Journal of Intelligent Manufacturing (2023) 34:633–649
https://doi.org/10.1007/s10845-021-01815-x

An individualized system of skeletal data-based CNN classifiers for action recognition in manufacturing assembly

Md. Al-Amin[1] · Ruwen Qin[2] · Md Moniruzzaman[3] · Zhaozheng Yin[4] · Wenjin Tao[5] · Ming C. Leu[5]

Received: 24 October 2020 / Accepted: 6 July 2021 / Published online: 26 July 2021

Abstract

Real-time Action Recognition (ActRgn) of assembly workers can timely assist manufacturers in correcting human mistakes and improving task performance. Yet, recognizing worker actions in assembly reliably is challenging because such actions are complex and fine-grained, and workers are heterogeneous. This paper proposes to create an individualized system of Convolutional Neural Networks (CNNs) for action recognition using human skeletal data. The system comprises six 1-channel CNN classifiers that each is built with one unique posture-related feature vector extracted from the time series skeletal data. Then, the six classifiers are adapted to any new worker through transfer learning and iterative boosting. After that, an individualized fusion method named Weighted Average of Selected Classifiers (WASC) integrates the adapted classifiers as an ActRgn system that outperforms its constituent classifiers. An algorithm of stream data analysis further differentiates the actions for assembly from the background and corrects misclassifications based on the temporal relationship of the actions in assembly. Compared to the CNN classifier directly built with the skeletal data, the proposed system improves the accuracy of action recognition by 28%, reaching 94% accuracy on the tested group of new workers. The study also builds a foundation for immediate extensions for adapting the ActRgn system to current workers performing new tasks and, then, to new workers performing new tasks.

Keywords Convolutional neural network · Action recognition · Transfer learning · Iterative boosting · Classifier fusion · Smart manufacturing · Deep learning

Introduction

Assembly is a process of coupling multiple workpieces together to produce a product of full functionality. It accounts for 20% of total production cost and 50% of total production time, respectively. In the automotive industry, the direct labor cost spent on assembly is ranged from 20 to 70% (ElMaraghy and ElMaraghy 2016). Therefore, the efficiency and quality of assembly are critical to manufacturers. The ability to recognize actions of assembly workers in real-time provides an opportunity to timely correct human mistakes and facilitate workers to operate effectively based on their particular needs (Zhou et al. 2013; Wang et al. 2021). Production innovations are occurring faster than ever. Manufacturing workers thus need to frequently learn new methods and skills. While vigorous efforts have been devoted to human action recognition (ActRgn) for various purposes (e.g., Pham et al. 2018; Moniruzzaman et al. 2021), action recognition for manufacturing assembly is rather limited for a few reasons. Such actions are complex, involve many fine motions, require interactions with various tools and parts, and have between-action similarity. Recognizing the detail of such actions in high accuracy is challenging. The lack of publicly available datasets on worker actions in manufacturing assembly

✉ Ruwen Qin
ruwen.qin@stonybrook.edu

1 Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA

2 Department of Civil Engineering, Stony Brook University, Stony Brook, NY 11794, USA

3 Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

4 Department of Biomedical Informatics, Department of Computer Science, and AI Institute, Stony Brook University, Stony Brook, NY 11794, USA

5 Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA

is another obstacle facing the manufacturing research community (Al-Amin et al. 2020b).

RGB image-based action recognition has some limitations such as occlusion, luminosity, and the privacy concern (Chen et al. 2017). Therefore, wearable sensors have been predominantly used to recognize actions in manufacturing (Stiefmeier et al. 2008; Tao et al. 2018; Kong et al. 2019). To capture the movement of different body parts, a worker may need to wear multiple wearable devices on the body. This can cause discomfort and pressure to workers in some circumstances, negatively impacting their productivity. Some depth sensors such as the Microsoft Kinect can extract the 3D skeletal data of humans from the depth images they capture, thus being an alternative when RGB and wearable sensors are limited for use. Skeletal data provide a lower-dimensional representation of actions than other sensor data, which make action recognition faster, computationally efficient, and better in accommodating the real-time inference (Pham et al. 2018). The spatially distributed body joints of a worker indicate the posture of the worker. The temporal dynamics of the posture contain features of worker actions (Du et al. 2015). Various approaches were proposed to capture human actions from 3D skeletal data, including hidden Markov models (Rude et al. 2018), dividing the posture into body parts and encoding them into images(Khaire et al. 2018), using the coordinates of body joints directly (Pham et al. 2018), and extracting statistical features from skeletal data (Shen et al. 2020). These methods nonetheless neglect some useful information that can be extracted from skeletal data.

Following the success of Convolutional Neural Network (CNN) in image analysis (Krizhevsky et al. 2012), skeletal data are presented as images and processed by ActRgn CNNs (e.g., Al-Amin et al., 2019; Li et al., 2017; Kamel et al., 2019). Compared to Multilayer Perceptrons (MLP) and Recurrent Neural Networks (RNN), CNN can automatically extract discriminative features of subtle, complex actions from the spatial and temporal relations of body joints, which can be obtained from the time series skeletal data (Al-Amin et al. 2020b). Thus, CNN is an attractive candidate classifier for skeletal data-based action recognition. Yet, challenges are also identified from those pioneer studies. First, how to translate the time series skeletal data into a set of images that capture temporal and spatial cues for action recognition? Second, how to address the negative impact of human heterogeneity on the ActRgn performance, including both the within-subject and the between-subject variances. Third, how to fuse multiple features or classifiers that can be developed from the skeletal data to provide more reliable ActRgn performance? Last but not the least, how to address the limitation of CNN classifiers in analyzing the stream data in real-time? Answers to these questions will advance the knowledge of skeletal data-based human action recognition in manufacturing assembly.

To address the above-discussed challenges, this paper proposes an individualized system of skeletal data-based CNN classifiers for recognizing worker actions in manufacturing assembly. Efforts to create this system are the development and integration of the following capabilities:

- System architecting that involves extracting feature images from the time series skeletal data to build independent, complementary constituent classifiers and fusing them as an ActRgn system;
- A method to adapt ActRgn classifiers to individual workers, which addresses the issues of between-subject heterogeneity and within-subject variance;
- A fusion method named Weighted Average of Selected Classifiers (WASC) which maximizes the ActRgn performance at the system's level for any individual worker;
- A data analysis algorithm that improves the ActRgn result from analyzing untrimmed stream data.

The remainder of the paper is organized as the following. The Literature section summarizes the related literature, followed by an elaborated description of the proposed approach to developing the ActRgn system in the Methodology section. Then, an illustrative example and the assessment of the proposed ActRgn system are presented. The conclusion from this study and future work are summarized at the end.

The literature

The prior work on image analysis using CNN, transfer learning, classifier fusion, and temporal coherence information build the foundation for the proposed ActRgn system. Gaps in the literature inspire the technical approach to creating the system.

CNN is a feed forward neural network that works well in image analysis. When using it for action recognition, spatial features of the skeleton in actions are presented as images and a CNN is trained to classify the images (Khaire et al. 2018). The skeleton optical spectra (Li et al. 2019) and the graph convolution (Hou et al. 2018) were proposed for learning dynamic features of the skeleton in actions. Moreover, to incorporate both the spatial and temporal information of actions, a multi-task learning network for action recognition was also developed to jointly process images in parallel (Ke et al. 2017). Long Short Term Memory (LSTM) is an advanced version of recurrent neural networks, which models long-term dependencies with memory cells. It has been applied to skeletal-based action recognition as well (Han et al. 2018). The body joints in each frame are of unequal importance, and so frames in a sequence. Therefore, certain weights are automatically assigned to dominant joints and frames (Song et al. 2017; Liu et al. 2017). CNN and LSTM were

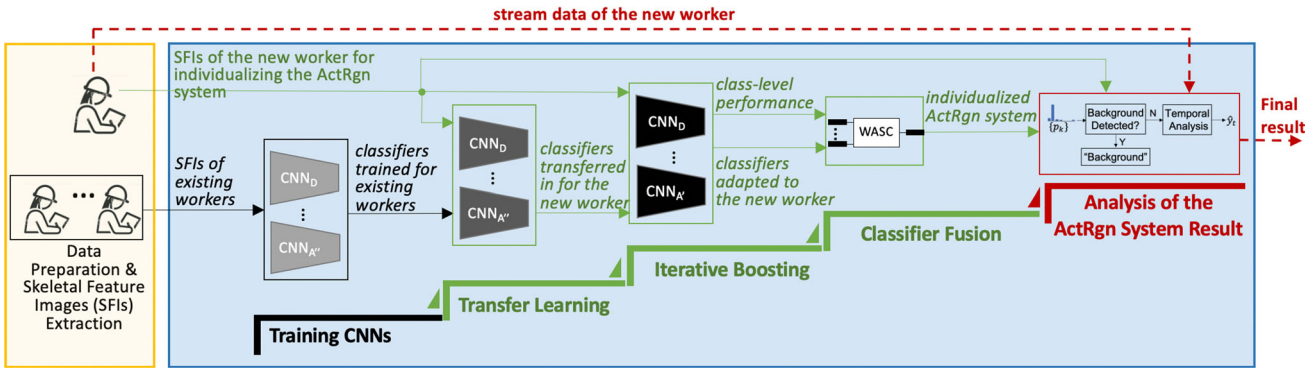

Fig. 1 The approach to creating an individualized action recognition (ActRgn) system of CNN classifiers

also used simultaneously through the score fusion (Li et al. 2017; Nunez et al. 2018). In this approach, spatial domain features and temporal domain features can be extracted and fed to the CNN and the LSTM, respectively. If the input to a CNN classifier captures the spatio-temporal features of the skeleton, the CNN by itself can learn features of actions well. This approach is simple, but not explored thoroughly in the literature.

