Vol.2022-UBI-74 No.12
2022/6/7

工場・物流センタにおける作業順序を考慮した少量学習データでの作業行動認識手法の検討

吉村 直也[1,a)]　前川 卓也[1,b)]　原 隆浩[1]　和田 篤[2]　浪岡 保男[2]

概要：
工場や物流センタではネジ締め作業や検品作業・梱包作業など，複雑で変則的な作業を人間の作業者が担っており，この作業者の作業状況を把握するために行動認識技術の適用が検討されている．しかし，作業員ごとに担当の作業が異なるため，ある作業員から取得した学習データを他の作業員に転用できないなど，学習データの収集コストに大きな課題がある．そこで本研究では，限られたラベル付きデータで作業行動を認識するためのニューラルネットワーク Lightweight Ordered-work Segmentation Network (LOS-Net) を提案する．既存のニューラルネットワークを用いた行動認識モデルは，高い認識性能を達成するためには大量のラベル付きデータを必要とする．しかし，作業工程には行動の順序が事前に決められているなどの特徴がある．LOS-Net では作業順序のような事前知識に着目し，(1) 効率的に長期的なコンテキストを抽出するためのモジュール，(2) 連続する行動ラベルの境界を正確に推定するためのモジュール，(3) 作業順序を用いて推定結果を修正するモジュールを導入し，限られた学習データでも高い精度で作業工程を認識できる．提案手法を実際の工場や物流センタで収集した 11 人分のデータを用いて評価し，提案手法の有効性を確認した．

キーワード：行動認識, ニューラルネットワーク, 可視化技術

1. はじめに

スマートフォンやスマートウォッチなど身体装着型加速度センサを用いた行動認識は，産業ドメインにおいても注目を集めている [2], [12]．工場のライン生産ラインや物流センタでの梱包作業など，工場や物流センタにおける作業は，人間の作業者に大きく依存している．人間の作業者は，機械化が難しい複雑で変則的な作業を担っていることが多く，今後も重要な役割を果たすことが予想される．このような手作業を行動認識技術で定量化することで，作業プロセスの合理化や作業者のパフォーマンス評価などが可能になり，工場の効率化に繋がることが期待される．

図 1 に，実際の工場で作業員の左右手首にスマートウォッチを装着して収集した加速度データの例を示す．一連の作業 (period) は複数の作業工程で構成されており，その順序は作業指示書などによってあらかじめ決められている．作業員はその指示に基づいて，同じ作業を繰り返しおこなっている．工場などにおける行動認識は，この一つ一つの作業工程を認識することが求められる．ただし，作業工程によって継続時間が大きく異なるため，タイムステップ毎に行動ラベル (作業工程) を推定することが求められる．

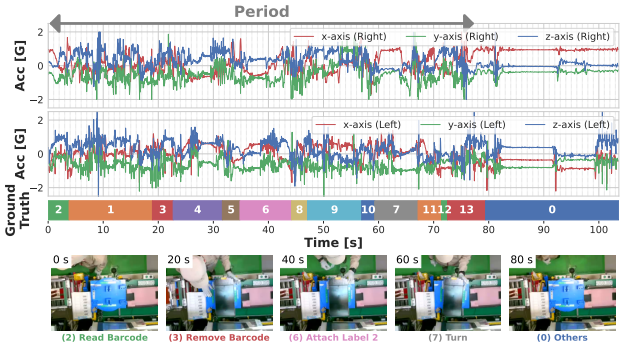


図 1: 工場における検査作業の例

工場や物流センタなどに行動認識技術を適用する際の大きな問題として，大量のラベル付きデータの収集コストがある．既存の行動認識モデルは，ニューラルネットワークなどを用いることで高い認識性能を達成しているが，一般に非常に多くの学習データを必要とする．しかし，工場では作業者毎に作業内容が異なる場合が多く，異なる作業者から取得したラベル付きデータの再利用が困難である．ま

1　大阪大学大学院情報科学研究科
2　株式会社東芝 生産技術センター
a)　yoshimura.naoya@ist.osaka-u.ac.jp
b)　maekawa@ist.osaka-u.ac.jp

た，製品の切り替えや製造プロセスの見直しによって，認識すべき作業工程は絶えず変化する．各作業者からその都度大量のデータを収集しラベリングするコストは，事業者にとって非常に大きな負担である．

図 1 に示すように，各行動の加速度データは細かい手の動きに対応する様々な波形で構成されており，複雑である．このような行動を正確に認識するためには，これらの小さな動作をすべて正確に検出することが理想的であるが，その方法では大規模なモデルと膨大な学習データが必要となる．しかし，作業工程の認識にはいくつか特徴がある．(1) 作業工程のセグメンテーションを考えると，各作業工程の最初と最後の動作を正確に検出することが重要であり，必ずしも作業工程を構成する小さな動作を全て捉える必要はない．(2) 基本的に作業工程にはあらかじめ実行の順序が決められている．注目する行動の前後の行動に関する情報大まかな情報さえ獲得できれば，現在の注目する行動を推定できる．これらの特徴を活用することで，必要とされるラベル付きデータの量を削減することができると考えられる．

本研究では，限られた学習データで作業工程の高精度な認識 (セグメンテーション) を行うため，重要な特徴を効率的に抽出することができるデコーダと，これを用いた時系列信号のセグメンテーションを行うニューラルネットワーク Lightweight Ordered-work Segmentation Network (LOS-Net) を提案する．前述の観点を踏まえ，LOS-Net のデコーダは 2 つの特徴を備える．まず，注目点の前後の行動など作業工程の長期的なコンテキストを少ないパラメータで抽出するため，Dilated Convolution を活用する．また，連続する行動の境界を正確に検出するため，時間的解像度が高い浅い層の中間出力を活用して境界の推定を行う．抽出した長期的なコンテキストと行動の境界に関する表現を統合することで，各タイムステップの行動を効率的に推定することを目指す．さらに認識精度を高めるために，作業順序に関する事前知識を活用してデコーダの出力を修正するモジュールを導入する．

