人間行動認識に対するクラス階層性を用いた補助学習

近藤 和真[1]　長谷川 達人[1]

概要：
深層学習を用いたセンサによる人間行動認識では，Convolutional Neural Network(CNN) をベースとした手法が多数提案されている．一般的な CNN ベースの行動認識モデルでは，各クラスの正解ラベルは One-hot ベクトルでエンコードされ，分類モデルの出力は正解クラスの One-hot ベクトルに近づくように訓練される．One-hot ベクトル表現では異なるクラスは互いに独立しているため，一般的な CNN ベースの行動認識モデルではクラス間の関係性を考慮せずに訓練が行われている．しかし，実際の行動クラスでは階層的な関係性を持つことが多く，クラス間の階層関係を補助的に利用することでモデルの認識性能を向上させることが期待できる．そこで本研究では，画像認識分野で提案されている Branch CNN(B-CNN) に対してクラス階層の自動構成手法を追加したクラス階層適応型 B-CNN を提案する．B-CNN は人が手動で設計したクラス階層を用いるため，クラス数が多い場合に適切なクラス階層を手動で設計することが困難であるという課題があった．そこで提案手法では，B-CNN の訓練に用いるクラスの階層構造自体を訓練データから自動で構成することで，クラス数が多い場合でも効果的に B-CNN を訓練することを可能とした．複数のベンチマークデータセットを用いた検証により，提案手法はクラス間の階層関係を考慮しない一般的な CNN モデルを上回る分類精度を達成し，加えて，提案手法の認識性能が事前知識をもとに人が手動で設計したクラス階層を用いた B-CNN モデルに匹敵することを確認した．

Auxiliary Learning using Class Hierarchy for Human Activity Recognition

KAZUMA KONDO[1]　TATSUHITO HASEGAWA[1]

1. はじめに

センサを用いた人間行動認識では，センサで観測したユーザの行動データからそのデータが捉えている行動や観測時のユーザの状況を自動で推定する．得られた推定結果はユーザ活動のサポートや組織の意思決定に利用することができ，人間行動認識は幅広い分野で応用が期待される分野となっている [1].

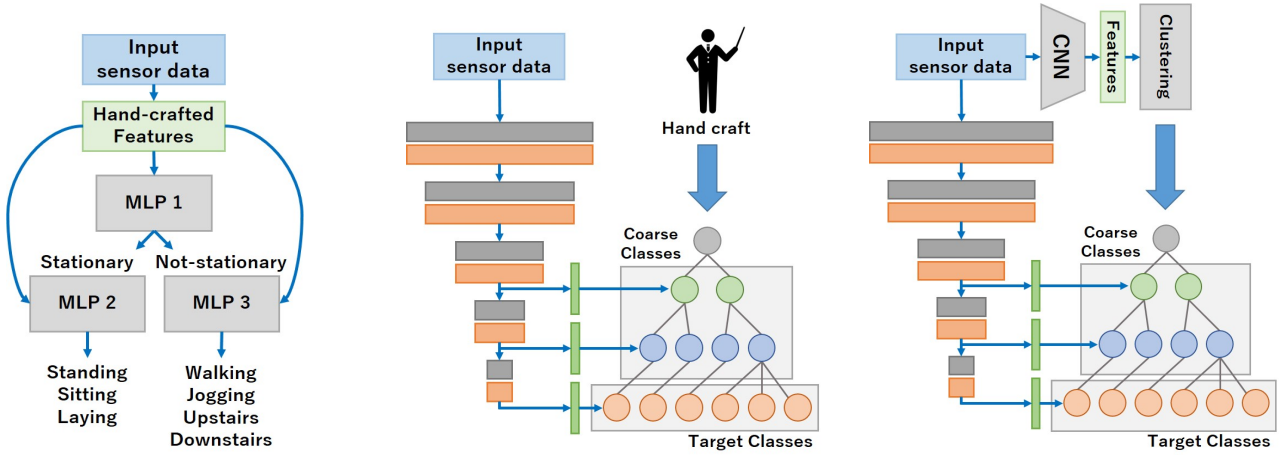
近年では様々な分野で深層学習の応用が行われており，人間行動認識分野においても深層学習を用いた手法が数多く提案されている．深層学習を用いた人間行動認識では Convolutional Neural Network(CNN) をベースとした手法が多数提案されている [2].

一般的な CNN によるクラス分類では，クラス間の関係性を考慮しない分類を行うことが多いが，実際にはクラス間には階層的な関係性が存在する [3]．例えば，「静止」，「歩行」，「階段を上る」，「階段を下る」の 4 つのクラスでは，「静止」以外の 3 クラスは「非静止」といったように，より抽象的なクラスを考えることができ，クラス間に階層的な構造が存在することが分かる．クラスの階層関係はモデルの分類パターンに影響を与えることが知られている [4]．上記の例では「静止」と「歩行」よりも「歩行」と「階段を上る」の方がクラス間の類似性は高く誤分類が多くなる．クラス間の階層関係が事前に分かっている場合はこの関係性を利用することで類似クラス間の誤分類を減らし，モデルの性能を向上させることが期待できる．

深層学習を用いた行動認識手法では，Fazli ら [5] が図 1(a) のような Multi Layer Perceptron(MLP) を用いた階層的分類手法 HHAR-net を提案している．階層的分類は分類問題を人が事前に定義したクラスの階層構造に従って

[1] 福井大学工学研究科知識社会基礎工学専攻
Graduate School of Engineering, Unversity of Fukui

© 2021 Information Processing Society of Japan



(a) HHAR-net[5] **(b)** B-CNN[6] **(c)** クラス階層適応型 B-CNN モデル

図 1: 関連手法 (a)，(b) と提案手法 (c) のモデル概要図

シンプルな分類問題に分解し，分解された問題を複数のモデルを用いて段階的に解く手法である．この手法では個々のモデルはクラスの階層構造によって単純化された分類問題のみを対象とするため，すべてのクラスをまとめて分類するモデルよりも複雑さが軽減され，分類性能の向上につなげることができる．しかし，このような階層的分類手法では，分類クラス数が多い場合や複雑な階層関係を持つクラスを扱う場合は，モデル数が増加する欠点がある．

一方で，画像認識分野では Zhu ら [6] が CNN モデルの構造にクラスの階層関係を反映させた Branch Convolutional Neural Network(B-CNN) を提案している．B-CNN は図 1(b) のように CNN の階層構造とクラスの階層構造を対応付けることでクラス間の階層関係を学習するように設計されたモデルである．クラスの階層構造では最下層のクラスを詳細クラスとすると，それ以外の階層は詳細クラスから得られる抽象的なクラスを表している．B-CNN モデルではモデルの中間で分岐させたパスに分類器を接続し，抽象クラス (上例における「静止」と「非静止」) を推定させる．B-CNN では抽象クラスと詳細クラス (上例における「静止」,「歩行」,「階段を上る」,「階段を下る」) の最適化を同時に行うことで，モデルにクラスの階層関係を学習させ性能向上を図っている．B-CNN は CNN モデルに対してクラス階層を反映させるシンプルな手法であり，行動分類モデルに対しても容易に適用することが可能である．B-CNN モデルは HHAR-net と異なり，単一のモデルでクラスの階層性を利用することができる手法であるが，訓練に用いるクラスの階層関係は HHAR-net と同様に事前知識として人が与える必要がある．クラス数が少ない場合，クラスの階層構造を設計することは比較的容易だが，クラス数が多い場合，クラス間の関係が複雑になるため階層構造の設計は容易ではない．また，クラス階層の設計には明確な基準がなく，人が事前知識に基づいて設計されたクラス階層が必ずしも有効に働くとは限らない．