Knowledge transfer is critical for action recognition due to the inevitable differences between the source and the target population of workers (Cook et al. 2013). Zhao et al. (2011) proposed a transfer learning algorithm named TransEMDT that integrates the decision tree with the k-means clustering algorithm to achieve personalized activity recognition. The use of transfer learning coupled with deep learning for action recognition is still limited. Recently, Al-Amin et al. (2020b) transferred an ActRgn CNN to new subjects through fine-tuning the model with a small amount of data from new subjects. While transfer learning is shown to work, rooms for improvement are noticed.

Classical methods of classifier fusion such as majority voting, Naive Bayes, Dempster-Shafer theory, average fusion, and random forests usually treat classifiers equally. Therefore, they overlook the strength and weaknesses of different classifiers. To overcome this limitation, classifiers may be assigned weights based on their abilities in a variety of approaches. For instance, Ward et al. (2006) ranked classifiers according to the highest rank, borda count, and logistic regression. Hierarchical fusion is another approach (Banos et al. 2013), and Guo et al. (2019) developed it based on the entropy weight. Weighted linear opinion pools and weighted logarithmic opinion pools were also implemented for the classifier fusion (Guo et al. 2012). Weights for classifiers are determined in various ways, including genetic algorithms (Chernbumroong et al. 2015) and classifier performance measurements such as the overall accuracy of classifiers (Chung et al. 2019) and class-level recall values (Tsanousa et al. 2019). However, strength and weaknesses of individual classifiers are not consistent among workers.

The temporal information of objects in successive images can help improve the object detection from video data. Examples include the use of temporal and contextual information from tubelets obtained from videos (Kang et al. 2018), the propagation of deep feature maps from key frames to other frames (Zhu et al. 2017b), and the flow-guided feature aggregation that integrates features from nearby frames (Zhu et al. 2017a). Likewise, the temporal coherence information of sequential actions in assembly can be used to improve the ActRgn accuracy. This method has not been thoroughly explored.

Methodology

The proposed approach to creating the ActRgn system is illustrated in Fig. 1. First, Skeletal Feature Images (SFIs) extracted from a group of existing workers are used to train a set of CNN classifiers that each captures a unique aspect of assembly actions. Then, these classifiers are refined with the SFIs of a new worker to adapt to that worker through transfer learning and iterative boosting. After that, the adapted classifiers are fused as a system for recognizing the actions of the new worker. The stream data analysis algorithm corrects possible mistakes that the ActRgn system made in analyzing the stream data in real-time. The description of the symbols used in this paper is presented in Table 1.

Data preparation

The study collects data from two mutually exclusive groups of subjects. The first group is a sample of workers currently assigned to perform the assembly operation of the study. The second group is a sample of new workers who will be joining the assembly line to perform this operation. To capture the within-subject variance in the operation, subjects are asked

Table 1 Nomenclature

ActRgn: action recognition
CNN: convolutional neural network
CRT: the group of current workers
DNN: deep neural network
IB: iterative boosting
IL: incremental learning
LSTM: long short-term memory
NEW: the group of new workers
ReLU: rectified linear unit
RGB: red green blue
SDA: stream data analysis
SDI: skeletal data image
SFI: skeletal feature image
TC: temporal coherence
TL: transfer learning
WASC: weighted average of selected classifiers
$\mathbf{A}_i$: angle feature vector in frame i
$\mathbf{A}'_i/\mathbf{A}''_i$: the first/second order derivatives of $\mathbf{A}_i$
$\text{CNN}_{m,n}$: the m^{th} CNN attained from iteration n
CNN^*_m: the m^{th} CNN that achieves the best performance from the boosting process
$\mathbf{D}_i$: distance feature vector in frame i
$\mathbf{D}'_i/\mathbf{D}''_i$: the first/second order derivatives of $\mathbf{D}_i$
Ft: the number of filters
J: total number of tracked joints
K: total number of action classes
$Kr/Pd/St$: kernel size/padding size/stride size
$L/\tilde{L}$: no. of distances/angles calculated from J joints
$\text{SFI}_{m,V}$: SFIs for evaluating CNNs in adaption process
$\text{SFI}_{m,n}$: SFIs for boosting CNNs in iteration n
$\text{SFI}^f_{m,n}$: a subset of $\text{SFI}_{m,n}$, which failed to be recognized by CNNs delivered from the last iteration
W: the span of the sliding window
$[L^{(k)}, U^{(k)}]$: the CI estimation of $p_l^{(k)}$
$a_{\tilde{l},i}$: the $\tilde{l}^{\text{th}}$ angle feature in frame i
$\mathbf{b}_{j,i}$: 3D coordinates of joint j at frame i
$d_{l,i}$: the l^{th} distance feature in frame i
i: index of sequential frames
i_s/i_e: the indices of the first/last frame of any SDI
Δi: the interval of frames for computing derivatives
j: index of tracked joints

to repeat the operation during the data collection. Each time of operation by a subject is considered as one experiment.

Microsoft Kinect, an infrared light sensor, is used for collecting data of individual workers in assembly operations at a frequency of 30 frames per second. The Kinect outputs the time series 3D coordinates of human body joints in a Euclidean space. The number of joints tracked in this study is J, and each joint has a unique index label, as Fig. 2 illustrates. Let i be the index of frames captured sequentially over time and j be the index of body joints. In any frame i, $\mathbf{b}_{j,i} = (x_{j,i}, y_{j,i}, z_{j,i})$ represents the 3D coordinates of joint j.

The assembly operation involves K classes of sequential actions, indexed by k. Therefore, the time series of body joint coordinates collected from each experiment are trimmed into K sequential segments with each pertaining to one and only

Table 1 continued

k: index of action class
$l/\tilde{l}$: index of distance/angle features
m: index of the classifiers
n: index of iterations for boosting
$p_t^{(k)}$: the k^{th} highest classification probability
$\overline{p}_{m,k}$: the weighted probability
$p_{m,k}$: prediction score of the m^{th} classifier on action class k
$r_{m,k}$:the recall value of CNN_m^* for action class k
$w_{m,k}$: the weight for $p_{m,k}$
$\widehat{y}$: the predicted action class
$\hat{y}_t$: classification of SDI_t
$\tilde{y}_t$: alternative classification of SDI_t
$\text{Tr}_{m,n}$: dataset for boosting CNN_m in iteration n
δ: overlap ratio between two successive SDIs
$\alpha_{m,n}$: classification accuracy of CNN_m on $\text{SFI}_{m,V}$
Γ_T/Γ_V: training/validation dataset for the adaptation
γ_0/γ_n: dataset used for TL/IB
ξ: binary filter

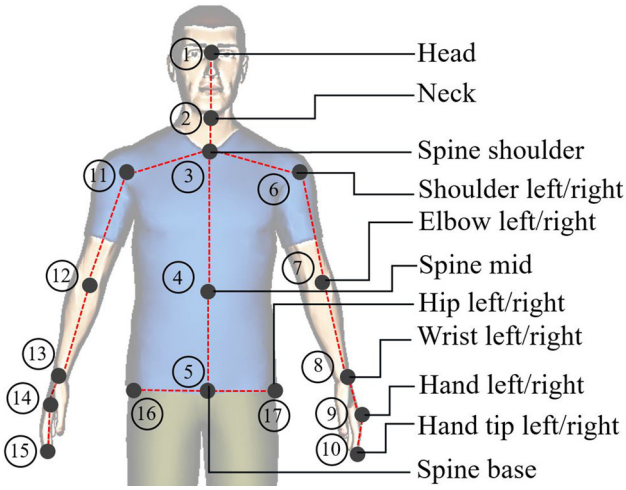

Fig. 2 Tracked body joints in this study

one action class. Then, a sliding window in a length of W frames is moving along the timeline at the stride size of δW frames to extract skeletal data pieces, named Skeletal Data Images (SDIs), from each time series segment. SDIs extracted from a segment are labeled with the action class of the segment. SDIs are in the size of $W \times J \times 3$ since each contains the 3D coordinates of the J joints over W successive frames. The selection of the window size W is crucial. SDIs with a very short time span lack sufficient information to capture features of the performed action. Those with a very long time span may contain data of more than one action. δ is chosen to be a positive decimal so that two successive SDIs extracted from a segment overlap with each other to capture their temporal connectivity.

Feature extraction from SDIs

This study calculates two categories of geometric features to capture the posture of workers in assembly. Derivatives of the features are further calculated to capture the temporal dynamics of the posture. Normalization is taken in calculating the features to make them invariant to the variations of the location and view of the Kinect in data collection and to the varied body size of subjects.

Geometric features of posture

Given J joints, $L = \binom{J}{2}$ joint-to-joint distances can be calculated, indexed by l. The distance features recorded over time form a time series of feature vector,

$$\mathbf{D}_i = \left[d_{1,i}, \ldots, d_{l,i}, \ldots, d_{L,i}\right], \ \forall i, \tag{1}$$

and the l^{th} distance, $d_{l,i}$, is calculated as

$$d_{l,i} = \|\mathbf{b}_{j,i} - \mathbf{b}_{j',i}\|_2 / \bar{d}_i, \tag{2}$$

where $j = \lceil l/J \rceil$ and $j' = l - \lfloor l/J \rfloor J$. $\bar{d}_i$ in Eq. (2) is the sum of three distances: the left shoulder (#11) to the right shoulder (#6), the spine shoulder (#3) to the spine mid (#4), and the spine mid to the spine base (#5),

$$\bar{d}_i = \|\mathbf{b}_{11,i} - \mathbf{b}_{6,i}\|_2 + \|\mathbf{b}_{3,i} - \mathbf{b}_{4,i}\|_2 + \|\mathbf{b}_{4,i} - \mathbf{b}_{5,i}\|_2, \tag{3}$$

which is used for normalizing the distance features.

