Multi-modal recognition of worker activity for human-centered intelligent manufacturing

Wenjin Tao [a],[*], Ming C. Leu [a], Zhaozheng Yin [b]

[a] Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA
[b] Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA

ARTICLE INFO

ABSTRACT

This study aims at sensing and understanding the worker's activity in a human-centered intelligent manufacturing system. We propose a novel multi-modal approach for worker activity recognition by leveraging information from different sensors and in different modalities. Specifically, a smart armband and a visual camera are applied to capture Inertial Measurement Unit (IMU) signals and videos, respectively. For the IMU signals, we design two novel feature transform mechanisms, in both frequency and spatial domains, to assemble the captured IMU signals as images, which allow using convolutional neural networks to learn the most discriminative features. Along with the above two modalities, we propose two other modalities for the video data, i.e., at the video frame and video clip levels. Each of the four modalities returns a probability distribution on activity prediction. Then, these probability distributions are fused to output the worker activity classification result. A worker activity dataset is established, which at present contains 6 common activities in assembly tasks, i.e., grab a tool/part, hammer a nail, use a power-screwdriver, rest arms, turn a screwdriver, and use a wrench. The developed multi-modal approach is evaluated on this dataset and achieves recognition accuracies as high as 97% and 100% in the leave-one-out and half-half experiments, respectively.

1. Introduction

Industrial big data has been increasingly accessible and affordable, benefiting from the availability of low-cost sensors and the development of Internet-of-Things (IoT) technologies (Jeschke et al., 2017; Lee et al., 2015), which builds up the data foundation to make the manufacturing more intelligent (Lee et al., 2018; Nagorny et al., 2017). With the recent fast growing of Artificial Intelligence (AI) technologies, especially deep learning (LeCun et al., 2015) and reinforcement learning (Kober et al., 2013) methods, AI boosted manufacturing has been increasingly attractive in both scientific research and industrial applications. In an intelligent manufacturing system involving workers, recognition of the worker's activity is one of the primary tasks. It can be used for quantification and evaluation of the worker's performance, as well as to provide onsite instructions with augmented reality. Also, worker activity recognition is crucial for human–robot interaction and collaboration. It is essential for developing human-centered intelligent manufacturing systems.

1.1. Related work

In the computer vision area, image/video-based human activity recognition using deep learning methods has been intensively studied

in recent years and unprecedented progress has been made (Ji et al., 2013; Carreira and Zisserman, 2017). However, visual-based recognition suffers from the occlusion issue, which affects the recognition accuracy. Wearable devices, such as an armband embedded with an Inertial Measurement Unit (IMU), directly sense the movement of human body, which can provide information on the body status. In addition, there are a lot of inexpensive wearable devices in the market, such as smart armbands and smartphones, which are widely used in activity recognition tasks. Wearable devices are directly attached to the human body and thus do not have the occlusion issue. Nevertheless, a wearable device can only sense the human body activity locally, and it is challenging to precisely recognize an activity involving multiple body parts. Although multiple devices can be applied to simultaneously sense the activity globally, it makes the system cumbersome and brings discomfort to the user.