本研究の研究の技術的貢献は以下の通りである．

- 産業ドメインにおける順序付き繰り返し作業のための新しい行動認識ネットワーク LOS-Net を提案する．提案モデルは作業順序などの事前知識を活用することで，限られた学習データでも学習することできる．
- 作業順序などに関する事前知識を活用するために，少ないパラメータで効率的に長期的な依存関係を抽出する WPCP モジュール，連続する行動の境界を正確に検出するための Boundary Detector，デコーダの出力を修正する Refinement モジュールを提案する．
- 実際の工場や物流センタから収集したセンサデータを用いて，提案手法の有効性を実証した．

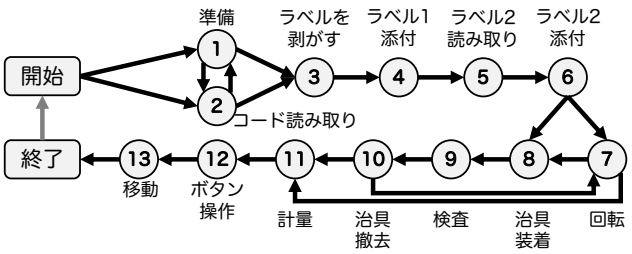


図 2: 作業順序の例 (検査作業)

2. 関連研究

近年，健康意識の高まりからスマートウォッチなどのウェアラブルデバイスの普及が進んでいる．ウェアラブルデバイスには，IMU，心拍センサー [9]，皮膚電気活動センサー [4]，マイク [3] など様々なセンサーが搭載されている．これらの研究は主に掃除や料理など日常生活行動やスポーツなどの認識に注目しているが [9]，工場や物流センタ・病院介護施設などの産業ドメインにおける行動認識も注目されている [6], [12]．これらの領域では，作業時間の計測，作業抜け検知，作業者のパフォーマンス評価など，行動認識をベースにした様々なアプリケーションが研究されている [2]．Xia ら [12] は，作業指示書の情報を活用し，教師なし学習で作業工程の認識を行なった．本研究でも，作業指示書から得られる作業順序の情報を活用し，加速度データを用いた作業行動の認識を行う．

ニューラルネットワークを用いた行動認識手法も盛んに研究されている．LSTM や GRU（gated recurrent units）などのリカレントモデルを用いる手法や，畳み込みニューラルネットワーク（CNN） [5], [7] を採用する手法もある．また，CNN と LSTM を組み合わせた手法もあり，特に DeepConvLSTM [8] は多くの行動認識研究においてベースラインとして利用されている．しかし，構造上の問題や学習データ量の観点で，考慮できる時間方向の幅には限界がある．本研究でもこれらの手法をベースラインとして用いた上で，より長期的なコンテキストを効率的に抽出するモジュールを提案する．

加速度データを用いて行動を認識する場合，多くの研究はスライディングウィンドウを適用し，ウィンドウごとに行動クラスを推定する [5]．一方，本研究が着目する産業領域における人間の作業の多くは，短時間と長時間の行動が混在しており，さらに間隔を空けずに順次行われるため．画一的なウィンドウサイズで認識することは困難である．例えば，継続時間が短い行動には小さなウィンドウサイズが効果的であるが，継続時間が長い行動では複数のウィンドウに行動が分割されてしまうため認識が難しくなる．この問題に対し Yao ら [13] は，入力時系列の各データ点に対して行動ラベルを予測するアプローチを提案した．この研

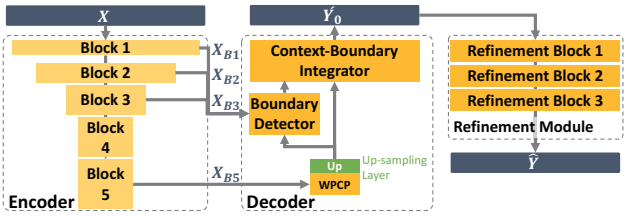


図 3: LOS-Net の構成図

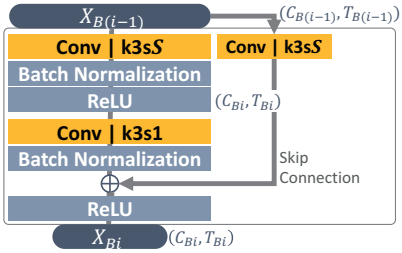


図 4: Residual Block

究に続き U-Net を用いた時系列信号のセグメンテーションモデルも提案されている [15]．本研究では，エンコーダ・デコーダに基づく軽量なセグメンテーションモデルを提案する．また，あらかじめ設定された作業順序に関する事前知識を活用して誤認識を修正する方法を提案する．

3. LOS-Net

本研究では，作業者の身体装着型 3 軸加速度センサのデータを用いた行動認識を想定する．図 1 のように，作業者は両手首にスマートウォッチを装着し，あらかじめ決められた作業順序で繰り返し作業を行う．作業順序は作業指示書を参考に決定され，図 2 のように有向グラフで表現できる．ただし，品物の状態などによって手順が変化することがある．本研究では，対象となるユーザーのラベル付きデータが少量与えられた場合に，各タイムステップの作業工程 (行動クラス) を推定することを目的とする．