Angle features are also calculated to supplement distance features. With J body joints, $\tilde{L} = J\binom{J-1}{2}$ angle features can be calculated, indexed by $\tilde{l}$. The angle features recorded over time form a time series of feature vector,

$$\mathbf{A}_i = \left[a_{1,i}, \ldots, a_{\tilde{l},i}, \ldots, a_{\tilde{L},i}\right], \ \forall i, \tag{4}$$

and the $\tilde{l}^{\text{th}}$ angle, $a_{\tilde{l},i}$, is calculated as

$$a_{\tilde{l},i} = \arccos \frac{(\mathbf{b}_{j',i} - \mathbf{b}_{j,i}) \cdot (\mathbf{b}_{j'',i} - \mathbf{b}_{j,i})}{|\mathbf{b}_{j',i} - \mathbf{b}_{j,i}\|_2 \cdot |\mathbf{b}_{j'',i} - \mathbf{b}_{j,i}\|_2}, \tag{5}$$

where $\mathbf{b}_{j,i}$, $\mathbf{b}_{j',i}$, and $\mathbf{b}_{j'',i}$ are three different body joints in frame i. $(j, j', j'') \rightarrow \tilde{l}$ is a bijection.

Temporal dynamics of the geometric features

The study calculates the first order and second-order derivatives of the distance features, respectively, to capture the temporal dynamics (linear speed and acceleration) of any subject's posture in the operation. The speed of distance change is approximated by the first-order difference equations:

$$\mathbf{D}'_i = \begin{cases} \dfrac{\mathbf{D}_{i+\Delta i} - \mathbf{D}_i}{\Delta i}, & \text{if } i_s \le i \le i_e - \Delta i \\ \dfrac{\mathbf{D}_i - \mathbf{D}_{i-\Delta i}}{\Delta i}, & \text{if } i_e - \Delta i < i \le i_e. \end{cases} \tag{6}$$

where i_s and $i_e (= i_s + W - 1)$ are the indices of the first frame and the last frame of any SDI, and Δi is the interval of frames for calculating changes in distance features.

The acceleration of distance change is approximated by the second-order difference equations:

$$\mathbf{D}''_i = \begin{cases} \dfrac{\mathbf{D}_i - 2\mathbf{D}_{i+\Delta i} + \mathbf{D}_{i+2\Delta i}}{\Delta i^2}, & \text{if } i_s \le i < i_s + \Delta i \\ \dfrac{\mathbf{D}_{i+\Delta i} - 2\mathbf{D}_i + \mathbf{D}_{i-\Delta i}}{\Delta i^2}, & \text{if } i_s + \Delta i \le i \le i_e - \Delta i \\ \dfrac{\mathbf{D}_i - 2\mathbf{D}_{i-\Delta i} + \mathbf{D}_{i-2\Delta i}}{\Delta i^2}, & \text{if } i_e - \Delta i < i \le i_e. \end{cases} \tag{7}$$

Angle related dynamic feature vectors, $\mathbf{A}'_i$ and $\mathbf{A}''_i$, are similarly calculated. Therefore, six Skeletal Feature Images (SFIs) are calculated from each SDI, and each SFI is one feature vector that spans W frames. The width of SFIs is equal to W and the height, denoted by H, is the dimension of the feature vector. H is equal to L for the distance related feature vectors and $\tilde{L}$ for the angle related feature vectors.

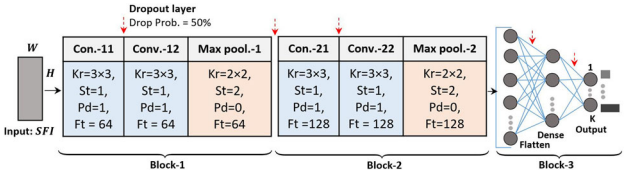

Fig. 3 The architecture of the proposed CNNs

Training CNNs for action recognition

This study trains six CNNs for recognizing worker actions in assembly. They respectively read one of the six SFIs extracted from a SDI to predict the action class of the SDI. The six CNNs share the same architecture illustrated in Fig. 3, which is composed of three blocks in sequence. Each of the first two blocks contains two convolutional layers followed by a max-pooling layer. The kernel size (Kr), stride size (St), padding size (Pd), and the number of filters (Ft) for each convolution and pooling operation are displayed in Fig. 3. A feature map is generated using the ReLU function from each of these layers. The last feature map generated by the second block is flattened and densified into a $1 \times K$ score vector in the third block, which is converted to a probabilistic prediction of the action class for the SDI using the softmax function. To prevent over-fitting, the dropout technique is applied to drop 50% neurons.

The six SFIs extracted from each SDI, indexed by m, are respectively entered into the six CNNs to generate six probabilistic predictions of the action class for the SDI. Let $\{p_{m,k}|k = 1, \ldots, K\}$ be the probabilistic prediction made by the CNN classifier that analyzes the mth SFI, where $p_{m,k}$ is the probability that the SDI would be action class k.

Adapting the CNNs to individual new workers

When new workers join the assembly line, the trained ActRgn CNN classifiers need to adapt to each of the new workers using transfer learning followed by iterative boosting. The approach is summarized in Algorithm 1 and discussed below.

For each new worker, the study collects a training dataset Γ_T for adapting the classifiers to the worker and a validation dataset Γ_V for evaluating the CNNs during the adaption process. The training dataset is split into a number of smaller mutually exclusive and collectively exhaustive subsets that each contains data from a few experiments:

$$\Gamma_T = \bigcup_{n=0}^{N} \gamma_n, \tag{8}$$

where γ_0 is used for transfer learning and γ_n is for the nth iteration of the boosting process, for $n = 1, \ldots, N$.

Algorithm 1 Adapting ActRgn CNN Classifiers to a New Worker

// Notations
m: index of the six classifiers, $\text{CNN}_m \in \{\text{CNN}_D, \text{CNN}_{D'}, \text{CNN}_{D''}, \text{CNN}_A, \text{CNN}_{A'}, \text{CNN}_{A''}\}$;
n: index of iterations for boosting, $n = 1, \ldots, N$;
$\{\text{CNN}_{m,n}\}$: CNNs obtained from iteration n;
$\{\text{SFI}_{m,V}\}$: SFIs for evaluating CNNs in adaption process;
$\{\text{SFI}_{m,n}\}$: SFIs for boosting CNNs in iteration n;
$\{\text{SFI}^f_{m,n}\}$: a subset of $\{\text{SFI}_{m,n}\}$, which failed to be recognized by CNNs delivered from the last iteration;
$\text{Tr}_{m,n}$: training dataset for boosting CNN_m in iteration n;
$\alpha_{m,n}$: classification accuracy of CNN_m on $\text{SFI}_{m,V}$.
// Initialization Through Transfer Learning
    $\{\text{CNN}_{m,0}\}$: CNNs obtained through transfer learning;
    $\text{Tr}_{m,0}(=\{\text{SFI}_{m,0}\})$: initial training dataset for boosting;
    $\alpha_{m,0}$: the accuracy of $\text{CNN}_{m,0}$ on $\{\text{SFI}_{m,V}\}$.
// Iterative Boosting
for $n = 1, \ldots, N$ **do**
    for any m **do**
        Evaluate $\text{CNN}_{m,n-1}$ on $\{\text{SFI}_{m,n}\}$ and obtain $\{\text{SFI}^f_{m,n}\}$;
        update training dataset: $\{\text{SFI}^f_{m,n}\} \cup \text{Tr}_{m,n-1} \rightarrow \text{Tr}_{m,n}$.
        Refine $\text{CNN}_{m,n-1}$ using $\text{Tr}_{m,n}$;
        update the classifier: $\text{CNN}_{m,n-1} \rightarrow \text{CNN}_{m,n}$;
        evaluate $\text{CNN}_{m,n}$ on $\{\text{SFI}_{m,V}\}$ to find $\alpha_{m,n}$.
    end for
end for
// Model Selection
for any m **do**
    find the best model $\text{CNN}_{m,n^*} \rightarrow \text{CNN}^*_m$ where $n^* = \text{argmax}_n\{\alpha_{m,n}\}$.
end for

Initial adaption by transferring learning

Transfer learning can adapt CNN classifiers a new worker who performs the same actions. The experimental study of this paper found that low- and medium-level features of assembly actions are well captured by the first two blocks of the ActRgn CNNs in Fig. 3, and distinct features of heterogeneous workers are mainly captured by the third block. Therefore, during transfer learning, the first two blocks of any CNN_m are frozen and the third block is retrained using the SFIs extracted from the training dataset γ_0, denoted as $\{\text{SFI}_{m,0}\}$. After the initial adaption through transfer learning, the classifiers become $\{\text{CNN}_{m,0}\}$, which are evaluated using the SFIs extracted from the validation dataset Γ_V, designated by $\{\text{SFI}_{m,V}\}$, to find their classification accuracy $\{\alpha_{m,0}\}$. A study by Al-Amin et al. (2020b) showed that an initial adaption is not sufficient for achieving a satisfactory result because of the within-subject variance.

