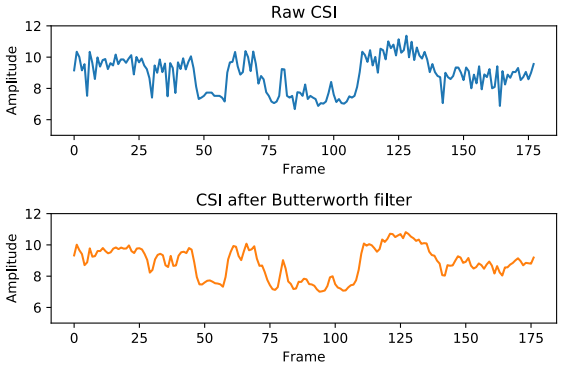

図 3 元データ（上）とバターワースフィルタ後のデータ（下）

時系列情報を保持した CSI への機械学習として，RNN[*9] を使った手法も提案されている．Ohara らは，CSI に畳み込みを行ったのちに RNN を使うことで，時系列情報を保持した状態で，室内の家具の状態を認識している [20]．Yang らでは，CSI から主成分分析した結果を RNN に用いることで，人の歩く方向を認識している [21]．RNN を利用するためには，入力データへ畳み込みや主成分分析などの別手法を併用して，1 次元データに特徴量を抽出しなければならない．

4. 実装

4.1 CSI の取得

Wi-Fi NIC から 0.01 sec 間隔で CSI を取得する．取得した CSI から振幅を抽出して学習に利用する．先行研究 [6] では，ローパスバターワースフィルタによるノイズ除去をすることで，CNN による学習結果が向上するとしている．よって，提案手法においてもローパスバターワースフィルタを適用した．図 3 は，ノイズフィルタ前と後の比較である．

[*8] Angle of Arrival; 電波が送られてくる方向

[*9] Recurrent Neural Network; 再帰型ニューラルネットワーク

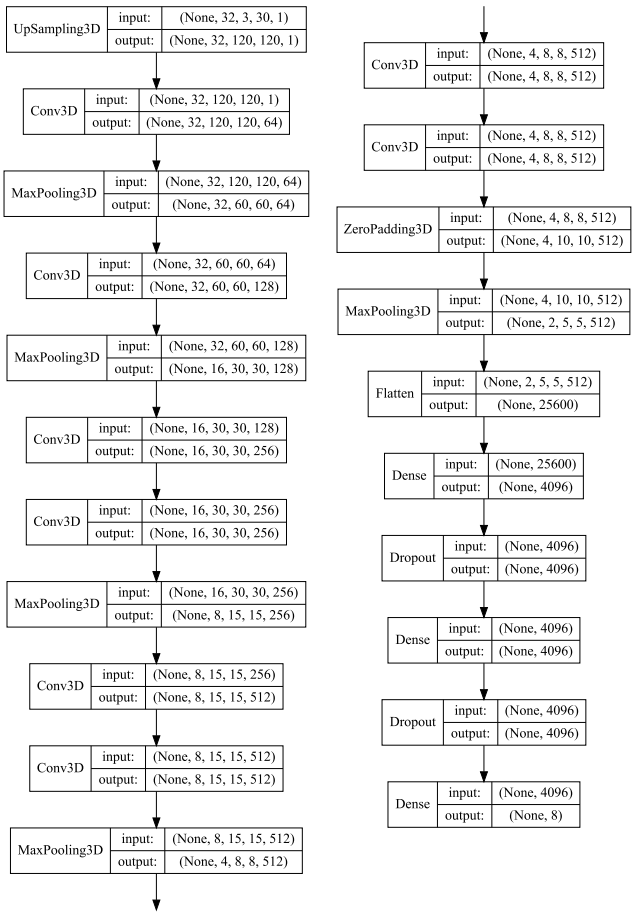

図 4 モデル概略

4.2 3 次元 CNN

時系列情報を含めるために，一定の N_{frame} frames ずつで CSI をまとめる．このとき，1 回のジェスチャから多くのデータを生成するために，8 frames ずつずらして抽出する．例えば，$N_{frame} = 32$ のとき，1 〜 32 frame を抽出した次は，9 〜 40 frame を抽出することになる．今回は，受信側のアンテナ $N_{\mathrm{Rx}} = 3$，送信側のアンテナ $N_{\mathrm{Tx}} = 1$ である．そのため，伝播経路は $N_{\mathrm{Rx}} \times N_{\mathrm{Tx}} = 3$ となり，1 sample は $N_{frame_{[\mathrm{frames}]}} \times 3_{[\mathrm{routes}]} \times 30_{[\mathrm{subcarriers}]}$ の多次元行列となる．