LOS-Net の構成を図 3 に示す．ネットワークの入力は，長さ N_T の 6 軸加速度データ $\boldsymbol{X}$ である．また，ネットワークの出力は長さ N_T の行動ラベル $\boldsymbol{Y}$ であり，以下のように表現できる．

$$\boldsymbol{X} = [x_1, x_2, ..., x_t, ..., x_{N_T}]$$
$$\boldsymbol{Y} = [y_1, y_2, ..., y_t, ..., y_{N_T}]$$

y_t は N_C 次元のベクトルで，c 番目の要素が c 番目の行動クラスの確率を示す．N_C は行動クラス数である．本研究では $N_T = 1800$pt (60 秒) を用いた．LOS-Net は，エンコーダ，デコーダ，Refinment モジュールの 3 つのコンポーネントから構成される．これら LOS-Net の主要コンポーネント 3 つについて詳しく説明する．

3.1 エンコーダ

LOS-Net のエンコーダは，1 つの畳み込みブロックと 4 つの残差ブロック (Residual Block) から構成される．ブロック 1 は，カーネル長 3 の 64 個のカーネルをもつ 1 次元畳み込み層で構成される．ブロック 2–5 は，図 4 に示す 2 つの畳み込み層とスキップコネクションによって構成される．図中の $\boldsymbol{X}_{Bi}$ は，i 番目のブロックの出力を示す．ブロック 2–4 はストライドを 2 に設定し，抽出した特徴を時間方向に圧縮する．この時間方向の圧縮を繰り返すことで受容野が広がり，深い層ではより長期的なコンテキストを抽出することができる．

3.2 デコーダ

デコーダは 3 つのコンポーネント (WPCP モジュール，Boundary Detector，Context–Boundary Integrator) で構成される．これらについて詳細に説明する．

3.2.1 WPCP モジュール

作業工程の認識では，注目する行動の前後の行動に関する長期的なコンテキストを捉えることが重要である．WPCP (Work-Process Context Pooling) モジュールは，Dilated Convolution (膨張畳み込み) [1] を用いることで，エンコーダの出力 $\boldsymbol{X}_{B5}$ から効率的に長期的なコンテキストを抽出する．このモジュールは Xi と Chen ら [1], [11] による画像セグメンテーションに関する研究を参考にした．

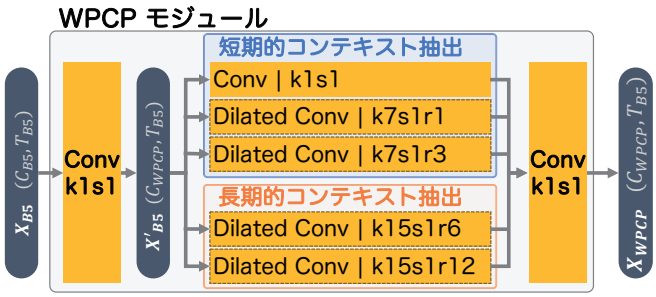
Dilated Convolution で用いられるフィルタの例を図 5(a) に示す．Dilated Convolution の畳み込みフィルタは，カーネルサイズ k と膨張率 r によって定義される．膨張率に合わせてサイズ k の畳み込みフィルタに穴を挿入することで，学習するパラメータ数を増やさずにより離れたタイムステップから情報を収集することができる．

図 5(b) に WPCP モジュールの構成を示す．まずエンコーダの出力 $\boldsymbol{X}_{B5}$ に，次元削減のためにカーネルサイズ 1 の畳み込みを行い，$\boldsymbol{X}'_{B5}$ を計算する．次に，$\boldsymbol{X}'_{B5}$ に膨張率の異なる複数の Dilated Convolution を適用し，再びカーネルサイズ 1 の畳み込みで並列して実行した畳み込み層の出力を統合することで $\boldsymbol{X}_{WPCP}$ を得る．3 つの Dilated Convolution には小さな膨張率を使用し，小さな動作などの短期的なコンテキストを抽出する．残り 2 つの Dilated Convolution には大きな膨張率を使用し，作業工程間の長期的なコンテキストを抽出する．これにより，注目点付近の動作と，その前後の作業工程の情報を抽出することができる．例えば本研究の設定の $(k, r) = (15, 12)$ では，最大 22.4 秒前と後の情報を抽出する．

3.2.2 Boundary Detector

Boundary Detector は，エンコーダの中間出力と WPCP モジュールの出力を用いて，連続する行動の境界に関する情報を抽出する．具体的には，各行動の開始/終了時刻をマルチラベル分類によって推定する．図 6 に Boundary Detector の構成を示す．入力はエンコーダの出力 $\boldsymbol{X}_{B1}$・

(a) Dilated Convolution



(b) WPCP モジュール

図 5: (a) Dilated Convoltion と (b) WPCP モジュールの構成．k・s・r はそれぞれカーネルサイズ，ストライド，膨張率を表す．

$\boldsymbol{X}_{B2}$・$\boldsymbol{X}_{B3}$ と，WPCP モジュールの出力 $\boldsymbol{X}_{WPCP}$ を入力信号と同じ長さ N_T にアップサンプルしたテンソルである．出力は各行動の開始/終了時刻を表す長さ N_T，チャンネル数 $2N_C$ のテンソルである．c 番目の行動クラスの開始時刻の推定値の系列は，以下のように表される．

$$\hat{\boldsymbol{b}}_c^{st} = [b_1^{st,c}, b_2^{st,c}, ..., b_t^{st,c}, ..., b_{N_T}^{st,c}],$$

ここで，$b_t^{st,c}$ は時刻 t に c 番目の行動の開始時刻が存在する確率を表す．したがって，Boundary Detector の出力は以下のように表される．

$$\hat{\boldsymbol{B}} = [\hat{\boldsymbol{b}}_1^{st}, \hat{\boldsymbol{b}}_1^{ed}, ..., \hat{\boldsymbol{b}}_c^{st}, \hat{\boldsymbol{b}}_c^{ed}, ..., \hat{\boldsymbol{b}}_{N_C}^{st}, \hat{\boldsymbol{b}}_{N_C}^{ed}]^\top,$$