Improving accuracy through iterative boosting

The performance of the initially adapted CNNs from transfer learning, $\{\text{CNN}_{m,0}\}$, can be further boosted iteratively. Let $\{\text{SFI}_{m,1}\}$ be the SFIs extracted from the training dataset γ_1 for the 1st iteration of boosting. $\{\text{CNN}_{m,0}\}$ are tested on $\{\text{SFI}_{m,1}\}$, and misclassified SFIs are denoted by $\{\text{SFI}^f_{m,1}\}$. The training dataset for the 1st iteration of boosting, $\text{Tr}_{m,1}$, is the union of $\{\text{SFI}^f_{m,1}\}$ and the initial training dataset $\text{Tr}_{m,0}=\{\text{SFI}_{m,0}\}$ that has been used for transfer learning. $\{\text{CNN}_{m,0}\}$ are refined with the updated training dataset $\text{Tr}_{m,1}$ to obtain the updated classifiers $\{\text{CNN}_{m,1}\}$. Evaluated on $\{\text{SFI}_{m,V}\}$, the performance of the boosted classifiers from this iteration, $\{\alpha_{m,1}\}$, is determined. This approach aims to boost the performance of classifiers by letting them learn from their weakness. This process continues for sufficient number of iterations to assure that a satisfied performance of the classifiers has been attained. The classifiers achieving the best performance from the boosting process are chosen as the final constituent classifiers of the ActRgn system for the worker, denoted by $\{\text{CNN}^*_m\}$. That is,

$$\text{CNN}^*_m = \text{CNN}_{m,n^*}, \text{ where } n^* = \text{argmax}_n\{\alpha_{m,n}\}. \tag{9}$$

The fusion of classifiers

The method to fuse the results of the six already adapted CNN classifiers is critical because it directly impacts the performance of the ActRgn system. This study proposes a fusion method, named Weighted Average of Selected Classifiers (WASC).

For a SDI, CNN^*_m classifies it as action class k with the probability $p_{m,k}$, for any class k. The six classifiers have unequal abilities to predict an action. Therefore, an individualized weight matrix is developed, where the element $w_{m,k}$ is the weight for $p_{m,k}$. Let $r_{m,k}$ be the recall value of CNN^*_m in recognizing action class k of the worker, obtained from evaluating $\{\text{SFI}_{m,V}\}$. $r_{m,k}$ measures the ability of CNN^*_m in recognizing action class k, which varies across the six classifiers. First, the study normalizes $r_{m,k}$'s across the six classifiers to determine the weight $w_{m,k}$:

$$w_{m,k} = r_{m,k} / \sum_m r_{m,k}, \ \forall k. \tag{10}$$

Then, the weighted probability of classification by CNN^*_m is:

$$\overline{p}_{m,k} = w_{m,k} \cdot p_{m,k}, \ \forall k. \tag{11}$$

For any action class k, a binary filter ξ is defined below to select classifiers that assign the largest weighted probability to it:

$$\xi(\overline{p}_{m,k}) = \begin{cases} 1, & \text{if } \overline{p}_{m,k} = \max\limits_k\{\overline{p}_{m,k}\}; \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Finally, the ActRgn system predicts the SDI as action class k with the probability p_k, which is the weighted average of the classification probabilities of selected classifiers:

$$p_k = \begin{cases} \frac{\sum_m \xi(\overline{p}_{m,k}) \cdot \overline{p}_{m,k}}{\sum_m \xi(\overline{p}_{m,k})}, & \text{if } \sum_m \xi(\overline{p}_{m,k}) > 0; \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

for any action class k. The final classification result by the ActRgn system, $\widehat{y}$, is:

$$\widehat{y} = \text{argmax}_k\{p_k\}. \tag{14}$$

Stream data analysis

Besides taking the designated actions in the assembly, workers may be idle or do something else for a variety of reasons such as loss of attention, fatigue, lack of knowledge or information, and so on. SDIs that are irrelevant to the actions for assembly are background SDIs. In processing stream data, the ActRgn system may read background SDIs and mistakenly recognize them as action SDIs. The ActRgn system may have mistakes in classifying action SDIs too. This study develops Algorithm 2 below that analyzes the probabilistic classification result of the ActRgn system in processing stream data to attempt to correct these two types of mistakes.

Algorithm 2 Stream Data Analysis

```
// Notations
{SDI_t | t = 1, 2, ...}: SDIs sequentially extracted from stream data
ŷ_t: classification of SDI_t
ỹ_t: alternative classification of SDI_t
p_t^(k): the k^th highest classification probability
[L^(k), U^(k)]: the CI estimation for the k^th highest classification probability of background SDIs
for t = 1, 2, ... do
    // Action Recognition
    The ActRgn system classifies SDI_t to yield ŷ_t and ỹ_t.
    // Background Detection
    if Σ_{(k)=1}^{4} 1{p_t^(k) ∈ [L^(k), U^(k)]} ≥ 2 then
        ŷ_t="Background";
        // Temporal Analysis
    else if t ≥ 2 & ỹ_t = ŷ_{t−1} then
        ŷ_t = ŷ_{t−1}.
    end if
end for
```

From testing the ActRgn system with background SDIs, it is noticed that the highest probability of classification is not dominantly high, and quite a few action classes (2∼4) are assigned with a non-trivial classification probability. This pattern of background SDIs is quite different than that of action SDIs, where one action class usually receives a dominantly high probability than other classes do. Accordingly, this study establishes a method to detect background SDIs from untrimmed data. For any SDI, let $p^{(k)}$ be the k^{th} highest classification probability. Using the probabilistic classification result of the background SDIs extracted from the validation dataset, a 99% confidence interval (CI) is established for the $(k)^{\text{th}}$ highest probability, named the $(k)^{\text{th}}$ CI and denoted as $[L^{(k)}, U^{(k)}]$, for $(k) = 1, \ldots, 4$. A SDI is classified as a background SDI and labeled as "Background" if two or more than two of the top four classification probabilities fall in their respective CI of classification probabilities for background SDIs.

Actions for assembly are sequential. Therefore, their temporal relationship may help correct some of the classification mistakes. If a SDI is classified as an action class different than that of the preceding SDI, this inconsistency may happen in a transition to the next action or it is a mistake. Observing an inconsistency, the temporal analysis yields an alternative classification result $\tilde{y}_t$, which is the action class with the second-highest probability. The assumption is that the alternative classification result may contain partial information of the SDI. If the alternative result is consistent with the classification of the preceding SDI, the temporal analysis considers the classification from the ActRgn system as a mistake and thus accepts the alternative classification result to rectify it. Otherwise, the classification result from the ActRgn system is accepted.

Illustrative example and assessment

Experiment design

To demonstrate and assess the proposed ActRgn system, this study analyzes one step in assembling the Bukito 3D printer in a lab setting, which is "putting on the handle". This step involves seven sequential actions (i.e., K=7) that are described in Fig. 4. A Microsoft Kinect is used to output the time series 3D coordinates of 17 body joints (i.e., J=17) displayed in Fig. 2 and the RGB images. The RGB images are annotated with corresponding frame numbers and are used as a reference for the data preparation described in the Methodology section.

15 subjects are recruited including both males and females. They are split into two mutually exclusive groups. The group CRT has 10 subjects who perform the assembly step for 10 times. Out of these, 8 times were used to create the dataset for training the base classifiers, whereas the remaining 2 times were for testing the classifiers. The group NEW has 5 subjects who represent new workers coming to perform the assembly. The group NEW repeats the assembly step for 40 times. Among these, 20 times are training data for adapting the classifiers to new workers through transfer learning and iterative boosting (i.e., Γ_T); 10 times are the

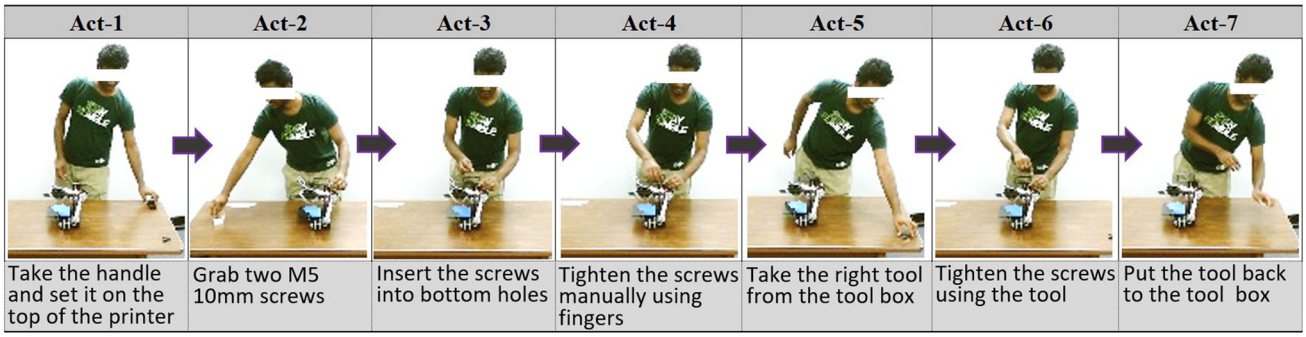

Fig. 4 The seven actions involved in the step "putting on the handle" for assembling the Bukito 3D printer

Table 2 Dataset summary: the size of SDIs in groups CRT and NEW

	Act-1	Act-2	Act-3	Act-4	Act-5	Act-6	Act-7	Total
CRT	176	280	340	544	362	1,012	496	3,210
NEW	688	1,018	1,270	1,860	1,460	3,758	1,320	11,374
Sub-1	118	184	234	376	268	672	210	2,062
Sub-2	110	152	208	324	240	684	208	1,926
Sub-3	114	192	236	318	300	772	290	2,222
Sub-4	178	230	304	386	300	722	308	2,428
Sub-5	168	260	288	456	352	908	304	2,736

data for evaluating the classifiers during the adaption process (i.e., Γ_V); the remaining 10 times are used to create the dataset for testing the proposed ActRgn system. Partial data can be accessed at Al-Amin et al. (2020a).

To extract SDIs from the time series of skeletal data, a sliding window in the length of 30 frames (i.e., W=30) and the stride size of 15 frames (i.e., δ=0.5) are used. That is, for every 0.5 seconds the ActRgn system reads the most recent 30 frames to classify the action of the worker during the past one second. Table 2 summarizes the distribution of SDIs in groups CRT and NEW.