そこで本研究では，B-CNN で用いるクラス階層をデータから適応的に決定するクラス階層適応型 B-CNN モデルを新たに提案する．図 1(c) は提案手法を表した図である．図 1(c) の左側は B-CNN モデルを表しており，提案手法は Zhu らが提案する B-CNN に対してクラス階層の自動構成手法を追加した手法である．提案手法では訓練データからクラスの階層構造を自動構成し，構成したクラス階層を用いて B-CNN を訓練することでクラス間の関係性を考慮した行動分類を行う．本稿では，B-CNN モデルが行動認識においても有効に働くことを示した上で，B-CNN の課題であったクラスの階層構造を人が設計しなければならない点を改良する．

以上より本研究の貢献は以下の 4 点である．

- センサによる行動認識において B-CNN モデルの有効性を示した．また，訓練に用いる被験者数を変化させたときのモデルの認識性能への影響を検証することで，B-CNN モデルは訓練データが比較的少ない場合に特に有効であることも明らかとなった．
- B-CNN に与えるクラス階層と認識性能の関係を検証することで，不適切なクラス階層がモデルの認識性能の低下を引き起こすことが分かり，クラス階層が B-CNN の性能を左右する重要なパラメータであることが明らかとなった．
- また，上記の検証では人が設計したクラス階層が必ずしも最適であるとは限らないことも判明し，クラス階層を利用する上で，適切なクラス階層を設計する手法が重要であることを示した．
- B-CNN に対して有効に働くクラス階層を構成するために，クラス間の特徴空間上の距離に基づいたクラス階層自動構成手法を新たに提案し，その有効性を示した．

© 2021 Information Processing Society of Japan

2. 関連研究

2.1 深層学習を用いたセンサによる行動認識

深層学習を用いたセンサによる行動認識では，センサで観測されたデータから特徴抽出と行動分類をまとめて実施する．深層学習を用いない従来手法では観測データから人が手動で設計した特徴量を用いて分類を行っていたが，深層学習を用いた手法ではこの特徴量の設計も自動的に行う．深層学習を用いた行動認識手法では CNN をベースとした手法が多数提案されている [7], [8], [9], [10], [11], [12]．多くの手法は数層の畳み込み層と Pooing 層を階層的に接続したシンプルなモデル構造を用いている [7], [8], [9]．この他には CNN で抽出された特徴量を Recurrent Neural Network(RNN) に入力して行動を分類するモデル [10], [11] や，その逆で RNN を用いて波形データの時間依存性をエンコードした後に CNN によって空間方向の特徴抽出を行うモデル [12] も提案されている．また，最近では自然言語処理分野や画像認識分野で注目されている Attention 機構を用いた手法も提案されている [13], [14]．

2.2 行動認識におけるクラス階層の利用

行動認識分野では，分類問題をクラスの階層構造に従ってよりシンプルな分類問題に分解し，分解された分類問題を複数のモデルを用いて段階的に解く手法が多数提案されている [5], [15], [16], [17], [18]．例えば，「立つ」,「座る」,「歩行」,「階段を上る」,「階段を下る」という 5 種類のクラスを考えると，これらは「静止」と「非静止」の 2 つのクラスに大きく分けることができ，「立つ，座る」は「静止」,「歩行，階段を上る，階段を下る」は「非静止」に区別することができる．例えば，上例の 5 種類のクラスの分類を考えると，階層的分類ではまず一つのモデルで「静止」と「非静止」を分類する．このとき推定クラスが「非静止」であった場合は別のモデルを用いて「非静止」クラスに含まれる「歩行」,「階段を上る」,「階段を下る」の 3 つのクラスで分類を行う．最初の推定結果が「静止」であった場合も同様の分類を行う．このような階層的分類手法では複数のモデルを用いて分類を行う上に，分類クラス数が増えるにつれて推論に用いるモデル数も増加することから分類モデルには深層学習でない機械学習手法を用いることが多い．深層学習モデルを用いた行動認識手法では，Fazli ら [5] や Cho ら [19] らによって深層学習ベースの階層的分手法が提案されている．

2.3 画像認識におけるクラス階層の利用

画像認識分野では Zhu ら [6] が CNN モデルの構造にクラスの階層構造を反映させた B-CNN を提案している．彼らは CNN モデルを途中で分岐させ，分岐させたパスでクラス階層に基づく抽象クラスを分類させることでモデルにクラスの階層関係を学習させモデルの認識性能を向上させている．クラスの階層構造を CNN モデルに反映する手法は B-CNN 以外にも様々な手法が提案されている [20], [21], [22], [23]．一方で，Deng ら [24] はクラス間の階層関係を HEX グラフと呼ばれる関係グラフで表現し，HEX グラフを用いてクラス間の階層的な関係自体を学習させる手法を提案している．この手法は B-CNN のようなクラスの階層構造をモデルに反映する手法と異なり，クラス間の階層関係を明確にモデル化し認識モデルに学習させている．また，Koo ら [25] は CNN を用いて画像の階層表現を抽出し，Recurrent Neural Network(RNN) を用いてクラスの階層関係を捉える手法を提案している．

2.4 クラス階層の自動構成

先に挙げた手法はクラスの階層構造を事前知識として人が与えることを想定した手法が多いが，クラス階層自体をデータから自動で構成する取り組みも行われている．クラスの階層構造を構成する手法は主に top-down 方式 [26], [27], [28], [29], [30] と bottom-up 方式 [30], [31], [32], [33], [34] に区別される．top-down 方式では，すべてのクラスをまとめた仮想的な抽象クラスを木構造のルートノードとして扱い，このルートノードをすべてのクラスが葉ノードとなるまで再帰的に分割することでクラス階層を構成する手法である．top-down 方式では K-means 法や Normalized cuts[35] などのクラスタリング手法を用いて分割を行う．一方で，bottom-up 方式では，すべてのクラスが一つの抽象クラスに統合されるまで，一定の基準で各クラスを再帰的に統合していく手法である．クラスの統合は階層クラスタリングを用いた手法 [31], [33] や混同行列を用いた手法 [30]，グラフを用いた手法 [34] が提案されている．

本稿と特に関連する取り組みとして，Jin ら [34] の取り組みが挙げられる．Jin らは bottom-up 方式のクラス階層構成手法を提案しており，彼らの手法では，CNN モデルが出力するクラス分類スコアから Confusion Graph と呼ばれるグラフを作成し，作成された Confusion Graph から Louvain 法と呼ばれるコミュニティ検出アルゴリズム [36] を用いて，クラス階層の構成を行っている．この手法は ImageNet や CIFAR-100 のように分類クラスが多いデータセットに対しては非常に有効な手法であるが，小規模データセットやクラス数が少ないデータセットではクラスの階層構造を捉えることが難しいと考えられる．

そこで本研究では CNN モデルをベースとした手法において，少ないクラス間の階層構造を捉えることができる手法を新たに提案する．提案する手法は特徴空間上のクラス間の距離に基づいて再帰的クラスを統合していくため，bottom-up 方式に分類される．

© 2021 Information Processing Society of Japan

2.5 階層的マルチラベル分類

クラス間の階層関係を扱う取り組みとしては，Hierarchical Multi-label Classification(HMC) を挙げることができる．HMC はクラス間の階層関係自体を推定することを目的としたタスクである [37], [38]．HMC では，複数クラスの推定に加えてクラス間の階層関係まで推論を行うため，一般的な分類タスクより複雑で困難なタスクであることが知られている．HMC はクラス階層の自動構成手法と概念が類似しているが，両者はその目的が異なる．クラス階層の自動構成手法では，クラス階層を別の課題に用いることを想定して階層構造の構成を行うため，階層構造の質は主要な評価対象ではないのに対し，HMC ではクラスの階層構造を推定することを目的とし，推定された階層構造の質が手法の主な評価対象である．

本研究はクラスの階層構造を利用することで特定の行動分類問題の性能を向上させることを目的としているため，HMC とは立ち位置が異なる．

2.6 本研究の位置づけ

以上を踏まえ本研究の位置づけは以下の通りである．

- Fazli ら [5] が提案する HHAR-net とは異なり，深層学習を用いた行動認識において単一のモデルでクラスの階層構造を補助的に利用した分類を行う．
- Jin ら [34] と同様に，特定の分類問題の性能向上のために，深層学習モデルを用いて既知の分類クラスからクラスの階層構造を自動構成する手法を新たに提案する．ただし，提案手法は Jin らとは異なり，ImageNet や CIFAR-100 と比較してクラス数やデータ数が小さいデータセットに対して粒度の高いクラス階層を設計することを目的としている．