ここで，$\hat{\boldsymbol{b}}_c^{ed}$ は c 番目の行動の終了時刻の存在確率の系列を表す．

Boundary Detector の構成は，Yu らの画像に対する境界検出研究 [14] を参考に，マルチラベル分類を行うために Shared Concatenation を応用する．まず，エンコーダの中間出力から生成した時間解像度が高い特徴表現 $\boldsymbol{X}_{SF}$ と，WPCP モジュールの出力から作成した $\boldsymbol{X}_{CF}$ の i チャネル目を結合し，$2N_C$ 個のテンソルを作成する (Shared Concatenation)．次に，各テンソルに対して畳み込み演算を適用し，$b_t^{st/ed,c}$ を出力する．このような構成をとることで，出力 $\hat{\boldsymbol{B}}$ は各行動クラスの開始/終了点の情報を抽出することができる．

行動の開始/終了時刻を推定するために，モデル学習において Tversky 損失 [10] を用いた．境界検出タスクは，負のサンプル数が正のサンプル数よりも圧倒的に多いため，クラス不均衡の問題が発生する．Tversky 損失は，適合率と再現率の両方の重みを独立して調整することができるため，このような不均衡データの学習に適した損失関数である．マルチラベル分類のための Tversky 損失は，以下のように計算することができる．

$$\mathcal{L}_b = \sum_{c,p\in\{st,ed\}} \mathrm{TL}(\hat{\boldsymbol{b}}_c^p, \boldsymbol{b}_{c,GT}^p, \beta_{fp}, \beta_{fn}),$$

$\hat{\boldsymbol{b}}_c^p$ は c 番目の行動の開始/終了時刻の推定値，$\boldsymbol{b}_{c,GT}^p$ は c 番目の行動の開始/終了時刻の真値を表す．また本研究では，

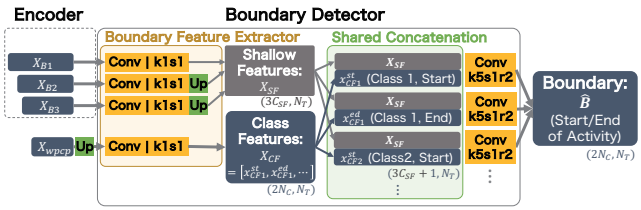


図 6: Boundary Detector の構成

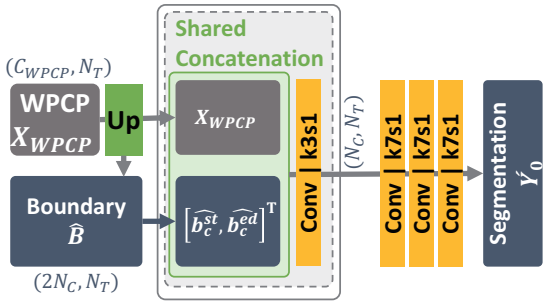


図 7: Context–Boundary Integrator の構成

行動の遷移があった時刻の前後 $2T_{bd}$ 秒の領域を境界として定義した．本研究の実験では $T_{bd} = 15$pt とした．

3.2.3 Context–Boundary Integrator

WPCP モジュールと Boundary Detector の出力を統合し，各タイムステップの行動クラスの推定を行う．Context–Boundary Integrator の構成を図 7 に示す．まず WPCP モジュールの出力をアップサンプルしたテンソルと Boundary Detector の出力を，Shared Concatenation によって連結し，行動クラスごとに特徴抽出を行う．次に，クラスごとに抽出したテンソルを結合し，畳み込み層で処理することで，クラス間の関係を考慮した多クラス分類の結果 $\acute{\boldsymbol{Y}}_0$ を得る．

3.3 Refinement Module

Refinement モジュールは，デコーダの出力に含まれる散発的な誤認識を，作業順序に関する事前知識を用いて修正する．Refinement モジュールの入力はデコーダの出力である行動クラスの推定値 $\acute{\boldsymbol{Y}}_0$ であり，出力は入力を補正した系列 $\hat{\boldsymbol{Y}}$ である．入力と出力はともに長さ N_T の N_C 次

元の時系列データである．時系列中の時刻 t における N_C 次元のベクトルは，時刻 t における各行動クラスの確率を表す．図 8(a) にデコーダの出力例を示す．推定値に多くの散発的な誤認識があることがわかる．

Refinement モジュールは $\acute{\boldsymbol{Y}}_0$ に含まれる想定外の作業順序の遷移の修正を試みる．図 2 のように，作業順序は有向グラフで表現することができ，Refinement モジュールではこれを遷移行列 $\boldsymbol{M}$ に変換し利用する．遷移行列 $\boldsymbol{M}$ は $N_C \times N_C$ の行列で，(i, j) 番目の要素 $m_{i,j}$ は i 番目から j 番目の行動への遷移が可能か否かを示している．つまり，有効グラフ上に辺が存在すれば $m_{i,j} = 1$，なければ $m_{i,j} = 0$ である．ただし自己遷移を表す対角成分 $m_{i,i}$ は未定義とする．作業順序は，行動認識研究者以外が作成した情報を元にするため，このように経験や専門知識を必要としない，可能な限りシンプルな定義を用いる．