The assembly operation mainly involves the 17 joints of the upper body shown in Fig. 2. Given 17 joints, $L = \binom{17}{2} = 136$ distance features can be calculated. Angles that can be formed by the 10 joints of the upper extremity (i.e., joints #6-#15) are calculated because the assembly operation mainly involves the worker's upper extremity. The 10 joints provide $\tilde{L} = 10\binom{9}{2} = 360$ angle features. Therefore, the dimension of the three distance related SFIs (i.e., SFI_D, $\text{SFI}_{D'}$, and $\text{SFI}_{D''}$) is 30×136. The dimension of three angle related SFIs (i.e., SFI_A, $\text{SFI}_{A'}$, and $\text{SFI}_{A''}$) is 30×360. SFIs are all normalized to take values within $[-1,1]$ before being used for training, validating, and testing the ActRgn CNN classifiers.

In training the ActRgn CNN classifiers, the adaptive learning rate optimizer (Adam) along with the cross-entropy loss function is used. Initially, the learning rate of Adam is set as 0.001. It is dynamically decreased over iterations. To avoid the issue of overfitting, L2 and dropout regularization are implemented.

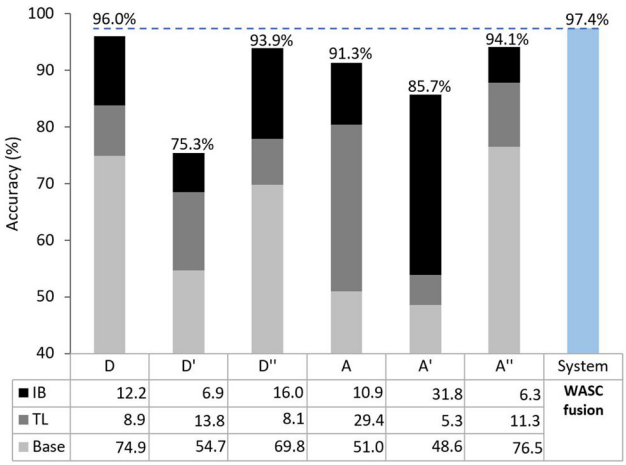

Fig. 5 ActRgn accuracy achieved by the individualized ActRgn system and its constituent classifiers: An illustrative example (the new worker Sub-1)

An illustrative example

Using a worker in the group NEW (Sub-1) as an example, the ActRgn system individualized for this worker achieves 97.4% accuracy on the testing dataset. Figure 5 describes how this ActRgn performance is achieved. The six base classifiers,

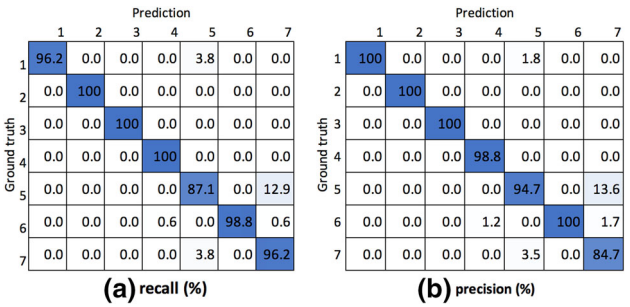

Fig. 6 The class-level performance of the ActRgn system individualized for the new worker Sub-1: **a** recall matrix and **b** precision matrix

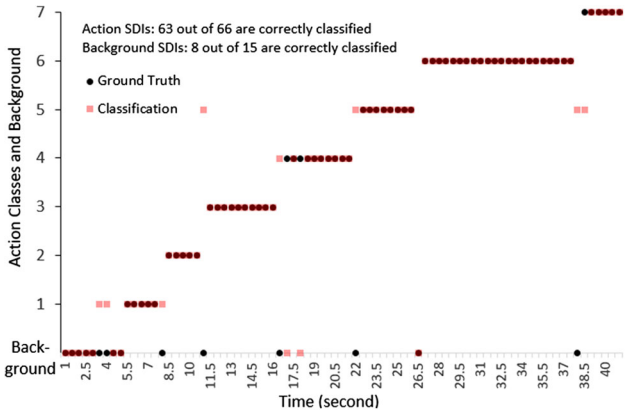

Fig. 7 An illustration of the stream data analysis

trained on the dataset of group CRT, achieve an accuracy ranging from 48.6% (classifier A') to 76.5% (classifier A'') in recognizing the actions of this worker. The performances of the base classifiers are far below the satisfaction and vary largely. Transfer Learning (TL) is implemented for achieving an initial adaption of the base classifiers to the worker, which increases the accuracy of the six classifiers by 5.3% (classifier A') to 29.4% (classifier A). Then, the initially adapted classifiers are further improved through Iterative Boosting (IB), achieving an accuracy ranging from 75.3% (classifier D') to 96.0% (classifier D). While the ActRgn accuracy of the adapted classifiers still varies largely, four out of six classifiers have an accuracy higher than 90%. The ActRgn system, as an ensemble of the already adapted classifiers using the WASC fusion, achieves 97.4% accuracy, higher than the accuracy of any constituent classifier of the system by 1.4% (compared to classifier D) to 22.1% (compared to classifier D').

The study further reviews the recall and precision matrices in Fig. 6 to determine the class-level performance of the ActRgn system. For worker Sub-1, the SDIs of classes 2, 3, and 4 are perfectly recognized; the SDIs of classes 1, 6, and 7 SDIs are recognized with a recall value greater than 96%; only class 5 has a relatively low recall value, 87.1%. The incorrectly classified class 5 SDIs are all recognized as class 7, which is a major reason for the low precision for class 7. Some action classes share a certain similarity, thus causing confusions. For example, action classes 5 and 7 all involve taking the tool from, or returning it to, a similar location using the same hand. The confusion matrix is found to vary among the tested subjects in this study, because different workers may perform the same action in a slightly different way.

Figure 7 illustrates the result of the ActRgn system in analyzing the untrimmed stream data of the new worker Sub-1 in an experiment. The experiment lasts 41 seconds and 81 SDIs in total are extracted from the stream data, with 66 action SDIs (action classes 1 to 7) and 15 background SDIs (labeled as "Background"). 63 out of 66 (95.5%) action SDIs are correctly recognized, and so 8 out of 15 (53.3%) background SDIs. Among the 7 misclassified background SDIs, 5 SDIs occur during the transition from one action to the next action and other 2 SDIs are 1.5∼2 seconds before Act-1 takes place. Moreover, 5 out of the 7 misclassified background SDIs are classified as the preceding or the succeeding action. This is due to the fact that SDIs during the transition may contain data either from the preceding or the succeeding action.

Comparison of basic ActRgn classifiers

This study chose CNN as the basic classifier for building the ActRgn system. To verify the rationale of choosing CNN, this study compares the performances of CNN, MLP, and LSTM as the basic classifiers. Here, both the SDIs and the six types of SFIs are considered as the input to the classifiers. The experimental result in Table 3 shows that CNN outperforms MLP and LSTM in analyzing any of the seven inputs on the CRT testing dataset. The result on the NEW testing dataset is similar except for one exception; that is, the accuracy of CNN in analyzing $\text{SFI}_{D'}$ is 0.56% lower than that of LSTM. This comparative study verifies the advantage of using CNN as the underlying classifiers for building the ActRgn system.

Advantages of the ActRgn system architecture

This study proposes extracting six posture related feature vectors to respectively create six 1-channel (1-C) ActRgn CNN classifiers and then fusing them as an ActRgn system. This architecture is based on two hypotheses. On one hand, each of the six feature vectors conveys unique information of actions to independently support action recognition to a certain degree. On the other hand, the six individual classifiers have complementary strengths. To demonstrate its advantage, the proposed system architecture is compared to a single CNN built on SDIs (i.e., the raw data) and a system of two 3-channel CNNs with one built with the three distance related features [SFI_D, $\text{SFI}_{D'}$, $\text{SFI}_{D''}$] and the other built with the

Table 3 The accuracy (%) of CNN, MLP, and LSTM as the basic classifiers, on the CRT and NEW testing datasets, respectively

		CRT			NEW		
		CNN	MLP	LSTM	CNN	MLP	LSTM
Classifiers	SDIs	76.26	74.33	70.03	66.30	59.33	51.56
	SFI_D	83.53	60.39	76.71	70.40	57.49	64.34
	$SFI_{D'}$	77.30	51.78	71.51	54.30	44.36	54.86
	$SFI_{D''}$	82.49	60.09	69.44	67.40	56.32	59.79
	SFI_A	80.56	55.64	57.99	61.50	53.69	46.88
	$SFI_{A'}$	78.48	50.89	49.26	51.40	47.30	47.30
	$SFI_{A''}$	81.15	52.52	71.66	62.90	51.53	59.26

three angle related features [SFI_A, $SFI_{A'}$, $SFI_{A''}$]. Table 4 shows the ActRgn accuracy of the raw data CNN, the six 1-channel (1-C) CNNs, the two 3-channel CNNs, and the 1-C and 3-C ActRgn systems respectively built with four fusion methods—maximum, average, product, and majority voting (Maj Vot). All CNNs in Table 4 are base classifiers that have not been individualized yet. They are firstly tested on the CRT testing dataset and then on the NEW testing dataset to verify the challenge of worker heterogeneity on action recognition. When tested on the group of new workers, both the subject-level and the group-level accuracy are provided.