3. クラス階層適応型 B-CNN モデル

本章では提案手法の主要な構成要素である B-CNN と本研究で提案するクラス階層の自動構成手法についてより詳細な説明を行う．

提案手法では訓練データからクラスの階層構造を自動構成し，構成したクラス階層を用いて B-CNN を訓練することでクラス間の関係性を考慮した行動分類を行う．図 1(c) は提案手法を表している．提案手法は 2 ステップで構成され，Algorithm1 の流れで実行される．提案手法では，まず訓練を行った標準的な CNN モデルにデータを入力して特徴ベクトルを得る．次に，得られた特徴ベクトルから特徴空間上のクラス間の重心ベクトルを算出し，算出された重心ベクトルから階層クラスタリングを用いて類似クラスを統合することでクラス階層を構成する．そして，構成されたクラス階層を用いて B-CNN の訓練を行う．クラス階層の作成と B-CNN モデルの訓練では 2 つの異なるモデル M_{std} と M_{branch} を使用しており，M_{branch} の訓練では訓練済み M_{std} のパラメータを使用せずに初期状態から訓練を行っている．

本章は 3.1 節で Zhu ら [6] が提案している B-CNN の説明を行い，3.2 節では本稿で提案するクラス階層の自動構成手法を示す．

3.1 B-CNN

B-CNN は一般的な CNN モデルに対して，図 1(b) ようにクラスの階層構造に基づいてモデルを複数のパスに分岐させ，クラス階層における上位の階層から順に分類を行う．B-CNN は一般的な CNN モデルと同様に，Softmax 関数によって計算されたクラススコアを用いて分類を行い，各階層の分類は独立して行われる．

B-CNN の訓練は確率的勾配降下法を用いて行われる．損失関数は各階層の Softmax Cross Entropy Loss の重み付き和であり，i 番目のサンプルに対するモデル全体の損失関数 L_i は

$$L_i = -\sum_{k=1}^{K} w_k \sum_{c=1}^{C_k} t_{i,c}^k \log \frac{\mathrm{e}^{y_{i,c}^k}}{\sum_{j=1}^{C_k} \mathrm{e}^{y_{i,j}^k}} \qquad (1)$$

と表される．ただし，K は階層数，w_k は第 k 階層目の損失の重み，C_k は第 k 階層の分類クラス数，$y_{*,c}^k$ は第 k 階層のクラス c の分類スコアを表す．t^k は第 k 階層の正解クラスを表しており，

$$t_{*,c}^k = \begin{cases} 1 & (c \text{ is ground truth}) \\ 0 & (\text{otherwise}) \end{cases} \qquad (2)$$

である．

B-CNN モデルにおける分岐位置は数層の畳み込み層とプーリング層からなる畳み込みブロックを分岐位置の単位として用いる．一般的な CNN モデルは複数の畳み込みブロックを階層的に接続した構造をもつため，分岐位置に様々なパターンが考えられる．本稿では分岐位置のパターンをハイパーパラメータとして扱いチューニングを行った．

Algorithm 1 Class Hierarchy Adaptive B-CNN

Input: Train Dataset for B-CNN $D_{train} = \{(x_i, y_i)\}_{i=0}^{N-1}$; Split rate r_{pre}; Dimension for PCA d; The number of corase levels in class hierarchy L;
Output: Trained B-CNN Model M_{branch}

```
// (1) Construct Class Hierarchy(see Algorithm 2)
P ← ConstructClassHierarchy(D_train, r_pre, d, L)
P^(0), P^(1), ··· , P^(L−1) ← P
// (2) Train B-CNN Model
Initialize B-CNN M_branch
let be p_i^(k) ∈ P^(k)(i = 0, ··· , N − 1, k = 0, 1, ··· , L − 1).
D_train^branch ← {(x_i, p_i^(0), p_i^(1), ··· , p_i^(L−1), y_i)}_{i=0}^{N−1}
Train M_branch using D_train^branch
return M_branch
```

© 2021 Information Processing Society of Japan

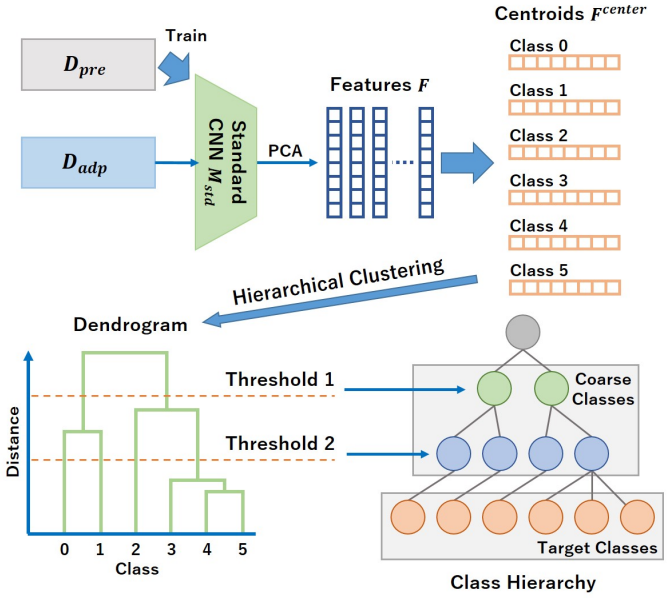


図 2: クラス階層構成手法の全体図

3.2 クラス階層の自動構成

図 2 は今回提案するクラス階層の自動構成手法を示している．クラス階層の作成手順を Algorithm2 で示す．

クラス階層の作成では，まず Branch 構造を持たない標準的な CNN モデル M_{std} を訓練し，特徴ベクトル F を得る．M_{std} の訓練と特徴ベクトル F の取得に用いられるデータ D_{pre}, D_{adp} は B-CNN モデル M_{branch} の訓練に用いるデータ D_{train} を分割することで作成する．r_{pre} は D_{train} に対する D_{pre} の割合をしており，本稿では $r_{pre} = 0.5$ としている．M_{std} に D_{adp} を入力することで得られた特徴ベクトルは主成分分析 (PCA) を用いて，次元数を d まで削減する．本稿ではこの次元数を $d = 64$ と設定している．次に，次元削減された特徴ベクトル F から特徴空間における各クラスの重心ベクトル F^{center} を計算する．そして，F^{center} を各クラスの代表ベクトルとし，F^{center} に対して階層クラスタリングを適用しデンドログラム H を得る．デンドログラムの作成には様々な手法が考えられるが，本稿では Ward 法を用いている．

最後に，クラス階層を作成するために，統合されたクラスタ間の距離に基づいて閾値を設定する．階層クラスタリングでは一定の基準で 2 つのデータ (あるいはクラスタ) を再帰的に統合していき，最終的にすべてのデータは一つのクラスタに統合される．図 2 ではまず class 4 と class 5 が一つのクラスタとして統合され，続いて class 3 と cluster $\{4, 5\}$ が統合される．デンドログラムからは統合されたクラスタ間の距離を得ることができる．今回提案する手法では統合されたクラスタ間の距離を統合された順に $d_0, d_1, d_2, \cdots, d_k$ とした上で，$s_i = d_i - d_{i-1}$ $(i = 1, 2, \cdots, k)$ を算出する．算出された $S = \{s_i\}_{i=0}^{k-1}$ から上位 L 要素を選択し，選択された各要素に対し，同じインデックスのクラスタ間距離 d_j から $d_j + \epsilon$ を計算しこれを閾値として設定する．そして，この閾値を用いてクラスタリングを行い，クラス階層に基づく各クラスの Coarse Label を得る．

Algorithm 2 Construct Class Hierarchy

Input: Train Dataset for B-CNN D_{train}; Split rate r_{pre}; Dimension for PCA d; Number of target classes C; The number of corase levels in class hierarchy L;
Output: Hierarchical Multi-labels based on constructed class hierarchy, P

