

スマートフォンの背面カメラ画像を用いた 空中人差し指ジェスチャ認識の精度評価

清 佑輔^{1,a)} 志築 文太郎^{2,b)}

概要:我々は、スマートフォン背面の空中にて実行される人差し指によるジェスチャ（空中人差し指ジェスチャ）を認識する手法を開発している。本手法では、畳み込みニューラルネットワークを用いた認識モデルを使用する。この認識モデルでは、背面カメラ画像から作製された Motion History Image を特徴量とする。今回、我々は、本手法にて認識する 7 種類の空中人差し指ジェスチャを設計した。さらに、4 人の参加者から取得したデータにて認識精度の評価を行ったところ、5 分割交差検証の結果、認識精度の平均は 81.9%であった。また、leave-one-participant-out 交差検証の結果、認識精度の平均は 60.1%であった。これに対して、空中人差し指ジェスチャを 4 種類に限定したところ、5 分割交差検証の結果、認識精度の平均は 89.7%であった。また、leave-one-participant-out 交差検証の結果、認識精度の平均は 74.9%であった。

1. はじめに

タッチスクリーンを備えたモバイル端末であるスマートフォンは、タップおよびスワイプなどのタッチジェスチャを用いて操作される。さらに、複数本の指によるタッチを検知できるマルチタッチスクリーンを搭載したスマートフォンに対して、ユーザはマルチタッチジェスチャを用いて操作できる。例えば、ピンチインおよびピンチアウトは、文書、画像および地図といったコンテンツを閲覧する際に、拡大および縮小の操作として実行される。

一方で、片手にてスマートフォンを持ち、スマートフォンを持った手にて操作（片手使用）する際には、マルチタッチジェスチャの実行が難しい [1]。よって、片手使用時には、スマートフォンを持った手と異なる手にて操作（両手使用）する際に比べて、容易に実行可能なタッチジェスチャの種類は少なくなる。

さらに、タッチジェスチャを行う指の接触面積がスクリーン上の UI に比べて大きいために、誤タッチが発生する問題が指摘されている [1]。また、タッチジェスチャを行う指によってスクリーン上のコンテンツが遮られる問題が指摘されている [2], [3]。

これらの問題を解決するために、スマートフォンにおける操作の種類を拡張する研究が行われている。そのアプローチの 1 つとして、スマートフォン背面における操作が



図 1 空中人差し指ジェスチャの認識の概観。

研究されてきた [2], [4], [5], [6], [7], [8], [9]。これらの研究では、指による操作領域を背面に拡張することにより、片手かつスクリーンを遮らずに実行可能な操作を実現している。

我々は、片手使用時の操作の種類をさらに拡張するために、操作領域をスマートフォン背面周辺の空中にまで拡張することを検討している。本研究では、スマートフォン背面の空中にて実行される人差し指によるジェスチャ（以降、空中人差し指ジェスチャ）を操作に用いる。人差し指は、スマートフォン背面において、安定した持ち方（以降、把持姿勢）にて動かせる領域が他の指と比べて広いことが報告されている [10]。よって、人差し指によるジェスチャを用いることにより、安定した把持姿勢にて実行可能な操作の実現が期待できる。

本研究にて、我々は、背面カメラ画像を用いて空中人差し指ジェスチャを認識する手法を開発している（図 1）。背面カメラ画像を用いることにより、背面カメラを備えた既

¹ 筑波大学情報理工学位プログラム

² 筑波大学システム情報系

^{a)} sei@iplab.cs.tsukuba.ac.jp

^{b)} shizuki@cs.tsukuba.ac.jp

製のスマートフォンに容易に導入可能な手法を目指している。我々は、これまでに、ジェスチャ中に取得された画像内にて指の領域が移動することに基づいて、閾値にて空中人差し指ジェスチャを認識する手法を作製した [11], [12]。さらに、畳み込みニューラルネットワーク（以降、CNN）を用いた認識モデルを使用して、空中人差し指ジェスチャを認識する手法を作製した [13]。この認識モデルでは、背面カメラ画像から作製された Motion History Image [14]（以降、MHI）を特徴量とする。本稿では、まず、CNNを用いた認識モデルを使用して、空中人差し指ジェスチャを認識する手法 [13] に関して述べる。その後、本手法の認識精度を調査した結果を示す。

2. 関連研究

本節ではスマートフォン背面におけるジェスチャを認識する手法と本研究の位置付けを示す。さらに、スマートフォンのカメラを用いてジェスチャを認識する手法と本研究の位置付けを示す。