In recognizing the actions of any group or any individual in Table 4, at least one 1-C CNN outperforms the raw data CNN, and at least one 3-C CNN is as good as, or better than, the raw data CNN. For example, the six 1-C CNNs and the two 3-C CNNs all outperform the raw data CNN in recognizing the actions of Sub-3. For Sub-2, the 1-C CNN built with $SFI_{A''}$ is better than the raw data CNN and the 3-C CNN with [SFI_A, $SFI_{A'}$,, $SFI_{A''}$] is as good as the raw data CNN. The observation suggests that some of the feature vectors calculated from SDIs convey information more useful for action recognition than SDIs.

No single feature-based classifier dominates other classifiers for all the five individual subjects. For example, the best 1-C CNN for Sub-1 is the classifier built with $SFI_{A''}$ and the best 3-C CNN is the classifier built with [SFI_D, $SFI_{D'}$,, $SFI_{D''}$]. But for Sub-3, the best 1-C CNN is the classifier with SFI_A and the best 3-C CNN is the one built with [SFI_A, $SFI_{A'}$,, $SFI_{A''}$]. Therefore, a system of fused classifiers is more robust to worker heterogeneity than an individual classifier.

Both the 1-C CNN system and the 3-C CNN system can be built through the maximum fusion, average fusion, and product fusion. The 1-C CNN system can also be built through the majority voting. These seven configurations have varied performance, displayed in the seven rows from the bottom in Table 4. At the group level, the seven configurations all outperform the raw data CNN. At the individual level, all the seven configurations are better than the raw data CNN in recognizing the actions of Sub-1, -3, and -4. For Sub-2, the 1-C CNN systems with the average fusion, product fusion, and majority voting outperform the raw data CNN. For Sub-5, the 1-C CNN systems with the average fusion and product fusion achieve higher accuracy than the raw data CNN. This comparison further confirms the advantage of posture-related feature images over the raw data images.

At least one fusion method can build a 1-C CNN system better than all the constituent classifiers, but this is not true for the 3-C CNN system. For example, no 3-C CNN system outperforms all its constituent classifiers for Sub-1 and Sub-2. This indicates the fusion of six 1-C CNN classifiers is a better system architecture than the fusion of two 3-C CNN classifier. The 1-C CNN system indeed is a better ActRgn system than the 3-C CNN system, evidenced by the comparison in Table 4. When testing them on the group CRT, the 1-C CNN system always achieves higher accuracy than the 3-C CNN system, ranging from 5.7% (maximum fusion) to 8.8% (average fusion). Similarly, on the group NEW, the 1-C CNN system has higher accuracy than the 3-C CNN system built with any fusion method. The improvement is up to 6.1% (product fusion). At the individual level, the 1-C CNN system outperforms the 3-C CNN system built with any fusion method, with only one noticed exception. That is, the 1-C CNN system for Sub-2 is outperformed by the 3-C CNN system if the maximum fusion is used. The comparative study in Table 4 supports the use of the proposed ActRgn CNN architecture of this paper.

Effectiveness of transfer learning and iterative boosting

Table 5 evaluates and confirms the effectiveness of transfer learning and iterative boosting for adapting the six classifiers to individual workers. The group-level assessment is also provided at the bottom. The ActRgn accuracy before the adaption (Bfr) is the accuracy of the base classifiers in Table 4. Then, increment due to transfer learning (TL) and that from iterative boosting (IB) are determined. The accu-

Table 4 The accuracy (%) of ActRgn systems and their constituent CNN classifiers, respectively tested on groups CRT and NEW

			CRT	NEW	Sub-1	Sub-2	Sub-3	Sub-4	Sub-5
Classifiers	SDIs		76.3	66.3	74.9	67.0	51.2	70.3	66.4
	SFI_D		83.5	70.4	74.9	63.8	64.8	80.0	68.2
	$SFI_{D'}$		77.3	54.3	54.7	51.7	53.8	56.0	54.5
	$SFI_{D''}$		82.5	67.4	69.8	64.4	61.3	76.8	64.5
	SFI_A		80.6	61.5	51.0	65.1	65.3	70.2	56.1
	$SFI_{A'}$		78.5	51.4	48.6	46.5	51.4	58.5	50.6
	$SFI_{A''}$		81.2	62.9	76.5	67.7	56.6	58.0	59.6
	$[SFI_D, SFI_{D'}, SFI_{D''}]$		78.2	67.3	77.7	65.3	54.7	74.5	65.8
	$[SFI_A, SFI_{A'}, SFI_{A''}]$		77.6	62.9	73.7	67.0	55.4	63.5	58.3
Systems	max	1-C	85.5	70.6	75.9	64.9	67.9	78.0	66.5
		3-C	79.8	67.2	74.7	66.8	57.3	73.2	65.1
	Average	1-C	87.7	73.5	82.8	69.0	67.5	80.0	69.0
		3-C	78.9	67.7	77.1	66.8	57.6	74.0	64.5
	Product	1-C	87.8	73.8	82.2	70.1	67.0	80.7	70.0
		3-C	79.2	67.7	77.1	66.4	57.8	73.7	65.1
	Maj Vot	1-C	88.4	70.7	75.7	68.6	65.5	79.0	65.5

racy after the adaption (Aft) and the corresponding change in accuracy (chg) due to the classifier adaption are calculated too.

Using the new worker Sub-1 as an example, transfer learning increases the ActRgn accuracy by an amount ranging from 5.3% (classifier of $SFI_{A'}$) to 29.4% (classifier of SFI_A). Iterative boosting contributes an additional 6.3% (classifier of $SFI_{A''}$) to 31.8% (classifier of SFI'_A). The classifier adaption improves the accuracy by 17.6% (classifier of $SFI_{A''}$) to 40.3% (classifier of SFI_A). Helped by transfer learning and iterative boosting, the classifier D has reached 96% accuracy, becoming the best 1-C CNN classifier for Sub-1. The classifier D', though being the least capable classifier for Sub-1, still reaches 75.3% accuracy. Both transfer learning and iterative boosting effectively adapt the six base classifiers to other new workers too. Yet their contributions vary among workers and classifiers. The most improved classifier is the classifier of SFI_A for Sub-1, whose accuracy is increased by 40.3%. The least improved classifier is the classifier of SFI_D for Sub-4, with an increase of 12.8%. At the group level, the classifier adaption increases the ActRgn accuracy by 17.8% to 25.1%. Among the six already adapted classifiers, the classifiers built on the velocity features (i.e., $SFI_{D'}$ and $SFI_{A'}$) are usually not among the top classifiers in terms of accuracy.

Effectiveness of the WASC fusion

To verify the advantage of the proposed Weighted Average of Selected Classifiers (WASC) fusion, it is compared to other five methods: maximum, product, average, majority voting (Maj Vot), and weighted average (Wgt Avg). The weighted average fusion is similar to the proposed WASC fusion except that it does not implement the filter in Eq. (12) to select classifiers. The comparative study is performed at both the individual level and the group level, with the result summarized in Table 6 and visualized in Fig. 8. Figure 8 shows only the weighted average fusion and the WASC fusion outperform all the constituent classifiers across the five tested workers. This verifies the helpfulness of discriminating individual classifiers by their strength at the class level. Furthermore, the WASC is better than the weighted average fusion for recognizing the actions of every individual in the group NEW, increasing the accuracy by 0.4% (Sub-1) to 2.5% (Sub-5). The higher accuracy of the WASC method over the weighted average method confirms the unique strength of the WASC fusion.

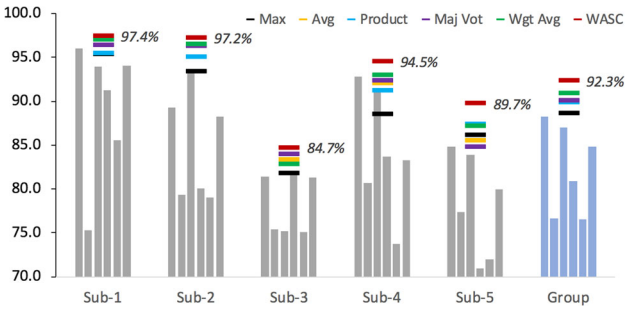

Fig. 8 A comparison of fusion methods. The accuracy of constituent classifiers is indicated by grey columns (for individuals) and blue columns (for the group); the accuracy of ActRgn systems built with different fusion methods is represented by bars on top of them

Table 5 Effectiveness of transfer learning (TL) and iterative boosting (IB) for adapting the constituent classifiers: the before (Bfr)-after (Aft) comparison of ActRgn accuracy on group NEW