Worker activity recognition in the manufacturing area is still an emerging topic and few studies have been made. Stiefmeier et al. (2006) utilized ultrasonic and IMU sensors for worker activity recognition in a bicycle maintenance scenario using a Hidden Markov Model classifier. Later they proposed a string-matching based segmentation and classification method using multiple IMU sensors for recognizing
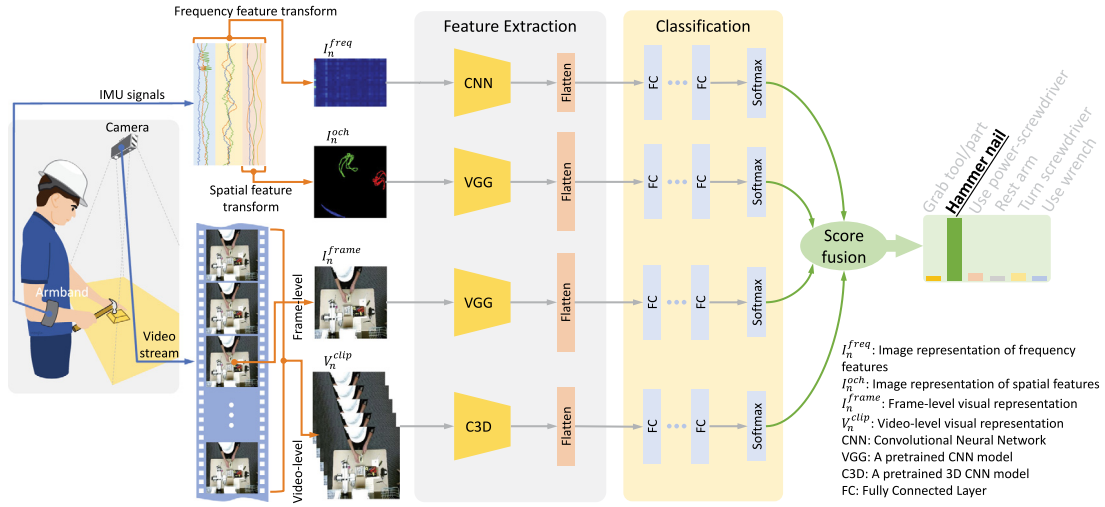
Fig. 1. Overview of our multi-modal approach for worker activity recognition.

worker activity in car manufacturing tasks (Stiefmeier et al., 2007, 2008). Koskimaki et al. (2009) used a wrist-worn IMU sensor to capture the arm movement and a K-Nearest Neighbor model to classify five activities for industrial assembly lines. Maekawa et al. (2016) proposed an unsupervised measurement method for lead time estimation of factory work using signals from a smartwatch with an IMU sensor. Recently, deep learning methods have been introduced to recognize worker activity in human–robot collaboration studies (Wang et al., 2018; Petruck and Mertens, 2018).

In general, the activity recognition task can be broken down into two subtasks: feature extraction and subsequent multiclass classification. To extract more discriminative features, various methods have been applied to the raw signals in the time or frequency domain, e.g., mean, correlation, and Principal Component Analysis (Anguita et al., 2013; Peterek et al., 2014; Chang et al., 2015; Ronao and Cho, 2014). Different classifiers have been explored on the features for activity recognition, such as the Support Vector Machine (Anguita et al., 2013; Chang et al., 2015), Random Forest, K-Nearest Neighbors, Linear Discriminant Analysis (Peterek et al., 2014), and Hidden Markov Model (Ronao and Cho, 2014). To effectively learn the most discriminative features, Jiang and Yin (2015) proposed a method based on Convolutional Neural Networks (CNN). They assembled the raw IMU signals into an activity image, which enabled the CNN model to automatically learn the discriminative features from the activity image for classification.

1.2. Proposed method

Few attempts have been made for the worker activity recognition in the manufacturing field, and most of them only use single sensing modality, which cannot guarantee robust recognition under various circumstances. In the present research, to comprehensively perceive the worker, we choose a smart armband to acquire the Inertial Measurement Unit (IMU) signals and a visual camera to capture the image sequence of the worker's activity. An overview of our method is illustrated in Fig. 1. For the IMU signals, we design two novel mechanisms, in both the frequency and spatial domains, to assemble the captured IMU signals as images. The assembled signal representation allows us to use Convolutional Neural Networks to explore the correlation among time-series signals and learn the most discriminative features for worker activity recognition. As for the video data, we propose two modalities, i.e., at the frame and video-clip levels. Overall, we have four modalities in parallel and each of the four modalities can return a probability distribution on the activity recognition. Then these probabilities are fused to output the worker activity classification result. To

Table 1
Tasks for collecting worker activity.

No.	Tasks	Activities
1	Grab 30 tools/parts from the 3 containers	GT
2	Hammer 15 nails into the wooden dummy	HN
3	Tighten 20 screws using a power-screwdriver	UP
4	Rest arms for about 60 s	RA
5	Tighten 10 nuts using a screwdriver	TS
6	Tighten 10 nuts using a wrench	UW

evaluate the method, a worker activity dataset containing 6 common activities in assembly tasks is established.