// (1) Pre-training phase
Initialize Standard CNN M_{std}
$D_{pre}, D_{adp} \leftarrow \mathrm{SplitDataset}(D_{train}, r_{pre})$
Train M_{std} using D_{pre}
$F \leftarrow \mathrm{ExtractFeature}(M_{std}, D_{adp})$
$F \leftarrow \mathrm{DecompositionWithPCA}(F, d)$

// (2) Calculate centroids for each target classes
for $c = 0, \cdots, C-1$ **do**
    $N_c \leftarrow$ The number of samples in class c
    $F_c \leftarrow \{f_c \in F; f_c \text{ belonging to class } c\}$
    $f_c^{center} \leftarrow \frac{1}{N_c} \sum_{f_c \in F_c} f_c$
end for
$F^{center} \leftarrow \{f_c^{center}\}_{c=0}^{C-1}$

// (3) Construct class hierarchy
$H \leftarrow \mathrm{CreateDendrogram}(F^{center})$
let $d0, d1, \cdots, dk$ be the distances between the clusters in the order of integration in the dendrogram H.
k denotes the number of clusters in dendrogram H.
for each $i = 0, 1, \cdots, k-1$ **do**
    $s_i = d_{i+1} - d_i$
end for
$S \leftarrow \{s_i\}_{i=0}^{k-1}$
for each $i = 0, 1, \cdots, L-1$ **do**
    $j \leftarrow \underset{i}{\mathrm{ArgMax}}(S)$
    $t_i \leftarrow d_j + \epsilon$ $(\epsilon > 0)$
    Pop j from S
end for
$T \leftarrow \{t_i\}_{i=0}^{L-1}$
let Y be target class labels included in D_{train}.
for each $i = 0, \cdots, L-1$ **do**
    // m_i denotes a mapping from target label to corse label.
    $m_i \leftarrow \mathrm{HierarchicalClustering}(H, t_i)$
    // $P^{(i)}$ denotes abstract class labels of target classes.
    $P^{(i)} \leftarrow \mathrm{Relabel}(Y, m_i)$
end for
$P \leftarrow \{P^{(i)}\}_{i=0}^{L-1}$
return P

4. 実験設定

提案手法の有効性を検証するために，3 種類のベンチマークデータセットを用いて，推定精度評価実験を行った．本章では実験設定の詳細を述べる．

© 2021 Information Processing Society of Japan

4.1 データセット

検証実験で使用するデータセットは HASC[39], WISDM[40], UniMib SHAR[41] の 3 種類である.

HASC[39] はスマートフォンなどのウェアラブルデバイスを用いて人の基本行動を計測したベンチマークデータセットである. HASC には静止 (stay), 歩行 (walk), 走行 (jog), スキップ (skip), 階段上り (stup), 階段下り (stdown) の 6 種類の行動を観測したセンサデータが記録されている. センサの種類には加速度センサ, ジャイロセンサ, 磁気センサなどが含まれるが, 本稿では 3 軸の加速度センサデータのみを使用した. 検証では, BasicActivity データの内, サンプリング周波数が 100 Hz のデータを使用し, 所持位置や端末の種類は特に制限していない. 使用するデータは計測ファイルごとに前後 5 s をトリミングした上で, ウィンドウサイズを 256 サンプル, ストライド幅を 256 サンプルとして, sliding-window 方式でモデルの入力データを作成した. また, 検証で用いるデータは, HASC に収録されている全被験者のうち, 十分な計測データのある 176 名の加速度センサデータを使用している.

WISDM[40] は HASC と同様にスマートフォンを用いて人の日常行動を計測したベンチマークデータセットである. WISDM には, 立つ (stand), 座る (sit), 歩行 (walk), 走行 (jog), 階段上り (stup), 階段下り (stdown) の 6 種類の行動を観測した 3 軸加速度センサデータが記録されている. WISDM は HASC と類似した行動が記録されているが, HASC よりも被験者やデータ数が少ないデータセットとなっている. 検証では, 加速度センサデータを被験者と行動の種類から複数のセグメントに分割し, 各セグメントの前後 3 s をトリミングした上で, sliding-window 方式でモデルの入力データを作成した. ウィンドウサイズ及びストライド幅は 256 サンプルと設定した.

UniMib SHAR[41] はスマートフォンの加速度センサを用いて人の日常行動と転倒シーンを計測したデータセットである. 計測された行動と転倒シーンは, 座っている状態から立ち上がる (standFS), 寝ている状態から立ち上がる (standFL), 歩行 (walk), 走行 (jog), ジャンプ (jump), 階段を上る (stup), 階段を下りる (stdown), 立っている状態から横になる (layFS), 座る (sit), 前方に倒れる (fallF), 後方に倒れる (fallB), 右方向に倒れる (fallR), 左側に倒れる (fallL), 落下時に障害物にぶつかる (hitO), 保護対策をして落下する (fallPS), 後ろに倒れ椅子に座る (fallBSC), 失神 (syncope) の計 17 種類である. UniMib は 3 軸の加速度センサデータが記録されており, 151 サンプルのフレームに分割された状態で提供されている. 検証はこの分割されたデータを用いて行った.

4.2 モデル構造

提案手法では標準的な CNN モデル (以降このモデルを std モデルと呼称する) と B-CNN モデルの 2 種類のモデルを用いている. 各モデルの中心となる Backbone は様々なモデルを採用可能であるが, 本稿ではいずれのモデルもベースモデルとして VGG モデル [42] を採用した. VGG モデルは Hasegawa ら [43] によってセンサベースの行動認識における有効性が確認されたモデルあり, VGG のシンプルなモデル構造は B-CNN のベースモデルとして適切であると判断した. 図 3 は検証で用いた B-CNN モデルの構造を示している. Conv Block は数層の畳み込み層と Max Pooling 層から構成され, Classifers はそれぞれ Global Average Pooling 層と 1 層の全結合層から構成される. Simonyan ら [42] が提案する VGG モデルでは畳み込み層から得られた特徴マップをサイズが 7×7 となるように Average Pooling を行い, Pooling された特徴量を数層の全結合層と Dropout 層からなる分類器に入力して分類を行うが, 本稿では B-CNN による特徴空間への作用とそれに伴う認識性能の変化を検証することを目的としているため, 分類器の規模を小さくし, モデルの性能における分類器の影響を小さくしている. 加えて, 標準的な VGG モデルは画像を入力として扱うために 2 次元の畳み込み層を用いているが, 本稿では 1 次元のセンサデータを入力として扱うため, すべての畳み込み層を 1 次元のものに変更している. また, B-CNN の分岐位置はハイパーパラメータとして各データセットでそれぞれチューニングを行った. 図 3 の Conv Block を左から 1, 2, 3, 4, 5 とすると, HASC では 1, 3, WISDM では 1, 2, UniMib では 3, 4 の Conv Block の出力位置を分岐位置として設定した.