2.1 スマートフォン背面におけるジェスチャを認識する手法

これまでに、スマートフォン内蔵および追加のセンサを用いて、スマートフォン背面におけるジェスチャを認識する手法が研究されてきた。Seipp および Devlin [4] は、人差し指、中指、および親指を用いてスマートフォン背面および側面を叩くジェスチャを認識する手法を示した。この手法において、叩くジェスチャの認識にはスマートフォンに内蔵されたマイクおよびジャイロセンサが用いられる。Zhang ら [5] は、スマートフォン内蔵の加速度センサ、ジャイロセンサおよびマイクを用いて、片手使用時の背面における人差し指のタップ、人差し指のスワイプ、親指のタップを認識する手法を示した。Huang ら [6] は、CNNを用いてタップの方向、指の接触部位、タップの位置を認識する手法を示した。提案された CNN モデルにおける入力は、スマートフォン内蔵の慣性計測センサから得られたデータおよびスマートフォンのフォームファクタである。Hakoda ら [7] は、スマートフォンの背面にフォトリフレクタを埋め込んだ穴を配置し、その穴を塞ぐジェスチャを認識した。これらの手法は、スマートフォン内蔵の慣性計測センサ、マイク、または追加のセンサをジェスチャの認識に用いる。一方で、我々の手法は、背面カメラを用いて空中人差し指ジェスチャを認識する。

スマートフォン背面におけるジェスチャを認識する手法として、背面カメラを用いる手法が提案されている。Yamada ら [2] は、スマートフォンの背面カメラ上にて実行されるジェスチャを認識した。Wong ら [8] はスマートフォンのカメラと反射鏡を用いて、スマートフォン背面における指のジェスチャを認識した。この認識は、スマート

フォン背面に取り付けた視覚的パターンに基づく。Xiao ら [9] は、スマートフォンの背面カメラを覆うジェスチャを認識する手法を示した。この手法を用いることにより、カメラ上における人差し指によるスワイプジェスチャも認識可能である。これらの手法は背面カメラを用いてスマートフォン背面に対する操作を実現している。ただし、追加の器具にてカメラが覆われる、またはカメラが直接触れられることによりカメラレンズが汚れるという制限がある。一方で、我々の手法は、追加の機器を必要とせず、空中にて実行されるジェスチャを用いたため、カメラに触れる必要もない。

さらに、Oh および Hong [15] は、我々と同様に空中にて実行される人差し指ジェスチャを認識する手法を示した。この手法におけるジェスチャの認識は、背面カメラ画像を特徴量とする機械学習モデルにて認識した指先の位置および画像中の面積に基づく。一方で、我々の手法では、背面カメラ画像から生成された MHI を特徴量とする認識モデルを用いてジェスチャを認識する。

2.2 スマートフォンのカメラを用いてジェスチャを認識する手法

Song ら [16] は、スマートフォン内蔵のカメラを用いて、手のジェスチャを認識するアルゴリズムを提案した。さらに、このアルゴリズムを用いて既存のタッチジェスチャに空中ジェスチャを組み合わせるアプリケーションを示した。Ishii ら [17] は、スマートフォンを用いたバーチャルリアリティ環境向けのターゲット選択手法を提案している。この手法において、ポインタは背面カメラにて撮影した手の位置に応じて表示される。また、ユーザは手のジェスチャによりターゲットの選択が可能である。Yu ら [3] は、直角プリズムミラーを取り付けたフロントカメラを用いて、スマートフォンを持つ手およびタッチする手の状態の認識を可能にした。Yang ら [18] は、スマートフォンのカメラに 360 度撮影可能なレンズを取り付けることにより、スマートフォン周囲における手の認識と空中ジェスチャの認識を実現した。これらはスマートフォンを把持した手と異なる手のジェスチャを認識する手法である。一方で、我々の手法はスマートフォンを把持している手の人差し指によるジェスチャを認識する手法である。

3. ジェスチャ認識手法

我々は、スマートフォンの背面カメラにて取得した画像から、空中人差し指ジェスチャを認識する手法を開発している。本節では、まず、本手法にて認識する空中人差し指ジェスチャを述べる。さらに、空中人差し指ジェスチャの事前動作の認識、および空中人差し指ジェスチャの認識を述べる。

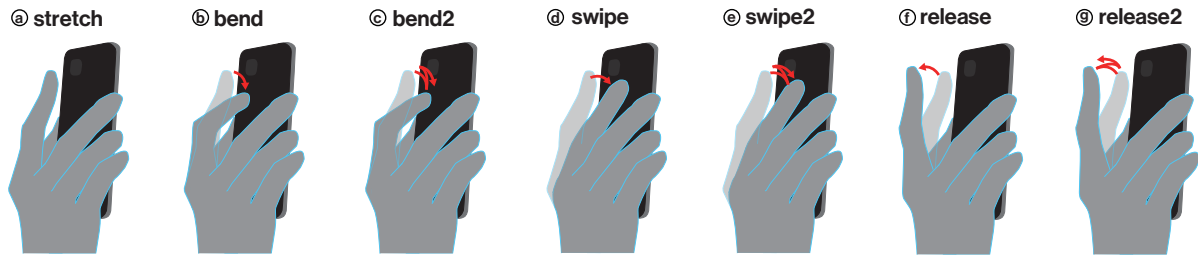


図 2 7 種類の空中人差し指ジェスチャ.