The main contributions of our work are as follows:

- We propose a multi-modal approach for the worker activity recognition in manufacturing, using both wearable devices and visual cameras.
- To take advantage of the powerful learning ability of CNN on images, we design two novel mechanisms to produce 2D signal representations of the IMU signals from wearable devices, in both the frequency and spatial domains.
- To synthesize more physical-realistic variations in the training dataset, we propose a kinematics-based data augmentation method for the wearable sensor data. It generates more data by spatial rotation and mirroring, in order to augment variations that cannot be achieved using traditional image augmentation methods.

The remainder of this paper is organized as follows. Section 2 discusses how we build up the worker activity dataset. Section 3 focuses on the novel feature representation and data augmentation. Section 4 describes the details of neural network architectures, training and testing of the multi-modal activity recognition. The experimental setups and results are described in Sections 5 and 6, respectively. Finally, Section 7 provides the conclusions of this research.

2. Multi-modal sensing and data acquisition

To establish our dataset of worker activity, six activities commonly performed in assembly tasks are chosen, which are: grab a tool/part (GT), hammer a nail (HN), use a power-screwdriver (UP), rest arms (RA), turn a screwdriver (TS), and use a wrench (UW). There are 8 subjects recruited to conduct a set of tasks (listed in Table 1) containing the 6 activities.

As demonstrated in Fig. 2(a), the subject is asked to stand in front of the workbench, wear a smart armband (Thalmic Labs Inc., 2019)
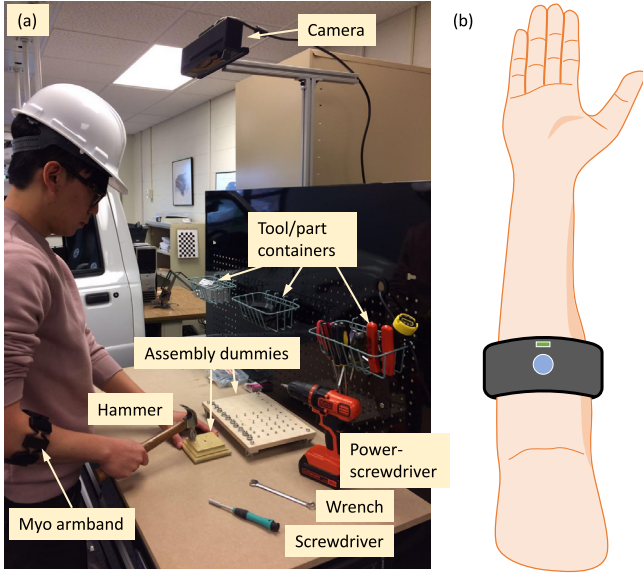
Fig. 2. (a) Data collection setup; (b) Wearing orientation of a right-hand.

Table 2
Number of data samples for each activity of different subjects.

Subject No.	GT	HN	UP	RA	TS	UW
1	193	140	364	266	222	442
2	302	408	195	56	274	751
3	198	183	171	251	214	567
4	204	172	188	29	82	344
5	187	204	142	43	213	372
6	216	77	179	47	129	301
7	213	196	203	254	231	576
8	200	184	262	145	148	273
Total	1713	1564	1704	1091	1513	3626

on his/her right forearm with a fixed orientation (Fig. 2(b)), and perform the tasks on assembly dummies in a natural way. The armband from Thalmic Labs is equipped with IMU sensors for wearable sensor data acquisition. The IMU returns three types of signals (3-channel acceleration, 3-channel angular velocity, and 4-channel orientation) at the sample rate of 50 Hz. These 10-channel signals captured on a worker are transmitted via Bluetooth to the workstation in real time.

While collecting wearable sensor data from the armband, an overhung camera is used to record the assembly tasks simultaneously for monitoring the process. Examples of the 6 activities are shown in Fig. 3, which are taken from the overhung camera.

3. Data preprocessing, signal representation and data augmentation

Convolution-based deep learning methods need the input data to be formatted as tensors, for example, with a fixed size of $h \times w \times c$ for images or with a fixed size of $h \times w \times c \times l$ for image sequences (video clips) where h, w and c are the height, width and the number of channels of the image, respectively, and l is the image sequence length. Therefore, some preprocessing steps are necessary before the data can be fed into a convolutional neural network. In this section we give a detailed description of the pipeline for data preprocessing and the new methods for signal representation. Furthermore, to generate more realistic data, we propose a kinematics-based augmentation method which is also presented in this section.