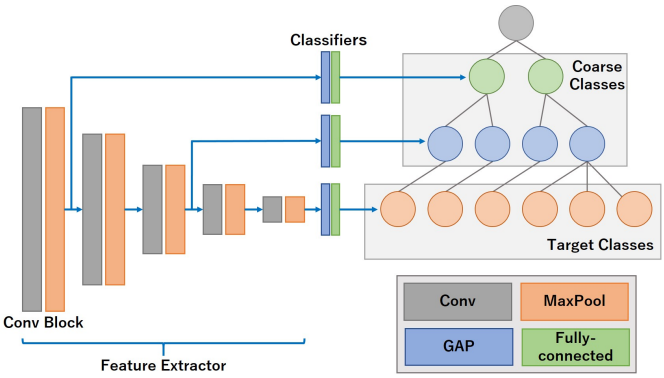


図 3: VGG16 をベースモデルとした B-CNN のモデル構造

4.3 モデルの訓練

モデルの訓練はすべて Adam を用いて行い, 初期学習率は 0.001 に統一した. エポック数は 300 を基本としているが, クラス階層の自動構成で利用する std モデル M_{std} の訓練では 200 エポックと設定した. モデルの訓練の際は RICAP[44], センサデータの軸の入れ替え, センサデータの軸の反転の 3 種類のデータ拡張を併用している. RICAP は画像認識分野で提案された手法であるが, センサデータのよ

© 2021 Information Processing Society of Japan

表 1: Hold-out 法における各データセットの被験者数の内訳

Dataset	Train	Validation	Test
HASC	10	50	50
WISDM	12	12	12
UniMib SHAR	10	10	10

うな 1 次元データに対しても適応可能であり，Hasegawa[45] によってセンサデータの軸の入れ替え (Rotation) と共にその有効性が確認されている．RICAP にはデータの切り出し位置に関連するハイパーパラメータ β が存在し，本稿では $\beta = 0.5$ と設定した．センサデータの軸の反転はセンサデータの軸の入れ替えと同様に端末位置や向きに関して入力データの分布を広げることができ，HASC のように端末の所持位置や向きが固定されていないデータセットで有効であると考えられる．

また，B-CNN の訓練では各階層の損失に対して重み w_k を設定する必要があり，今回は各階層の重みの総和が 1 であるという制約を与えたうえで，各階層の重みを均等に設定した．本研究ではクラス階層の違いによる影響に焦点をあてており，重み w_k による影響を最小限にするためにこのように設定している．

4.4 評価方法

モデルは Hold-out 法を用いて評価を行い，データセットは被験者単位で分割した．訓練 (Train)，検証 (Validation)，テスト (Test) に含まれる被験者数の内訳は表 1 の通りである．検証実験では，データセットの分割，モデルの訓練，モデルの評価を 1 セットとして 20 回の試行を行い，全試行の結果の平均で評価を行う．評価に用いる指標を Accuracy と平均 F-Score である．

また，今回提案するクラス階層の自動構成手法では，M_{std} の訓練データ D_{pre} とクラス階層を作成する際に用いるデータ D_{adp} の 2 種類のデータセットを用いている．本稿では提案手法と他のベースライン手法と条件を統一するために提案手法の検証では，Hold-out 法にて分割された訓練用データ D_{train} を等分することで D_{pre} と D_{adp} を作成し，これらを用いて作成されたクラス階層と D_{train} を用いて B-CNN モデル M_{branch} の訓練を行っている．

5. 実験結果

表 2 は提案手法とベースライン手法の Accuracy 及び平均 F-Score を示している．表中の stdvgg16 は VGG 構造を持つ標準的な CNN モデルを表し，branchvgg16 は VGG 構造に B-CNN の分岐を追加した CNN モデルである．本章ではまず B-CNN モデルの有効性に関する検証結果を示し，得られた結果に対して考察を行う．その後，クラス階層の違いによる B-CNN モデルの認識性能への影響とクラス階層の探索コストに関して考察を行う．そして上記 2 点を踏まえ，提案手法の有効性評価と提案手法によって作成されたクラス階層の分析を行う．

5.1 B-CNN の有効性に関する考察

5.1.1 B-CNN の有効性

表 2 では各データセットに対して，std モデルと B-CNN モデルの評価結果を示している．B-CNN では設定するクラス階層の作成方法に関して，hand-crafted，Jin et al.[34]，今回の提案手法 (ours) の 3 つの手法を比較している．hand-crafted では人が手動で設計したクラス階層を用いて B-CNN の訓練を行う．手動で設計したクラス階層については 5.3 節で詳細に述べる．

この結果から，手動設計したクラスタ階層を用いた場合，すべてのデータセットにおいて B-CNN モデルは std モデルの F-Score を上回っている．UniMib では Accuracy で std モデルに劣っているがその差は 0.002 と小さい．従って，B-CNN モデルはセンサによる行動認識においても有効であることが確認できる．また，図 4 では，HASC データセットにおいて訓練に用いる被験者数を 10 人から 50 人まで 10 人ずつ増やした際の Accuracy の変化を示している．手動設計されたクラス階層を用いた B-CNN の結果に注目すると，被験者数を増加させた場合においても B-CNN モデルの Accuracy は std モデルを上回っている．ただし，訓練に用いる被験者数が増加するにつれて B-CNN モデルと std モデルの Accuracy の差は小さくなっており，B-CNN モデルは訓練データが比較的少ない場合に特に有効な手法であることが明らかとなった．

表 2: 3 種類の行動認識データセットに対する推定精度の比較結果

Model * Class Hierarchy	Branch	HASC		WISDM		UniMib	
		Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
stdvgg16 (not use class hierarchy)	-	0.803	0.806	0.866	0.799	0.725	0.607
branchvgg16 * hand-crafted	✓	**0.819**	**0.823**	**0.887**	**0.830**	0.723	**0.620**
branchvgg16 * Jin et al.[34]	✓	0.810	0.814	0.870	0.801	0.719	0.607
branchvgg16 * ours	✓	0.814	0.817	0.881	0.827	**0.728**	0.614

© 2021 Information Processing Society of Japan

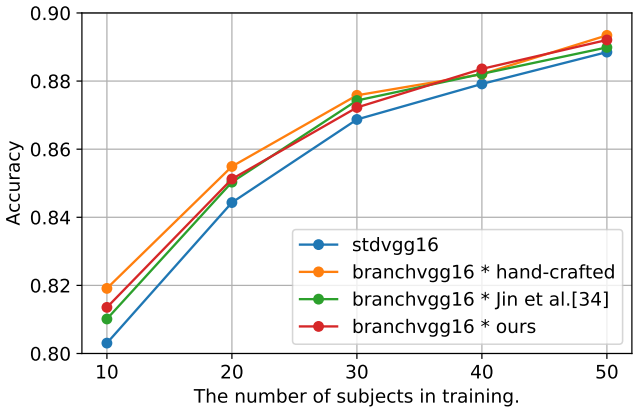


図 4: 訓練に用いる被験者数を変化させた際の Accuracy への影響

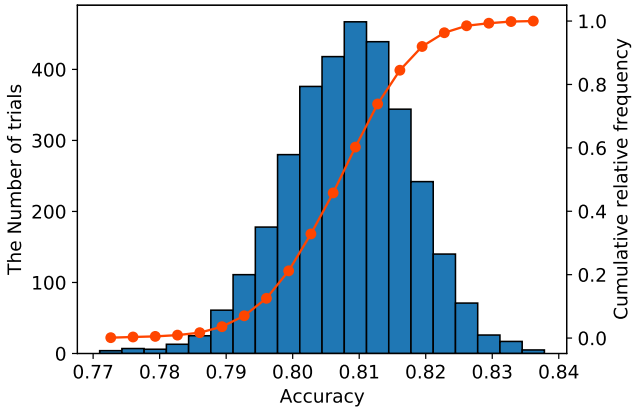


図 5: HASC データセットにおいて作成可能な全パターンのクラス階層を試行した際の Accuracy の分布

5.1.2 クラス階層の違いによる B-CNN の認識性能への影響に関する考察

次に，B-CNN に与えるクラス階層とモデルの認識性能の関係を検証する．検証では HASC データセットを用いて作成可能なすべてのクラス階層作成パターンを試行する．クラス階層の作成パターン数は 3230 通りと非常に多いため，各クラス階層に対して 1 回ずつモデルの訓練と評価を行い，得られた評価結果をクラス階層のスコアとして扱う．

図 5 は全試行の Accuracy をヒストグラムで集計したものである．青の縦棒は各階級の度数を表し，赤線は累積相対度数を表している．この結果から，全試行の中で Accuracy は 0.808 の付近に集中しており，さらに Accuracy が 0.808 以上の試行は全体の 50%を占めていることが分かる．std モデルの Accuracy は 0.805 であるため，ランダムにクラス階層を作成したとしても std モデルより高い精度を達成する可能性が高い．また，全試行で最小の Accuracy は 0.771，最大の Accuracy は 0.838 となっており，クラス階層の設計方法によって B-CNN の認識性能が大きく変わることも明らかとなった．