図 9(a) に Refinement モジュールの構成を示す．Refinement モジュールは，図 9(b) に示す Refinement ブロックで構成される．Refinement ブロックでは，まず入力された行動クラスの系列に対して作成した遷移行列 $\boldsymbol{M}$ を用いて，以下の式に基づき，時刻 t における遷移コスト $\boldsymbol{c}_t$ を計算する．遷移コストは，想定される遷移では小さく，想定外の遷移には大きなコストを割り当てる．

$$\boldsymbol{c}_t = \begin{cases} 0 & \text{if } \arg\max(\acute{y}_t) = \arg\max(\acute{y}_{t+1}) \\ \alpha & \text{else if } m_{i,j} == 1 \\ 1 & \text{else if } m_{i,j} == 0 \end{cases},$$

α は可能な遷移に対するコスト $(0 < \alpha < 1)$，$\acute{y}_t$ は時刻 t における Refinement ブロックへの入力である．ただし，遷移コスト C はネットワークに投入する前に，窓サイズ w_{cost} の移動和を適用して平滑化を行う．その後，入力 $\acute{\boldsymbol{Y}}_n$ の各チャネルに対して計算した遷移コストを結合し，異なる膨張率の Dilated Convolution に入力する．例えば，図 8(a) の 4 秒付近の誤認識など，比較的継続時間が長い誤認識を修正するためには直前直後の情報だけではなく，ある程度離れた時刻の情報も参考にする必要がある．Refinement ブロックでは，様々な膨張率の Dilated Convolution を並列的に適用することで，修正に必要な情報を収集している．最後に並列して計算した Dilated Convolution の出力と $\acute{\boldsymbol{Y}}_n$ を結合し，畳み込み層で処理することで修正を行う．$\acute{\boldsymbol{Y}}_n$ と $\acute{\boldsymbol{Y}}_{n+1}$ は，誤認識箇所以外は基本的には一致する．そこで，Skip Connection を用いて $\acute{\boldsymbol{Y}}_n$ を畳み込み層に入力することで，修正の必要がない箇所を保存している．図 8(c) に Refienment モジュールによる修正結果を示す．散発的な誤認識が修正されていることが分かる．本研究では，$\alpha = 0.1$，$w_{cost} = 30$pt を使用した．

Refinement モジュールの学習には，以下の損失関数を用いる．

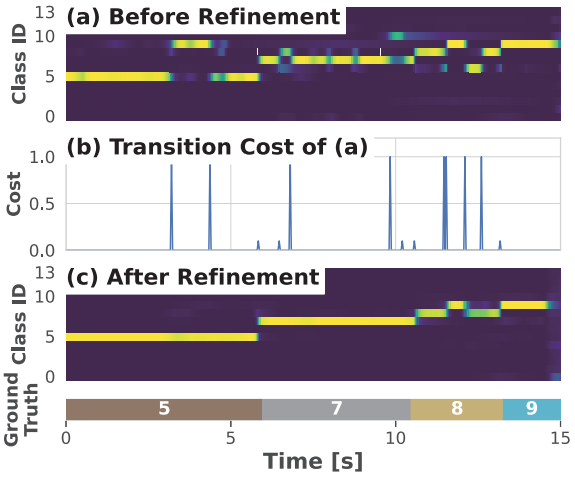
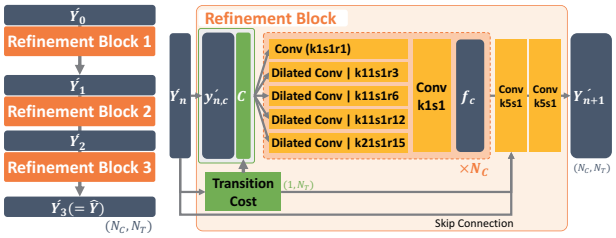


図 8: Refinement を行なった予測結果の例



(a) (b)

図 9: Refinement モジュール (a) と Refinement ブロックの構成 (b)

$$\mathcal{L}_r = \gamma_1 \mathrm{CE}(\acute{\boldsymbol{Y}}_1, \boldsymbol{Y}) + \gamma_2 \mathrm{CE}(\acute{\boldsymbol{Y}}_2, \boldsymbol{Y}) + \gamma_3 \mathrm{CE}(\acute{\boldsymbol{Y}}_3, \boldsymbol{Y}),$$

ここで，CE() は log–softmax crossentropy を計算し，γ_1, γ_2, γ_3 はトレードオフ・パラメータである．$\mathcal{L}_r$ の最初の CE() は最初の Refinement ブロックに対して，出力 $\acute{\boldsymbol{Y}}_1$ と真値 $\boldsymbol{Y}$ の間の誤差を計算する．損失関数の 2 項目と 3 項目の CE() は，それぞれ同様に 2 番目と 3 番目のブロックの出力を評価する．また，深いブロックほど大きな γ_n を用いることで，デコーダの予測値の誤差を深くなるにつれ緩やかに補正することができる．本研究では $\gamma_2 = 0.95$, $\gamma_3 = 1.0$ とした．

また，学習データ量が限られているため，Refinement モジュールが誤差パターンを網羅的に学習できない可能性がある．そこで，デコーダの出力 $\acute{\boldsymbol{Y}}_0$ と真値 $\boldsymbol{Y}$ を用いて，(i) 行動の伸長・短縮，(2) 散発的な誤認識をランダムに挿入し，データオーグメンテーションを行う．

3.4 Network Training

LOS-Net の学習では，モーメント付き SGD を用いて以下の損失関数を最小化する．

$$\mathcal{L}(\theta_e, \theta_d, \theta_r) = \gamma_0 \mathcal{L}_c(\theta_e, \theta_d) + \gamma_{bd} \mathcal{L}_b(\theta_e, \theta_d, \theta_r) + \mathcal{L}_r(\theta_e, \theta_d, \theta_r),$$