3.1 認識する空中人差し指ジェスチャ

我々は、本手法にて認識する 7 種類の空中人差し指ジェスチャを設計した (図 2)。これらは、スマートフォンを把持しながら実行可能なジェスチャおよびそれを 2 回繰り返すジェスチャから構成されている。なお、我々は、人差し指が背面カメラの前にて動かされることを前提に空中人差し指ジェスチャを設計した。そのため、これらの空中人差し指ジェスチャを実行する際には、スマートフォンの背面カメラに向けて人差し指を伸ばす事前動作が必要である。

7 種類の空中人差し指ジェスチャについて、ユーザは人差し指を事前動作の後に次のように動かす。

- stretch: 伸ばしたまま保持する (図 2a)
- bend: 端末方向に曲げる (図 2b)
- bend2: bend を 2 回行う (図 2c)
- swipe: 中指方向に払う (図 2d)
- swipe2: swipe を 2 回行う (図 2e)
- release: 端末反対方向に伸ばす (図 2f)
- release2: release を 2 回行う (図 2g)

空中人差し指ジェスチャを用いた操作の利用例として、通知の表示および端末設定の変更のショートカットが挙げられる (図 3)。例えば、ユーザは、bend を実行することによって通知を表示できる。さらに、release を実行することによって Wi-Fi 接続機能の ON/OFF を切り替えられる。我々は、ユーザが誤って事前動作を行った際にキャンセルするための操作として、stretch を用いることを想定している。よって、残りの 6 種類の空中人差し指ジェスチャを、通知の表示および端末設定の変更のショートカットに割り当てられる。

通知の表示および端末設定の変更は、多くのスマートフォンにて、ステータスバーに収められている機能である。ただし、ステータスバーは片手使用時に親指の届きにくい画面上部にあるため、ユーザがこれらの機能を使用する際に、不安定な把持姿勢を伴う。一方で、本ショートカットを用いることにより、ユーザは届きにくい画面上部に親指を伸ばすことなく、これらの機能を使用できる。

3.2 事前動作の認識

本手法では、まず空中人差し指ジェスチャの事前動作の

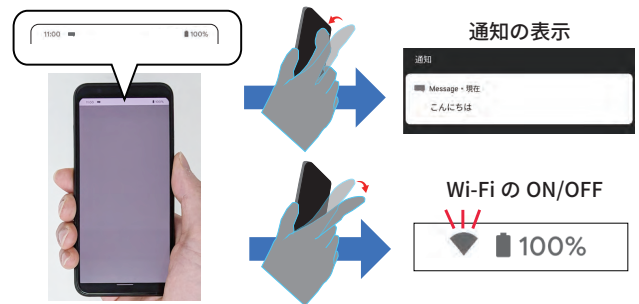


図 3 空中人差し指ジェスチャを用いた操作の利用例. ユーザは空中人差し指ジェスチャを実行することにより、通知の表示および端末設定の変更を行える。

認識を行う。この認識により、空中人差し指ジェスチャおよび日常の手の動作を切り分ける狙いがある。

事前動作の認識の流れを図 4 に示す。事前動作の認識は、スマートフォンの背面カメラを用いて、50 Hz にて取得された画像 (240 × 320 pixels) に基づく。まず、取得した画像内の人差し指の領域を特定するために、肌の領域の抽出およびグレイスケール化の処理を行う。肌の領域を抽出する処理には、Song ら [16] のアルゴリズムを使用する。次に、画像内における肌の領域の面積、つまり、人差し指の面積を算出する。

事前動作の認識には、ある一定時間取得した画像を使用する。取得したすべての画像について、人差し指の面積の最大値および最小値を取得する。この最大値および最小値が閾値以内である場合に、事前動作として認識する。なお、画像を取得する時間および認識に用いる閾値は、利用時にユーザによって調節されることを想定している。

3.3 空中人差し指ジェスチャの認識

空中人差し指ジェスチャの認識の流れを図 5 に示す。空中人差し指ジェスチャの認識は、事前動作を認識した直後の一定時間にて取得した画像に基づく。この一定時間は、7 種類の空中人差し指ジェスチャを実行可能な時間として、著者の経験に基づき 700 ms とした。本手法では、まず、700 ms 間に 35 枚の画像 (240 × 320 pixels) を取得する。次に、この画像に対して、肌の領域の抽出およびグレイスケール化の処理を行う (図 5a)。肌の領域を抽出する処理は、事前動作の認識と同じである。さらに、これらの