実際のモデルの概略を図 4 に示す．サンプルに対して，C3D [9] を参考に 3 次元に畳込み処理を行う．しかし，取得したサンプルでは畳込みするデータが少ないため，アップサンプリングによって，$N_{frame} \times 120 \times 120$ になるようデータを反復する．Conv3D では，畳み込み処理を行う．活性化関数に ReLU，ストライドは $1 \times 1 \times 1$，ゼロパディングを有効にした．MaxPooling3D では，Max Pooling 法で Pooling を行う．初回のカーネルサイズ $1 \times 2 \times 2$，以降は $2 \times 2 \times 2$ とし，ストライドはカーネルサイズと同等にした．また，ゼロパディングを有効にした．Flatten では，データの平滑化を行う．Dense では，全結合を行う．活性化関数は ReLU を採用した．Dropout では，過学習を抑えるためにデータを一定の確率で無効にする [22]．今回は 0.5 の確率で Dropout を実施した．最適化手法には，SGD[*10] を採用した．SGD では，学習率 1.0×10^{-4}，慣性 0.9 にし，Nesterov の加速勾配降下法を有効にした．

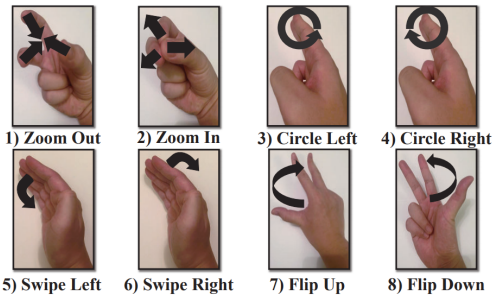

図 5 8 つのジェスチャ [4]

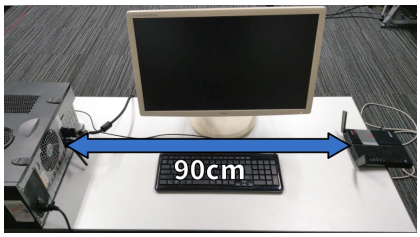

図 6 送信機（右）と受信機（左）

5. 評価実験

5.1 実験手順

23 〜 24 歳の大学院生 5 人（うち男性 5 人）に指定した 8 つのジェスチャ（図 5）を各 50 回試行させ，そのときの CSI を 0.01 sec 間隔で取得した．ジェスチャは先行研究 [4] に倣った．実験では，図 6 で示すように，デスクでの作業を想定して机上に 90 cm 離して送信機と受信機を設置し，実験参加者は送受信機の中間に座らせた．CSI を取得しているときは，実験を行う部屋には実験参加者以外誰もいない状態にした．

CSI を取得する際には，ジェスチャの開始と終了時に，実験参加者へキーボードのスペースキーを押下させた．ジェスチャを行う場所は，送信機と受信機の直線上になるように指示した．実験参加者には，各ジェスチャごとに動きを実際に見せたのちに，50 回ジェスチャを試行させた．実装システムの都合上，CSI が取得できなかった場合は，その都度再試行させた．

CSI には，4.1 節で述べた前処理を行った．各ジェスチャで，前半 9 割に当たる 45 回のデータを学習データに，後半 1 割に当たる 5 回のデータを検証データとして利用した．学習回数は最大 25 epochs とし，過学習を防止するため，検証データの損失を基準とした Early Stopping を有効にした．Early Stopping で停止した epoch の 1 つ前の学習結果を，最終的な学習結果とした．損失の計算には，

[*10] Stochastic Gradient Descent; 確率的勾配降下法

表 1 実験参加者 A のデータに対する 1 sample 当たりのフレーム数による正解率と損失

	サンプル数 (学習/検証)	Epoch 数	正解率	損失
16 frames	6359/717	4 epoch	0.845	0.227
24 frames	5951/673	6 epoch	0.975	0.092
32 frames	5539/627	5 epoch	0.979	0.078

クロスエントロピー誤差を使用した.