ここで, $\theta_e, \theta_d, \theta_r$ はそれぞれエンコーダ, デコーダ, Refinement モジュールのネットワークパラメータである．また, γ_0 と γ_{bd} は，トレードオフ・パラメータである．また, $\mathcal{L}_c(\cdot)$ はデコーダの行動クラス推定値を評価する Cross–Entropy 損失である．本研究では $\gamma_0 = 0.95$，$\gamma_1 = 1.0$ を使用した．

3 つの Refinement モジュールを同時に学習することは困難であるため，2 段階に分けてモデルを学習する．まず，エンコーダ，デコーダ，1 つ目の Refinement ブロックを，N_{e1} エポック，学習率 lr_1 で，$\mathcal{L}_c$，$\mathcal{L}_b$，および $\mathcal{L}_r$ の第一項を用いて学習する．続いて，2・3 つ目の Refinement ブロックをかさ増ししたデータを用いて，lr_2 で N_{e2} エポック，損失関数 $\mathcal{L}_r$ で学習する．学習を安定させるため，二段目の学習ではエンコーダから 1 つ目の Refinement ブロックのパラメータは固定し，2・3 つ目の Refinement ブロックのパラメータは学習済みの 1 つ目の Refinement ブロックの重みで初期化した．

4. 評価

4.1 データセット

実際の工場や物流センタで働く 11 人の作業者から収集した加速度データを用いて，LOS-Net の評価を行った．データは両手に装着したスマートウォッチ (Sony SmartWatch3 SWR50) を用いて，サンプリングレート約 30Hz で加速度データを収集した．行動クラスは現場の管理者が作成した作業指示書に基づいて定義し，ビデオを参照してアノテーションを行い正解ラベルを作成した．データセットの概要を表 1 に示す．作業者によって作業工程は異なる．「SCREW」はネジ締め作業,「CHECK」は完成品の検査作業,「PACK」は物流センタにおける梱包作業である．前処理として，測定誤差に対応するため，$[-3\mathrm{G}, +3\mathrm{G}]$ の範囲外の加速度値にはクリッピングを行い，各軸ごとに正規化を行った．

4.2 評価手順

本研究では，学習データ量が限定された環境を想定している．そこで，各作業者のデータを，繰り返し作業 10 period を 1 セットとして，N_{fold} 個に分割した．次に，分割 1 つ分を学習データとしてモデルを学習し，残りの分割 (すなわち，$N_{fold} - 1$ 分割) に対して評価を行った．この手順を繰り返すことで，各分割が 1 回ずつ学習データとなるようにした．よりロバストな評価を行うため，ランダムシードを変えて，同様のクロスバリデーションを 5 回繰り返した．評価指標には，各タイムステップを 1 サンプルとして計算した F1-measure (マクロ平均) を用いた．マクロ平均 F1 値は，各クラスで計算された F1 値を平均することで計算され，クラス不均衡問題の影響を受けにくい．

LOS-Net の有効性を評価するために，次の手法と比較した．

- **ConvLSTM**: [8] で提案されたモデル．畳み込み層 5 層の後に，LSTM 層 2 層で構成される．ユニット数とカーネルサイズは関連研究 [8], [15] と同様の設定を用いた．
- **U-Net**: [15] で提案されたモデル．このモデルはそれぞれ 5 段のエンコーダ・デコーダブロックから構成される．ユニット数とカーネルサイズは [15] と同様の設定を用いた．
- **LOS-Net**: 提案手法.

また，LOS-Net の各機能の評価のため，以下のモデルを用いた．

- **LOS-Net(-WPCP)**: LOS-Net から WPCP モジュールと Refinement モジュールを除去したモデル．$\boldsymbol{X}_{B5}$ を直接 Context-Boundary Integrator に入力する．
- **LOS-Net(-B)**: LOS-Net から Boundary Detector と Refinement モジュールを除去したモデル．Context–Boundary Integrator の入力は，WPCP モジュールの出力だけである．
- **LOS-Net(-R)**: LOS-Net から Refinement モジュールを除去したモデル．このモデルの出力はデコーダの出力 $\acute{\boldsymbol{Y}}_{\boldsymbol{0}}$ と同じである．

上記手法は PyTorch と Sklearn を用いて実装し，GPU クラスタ上で実験を行った．

提案手法および比較手法のモデルのパラメータ数を表 2 に示す．LOS-Net は U-Net に比べ、デコーダ，エンコーダともにパラメータ数が少なく、全体として U-Net の約 1/10 であった。LOS-Net は WPCP モジュールで長期的なコンテキストを抽出するため，U-Net よりもブロック数の少ないエンコーダを使用した．なお，U-Net で長期的コンテキストを捉えるためには，ブロック数を増やす必要があり，結果的にパラメータが多くなる．

5. 結果

5.1 定量的評価

図 10 に，作業者ごとに集計した各手法の認識精度 (F 値マクロ平均) の 5 回の平均値 (棒グラフ) と標準偏差 (誤差棒) を示す．図より LOS-Net は全ての作業者において，既存モデル (ConvLSTM、U-Net) を大きく上回った．ConvLSTM では LSTM 層，U-Net ではダウンサンプリングとアップサンプリングといった長期的なコンテキストを抽出するための仕組みが入っている．しかし，学習データ量の制約や受容野の広さにより，作業行動の認識のために必要な長期的なコンテキストの抽出が困難だったと考えられる．

また図 10 に示した，LOS-Net(-WPCP)，LOS-Net(-B)，LOS-Net(-R) と LOS-Net の比較より，WPCP Module が認識精度の向上に最も貢献していることがわかる．例えば作業者 A で WPCP モジュールの有無を比較すると, WPCP