3.1. Data sampling

Although the data (i.e., IMU sensor signals and videos) are collected simultaneously for all tasks and each task consists of only one activity, there still might be some unrelated activities inside the data, such as preparing activities before hammering nails. To address it, the recorded videos are manually annotated to locate the time durations (i.e., the starting and ending timestamps), each of which contains only one of the six activities. These durations are used to segment the raw data (IMU sensor signals and videos).

Usually, the duration of an activity instance ranges from a few seconds to more than one minute. Thus, sampling is needed to prepare the data samples for recognition. As depicted in Fig. 4, the 10-channel IMU signals and the video recording are synchronized with the timestamps.

Then the 50 Hz IMU signals are sampled using a temporal sliding window with the width of $T = 64$ timestamps and 75% overlap between two windows. Thus, each IMU sample lasts for about 1.3 s, which covers at least one activity pattern. After sampling the IMU signals, the video recordings are sampled according to the time durations of the IMU samples. Then, each video clip has an approximate length of 38 frames.

After sampling, we denote our dataset as $\mathbb{D} = \{D_1, \ldots, D_n, \ldots, D_N\}$

$$D_n = \{[s_n, v_n], y_n\}, n \in [1, N] \tag{1}$$

where s_n is a sample set of time-series IMU signals, v_n is the corresponding video clip sample, and y_n is the manually labeled ground truth of the activity class. More specifically, s_n is a sequence of discrete-time data over T timestamps, $s_n = \{s_{n,1}, \ldots, s_{n,t}, \ldots, s_{n,T}\}$, and each element is elaborated as

$$s_{n,t} = [\underbrace{a_{n,t}^x, a_{n,t}^y, a_{n,t}^z}_{a_{n,t}:\ \text{acceleration}}, \underbrace{g_{n,t}^x, g_{n,t}^y, g_{n,t}^z}_{g_{n,t}:\ \text{gyro}}, \underbrace{q_{n,t}^x, q_{n,t}^y, q_{n,t}^z, q_{n,t}^w}_{q_{n,t}:\ \text{orientation}}],$$
$$t \in [1, T], \tag{2}$$

where a, g, and q are acceleration, angular velocity, and orientation in quaternion, respectively.

After sampling, the quantitative information of the dataset is listed in Table 2. There are 11,211 data samples in total. The eight subjects use different amounts of time to finish each task, therefore they have different numbers of data samples for each activity.

3.2. Wearable sensor signal representation

To take advantage of the powerful learning ability of CNNs on images, we propose to transfer the time-series IMU sensor signals to the image representation. As shown in Fig. 5 , the frequency feature transform assembles the sensor signals in a special pattern such that the hidden correlations among different channels of sensor signals are revealed; and the spatial feature transform uncovers the changing history of orientation signals in the spatial domain. Both feature transform mechanisms enable a CNN model to learn the most discriminative features from images, which are less visible or hidden in the original time-series sensor signals.