CSI の取得には，Halperin らが公開している CSI-Tools [12] を使った．CSI の取得で使う Wi-Fi NIC は，Intel 5300 NIC を使用した．CNN の構築には， Keras[*11] と TensorFlow[*12] を利用した．学習に使用した GPU は，GeForce GTX 980M であり，NVIDIA のドライババージョンは 410.78 であった.

5.2 予備実験｜学習時のフレーム数

先行研究 [23] では，3 次元 CNN の 1 sample に含めるフレーム数が多いと，正解率が向上することが報告されている．4.2 節で述べたサンプル抽出において，1 sample に含める適切なフレーム数を調べるために予備実験を行った．前述の実験参加者のうち，実験参加者 A のデータを使って，16 / 24 / 32 frames ごとにサンプルを抽出し，それぞれのデータを学習させた.

各条件ごとの正解率と損失を表 1 に示す．16 frames のときが最も正解率が低く，32 frames のときが最も正解率が高い．また，フレーム数が多いほど損失が小さくなる傾向がみられた．よって，動画識別の先行研究 [23] と同様に，1 sample に含めるフレーム数を多くしたほうが，学習の精度が上がるといえる.

一方で，24 frames と 32 frames では，正解率が少ししか変化しなかった．しかし，学習回数をみると，32 frames のときが 24 frames のときより少ない学習回数で，十分な学習結果を得ることができている．よって，24 frames 以上にフレーム数を増やしたとしても，学習の精度が著しく向上することはないが，より効率のよい学習ができると推測される．以上のことから，今回の手法では 32 frames ごとにサンプルを抽出することにした.

5.3 評価実験 1 ｜実験参加者ごとに学習する場合

5 人の実験参加者ごとに自身のデータで学習したときの正解率と損失を調べた．5.2 節で述べたように，1 sample につき 32 frames を抽出してサンプルデータを作った．実際のサンプル数は，表 2 に示すとおりである.

正解率と損失を表 2 に示す．正解率の平均は 0.978 であり，先行研究 [4] の正解率が 0.93 であることを考慮すると，従来手法より高い正解率を実現できたといえる．また，

[*11] https://keras.io/
[*12] https://www.tensorflow.org/

表 2 評価実験 1 ｜各実験参加者ごとの正解率と損失

	サンプル数 (学習時/検証時)	Epoch 数	正解率	損失
実験参加者 A	5539/627	5 epoch	0.979	0.078
実験参加者 B	4727/555	7 epoch	1.000	0.000
実験参加者 C	7590/820	7 epoch	0.978	0.030
実験参加者 D	6065/676	6 epoch	0.932	0.151
実験参加者 E	4966/551	6 epoch	1.000	0.048

表 3 評価実験 2 ｜他実験参加者のデータで学習した場合の各実験参加者ごとの正解率と損失

学習 → 評価	Epoch 数	正解率 (学習時)	損失 (学習時)	正解率 (評価時)
BCDE → A	6 epoch	0.980	0.081	0.168
ACDE → B	8 epoch	0.964	0.077	0.174
ABDE → C	5 epoch	0.930	0.135	0.230
ABCE → D	7 epoch	0.968	0.067	0.296
ABCD → E	6 epoch	0.962	0.089	0.050

実験参加者 B と E については，正解率 1.000 となった.

各実験参加者における混同行列を図 7 に示す．実験参加者 A では，ジェスチャ 3 を 4 として，実験参加者 D では，ジェスチャ 2 を 1 として誤認識することがあった．図 5 のジェスチャをみると，それぞれ似通ったジェスチャであることがわかる．一方で，実験参加者 C では，ジェスチャ 4 を 2 として，実験参加者 D では，ジェスチャ 1 を 5 として誤認識することがあった．これらのジェスチャは，それぞれ互いに全く違うものであるが，誤認識が発生している．全く違うジェスチャが誤認識する理由については，今回の結果からは明らかにならなかった.

ジェスチャの間違った識別結果は，既存手法 [4] では複数の誤答があったことに比べ，本手法ではどのジェスチャをみても，誤答は 1 つ以下であった．今回のサンプルの作り方を考えると，ジェスチャの一部分を学習にかけており，ジェスチャ 1 つから複数回の認識をかけられるため，複数の認識結果を合わせることで精度が上がる可能性がある.