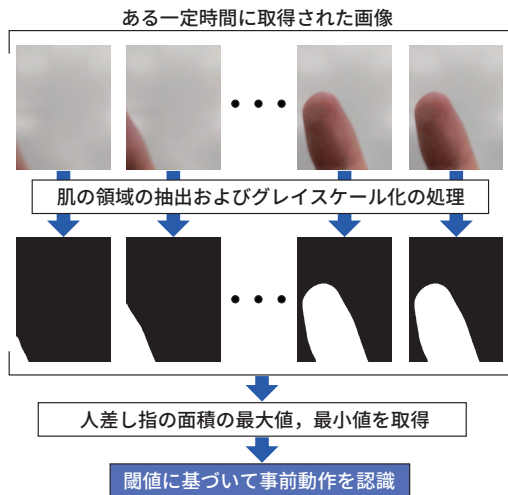


図 4 事前動作の認識の流れ。

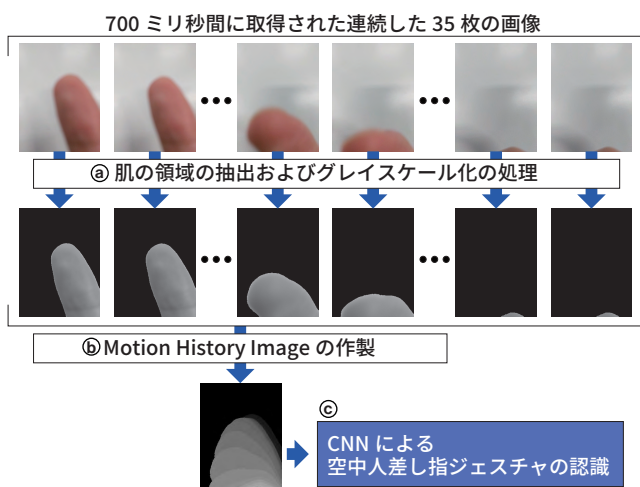


図 5 空中人差し指ジェスチャの認識の流れ。a: 取得した 35 枚の画像に対して、肌の領域の抽出およびグレースケール化の処理を行う。b: Motion History Image を作製する。c: CNN を用いて、Motion History Image から空中人差し指ジェスチャを認識する。

処理が行われた 35 枚の画像を用いて、MHI が生成される (図 5b)。MHI は動作履歴を輝度値にて表現する画像である。つまり、MHI にて新しい動作領域は輝度値が高く、古い動作領域は輝度値が低く示される。これらの処理を通して、空中人差し指ジェスチャの動作履歴を含む MHI を得る。

本手法では、MHI から空中人差し指ジェスチャを認識するために、CNN を用いる (図 5c)。空中人差し指ジェスチャの認識に使用される CNN のモデルを図 6 に示す。このモデルは、 240×320 pixels の MHI を入力として、認識したジェスチャの種類またはジェスチャが未実行であるかを出力する。このモデルのすべての畳み込み層において、フィルタ数は 32、活性化関数は ReLU、ストライドは 2×2 である。また、すべてのプーリング層におけるプーリングには、Max Pooling 法を採用した。プーリングにおけるカーネルサイズは 2×2 である。ドロップアウト層で

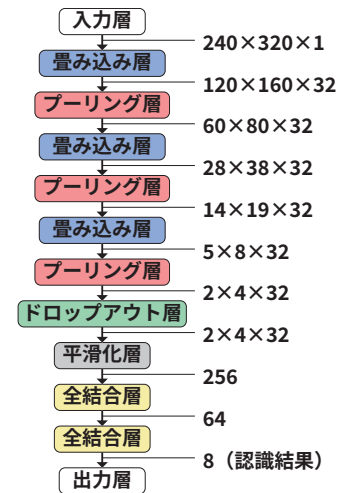


図 6 空中人差し指ジェスチャを認識する CNN モデルの概観。

は、データを 50% の確率にて無効にする。この層を用いることによって、過学習を抑える狙いがある。全結合層における活性化関数は ReLU である。このモデルにおける最適化手法には、Adam を採用した。なお、学習率は 0.001 である。

4. 認識精度の評価

我々はジェスチャ認識手法の認識精度を評価するために実験を行った。

4.1 参加者および実験環境

参加者はボランティアとして参加した研究室内の大学・大学院生 4 人 (平均年齢 22.5 歳, SD = 1.0 歳) である。参加者の内、男性が 3 人、女性が 1 人であり、全員が右利きであった。参加者の手の大きさは平均 184.5 mm (SD = 11.8 mm) であり、人差し指の長さは平均 70.2 mm (SD = 5.3 mm) であった。参加者全員が、同じ室内にて、椅子に腰掛けて、右手にて実験タスクを行った。実験中はブラインドが閉じられた上で、照明が点けられた。同室内にて、照度計 (サトテック デジタル照度計 CENTER530) を用いて計測した照度は、上向きにておよそ 620–650 lux, 下向きにておよそ 230–260 lux であった。