また，表 3 は全試行で精度が最大と最小のクラス階層をそれぞれ示している．最高精度となったクラス階層では，Level 2 で stay と skip が統合されており，Level 1 では stay，skip，walk，stup が統合されている．図 6 は上位 1%のクラス階層の Level 2 の階層において異なる 2 つのクラスが統合される割合を表している．この結果から，上位 1%の Level 2 の階層では，walk，stup，stdown の 3 クラスで互いに統合される割合が最も大きいことが分かる．さらに，skip と統合されているクラスに着目すると，最高精度となったクラス階層で見られた stay と skip が統合される割合が比較的大きいことも分かる．また，Level 1 では全体的に 2 つのクラスが統合される割合が増加しているが，おおむね Level 2 と同様の結果が見られた．手動でクラス階層を設計する際は stay と skip は互いの類似性が最も小さいクラスと考え，異なるクラスタに割り当て，stay と walk, stup も静止と非静止として異なるクラスタに割り当てる．このことから人が設計したクラス階層が必ずしも B-CNN において最適なクラス階層とは限らないことが分かる．

一方で，最低精度となったクラス階層では，Level 2 で stay と jog，walk と stdown，skip と stup が統合され，Level 1 では stay, jog, walk, stdown が一つのクラスタとして統合されている．図 7 では下位 1%のクラス階層の Level 2 の階層において異なる 2 つのクラスが統合される割合を表している．この結果に注目すると，下位 1%の Level 2 の階層では stay と jog，stay と stup，skip と stup，skip と stdown が統合される割合が比較的大きいことが分かり，特に stay と stup は表 3 の最下位のクラス階層にも現れるパターンである．この傾向は Level 1 の階層においても同様である．walk，stup，stdown は互いに誤分類が多いクラスであるため，最下位のクラス階層では，stay と stup の統合が Target Classes における walk，stup，stdown の分類に悪影響を与え，結果として分類精度の低下につながったと考えられる．これに対し，上位 1%のクラス階層で多く見られた stay と skip が統合されるパターンでは stay と skip は全く異なる行動クラスであるため，B-CNN の分岐された分類器で stay と skip を同じクラスとして最適化したとしても Target Classes の最適化によって分類に足りうる特徴表現を獲得できたと考えられる．

5.1.3 クラス階層の探索コスト

5.1.1 項では B-CNN を用いることで行動認識性能の向上を確認できたが，5.1.2 項では不適切なクラス階層が B-CNN モデルの認識性能の低下につながることが明らかになった．よって，クラス階層の設計は B-CNN の性能を左右する重要なパラメータであることが分かる．クラス階層の設計方法としては，まず人が事前知識をもとに手動で設計する方法が挙げられる．元のクラス数が少ない場合は手動で

© 2021 Information Processing Society of Japan

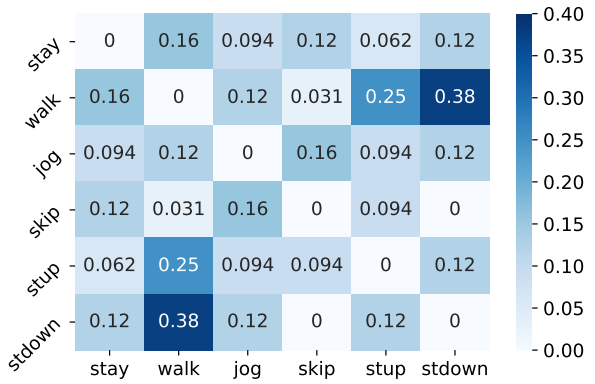


図 6: 上位 1%のクラス階層の Level 2 において 2 つのクラスが統合される割合



図 7: 下位 1%のクラス階層の Level 2 において 2 つのクラスが統合される割合

の設計は容易であるが，クラス数が多くなるとクラス間の関係性が複雑になる上に組み合わせ数も増加するため手動での設計は容易ではない．ここで，図 2 のような高さ 3 のクラス階層の作成パターン数を図 8 に示す．元のクラス数が 4 の場合，クラス階層の作成パターンは 18 通りと少なく，全件探索も実行可能である．しかし，元のクラス数が 9 の場合はクラス階層の作成パターンは 7226538 通りとなり，全件探索は現実的ではない．

クラス数が 9 の場合，対象クラスによっては手動での設計も可能であるが，17 種の行動クラスを持つ UniMib のようにクラス数がさらに多くなると，手動での設計も困難になってくる．例えば UniMib の場合，17 種類の行動クラスは日常行動と転倒シーンで，意味的に 2 クラスに大別することが可能だが，日常行動に含まれる「立っている状態から横になる (layFS)」という動作と転倒シーンに含まれる「後方へに倒れる (fallBSC)」というシーンは転倒か否かという差はあれど体の動きは類似している可能性がある．このようにクラス階層を手動で設計する際は各行動の類似性などを考慮して分析を行う必要があるため，クラス数が多い場合はクラス階層の手動設計は非常に困難な作業となる．

以上の議論から，クラス階層をデータから自動で設計する手法は特にクラス数が多い場合に有用である．本稿では B-CNN の訓練のためにクラス階層の自動設計を行っているが，大規模なデータセットに対してデータの類似性の観点から分析を行う際にもクラス階層の自動設計は有用な手段になり得る．

表 3: 全試行で精度が最大と最小のクラス階層

Level	HASC Best Hierarchy					
1	stay	skip	walk	stup	jog	stdown
2	stay	skip	walk	stup	jog	stdown
3	stay	skip	walk	stup	jog	stdown
	HASC Worst Hierarchy					
1	stay	jog	walk	stdown	skip	stup
2	stay	jog	walk	stdown	skip	stup
3	stay	jog	walk	stdown	skip	stup

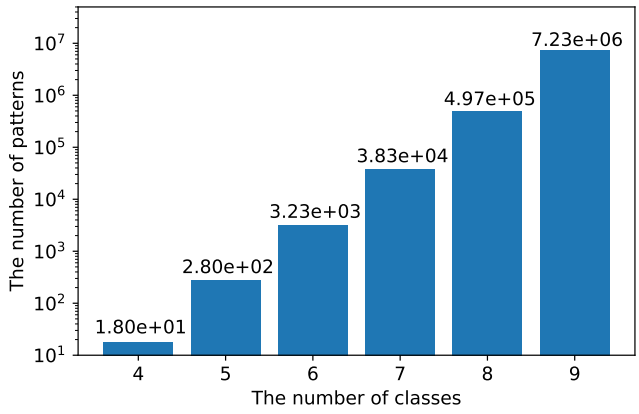


図 8: クラス階層の組み合わせ数

5.2 提案したクラス階層自動構成手法に関する考察

本節では以上の議論を踏まえたうえで提案手法の有効性を評価を行う．表 2 より，すべてのデータセットにおいて提案手法は Accuracy，F-Score のいずれの評価指標においても std モデルを上回る結果を達成した．加えて，提案手法は Jin らが提案している手法 [34] によって構成されたクラス階層を用いた B-CNN モデルよりも高いスコアとなっている．Jin らの手法は ImageNet や CIFAR-100 はクラス数やデータ数が多い大規模データセット向けに設計された手法であり，比較的小規模な行動認識ベンチマークデータセットでは，B-CNN において有効に働くクラス階層を作成できなかったと考えられる．また，提案手法と手動設計されたクラス階層を用いた B-CNN モデルを比較すると，HASC と WISDM において，提案手法は Accuracy と F-Score のいずれの指標に対しても手動設計のクラス階層を用いたモデルに匹敵する分類精度を達成した．UniMib