表 1: 実験で使用したデータセットの概要

Worker	A	B	C	D	E	F	G	H	I	J	K
# of periods	40	40	40	40	40	60	100	30	50	20	70
# of activities (N_C)	12	12	16	16	19	19	14	14	9	10	10
AVG period duration	117s	127s	132s	118s	104s	103s	81s	134s	68s	106s	135s
(SD of duration)	(15s)	(20s)	(41s)	(12s)	(64s)	(17s)	(11s)	(66s)	(27s)	(35s)	(54s)
data duration	78m4s	84m57s	88m33s	78m52s	69m31s	103m58s	136m23s	67m11s	57m7s	35m22s	158m0s
work type	SCREW	SCREW	CHECK	CHECK	SCREW	SCREW	CHECK	CHECK	PACK	PACK	PACK

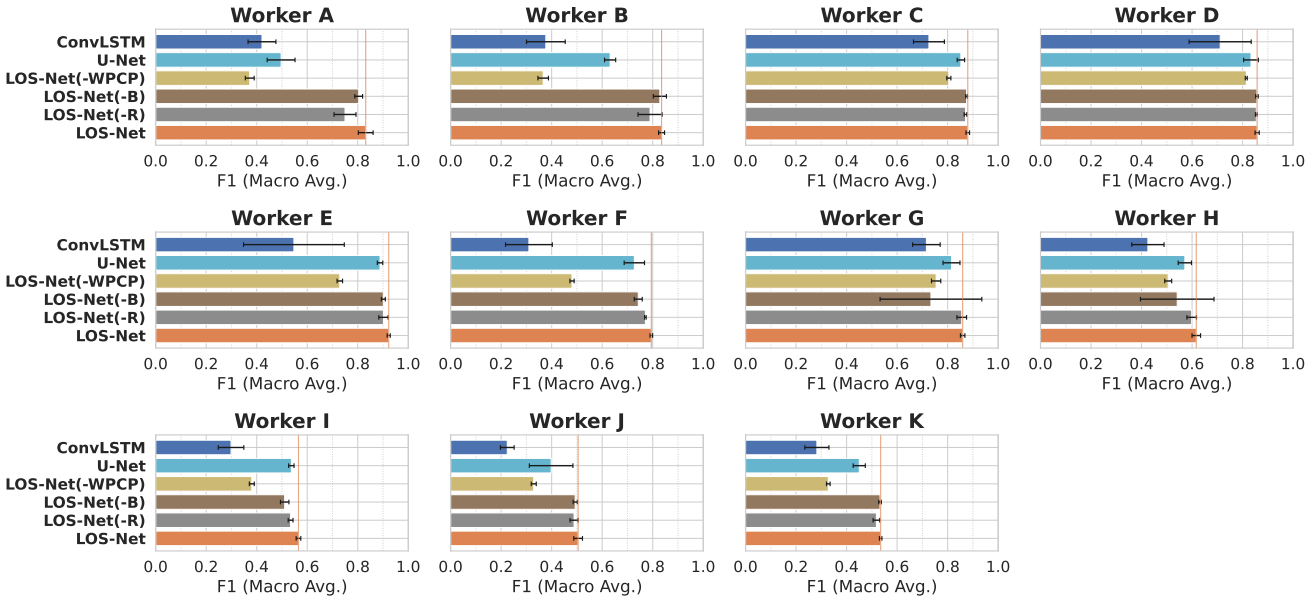


図 10: 認識精度 (F 値)

表 2: モデルのパラメータ数 ($N_T = 1800pt$, $N_C = 10$ で集計)

Model	Params	Model	Params
ConvLSTM	296,330	LOS-Net	963,620
		– Encoder	437,824
U-Net	11,537,162	– WPCP	442,368
– Encoder	6,296,096	– Boundary Detector	9,508
– Decoder	5,241,066	– CB Integrator	10,500
		– Refinement Module	21,140

モジュールがない LOS-Net(-WPCP) に対して WPCP モジュールがある LOS-Net(-R) は，F 値が 0.30 以上向上した．また，Boundary Detector と Refinement モジュールも，WPCP モジュールほど大きくはないが，認識精度向上への寄与が見られた．

5.2 定性的評価

図 11 に，ネジ締め作業 (SCREW) を行う作業者 A の LOS-Net，U-Net，ConvLSTM の出力例を示す．行動クラス 2–10 は，それぞれ異なるネジを締める動作であるが，U-Net と ConvLSTM では異なるネジを締める動作に誤認識されている．一方，提案手法 LOS-Net は，注目する行動の前後に何番目のネジを締めているかといった長期的なコンテキストを考慮することで，比較的正確に認識することができたと考えられる．ConvLSTM と U-Net はこのような長期的なコンテキストを捉えることができず，ネジ締めの作業工程を別のネジを締める行動クラスに誤認識したと考えられる．また，作業者 B，E，F，I，J，K についても，同様の誤認識が見られた．

図 12 に，ネジ締め作業 (SCREW) を行う作業者 E の LOS-Net(-B) と LOS-Net(-R) の出力例を示す．Boundary Detector を使用しない LOS-Net(-B) には，0–15 秒目付近の行動クラス 1 のように，ある行動の途中で散発的に誤認識が発生している．一方で Boundary Detector を使用する LOS-Net(-R) は，この散発的に発生する誤認識を抑制することができている．Boundary Detector を使用することで，連続する行動の境界を正確にするだけではなく，行動の境界の偽陽性を抑制する効果があった．

図 13 に，梱包作業 (PACK) を行う作業員 I の LOS-Net(-R)，LOS-Net の出力例を示す．Refinement モジュールを使用しない LOS-Net(-R) の出力結果には，断続的に作業順序を無視した散発的な誤認識が発生している．梱包作業は梱包するアイテムに影響を受ける変則的な作業であるため認識が難しい．このような散発的な誤認識は，比較的継続時間が長い作業工程の特徴的な動作に影響されていると考えられる．一方で，Refinement モジュールを用いる