実験では、スマートフォンとして Google Pixel 4 (147.1 mm $\times$ 68.8 mm $\times$ 8.2 mm, 5.7 インチ, Android 11) を用いた。このスマートフォンの背面カメラは背面上部の左側に配置されている。学習モデルの作製には Google Colaboratory [19] を使用した。この学習には、ライブラリとして TensorFlow 2.4.1 を用いた。

4.2 タスクおよび実験手順

本実験にて、参加者が行ったタスクは次の通りである。まず、空中人差し指ジェスチャの事前動作として、参加者は背面カメラの前に人差し指を伸ばす。参加者は、この動

作をタスク開始から1秒経過するまでに実行するように指示された。事前動作を行う1秒が経過した後に、参加者は空中人差し指ジェスチャを行う。参加者は、空中人差し指ジェスチャを1秒経過するまでに実行するように指示された。背面カメラの前に人差し指を伸ばした後に、空中人差し指ジェスチャを行うことを1試行とした。なお、参加者が事前動作または空中人差し指ジェスチャを正しく行えなかったと判断した場合、その試行をやり直すことを指示された。

さらに、参加者は、2秒間自由に把持姿勢を変える持ち替えタスクを行った。このタスクにて、ジェスチャ未実行の画像が取得された。

本実験において、1セッションは80試行（7種類のジェスチャおよび持ち替えタスク×繰り返し10回）であった。実行するタスクはランダムな順に提示された。参加者全員がそれぞれ3セッション分のタスクを行った。各セッション間に5分程度の休憩が設けられた。実験に要した時間は、参加者1人当たり45分程度であった。

4.3 画像データ

本実験において、1試行にて取得した画像の枚数は100枚である。これらの画像の内、時系列順に51枚目から94枚目までを、空中人差し指ジェスチャを実行中の画像として扱った。さらに、35枚の画像に対して、肌の領域の抽出およびグレイスケール化の処理、ならびにMHIの作製を行った。これらの処理によって、10枚のMHIを得た。実験を通して、画像データとして9,600枚（10枚×80試行×3セッション×4人）のMHIを得た。

4.4 結果1

取得した画像データを用いて、7種類の空中人差し指ジェスチャの認識精度を調査した。この調査では、5分割交差検証およびleave-one-participant-out交差検証を行った。

4.4.1 5分割交差検証

画像データ9,600枚を用いて5分割交差検証を行った。分割においては、9,600枚の画像データを1,920枚ずつの5つのデータセットとした。このとき、データセット間にて、空中人差し指ジェスチャの種類および画像の枚数のバランスを取った。また、同じ試行から得られた画像が同じデータセットに含まれるように分割を行った。交差検証においては、5つのデータセットのうち、1つを検証データ、残りの4つを学習データとした。ここで、学習モデルの汎用性を高めるために、学習データにのみ拡張を行った。データセット内の画像に対して、中心を基準として1.2倍に拡大した画像および0.8倍に縮小した画像を学習データに追加した。これにより、学習データに含まれる画像の枚数を3倍に拡張した。よって、検証に用いられる画像は1,920枚、学習に用いられる画像は23,040枚であった。交差検証

④ 5分割交差検証

真値\予測値	未実行	stretch	bend	bend2	swipe	swipe2	release	release2
未実行	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
stretch	1.1%	92.3%	0.0%	1.0%	0.3%	0.7%	2.7%	2.0%
bend	1.3%	0.0%	73.5%	8.6%	10.4%	1.4%	2.8%	2.0%
bend2	0.0%	0.4%	4.2%	66.1%	0.8%	14.5%	1.8%	12.2%
swipe	2.8%	0.1%	7.1%	2.3%	78.4%	6.9%	1.3%	1.0%
swipe2	0.0%	0.7%	1.0%	12.2%	3.4%	78.4%	0.7%	3.7%
release	0.9%	2.0%	1.5%	1.6%	0.3%	0.9%	85.9%	6.8%
release2	0.0%	0.7%	0.2%	11.7%	0.1%	1.5%	5.3%	80.7%

⑤ leave-one-participant-out 交差検証

真値\予測値	未実行	stretch	bend	bend2	swipe	swipe2	release	release2
未実行	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
stretch	0.8%	74.1%	0.0%	2.1%	0.5%	6.2%	13.8%	2.5%
bend	1.3%	0.2%	58.8%	7.2%	20.9%	3.7%	4.3%	3.8%
bend2	0.0%	0.6%	6.3%	30.3%	3.4%	22.7%	14.4%	22.3%
swipe	2.8%	0.2%	16.0%	5.3%	58.6%	11.3%	5.1%	0.8%
swipe2	0.0%	1.0%	0.8%	25.1%	5.0%	54.9%	3.0%	10.2%
release	3.0%	9.8%	18.7%	3.3%	0.3%	1.0%	43.7%	20.3%
release2	0.0%	0.5%	1.8%	15.2%	0.0%	9.2%	13.0%	60.4%