Frequency Feature Transform: Frequency domain analysis is a commonly used technique for signal pattern recognition. Rather than directly applying the frequency transform to time-series signals, we propose a new way to unveil the hidden correlations among sensor signals: (1) The 10-channel signals s_n in an IMU sample are stacked row by row as an image $I_n^{stacked}$ with the size of 10×64 (Fig. 5(a)); (2) We expand the 10-row image with a shuffling algorithm (Jiang and Yin, 2015) to form $I_n^{expanded}$ (Fig. 5(b)) with the size of 42×64. The idea here is to make every pair of 10 channels have the chance to be row-neighbors in the image, then the correlations among different channels can be exposed and be further detected by a CNN model; (3) Two-dimensional (2D) Discrete Fourier Transform (DFT) is applied to $I_n^{expanded}$ to get the representation in the frequency domain to analyze the frequency characteristics. Only its logarithmic magnitude is taken
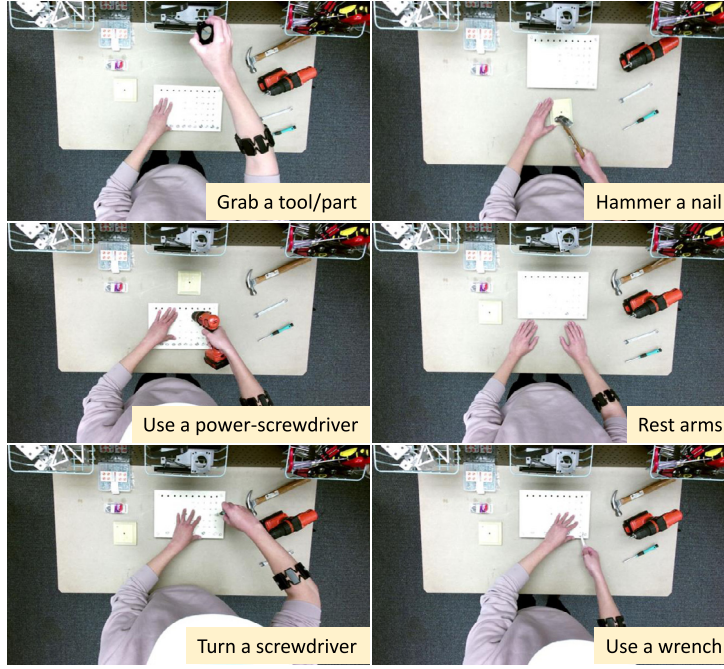
Fig. 3. Examples of the 6 activities captured from the overhung camera.
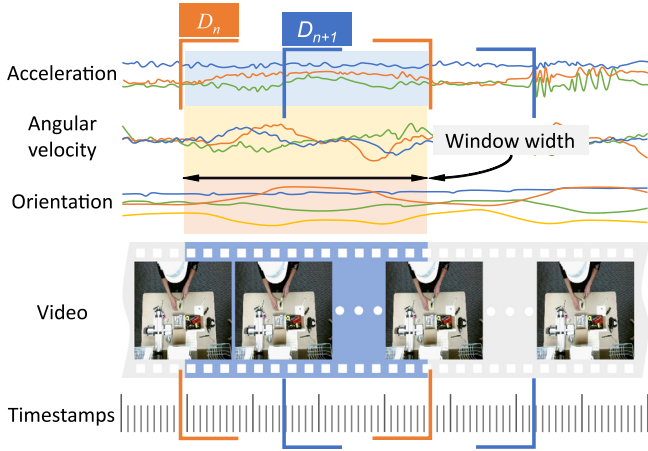

Fig. 4. Scheme of the signal sampling method.

to form the image I_n^{freq} (Fig. 5(c)); (4) Due to the conjugate symmetry of Fourier Transforms

$$\begin{cases} I_n^{freq}(u,v) = I_n^{freq}(-u,-v), \\ I_n^{freq}(-u,v) = I_n^{freq}(u,-v), \end{cases} \quad (3)$$

where u and v represent the two directions of an image, we can use only a half to represent the DFT image to remove the redundancy. This will reduce the architectural complexity and the number of training parameters for the CNN model. Here we keep using the notation I_n^{freq} to represent the one-half (the first and fourth quadrants) of DFT image for simplicity.

Spatial Feature Transform: Implementing feature transform in the frequency domain unavoidably abandons the spatial information from the signals, which motivates us to introduce the second mechanism to exploit the spatial information included in the raw signals. Since recovering the spatial trajectory from IMU data is not an easy task, here we develop an *orientation changing history (och)* image to represent the

pose-changing information of the subject in the spatial domain

$$I_n^{och} = \mathcal{T}_{och}(q_n) \quad (4)$$

where $\mathcal{T}_{och}$ is the spatial feature transform and I_n^{och} is the resulted image. In the spatial feature transform described below, only the orientation information q_n is considered.