© 2021 Information Processing Society of Japan

において提案手法は最も高い Accuracy であるが，手動設計したクラス階層を用いた B-CNN モデルに対して F-Score で 0.006 程劣る結果となった．加えて，図 4 の結果から，訓練に用いる被験者数を増加させた場合においても提案手法は std モデルの Accuracy を上回っているが，被験者数が増えるにつれて std モデルとの Accuracy の差は小さくなっている．手動設計されたクラス階層を用いる B-CNN と比較すると，訓練に用いる被験者数に関わらず，提案手法は手動設計のクラス階層を用いた B-CNN よりやや低い精度であるが，被験者数が増加するにつれてその差は小さくなっている．従って，B-CNN モデルにおけるクラス階層の違いによる影響は訓練データが少ない場合に顕著に現れることが分かる．

これらの結果から，人が事前知識をもとに手動設計したクラス階層は B-CNN に対して有効に働くことが分かる．ただし，提案手法は手動設計のクラス階層を用いた B-CNN モデルに匹敵する性能であることから，事前知識が少なく B-CNN に対して有効に働くクラス階層の設計が困難である場合，提案手法は特に有効である．

5.3 提案手法で設計されたクラス階層に関する考察

次に提案手法で設計されたクラス階層を具体的に示し，定性的な評価を行う．表 4，5 では，各データセットに対して，手動で設計したクラス階層とある特定の試行で提案手法によって自動で設計されたクラス階層を示している．表 5 は UniMib データセットに対して設定したクラス階層を表しているが，UniMib データセットはクラス数が多いため，表を途中で折り返して表示している．表中の level 1，level 2 は図 1(c) の Coarse Classes に該当し，level 3 は Target Classes に該当する．

まず HASC における結果に注目すると，Level 2 では walk と stup のみが統合されており，行動の類似性の観点では妥当な結果である．しかし，Level 1 では stay，walk，stup，stdown が一つのクラスとして統合されてしまっている．他の試行においてもこのように stay と walk，stup，stdown が一つのクラスに統合されるケースが多く見られた．しかし，行動の類似性で考えると，stay と walk，stup，stdown は分離されるべきである．よって，5.1.2 項の議論を踏まえると，これが手動設計したクラス階層を用いた B-CNN モデルと提案手法の認識性能の差につながっていると考えられる．

また，stay と walk 系 (walk，stup，stdown) が一つのクラスとして統合されてしまった要因としては，std モデルの訓練に用いた損失関数が挙げられる．提案手法では std モデルを訓練することで各クラスの特徴空間上での分布を設計し，各クラスの特徴空間上での分布を用いてクラス階層を作成する．設計される特徴空間は std モデルの訓練で最適化される損失関数に依存しており，今回用いた Softmax Cross Entropy Loss の最適化によって設計された特徴空間上では walk 系のデータが skip のデータよりも stay のデータに近い位置に分布していたと考えられる．従って，std モデルの訓練に用いる損失関数や std モデルの訓練方法を検討することでより適切なクラス階層の作成が可能となる可能性がある．

WISDM では，walk に対するクラスタリング結果が手動設計のクラス階層と大きく異なる．手動設計のクラス階層では動きの大きさの観点から，Level 2 で walk は stup，stdown と同じクラスタに統合され，Level 1 で jog，stup，stdown と統合されていたが，提案手法で作成されたクラス階層では，Level 2 で walk はいずれのクラスとも統合されず，Level 1 で jog と統合される．jog と walk はいずれも平らな地面の上で移動する動作であり，動きの大きさに差はあるがその動作も似ている．提案手法ではこのような類似性をから walk と jog を一つのクラスタに統合し，stup と stdown のクラスタと分離させたと考えられる．

UniMib では，standFS, standFL, layFS, sit は立ち上がる動作や座る動作など移動を伴わない行動，fallF, fallPS, fallR, fallL, fallB, fallBSC, hitO, syncope は転倒時の動作として一つの行動クラスにまとめられている．walk, jog, stup と jump, stdown は判別が難しいが体の動きの大きさで区別されているのではないかと考えられる．stup と stdown はそれぞれ階段を上る動作と階段を下る動作で類似する行動であるが，stdown は stup と異なり重力方向へ移動する行動である．そのため stdown は stup よりも階段を一段上り下りする際の体の移動が速くなり，それに伴いセンサで観測される加速度も大きくなる．これは飛び上がる際の加速度が大きい jump と共通する項であり，提案手法で加速度の大きさで walk, jog, stup と jump, stdown を分離させたと考えられる．

以上の議論から，提案手法で作成されたクラス階層は人が設計したクラス階層とは異なるものの，人とは異なる観点で行動の類似性を捉えて設計されたものが多く見られた．また，表 2 の結果から提案手法で作成されたクラス階層を用いて訓練した B-CNN モデルは std モデルを上回る認識性能を達成している．ただし，5.1.2 項の議論から B-CNN モデルの認識性能を向上させるより適切なクラス階層が存在する可能性があり，提案手法にはまだ改良の余地がある．

6. まとめ

本稿では，クラス間の階層関係を補助的に利用して訓練を行う B-CNN にクラス階層の自動構成手法を追加したクラス階層適応型 B-CNN モデルを提案した．提案手法では，訓練データから自動でクラス階層を作成し，作成されたクラス階層を用いて B-CNN を訓練することでクラス間の階層関係を考慮した分類モデルを作成する．検証実験では，まず手動で設計されたクラス階層を用いた B-CNN モデル

© 2021 Information Processing Society of Japan

表 4: HASC と WISDM に対して手動で設計したクラス階層と提案手法で設計されたクラス階層

(a) HASC

Level	Hand-crafted						Proposed Method					
1	stay	walk	stUp	stDown	jog	skip	stay	walk	stUp	stDown	jog	skip
2	stay	walk	stUp	stDown	jog	skip	stay	walk	stUp	stDown	jog	skip
3	stay	walk	stUp	stDown	jog	skip	stay	walk	stUp	stDown	jog	skip

(b) WISDM

Level	Hand-crafted						Proposed Method					
1	jog	walk	stUp	stDown	sit	stand	jog	walk	stUp	stDown	sit	stand
2	jog	walk	stUp	stDown	sit	stand	jog	walk	stUp	stDown	sit	stand
3	jog	walk	stUp	stDown	sit	stand	jog	walk	stUp	stDown	sit	stand

表 5: UniMib SHAR に対して手動で設計したクラス階層と提案手法で設計されたクラス階層

(a) Hand-crafted

Level	Hand-crafted									
1	jog	walk	stup	stdown	jump	standFS	standFL			
2	jog	walk	stup	stdown	jump	standFS	standFL			
3	jog	walk	stup	stdown	jump	standFS	standFL			
1	layFS	sit	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope
2	layFS	sit	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope
3	layFS	sit	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope

(b) Proposed Method

Level	Proposed Method								
1	jog	walk	stup	stdown	jump	standFS	standFL	layFS	sit
2	jog	walk	stup	stdown	jump	standFS	standFL	layFS	sit
3	jog	walk	stup	stdown	jump	standFS	standFL	layFS	sit
1	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope	
2	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope	
3	fallF	fallPS	fallR	fallL	fallB	fallBSC	hitO	syncope	

と標準的な CNN ベースの分類モデルを比較し，センサによる行動認識においても B-CNN モデルが有効であることを示した．加えて，訓練に用いる被験者数を変化させた際のモデルの認識性能への影響を検証することで，B-CNN モデルは訓練データが比較的少ない場合において特に有効であることが明らかとなった．次に本稿で提案した手法の評価を行い，提案手法が標準的な CNN モデルを上回る分類精度を達成し，さらに，手動で設計したクラス階層を用いた B-CNN モデルに匹敵する認識性能を達成することを確認した．B-CNN の訓練に必要なクラス階層の探索コストはクラス数に対して指数関数的に増加するため，事前知識が少なく適切なクラス階層の作成が困難な状況下において，提案手法は特に有効である．ただし，提案手法によって作成されるクラス階層は std モデルの訓練によって設計される特徴空間に大きく依存しており，クラス間の階層関係がこの特徴空間に反映されていない場合，提案手法では適切なクラス階層の設計が困難となる．また，5.1.1 項，5.1.2 項の検証から提案手法で自動で作成されるクラス階層よりも，行動認識精度を向上させるクラス階層が存在する可能性も示唆された．従って，クラス階層の作成に利用する特徴量の設計方法や特徴量からより適切なクラス階層を作成する手法の検討は今後の課題である．

謝辞

本研究の一部は科学研究費助成事業若手研究（19K20420）の助成によるものである．ここに謝意を表す．

参考文献

[1] Oscar Lara and Miguel Labrador. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Commun. Surv. Tut.*, Vol. 15, pp. 1192–1209, Jan 2013.