図7 交差検証にて得られた混同行列。横軸が真値、縦軸が予測値を表す。a: 5分割交差検証, b: leave-one-participant-out 交差検証。

の結果、分割されたデータセット毎の認識精度は、82.9%、81.6%、82.4%、83.1%、79.6% ($M = 81.9\%$, $SD = 1.43\%$)であった。図7aに混同行列を示す。

4.4.2 leave-one-participant-out 交差検証

画像データ9,600枚を用いてleave-one-participant-out交差検証を行った。交差検証においては、1人分の画像データを検証データ、残りの3人分の画像データを学習データとする。ここで、5分割交差検証と同じく、学習モデルの汎用性を高めるために、学習データにのみ拡張を行った。データセット内の画像に対して、中心を基準として1.2倍に拡大した画像および0.8倍に縮小した画像を学習データに追加した。これにより、学習データに含まれる画像の枚数を3倍に拡張した。よって、検証に用いられる画像は2400枚、学習に用いられる画像は21,600枚であった。交差検証の結果、分割されたデータセット毎の認識精度は、62.1%、49.1%、66.9%、62.3% ($M = 60.1\%$, $SD = 7.69\%$)であった。図7bに混同行列を示す。

4.5 結果2

同じ動作を繰り返すジェスチャ（bend2, swipe2, およびrelease2）を除いた、4種類の空中人差し指ジェスチャに限定して認識精度を調査した。この調査では、同じ動作を繰り返すジェスチャのMHIを除いた6,000枚の画像データを用いて、5分割交差検証およびleave-one-participant-out交差検証を行った。

4.5.1 4種類の空中人差し指ジェスチャに限定した5分割交差検証

画像データ6,000枚を用いて5分割交差検証を行った。画像データの分割および学習データ拡張の方法は、4.4.1節に示した方法と同じである。これにより、検証に用いられる画像は1,200枚、学習に用いられる画像は14,400枚

⑤ 5分割交差検証

真値\予測値	未実行	stretch	bend	swipe	release
未実行	100.0%	0.0%	0.0%	0.0%	0.0%
stretch	0.8%	93.5%	0.8%	0.8%	4.1%
bend	1.3%	0.2%	82.3%	10.3%	6.1%
swipe	2.7%	0.6%	13.7%	80.7%	2.4%
release	1.1%	2.8%	3.4%	0.6%	92.2%

⑥ leave-one-participant-out 交差検証

真値\予測値	未実行	stretch	bend	swipe	release
未実行	100.0%	0.0%	0.0%	0.0%	0.0%
stretch	0.8%	77.4%	1.2%	2.7%	17.9%
bend	1.3%	0.8%	62.6%	23.5%	11.9%
swipe	2.6%	0.8%	19.9%	70.5%	6.3%
release	2.3%	14.8%	15.8%	2.8%	64.2%

図 8 4 種類の空中人差し指ジェスチャに限定した交差検証にて得られた混同行列。横軸が真値、縦軸が予測値を表す。a: 5 分割交差検証, b: leave-one-participant-out 交差検証。

であった。交差検証の結果、分割されたデータセット毎の認識精度は、90.2%, 87.3%, 91.3%, 91.5%, 88.3% ($M = 89.7\%$, $SD = 1.86\%$) であった。図 8a に混同行列を示す。

4.5.2 4 種類の空中人差し指ジェスチャに限定した leave-one-participant-out 交差検証

画像データ 6,000 枚を用いて leave-one-participant-out 交差検証を行った。画像データの分割および学習データ拡張の方法は、4.4.2 節に示した方法と同じである。これにより、検証に用いられる画像は 1,500 枚、学習に用いられる画像は 13,500 枚であった。交差検証の結果、分割されたデータセット毎の認識精度は、85.4%, 82.9%, 71.6%, 59.8% ($M = 74.9\%$, $SD = 11.7\%$) であった。図 8b に混同行列を示す。

4.6 考察

7 種類の空中人差し指ジェスチャについて、5 分割交差検証の結果、認識精度の平均は 81.9% であった。一方で、ジェスチャを 4 種類に限定した場合、認識精度の平均は 89.7% であった。このことから、ある程度の性能にて 4 種類の空中人差し指ジェスチャを認識できることがわかった。また、leave-one-participant-out 交差検証の結果、認識精度の平均は、ジェスチャが 7 種類の場合に 60.1%, ジェスチャが 4 種類の場合に 74.9% であった。よって、いずれの場合においても汎用性が低いことがわかる。このことから、実利用の際はユーザへのパーソナライズが必要である。