5.4 評価実験 2 ｜全実験参加者を合わせて学習する場合

他人のジェスチャから得たデータが，自身のジェスチャ認識に使えるかを調べるため，全実験参加者のデータを合わせて学習を行った．学習時の学習データと検証データは，評価で使う実験参加者以外の 4 人のデータを使った．実際の学習結果を算出する際には，実験参加者の検証データのみを使った．例えば，実験参加者 A に対しては，実験参加者 B / C / D / E のデータから学習したモデルで評価する．実際のサンプル数は，表 3 に示すとおりである.

正解率と損失を表 3 に示す．学習時はどの場合も 9 割以上の正解率であったが，評価時における正解率の平均は

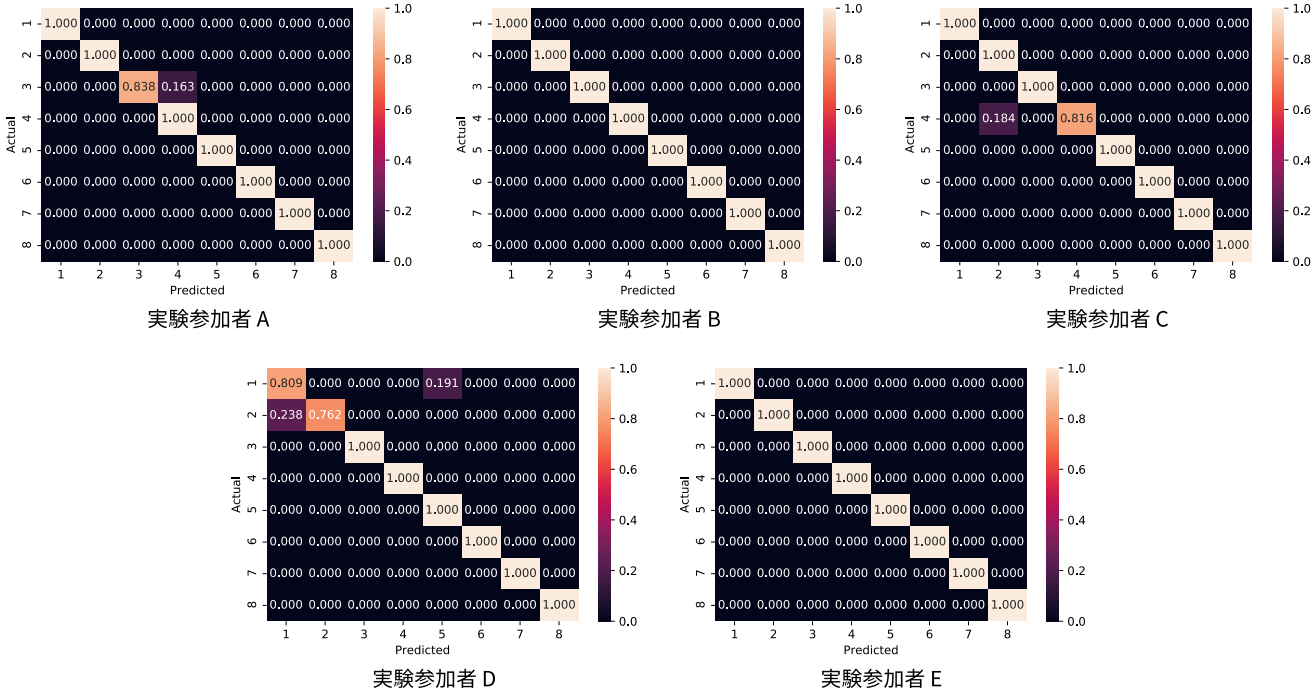

図 7 評価実験 1 ｜各実験参加者ごとの混同行列

0.184 であり，著しく精度が落ちる結果となった．これは，学習に使った実験参加者の動作に過学習が起きているといえる．換言すれば，学習データに自身のデータが含まれていれば，他人のデータが一緒に含まれる場合でも高精度の認識ができているといえる．

学習結果を混同行列を図 8 に示す．実験参加者 A のジェスチャ 6 や，実験参加者 D の ジェスチャ 6 と 8 では，正解率 1.000 であった．一方で，実験参加者 A のジェスチャ 5 や 7 など，全体的に誤答率が 1.000 になってしまっているジェスチャが多く見受けられた．よって，他人のデータを利用すると，自身と他人の動きの差が精度に大きく影響を与えると考えられる．