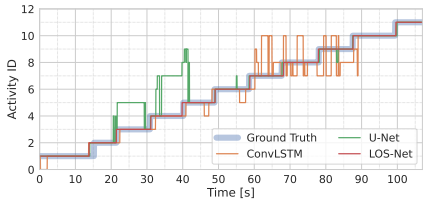


図 11: モデルの出力例 (作業員 A; ConvLSTM，U-Net，LOS-Net)

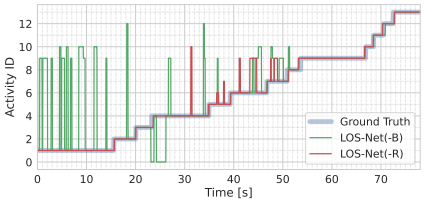


図 12: モデルの出力例 (作業員 E; LOS-Net(-B)，LOS-Net(-R))

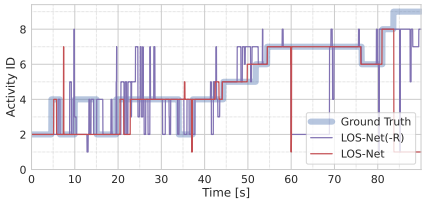


図 13: モデルの出力例 (作業員 I; LOS-Net(-R)，LOS-Net)

LOS-Net は，そのような誤認識を大きく抑制することができている．

6. 結論

本研究では，工場・物流センタなどにおける作業行動の認識を行うモデル LOS-Net を提案した．作業行動の認識システムを実用化するためには，学習データの収集コストが大きな問題となっている．LOS-Net では，作業行動の特徴である作業順序などの事前知識に着目し，前後の作業工程などの長期的なコンテキストを効率的に抽出する WPCP モジュールや，想定された作業順序の情報を用いて誤認識を修正する Refinement モジュールを提案した．また，連続する行動の境界を正確に認識するための Boundary Detector も採用している．実際の工場で 11 名の被験者から収集されたデータを用いて検証を行い，LOS-Net が既存手法の認識精度を大きく上回ることを確認した．今後は同じ作業工程の繰り返しを個別に検出する Panoptic Segmentation などに取り組んでいきたい．

謝辞

本研究の一部は JSPS 科研費 JP21H03428, JP21H05299, and JP21J10059, JST ACT-X JPMJAX200T の助成を受けて行われたものである．

参考文献

[1] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation, *Proceedings of the European Conference on Computer Vision*, pp. 801–818 (2018).

[2] Forkan, A. R. M., Montori, F., Georgakopoulos, D., Jayaraman, P. P., Yavari, A. and Morshed, A.: An industrial IoT solution for evaluating workers' performance via activity recognition, *Proceedings of the 39 th IEEE International Conference on Distributed Computing Systems*, pp. 1393–1403 (2019).

[3] Garcia-Ceja, E., Galván-Tejada, C. E. and Brena, R.: Multi-view stacking for activity recognition with sound and accelerometer data, *Information Fusion*, Vol. 40, pp. 45–56 (2018).

[4] Gashi, S., Di Lascio, E. and Santini, S.: Using unobtrusive wearable sensors to measure the physiological synchrony between presenters and audience members, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 3, No. 1, pp. 1–19 (2019).

[5] Hammerla, N. Y., Halloran, S. and Ploetz, T.: Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables, *CoRR*, Vol. abs/1604.08880 (2016).

[6] Inoue, S., Lago, P., Hossain, T., Mairittha, T. and Mairittha, N.: Integrating Activity Recognition and Nursing Care Records: The System, Deployment, and a Verification Study, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 3, No. 3 (online), DOI: 10.1145/3351244 (2019).

[7] Münzner, S., Schmidt, P., Reiss, A., Hanselmann, M., Stiefelhagen, R. and Dürichen, R.: CNN-based sensor fusion techniques for multimodal human activity recognition, *Proceedings of the ACM International Symposium on Wearable Computers*, pp. 158–165 (2017).

[8] Ordóñez, F. and Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, *Sensors*, Vol. 16, No. 1, p. 115 (2016).

[9] Reiss, A. and Stricker, D.: Introducing a new benchmarked dataset for activity monitoring, *Proceedings of the International Symposium on Wearable Computers*, pp. 108–109 (2012).

[10] Salehi, S. S. M., Erdogmus, D. and Gholipour, A.: Tversky loss function for image segmentation using 3D fully convolutional deep networks, *Proceedings of the International workshop on machine learning in medical imaging*, Springer, pp. 379–387 (2017).

[11] Xi, R., Hou, M., Fu, M., Qu, H. and Liu, D.: Deep dilated convolution on multimodality time series for human activity recognition, *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8 (2018).

[12] Xia, Q., Korpela, J., Namioka, Y. and Maekawa, T.: Robust Unsupervised Factory Activity Recognition with Body-worn Accelerometer Using Temporal Structure of Multiple Sensor Data Motifs, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 4, No. 3, pp. 1–30 (2020).

[13] Yao, R., Lin, G., Shi, Q. and Ranasinghe, D. C.: Efficient dense labelling of human activity sequences from wearables using fully convolutional networks, *Pattern Recognition*, Vol. 78, pp. 252–266 (2018).

[14] Yu, Z., Feng, C., Liu, M.-Y. and Ramalingam, S.: Casenet: Deep category-aware semantic edge detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5964–5973 (2017).

[15] Zhang, Y., Zhang, Z., Zhang, Y., Bao, J., Zhang, Y. and Deng, H.: Human activity recognition based on motion sensor using u-net, *IEEE Access*, Vol. 7, pp. 75213–75226 (2019).