図 7 および図 8 の混同行列が示すように、5 分割交差検証および leave-one-participant-out 交差検証のいずれにおいても、ジェスチャの未実行は正しく認識された。このため、事前動作が誤って認識された場合においても、空中人差し指ジェスチャが認識される可能性は低いと考える。

実験結果から、bend および swipe を誤認識しやすい傾向が確認された。これらのジェスチャについて、人差し指の移動方向は、ユーザの把持姿勢によって同様である可能性がある。人差し指の付け根がスマートフォンの側面に位置す

るように把持した場合、bend および swipe における人差し指の移動方向は、いずれも取得される画像中にて右上から左下となる。よって、bend および swipe の MHI から似た特徴が学習されたと考える。また、leave-one-participant-out 交差検証においては、stretch および release が誤認識される傾向がある。この傾向について、release が実行された際の人差し指の動作が小さかったため、stretch と同様な特徴が学習された可能性がある。

さらに、ジェスチャが 7 種類の場合、bend2, swipe2, および release2 にて誤認識されやすい傾向が見られる。この原因として、学習時に MHI からこれらのジェスチャの特徴が捉えられなかったことが考えられる。MHI において、同じ動作を正確に繰り返した場合、過去の動作が新しい動作に上書きされる。bend2, swipe2, および release2 は同じ動作を 2 回繰り返す必要があるため、生成される MHI において指の動作の情報が少なくなった可能性がある。

5. 議論および今後の課題

空中人差し指ジェスチャの認識精度を向上させるために、学習モデルの改善が必要である。これに対して、学習に用いるデータの多様性を高めることによって、より精度が向上する可能性がある。よって、追加のデータを参加者から取得すること、より多くの参加者からデータを取得すること、および学習データの拡張方法を増やすことを計画している。さらに、CNN の畳み込み層の深さおよび最適化手法といった、パラメータチューニングを行うことにより、性能の向上を目指す。また、スマートフォン内蔵センサにて計測された加速度および角速度を特徴量に含めることによって、より堅牢なモデルを作製できる可能性がある。

実験結果から、bend2, swipe2, および release2 の同じ動作を繰り返すジェスチャを加えた場合、認識精度が低くなることがわかった。よって、これらのジェスチャを認識するための処理を認識手法に追加することを検討している。追加する処理の候補として、ジェスチャが実行された後に、あらかじめ設定された猶予時間内にジェスチャが再実行された場合に、2 回目のジェスチャとして認識する処理を考えている。今後、この処理を追加した上で、認識精度を調査する。

スマートフォンの片手使用時において、人差し指は把持に用いられるため、ユーザが意図して人差し指を動かすことは少ない。そのため、スマートフォンを把持しながら実行する空中人差し指ジェスチャは難易度が高い可能性がある。容易に実行可能な操作を実現するために、空中人差し指ジェスチャの設計に改善の余地がある。アイデアの 1 つとして、より小さい動作のジェスチャにて設計することが挙げられる。現状の設計と比べて小さい動作にて実行可能であれば、ユーザの身体への負荷が小さくなる。これにより、実行難易度が低くなると考える。ただし、小さい動作

の場合, MHI における動作履歴の特徴が小さくなることが予想される. そのため, 現状の手法にて認識可能かを確認する必要がある.

6. おわりに

本稿では, CNN を用いた認識モデルにて, 空中人差し指ジェスチャを認識する手法を示した. 我々は, 本手法にて認識する 7 種類の空中人差し指ジェスチャを設計した. さらに, 4 人の参加者から取得したデータを用いて, 認識精度の評価を行った. その結果, 認識精度は, 5 分割交差検証にて平均 81.9%, leave-one-participant-out 交差検証にて平均 60.1%であった. また, 4 種類の空中人差し指ジェスチャに限定したところ, 5 分割交差検証にて平均 89.7%, leave-one-participant-out 交差検証にて平均 74.9%であった. このことから, 4 種類の空中人差し指ジェスチャをある程度の性能にて認識できる一方で, 本手法の汎用性は低いことがわかった. 今後は, ジェスチャを認識する手法の改善および空中人差し指ジェスチャの設計の改善を行う.