6. 考察

6.1 学習データの取得

今回の実験では，学習データをすべて取り終えるまで，ひとり当たり 1 時間半程度の時間が必要であった．これには，今回のデバイスに依存するドライバの再起動などが含まれているため，それらを除いた場合には 30 分程度と見込まれる．しかし，一般家庭に導入することを想定するならば，30 分の初期設定は現実的ではない．そのため，どのようにして学習データを用意するかを考える必要がある．ひとつの解決策は，既に学習済みのモデルを使う方法である．5.4 節の実験では，他人のジェスチャによるモデルでは 0.184 の正解率にしか達せず，他人のデータのみによる学習済みモデルを流用することは，提案手法では難しいと考えられる．しかし，自身のデータが含まれている場合は，他人のデータが含まれていても精度が高い結果であったため，自身のデータをある程度含めると精度は上がると考察される．自身のデータをどの程度含めるべきかは検証が必要であるが，学習済みモデルを使うことで，本来より少ないデータでモデルを構築できる可能性はある．

6.2 ジェスチャの動作速度

ジェスチャの動作速度がひとによって異なることは，先行研究においても指摘されている [4]．動作速度の違いによって，2 点の懸念点が挙げられる．1 つは，学習済みモデルの汎用性である．既に学習済みモデルをベースにして，他のユーザを学習するために必要なデータを少なくすることができる．今回の実験における実験参加者ごとの平均ジェスチャ時間を調べた結果，それぞれ違った時間をかけてジェスチャしていた（表 4）．よって，同じ 32 frames の中に含まれる動作が，実験参加者ごとに異なっていたと推察される．これは，5.4 節の結果にある，他人のデータで学習したモデルは適合しない原因のひとつと考えられる．このことから，個々人によってジェスチャの速度が違う場合には，一定区間ごとに含まれるジェスチャの動きが異なり，学習済みモデルの汎用性が下がることが懸念される．

もう 1 つは，区切る区間の長さである．提案手法では 32 frames (0.32 sec) ごとにデータを区切っている．この 32 frames がどのひとにおいても最適であるかどうかは，今回の実験結果からは明らかにはなっていない．今回は，ジェスチャの例を見せてから試行させたため，大きくジェスチャ速度が変わらなかったと考えられる．換言すれば，