[2] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recogn. Lett.*, Vol. 119, pp. 3–11, July 2017.

[3] Carlos Silla and Alex Freitas. A survey of hierarchical

© 2021 Information Processing Society of Japan

classification across different application domains. *Data. Min. Knowl. Disc.*, Vol. 22, pp. 31–72, Jan 2011.

[4] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE Trans. Vis. Comput. Graph.*, Vol. 24, No. 1, pp. 152–162, 2018.

[5] Mehrdad Fazli, Kamran Kowsari, Erfaneh Gharavi, Laura Barnes, and Afsaneh Doryab. HHAR-net: Hierarchical human activity recognition using neural networks. In *Proc. of the IHCI 2020*, pp. 48–58, 2020.

[6] Xinqi Zhu and Michael Bain. B-CNN: Branch convolutional neural network for hierarchical classification. *ArXiv*, 2017.

[7] Ming Zeng, Le T. Nguyen, Bo Yu, Ole Mengshoel, Jiang Zhu, and Pang Wu. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. In *Proc. of the MobiCASE 2014*, pp. 197–205, Nov 2014.

[8] Yuqing Chen and Yang Xue. A deep learning approach to human activity recognition based on single accelerometer. In *Proc. of the IEEE SMC 2015*, pp. 1488–1492, Oct 2015.

[9] 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 *Proc. of the ICOAI 2015*, IJCAI' 15, p. 3995–4001, 2015.

[10] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, Vol. 16, No. 1, 2016.

[11] Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. Innohar: A deep neural network for complex human activity recognition. *IEEE Access*, Vol. 7, pp. 9893–9902, 2019.

[12] Kun Xia, Jianguang Huang, and Hanyu Wang. Lstm-cnn architecture for human activity recognition. *IEEE Access*, Vol. 8, pp. 56855–56866, 2020.

[13] Wenbin Gao, Lei Zhang, Qi Teng, Hao Wu, FuHong Min, and Jun He. Danhar: Dual attention network for multimodal human activity recognition using wearable sensors. *ArXiv*, 2020.

[14] HaoJie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In *Proc. of the IJCAI 2019*, pp. 3109–3115, July 2019.

[15] Yuhuang Zheng. Human activity recognition based on the hierarchical feature selection and classification framework. *J. of Ele. Com. Eng*, Vol. 2015, No. 9, 2015.

[16] Aiguo Wang, Guilin Chen, Xi Wu, Li Liu, Ning An, and Chih-Yung Chang. Towards human activity recognition: A hierarchical feature selection framework. *Sensors*, Vol. 18, p. 3629, Oct 2018.

[17] Adil Khan, Soo-Young Lee, and Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE Trans. Inf. Technol. Biomed.*, Vol. 14, pp. 1166 – 1172, Oct 2010.

[18] Heike Leutheuser, Dominik Schuldhaus, and Bjoern M. Eskofier. Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset. *PLOS ONE*, Vol. 8, No. 10, pp. 1–11, Oct 2013.

[19] Heeryon Cho and Sang Min Yoon. Divide and conquer-based 1d cnn human activity recognition using test data sharpening. *Sensors*, Vol. 18, No. 4, p. 1055, 2018.

[20] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *Proc. of the ICCV 2015*, pp. 2740–2748, 2015.

[21] Yuntao Liu, Yong Dou, Ruochun Jin, and Peng Qiao. Visual tree convolutional neural network in image classification. In *Proc. of the ICPR 2018*, pp. 758–763, 2018.

[22] Ruigang Fu, Biao Li, Yinghui Gao, and Ping Wang. CNN with coarse-to-fine layer for hierarchical classification. *IET Computer Vision*, Vol. 12, No. 6, pp. 892–899, 2018.

[23] Yuqi Huo, Yao Lu, Yulei Niu, Zhiwu Lu, and Ji-Rong Wen. Coarse-to-fine grained classification. In *Proc. of the SIGIR 2019*, p. 1033–1036, 2019.

[24] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Proc. of the ECCV 2014*, pp. 48–64, 2014.

[25] Jaehoon Koo, Diego Klabjan, and Jean Utke. Combined convolutional and recurrent neural networks for hierarchical classification of images. In *Proc. of the Big Data 2020)*, pp. 1354–1361, 2020.

[26] Song Liu, Haoran Yi, Liang-Tien Chia, and Deepu Rajan. Adaptive hierarchical multi-class svm classifier for texture-based image classification. In *Proc. of the ICME 2005*, pp. 4 pp.–, 2005.

[27] Yu-Chiang Wang and D. Casasent. A hierarchical classifier using new support vector machine. In *Proc. of the ICDAR 2005*, pp. 851–855 Vol. 2, 2005.

[28] Xun Yuan, Wei Lai, Tao Mei, Xian sheng Hua, Xiu qing Wu, and Shipeng Li. Automatic video genre categorization using hierarchical svm. In *Proc. of the ICIP 2006*, pp. 2905–2908, 2006.

[29] Marcin Marszalek and Cordelia Schmid. Constructing Category Hierarchies for Visual Recognition. In *Proc. of the ECCV 2008*, Vol. 5305, pp. 479–491, Oct 2008.

[30] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *Proc. of the CVPR 2008*, pp. 1–8, 2008.

[31] Liu Zhigang, Shi Wenzhong, Qin Qianqing, Li Xiaowen, and Xie Donghui. Hierarchical support vector machines. In *Proc. of the IGARSS 2005*, Vol. 1, pp. 4 pp.–, 2005.

[32] Hakan Cevikalp. New clustering algorithms for the support vector machine based hierarchical classification. *Pattern Recogn. Lett.*, Vol. 31, No. 11, p. 1285–1291, Aug 2010.

[33] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proc. of the ECCV 2018*, pp. 269–285, Sept 2018.

[34] Ruochun Jin, Yong Dou, Yueqing Wang, and Xin Niu. Confusion graph: Detecting confusion communities in large scale image classification. In *Proc. of the IJCAI 2017*, pp. 1980–1986, 2017.

[35] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Int.*, Vol. 22, No. 8, pp. 888–905, 2000.

[36] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. of Stat. Mech. Theory Exp.*, Vol. 2008, p. 10008, Apr 2008.

[37] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In *Proc. of the ICML 2018*, Vol. 80, pp. 5075–5084, 10–15 Jul

© 2021 Information Processing Society of Japan

2018.

[38] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. In *Proc. of the NeurIPS 2020*, Vol. 33, pp. 9662–9673, 2020.

[39] Haruyuki Ichino, Katsuhiko Kaji, Ken Sakurada, Kei Hiroi, and Nobuo Kawaguchi. Hasc-pac2016: Large scale human pedestrian activity corpus and its baseline recognition. In *Proc. of the UbiComp 2016*, p. 705–714, 2016.

[40] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, Vol. 12, No. 2, p. 74–82, Mar 2011.

[41] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, Vol. 7, No. 10, p. 1101, 2017.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the ICLR 2015*, pp. 1–14, 2015.

[43] Tatsuhito Hasegawa and Makoto Koshino. Representation learning by convolutional neural network for smartphone sensor based activity recognition. In *Proc. of the ICCIIS 2019*, p. 99–104, 2019.

[44] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Proc. of the ACML 2018*, Vol. 95, pp. 786–798, 14–16 Nov 2018.

[45] Tatsuhito Hasegawa. Octave mix: Data augmentation using frequency decomposition for activity recognition. *IEEE Access*, Vol. 9, pp. 53679–53686, 2021.

© 2021 Information Processing Society of Japan