参考文献

- [1] Boring, S., Ledo, D., Chen, X. A., Marquardt, N., Tang, A. and Greenberg, S.: The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction, *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, New York, NY, USA, ACM, pp. 39–48 (online), DOI: 10.1145/2371574.2371582 (2012).
- [2] Yamada, W., Manabe, H. and Ikeda, D.: CamTrack-Point: Camera-Based Pointing Stick Using Transmitted Light Through Finger, *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, New York, NY, USA, ACM, pp. 313–320 (online), DOI: 10.1145/3242587.3242641 (2018).
- [3] Yu, C., Wei, X., Vachher, S., Qin, Y., Liang, C., Weng, Y., Gu, Y. and Shi, Y.: HandSee: Enabling Full Hand Interaction on Smartphone with Front Camera-based Stereo Vision, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, ACM, pp. 705:1–705:13 (online), DOI: 10.1145/3290605.3300935 (2019).
- [4] Seipp, K. and Devlin, K.: BackPat: One-Handed Off-Screen Patting Gestures, *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, MobileHCI '14, New York, NY, USA, ACM, pp. 77–80 (online), DOI: 10.1145/2628363.2628396 (2014).
- [5] Zhang, C., Guo, A., Zhang, D., Southern, C., Arriaga, R. and Abowd, G.: BeyondTouch: Extending the Input Language with Built-in Sensors on Commodity Smartphones, *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, New York, NY, USA, ACM, pp. 67–77 (online), DOI: 10.1145/2678025.2701374 (2015).
- [6] Huang, M. X., Li, Y., Nazneen, N., Chao, A. and Zhai, S.: TapNet: The Design, Training, Implementation, and Applications of a Multi-Task Learning CNN for Off-Screen Mobile Input, *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, ACM, pp. 282:1–11 (online), available from <https://doi.org/10.1145/3411764.3445626> (2021).
- [7] Hakoda, H., Fukatsu, Y., Shizuki, B. and Tanaka, J.: Back-of-Device Interaction Based on the Range of Motion of the Index Finger, *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, OzCHI '15, New York, NY, USA, ACM, pp. 202–206 (online), DOI: 10.1145/2838739.2838812 (2015).
- [8] Wong, P. C., Fu, H. and Zhu, K.: Back-Mirror: Back-of-Device One-Handed Interaction on Smartphones, *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*, SA '16, New York, NY, USA, ACM, pp. 10:1–5 (online), DOI: 10.1145/2999508.2999522 (2016).
- [9] Xiao, X., Han, T. and Wang, J.: LensGesture: Augmenting Mobile Interactions with Back-of-Device Finger Gestures, *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, New York, NY, USA, ACM, pp. 287–294 (online), DOI: 10.1145/2522848.2522850 (2013).
- [10] Le, H. V., Mayer, S., Bader, P. and Henze, N.: Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, ACM, pp. 31:1–12 (online), available from <https://doi.org/10.1145/3173574.3173605> (2018).
- [11] 清佑輔, 船越南斗, 志築文太郎: スマートフォン背面の人差し指を用いた片手操作向け入力拡張手法の検討, 第 28 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2020), 日本ソフトウェア科学会 (2020). 6 pages.
- [12] Sei, Y. and Shizuki, B.: Expanding One-Handed Input Vocabulary for Smartphone Using In-Air Gesture of Index Finger Captured by Rear Camera, *Asian CHI Symposium 2021*, New York, NY, USA, Association for Computing Machinery, pp. 61–63 (online), available from <https://doi.org/10.1145/3429360.3468181> (2021).
- [13] 清佑輔, 志築文太郎: スマートフォンの背面カメラを用いて認識可能な空中人差し指ジェスチャによるショートカットの検討, 第 29 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2021), 日本ソフトウェア科学会 (2021). 2 pages.
- [14] Bobick, A. F. and Davis, J. W.: The Recognition of Human Movement Using Temporal Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, pp. 257–267 (online), DOI: 10.1109/34.910878 (2001).
- [15] Oh, B.-H. and Hong, K.-S.: Finger Gesture-based Three-Dimension Mobile User Interaction Using a Rear-facing Camera, *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 8, pp. 119–130 (online), DOI: 10.14257/ijmue.2013.8.5.12 (2013).
- [16] Song, J., Sörös, G., Pece, F., Fanello, S. R., Izadi, S., Keskin, C. and Hilliges, O.: In-air Gestures Around Unmodified Mobile Devices, *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, New York, NY, USA, ACM, pp. 319–329 (online), DOI: 10.1145/2642918.2647373 (2014).
- [17] Ishii, A., Adachi, T., Shima, K., Nakamae, S., Shizuki, B. and Takahashi, S.: FistPointer: Target Selection Technique Using Mid-Air Interaction for Mobile VR Environment, *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing*

- tems*, CHI EA '17, New York, NY, USA, ACM, p. 474 (online), DOI: 10.1145/3027063.3049795 (2017).
- [18] Yang, X.-D., Hasan, K., Bruce, N. and Irani, P.: Surround-See: Enabling Peripheral Vision on Smartphones during Active Use, *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, New York, NY, USA, ACM, pp. 291–300 (online), DOI: 10.1145/2501988.2502049 (2013).
- [19] Google LLC: Colaboratory - Google (2021). <https://colab.research.google.com/> (最終参照日：2021 年 12 月 9 日) .