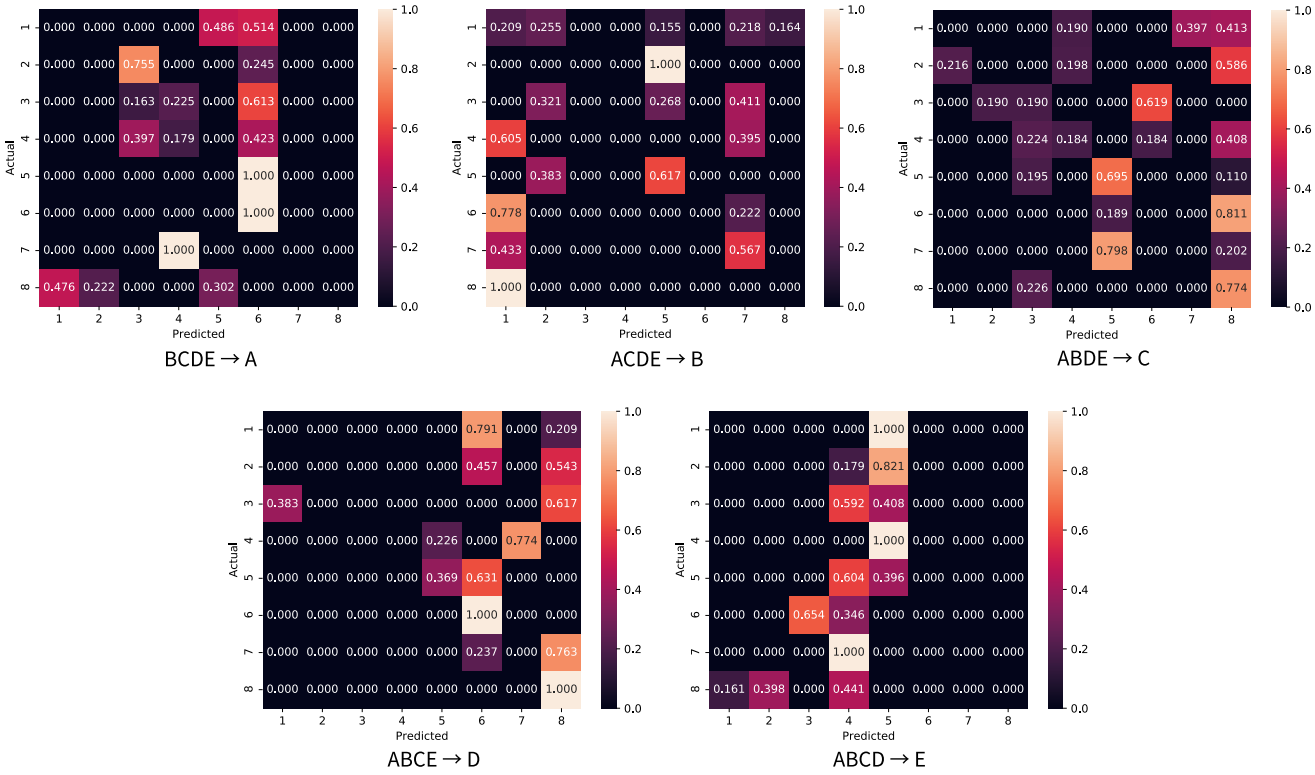

図 8 評価実験 2 ｜他実験参加者（ラベル左側）のデータで学習した場合の各実験参加者ごと（ラベル右側）の混同行列

表 4 実験参加者ごとの平均ジェスチャ時間

	平均ジェスチャ時間 (ms)
実験参加者 A	1105
実験参加者 B	962
実験参加者 C	1420
実験参加者 D	1231
実験参加者 E	1001

実際に導入する場合にも，単純なジェスチャであれば，事前に動作を見せるなど例示をすることで，動作速度を揃えることができるといえる．

6.3 アプリケーションへの活用

提案手法は，送信機と受信機を結ぶ直線上でジェスチャする必要がある．今回の実験では，一般的なオフィスデスクの幅に近い 90 cm を送受信機の間の距離とした．よって，実際の想定場面としては，デスクの端と端に送受信機がある状態となる．デスクの片端にデスクトップコンピュータを送信機として置き，もう片端には手持ちのスマートフォンを受信機として置くことで，新しく機材を用意することなく，即座にジェスチャ認識の環境を構築できる．

今回のジェスチャは，スマートホームの操作などに活用できると考えている．例えば，図 5 の “Swipe Left / Right” でスピーカから再生している曲を切り替えたり，“Circle Left / Right” で室内の温度を調整したりといった具合である．また，CNN を使って CSI を学習させる先行研究 [6] を同じ環境で利用できるメリットもある．先行研究 [6] の個人認識と，提案手法のジェスチャ認識を組み合わせたアプリケーションが構築できる．例えば，“Zoom Out / In” をユーザ P は『スピーカの音量調整』に割り当てているが，ユーザ Q は『室内の照明の明るさ調整』に割り当てたとする．同じ動作で違う挙動になるが，個人認識が働いているため，それぞれのユーザごとに挙動が切り替わり，意図した動作ができるようになる．

7. 結論

動画の内容推定に利用される 3 次元 CNN を応用し，時系列情報を保持したまま学習させることで，CSI からジェスチャを認識する手法を提案した．実験の結果から， 32 frames ごとにデータ抽出すると，正解率が高くなることを示した．また，実験参加者 5 人それぞれで，正解率 0.932 以上を得ることができ，従来手法と比較しても高い正解率を実現した．提案手法で他人のデータのみを学習したモデルは，正解率 0.184 でほとんど適合しなかった．

一方で，ジェスチャの動作速度が遅い場合には，正解率が変わる可能性があることが指摘された．事前に動作を見せる例示を設けることで，動作速度を合わせるなどの施策を考える必要がある．また，学習させるためには，大量の CSI データを用意する必要があり，学習データをどのよ

うに取得するかを検討する必要がある．学習済みモデルを使って，必要な学習データを減らすなどの施策で，現実的な初期設定を設計することが課題である．

参考文献

[1] Xiao, J., Wu, K., Yi, Y., Wang, L. and Ni, L. M.: Pilot: Passive Device-Free Indoor Localization Using Channel State Information, *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems*, ICDCS '13, Washington, DC, USA, IEEE Computer Society, pp. 236–245 (online), DOI: 10.1109/ICDCS.2013.49 (2013).

[2] Chen, H., Zhang, Y., Li, W., Tao, X. and Zhang, P.: ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information, *IEEE Access*, Vol. 5, pp. 18066–18074 (online), DOI: 10.1109/ACCESS.2017.2749516 (2017).