		SFI_D	$SFI_{D'}$	$SFI_{D''}$	SFI_A	$SFI_{A'}$	$SFI_{A''}$	Median	Min	Max
Sub-1	Bfr	74.9	54.7	69.8	51.0	48.6	76.5	62.2	48.6	76.5
	TL	8.9	13.8	8.1	29.4	5.3	11.3	10.1	5.3	29.4
	IB	12.2	6.9	16.0	10.9	31.8	6.3	11.5	6.3	31.8
	Aft	96.0	75.3	93.9	91.3	85.6	94.1	92.6	75.3	96.0
	chg	21.1	20.7	24.1	40.3	37.0	17.6	22.6	17.6	40.3
Sub-2	Bfr	63.8	51.7	64.4	65.1	46.5	67.7	64.1	46.5	67.7
	TL	11.6	15.3	4.8	1.8	7.9	6.8	7.3	1.8	15.3
	IB	14.0	12.2	24.2	13.3	24.7	13.8	13.9	12.2	24.7
	Aft	89.3	79.3	93.5	80.1	79.0	88.2	84.2	79.0	93.5
	chg	25.6	27.5	29.1	15.1	32.5	20.5	26.5	15.1	32.5
Sub-3	Bfr	64.8	53.8	61.3	65.3	51.4	56.6	58.9	51.4	65.3
	TL	8.3	5.4	11.6	8.9	9.2	16.2	9.0	5.4	16.2
	IB	8.3	16.2	2.3	7.5	14.5	8.5	8.4	2.3	16.2
	Aft	81.4	75.4	75.2	81.6	75.1	81.3	78.3	75.1	81.6
	chg	16.7	21.5	13.9	16.3	23.7	24.7	19.1	13.9	24.7
Sub-4	Bfr	80.0	56.0	76.8	70.2	58.5	58.0	64.3	56.0	80.0
	TL	6.0	12.8	0.3	2.3	4.0	18.8	5.0	0.3	18.8
	IB	6.8	11.8	14.2	11.2	11.2	6.5	11.2	6.5	14.2
	Aft	92.8	80.7	91.3	83.7	73.7	83.3	83.5	73.7	92.8
	chg	12.8	24.7	14.5	13.5	15.2	25.3	14.8	12.8	25.3
Sub-5	Bfr	68.2	54.5	64.5	56.1	50.6	59.6	57.8	50.6	68.2
	TL	6.8	6.8	7.8	4.5	8.1	8.1	7.3	4.5	8.1
	IB	9.9	16.1	11.6	10.3	13.3	12.3	12.0	9.9	16.1
	Aft	84.8	77.4	83.9	70.9	72.0	80.0	78.7	70.9	84.8
	chg	16.6	22.9	19.4	14.8	21.5	20.4	19.9	14.8	22.9
Group	Bfr	70.4	54.3	67.4	61.5	51.4	62.9	62.2	51.4	70.4
	TL	8.6	10.4	6.6	8.9	6.9	12.4	8.8	6.6	12.4
	IB	9.2	12.0	13.1	10.4	18.2	9.5	11.2	9.2	18.2
	Aft	88.3	76.6	87.0	80.9	76.5	84.8	82.8	76.5	88.3
	chg	17.8	22.3	19.6	19.3	25.1	21.9	20.7	17.8	25.1

Table 6 Comparing the accuracy (%) of ActRgn Systems built with different classifier fusion methods, tested on the group NEW

		Sub-1	Sub-2	Sub-3	Sub-4	Sub-5	Group
Classifiers	SFI_D	96.0	89.3	81.4	92.8	84.8	88.3
	$SFI_{D'}$	75.3	79.3	75.4	80.7	77.4	76.6
	$SFI_{D''}$	93.9	93.5	75.2	91.3	83.9	87.0
	SFI_A	91.3	80.1	81.6	83.7	70.9	80.9
	$SFI_{A'}$	85.6	79.0	75.1	73.7	72.0	76.5
	$SFI_{A''}$	94.1	88.2	81.3	83.3	80.0	84.8
Fusion methods	Maximum	95.3	93.4	81.8	88.5	86.1	88.6
	Product	95.5	95.0	82.8	91.2	87.4	89.9
	Average	96.4	95.0	83.3	92.0	85.5	89.9
	Maj Vot	96.4	96.3	83.9	92.3	84.8	90.1
	Wgt Avg	97.0	96.5	82.8	93.0	87.2	90.9
	WASC	97.4	97.2	84.7	94.5	89.7	92.3

Table 7 The accuracy (%) in detecting the background and recognizing actions by the stream data analysis: a before (Bfr) - after (Aft) comparison

		All	Bdg	Acts	Act-1	Act-2	Act-3	Act-4	Act-5	Act-6	Act-7
Sub-1	Bfr (%)	85.4	0.0	97.2	100.0	98.4	100.0	100.0	95.1	100.0	78.9
	Aft (%)	93.4	68.7	96.9	95.3	96.7	97.4	99.0	97.6	100.0	82.5
	Size	687	83	604	43	61	78	100	82	183	57
Sub-2	Bfr (%)	84.2	0.0	94.4	94.0	98.1	100.0	86.0	85.1	100.0	91.1
	Aft (%)	90.9	46.5	96.3	92.0	98.1	100.0	76.0	94.6	99.5	91.1
	Size	658	71	587	50	54	72	93	74	188	56
Sub-3	Bfr (%)	78.5	0.0	86.0	78.3	95.1	91.7	83.6	87.1	95.1	65.0
	Aft (%)	85.2	46.3	88.9	73.9	95.1	90.5	86.9	95.0	97.8	70.9
	Size	768	67	701	46	61	84	122	101	184	103
Sub-4	Bfr (%)	86.2	0.0	94.3	89.5	98.4	94.7	96.7	91.4	98.5	84.3
	Aft(%)	90.7	41.8	95.3	80.7	100.0	97.9	95.1	96.8	98.0	91.0
	Size	784	67	717	57	61	95	122	93	200	89
Sub-5	Bfr (%)	82.0	0.0	90.5	90.7	92.7	93.9	89.0	90.4	88.8	92.3
	Aft (%)	88.6	38.6	93.8	87.0	92.7	94.9	89.0	93.3	96.8	97.4
	Size	876	83	793	54	82	98	127	104	250	78
Group	Bfr (%)	83.2	0.0	92.2	90.4	96.2	95.8	91.0	89.9	96.0	80.9
	Aft (%)	89.6	48.8	94.1	85.6	96.2	96.0	92.4	95.2	98.3	85.6
	Size	3773	371	3402	250	319	427	564	454	1005	383

Stream data analysis

This paper performs a before-after study to demonstrate that Algorithm 2 can improve the results of the ActRgn system in analyzing stream data. Table 7 computes the accuracy (%) in recognizing all the testing SDIs (All), background SDIs (Bdg), action SDIs (Acts), and individual action classes (Act-1,…, -7). The study is performed using the untrimmed stream test data of the NEW group. At both the subject level and the group level, the comparison shows the ActRgn accuracy before implementing Algorithm 2 (Bfr) and after (Aft). The volume of testing SDIs (size) for the accuracy calculation is provided too as a reference. Although the ActRgn system has a good capability in classifying action SDIs, its accuracy in recognizing background SDIs is zero. This is because the ActRgn system is designed to classify SDIs into one of the seven action classes.

Using the new worker Sub-1 as an example, the accuracy in recognizing action SDIs is 97.2% whereas the accuracy in recognizing background SDIs is 0%. The accuracy in recognizing all the SDIs extracted from the stream data is only 85.4%. After implementing Algorithm 2, the accuracy in recognizing background SDIs is 68.7% and the accuracy in recognizing actions drops about 0.3%, to 96.9%. The accuracy in analyzing the stream data, including both the background and action SDIs, is 93.4%, which is an 8% increase compared to the accuracy before implementing the stream data analysis algorithm. For the other four subjects, the improved accuracy in analyzing the stream data is 6.7% (=90.9-84.2), 6.7% (=85.2-78.5), 4.5% (=90.7-86.2), and 6.6% (=88.6-82.0), respectively. At the group level, the accuracy is improved by 6.4%, from 83.2% to 89.6%.

The capability of the stream data analysis algorithm in detecting the background SDIs varies among the tested subjects. The accuracy is the highest in detecting the background SDIs of Sub-1 (68.7%) and it is the lowest for Sub-5 (38.6%). This variation is mainly caused by the heterogeneity of individual workers. At the group level, the algorithm detected 48.8% of background SDIs correctly, and other background SDIs are recognized as actions. The background SDIs recognized as actions are mainly in the transition of two successive actions.

Although the accuracy in recognizing actions drops about 0.3% for Sub-1, the accuracy in recognizing the actions for the other four subjects increases. Therefore, at the group level the accuracy in recognizing actions increases 1.8%, from 92.2% to 94.1%. This indicates the temporal analysis in Algorithm 2 helps improve the accuracy in recognizing worker actions. By comparing the change in accuracy at the class level in Table 7, it is noticed that the accuracy improvement for some classes may be at the cost of lowering the accuracy of others. Using the new worker Sub-5 as an example, the temporal analysis increases the accuracy in recognizing Act-3, -5 -6, and -7, but decreases the accuracy in recognizing Act-1.

Table 8 The impact of occlusion on the accuracy (%) in recognizing actions and background from stream data

Level	No.	Duration (sec)	STA	All	Bdg	Acts
Partial	2	3	Bfr	80.64	0.0	91.72
			Aft	81.66	19.28	90.23
		6	Bfr	73.65	0.0	83.77
			Aft	74.09	28.92	80.30
	3	3	Bfr	76.86	0.0	87.42
			Aft	78.31	15.66	86.92
		6	Bfr	70.45	0.0	80.13
			Aft	70.60	8.43	79.14
Full	2	3	Bfr	72.93	0.0	82.95
			Aft	73.36	16.88	81.12
		6	Bfr	57.93	0.0	65.89
			Aft	58.66	12.04	65.07
	3	3	Bfr	65.65	0.0	74.67
			Aft	67.54	15.66	74.67
		6	Bfr	43.38	0.0	49.34
			Aft	43.96	10.84	48.51

Impact of occlusion

Occlusion is one of the key challenges for skeletal data-based human action recognition. There are a few crucial factors that vary the impact of occlusion. To what extent does occlusion impact the skeletal data-based action recognition? To study this problem, an experimental study is performed to evaluate the impact of occlusion. In the design of experiments, three factors with two levels of each have been considered:

- The level of occlusion (Level): partial (the left hand) vs. full (the entire human body)
- The number of occlusions in each time of operation (No.): 2 vs. 3
- The duration of each occlusion (Duration (sec)): 3 vs. 6

to investigate the impact occlusion.