[3] Wang, X., Wang, X. and Mao, S.: CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi, *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6 (online), DOI: 10.1109/ICC.2017.7997235 (2017).

[4] Tan, S. and Yang, J.: WiFinger: Leveraging Commodity WiFi for Fine-grained Finger Gesture Recognition, *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '16, New York, NY, USA, ACM, pp. 201–210 (online), DOI: 10.1145/2942358.2942393 (2016).

[5] Ali, K., Liu, A. X., Wang, W. and Shahzad, M.: Keystroke Recognition Using WiFi Signals, *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, New York, NY, USA, ACM, pp. 90–102 (online), DOI: 10.1145/2789168.2790109 (2015).

[6] Wang, F., Han, J., Zhang, S., He, X. and Huang, D.: CSI-Net: Unified Human Body Characterization and Action Recognition, *arXiv preprint arXiv:1810.03064* (2018).

[7] Wang, Y., Jiang, X., Cao, R. and Wang, X.: Robust Indoor Human Activity Recognition Using Wireless Signals., *Sensors (Basel, Switzerland)*, Vol. 15, No. 7, pp. 17195–208 (online), DOI: 10.3390/s150717195 (2015).

[8] Aljumaily, M.: A survey on WiFi Channel State Information (CSI) utilization in Human Activity Recognition (2016).

[9] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: Learning Spatiotemporal Features with 3D Convolutional Networks, *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, Washington, DC, USA, IEEE Computer Society, pp. 4489–4497 (online), DOI: 10.1109/ICCV.2015.510 (2015).

[10] Yang, Z., Zhou, Z. and Liu, Y.: From RSSI to CSI: Indoor Localization via Channel Response, *ACM Comput. Surv.*, Vol. 46, No. 2, pp. 25:1–25:32 (online), DOI: 10.1145/2543581.2543592 (2013).

[11] IEEE Computer Society: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification (2012).

[12] Halperin, D., Hu, W., Sheth, A. and Wetherall, D.: Tool Release: Gathering 802.11N Traces with Channel State Information, *SIGCOMM Comput. Commun. Rev.*, Vol. 41, No. 1, pp. 53–53 (online), DOI: 10.1145/1925861.1925870 (2011).

[13] Xie, Y., Li, Z. and Li, M.: Precise Power Delay Profiling with Commodity WiFi, *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, New York, NY, USA, ACM, pp. 53–64 (online), DOI: 10.1145/2789168.2790124 (2015).

[14] Zhou, Z., Wu, C., Yang, Z. and Liu, Y.: Sensorless sensing with WiFi, *Tsinghua Science and Technology*, Vol. 20, No. 1, pp. 1–6 (online), DOI: 10.1109/TST.2015.7040509 (2015).

[15] Fukushima, K. and Miyake, S.: Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, *Competition and cooperation in neural nets*, Springer, pp. 267–285 (1982).

[16] 福島邦彦：Deep CNN ネオコグニトロンの学習，人工知能学会全国大会論文集，Vol. JSAI2016, No. 1A3-OS-27a-1, pp. 1–7（オンライン），DOI: 10.11517/pjsai.JSAI2016.0_1A3OS27a1 (2016).

[17] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324 (online), DOI: 10.1109/5.726791 (1998).

[18] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (online), DOI: 10.1109/CVPR.2016.90 (2016).

[19] Hara, K., Kataoka, H. and Satoh, Y.: Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?, *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '18, pp. 6546–6555 (2018).

[20] Ohara, K., Maekawa, T. and Matsushita, Y.: Detecting State Changes of Indoor Everyday Objects Using Wi-Fi Channel State Information, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, Vol. 1, No. 3, pp. 88:1–88:28 (online), DOI: 10.1145/3131898 (2017).

[21] Xu, Y., Chen, M., Yang, W., Chen, S. and Huang, L.: Attention-based Walking Gait and Direction Recognition in Wi-Fi Networks, *arXiv preprint arXiv:1811.07162* (2018).

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, Vol. 15, No. 1, pp. 1929–1958 (online), available from ⟨http://dl.acm.org/citation.cfm?id=2627435.2670313⟩ (2014).

[23] Varol, G., Laptev, I. and Schmid, C.: Long-Term Temporal Convolutions for Action Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 6, pp. 1510–1517 (online), DOI: 10.1109/TPAMI.2017.2712608 (2018).