The untrimmed streaming data of Sub-1 from the NEW group are assessed in the above-said eight experiments ($=2^3$). The data comprise 687 SFIs in total, with 604 action SFIs and 83 background SFIs, obtained from ten times of operation by the subject. Table 8 summarizes the accuracy in recognizing actions, the background, and all of them, before (Bfr) and after (Aft) the stream data analysis (SDA) is applied to the fusion result from the Weighted Average of Selected Classifier (WASC). From the table is can be seen that:

- The impact of full occlusion is more severe than partial occlusion. When a full occlusion happens, input SFIs contain no skeletal data. The ActRgn system lacks the ability to recognize them, which is reflected by the reduced overall accuracy, the accuracy in detecting the background (fully blocked SFIs reduce the ability of STA), and the accuracy in action recognition.
- When a partial occlusion happens, missing features in the affected SFIs are estimated as the average of the remaining features. The ActRgn system can still recognize some of the partially occluded SFIs.
- The accuracy declines when the number of occlusions and/or the duration of occlusion increase.
- Occlusion impairs the ability of stream data analysis on top of WASC. But the stream data analysis still helps improve the overall performance slightly, mainly due to its ability to detect the background.

Conclusion and future work

This paper proposes a system of skeletal data-based CNN classifiers for action recognition, which is individualized for heterogeneous workers to recognize their actions in assembly reliably. The paper demonstrates the advantage of the proposed system architecture that computes posture-related feature vectors using the skeletal data extracted from depth images, builds the constituent classifiers using individual feature vectors, and fuses them to become a system. The study further verifies the importance of individualizing the system for heterogeneous workers, which adapts the ActRgn system to individual workers through transfer learning, iterative boosting, and the WASC fusion method. The algorithm of stream data analysis not only improves the accuracy of the

individualized system in recognizing the worker's actions but differentiate background data and actions to some extent.

The paper builds a foundation for important extensions and future explorations. The revision or an update of the assembly process usually introduces additional actions. Adding new action classes to an existing ActRgn system is an important research problem. We plan on exploring this problem by developing a class incremental learning (Class-IL) strategy (Tao et al. 2020). When the classifiers learn to recognize new action classes, the classifiers might suffer from catastrophic forgetting, which is a long-standing challenge in class-IL that tends to override the previous classes when confronted with new classes. To address this challenge, multiple crucial components of a class-IL algorithm will be explored: including a memory buffer to store a few exemplars of old classes, a constraint on keeping previous knowledge in learning new classes, and a learning system that balances old and new classes (Mittal et al. 2021). Another immediate extension of the study is to adapt the ActRgn system to existing workers performing new tasks and then to new workers performing new tasks. This extension is critical to the scale-up of system implementation. While transfer learning and iterative boosting effectively adapt the constituent classifiers to individuals, a faster adaption is desired to accommodate the quickly changing, highly unpredictable condition of future manufacturing. A hypothesis is that layers of neural networks to refine is dependent on the new subject or new task that the system will adapt to. A method that can optimize the classifier refining process is needed. The current stream data analysis algorithm can detect some background data, but not all of them. The ability to detect background in high accuracy and to differentiate different types of background data is useful in applications. These exciting opportunities call for future research.

Acknowledgements All the authors of this paper received financial support from the National Science Foundation through the Award CMMI-1646162. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Al-Amin, M., Qin, R., Moniruzzaman, M., Yin, Z., Tao, W., & Leu, M. C. (2020). Data for the individualized system of skeletal data-based CNN classifiers for action recognition in manufacturing assembly.

Al-Amin, M., Qin, R., Tao, W., Doell, D., Lingard, R., Yin, Z., & Leu, M. C. (2020). Fusing and refining convolutional neural network models for assembly action recognition in smart manufacturing. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, page NA.

Al-Amin, M., Tao, W., Doell, D., Lingard, R., Yin, Z., Leu, M. C., et al. (2019). Action recognition in manufacturing assembly using multimodal sensor fusion. *Procedia Manufacturing*, *39*, 158–167.

Banos, O., Damas, M., Pomares, H., Rojas, F., Delgado-Marquez, B., & Valenzuela, O. (2013). Human activity recognition based on a sensor weighting hierarchical classifier. *Soft Computing*, *17*(2), 333–343.

Chen, C., Jafari, R., & Kehtarnavaz, N. (2017). A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, *76*(3), 4405–4425.

Chernbumroong, S., Cang, S., & Yu, H. (2015). Genetic algorithm-based classifiers fusion for multisensor activity recognition of elderly people. *IEEE Journal of Biomedical and Health Informatics*, *19*(1), 282–289.

Chung, S., Lim, J., Noh, K. J., Kim, G., & Jeong, H. (2019). Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning. *Sensors*, *19*(7), 1716.

Cook, D., Feuz, K. D., & Krishnan, N. C. (2013). Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, *36*(3), 537–556.

Du, Y., Fu, Y., & Wang, L. (2015). Skeleton based action recognition with convolutional neural network. In 3rd IAPR Asian conference on pattern recognition (ACPR), pp. 579–583.

ElMaraghy, H., & ElMaraghy, W. (2016). Smart adaptable assembly systems. *Procedia CIRP*, *44*, 4–13.

Guo, M., Wang, Z., Yang, N., Li, Z., & An, T. (2019). A multisensor multiclassifier hierarchical fusion model based on entropy weight for human activity recognition using wearable inertial sensors. *IEEE Transactions on Human-Machine Systems*, *49*(1), 105–111.

Guo, Y., He, W., & Gao, C. (2012). Human activity recognition by fusing multiple sensor nodes in the wearable sensor systems. *Journal of Mechanics in Medicine and Biology*, *12*(05), 1250084.

Han, Y., Chung, S. L., Chen, S. F., & Su, S. F. (2018) Two-stream LSTM for action recognition with RGB-D-based hand-crafted features and feature combination. In IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3547–3552. IEEE.

Hou, Y., Li, Z., Wang, P., & Li, W. (2018). Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(3), 807–811.

Kamel, A., Sheng, B., Yang, P., Li, P., Shen, R., & Feng, D. D. (2019). Deep convolutional neural networks for human action recognition using depth maps and postures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(9), 1806–1819.

Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., et al. (2018). T-CNN: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 2896–2907.

Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017). A new representation of skeleton sequences for 3D action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3288–3297. IEEE.

Khaire, P., Kumar, P., & Imran, J. (2018). Combining CNN streams of RGB-D and skeletal data for human activity recognition. *Pattern Recognition Letters*, *115*, 107–116.

Kong, X. T., Luo, H., Huang, G. Q., & Yang, X. (2019). Industrial wearable system: The human-centric empowering technology in Industry 4.0. *Journal of Intelligent Manufacturing*, *30*(8), 2853–2869.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Li, B., Li, X., Zhang, Z., & Wu, F. (2019). Spatio-temporal graph routing for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 8561–8568.

Li, C., Wang, P., Wang, S., Hou, Y., & Li, W. (2017). Skeleton-based action recognition using LSTM and CNN. In IEEE International conference on multimedia and expo workshops (ICMEW), pp. 585–590. IEEE.

Liu, J., Shahroudy, A., Xu, D., Kot, A. C., & Wang, G. (2017). Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 3007–3021.

Mittal, S., Galesso, S., & Brox, T. (2021). Essentials for class incremental learning. arXiv preprint arXiv:2102.09517.

Moniruzzaman, M., Yin, Z., He, Z. H., Qin, R., & Leu, M. (2021). Human action recognition by discriminative feature pooling and video segmentation attention model. *IEEE Transactions on Multimedia*.

Nunez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S., & Velez, J. F. (2018). Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, *76*, 80–94.

Pham, H. H., Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S. A. (2018). Exploiting deep residual networks for human action recognition from skeletal data. *Computer Vision and Image Understanding*, *170*, 51–66.

Rude, D. J., Adams, S., & Beling, P. A. (2018). Task recognition from joint tracking data in an operational manufacturing cell. *Journal of Intelligent Manufacturing*, *29*(6), 1203–1217.

Shen, C., Chen, Y., Yang, G., & Guan, X. (2020). Toward hand-dominated activity recognition systems with wristband-interaction behavior analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(7), 2501–2511.

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of AAAI Conference on Artificial Intelligence,* pp. 4263–4270.

Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., & Tröster, G. (2008). Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, *7*(2), 42–50.

Tao, W., Lai, Z.-H., Leu, M. C., & Yin, Z. (2018). Worker activity recognition in smart manufacturing using IMU and sEMG signals with convolutional neural networks. *Procedia Manufacturing*, *26*, 1159–1166.

Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., & Gong, Y. (2020). Few-shot class-incremental learning. In*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 12183–12192.

Tsanousa, A., Meditskos, G., Vrochidis, S., & Kompatsiaris, I. (2019). A weighted late fusion framework for recognizing human activity from wearable sensors. In 10th international conference on information, intelligence, systems and applications (IISA), pp. 1–8. IEEE.

Wang, K.-J., Rizqi, D. A., & Nguyen, H.-P. (2021). Skill transfer support model based on deep learning. *Journal of Intelligent Manufacturing*, *32*(4), 1129–1146.

Ward, J. A., Lukowicz, P., Troster, G., & Starner, T. E. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1553–1567.

Zhao, Z., Chen, Y., Liu, J., Shen, Z., & Liu, M. (2011). Cross-people mobile-phone based activity recognition. In *Twenty-second International Joint Conference on Artificial Intelligence,* pp. 2545–2550.

Zhou, F., Ji, Y., & Jiao, R. J. (2013). Affective and cognitive design for mass personalization: Status and prospect. *Journal of Intelligent Manufacturing*, *24*(5), 1047–1069.

Zhu, X., Wang, Y., Dai, J., Yuan, L., & Wei, Y. (2017). Flow-guided feature aggregation for video object detection. *Proceedings of the IEEE International Conference on Computer Vision*, *1*, 408–417.

Zhu, X., Xiong, Y., Dai, J., Yuan, L., & Wei, Y. (2017). Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 4141–4150.