First, a unit vector $\mathbf{v}_{ref} = [0,0,1]$ is rotated by q_n to generate a direction vector $\mathbf{v}_{n,t}$ by

$$\mathbf{v}_{n,t} = \mathbf{q}_{n,t} * \mathbf{v}_{ref} \quad (5)$$

where $*$ denotes the rotation operation defined as

$$\mathbf{q} * \mathbf{v} = [(\mathbf{q} \otimes [v^x, v^y, v^z, 0]) \otimes \mathbf{q}^*]_{1:3} \quad (6)$$

where $\otimes$ is the quaternion multiplication, defined as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} q_1^w q_2^x + q_1^x q_2^w + q_1^y q_2^z - q_1^z q_2^y \\ q_1^w q_2^y + q_1^y q_2^w + q_1^z q_2^x - q_1^x q_2^z \\ q_1^w q_2^z + q_1^z q_2^w + q_1^x q_2^y - q_1^y q_2^x \\ q_1^w q_2^w - q_1^x q_2^x - q_1^y q_2^y - q_1^z q_2^z \end{bmatrix}^T \quad (7)$$

where $\mathbf{q_1} = [q_1^x, q_1^y, q_1^z, q_1^w]$, $\mathbf{q_2} = [q_2^x, q_2^y, q_2^z, q_2^w]$, and $\mathbf{q}^*$ is the conjugate of $\mathbf{q}$:

$$\mathbf{q}^* = [-q^x, -q^y, -q^z, q^w]. \quad (8)$$

Then, the orientation changing history can be represented by a series of orientation vectors at different time steps.

$$\mathbf{v}_n^{och} = [v_{n,1}, v_{n,2}, \dots, v_{n,t}], t \in [1, T] \quad (9)$$

which is essentially a set of points on the unit sphere surface.

Secondly, these points are projected onto three orthogonal planes (Fig. 5(d)). On each plane, the points are connected with line segments sequentially to form orientation changing curves in an image.

Finally, these three projected images are stacked as a 3-channel image I_n^{och} which is represented in red, green and blue color, respectively (Fig. 5(e)). Fig. 6 shows some examples of image representations in the frequency and spatial domain, from one subject on six activities, from which we can observe unique patterns of each activity.
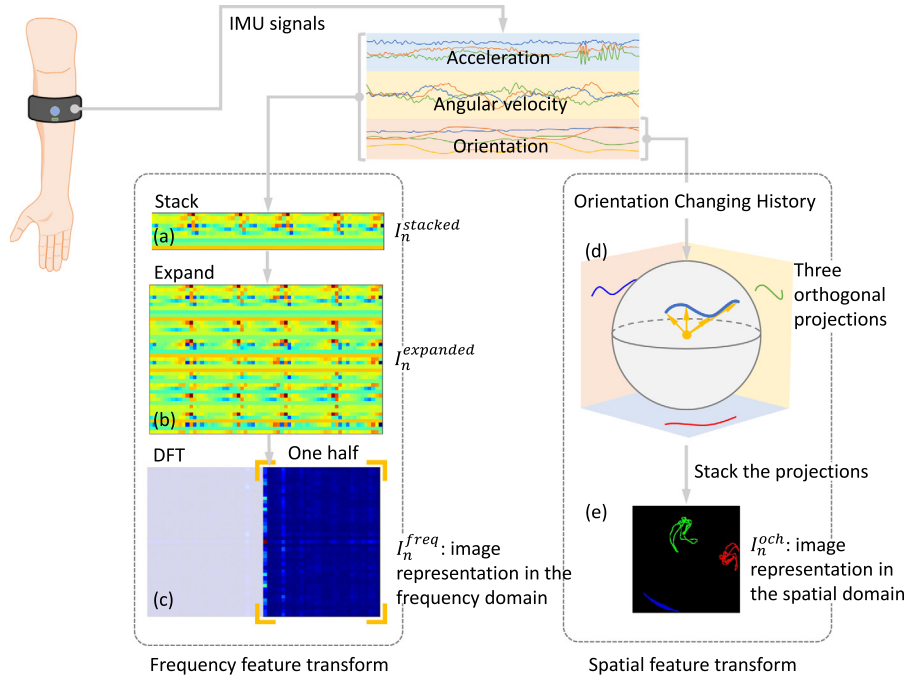
Fig. 5. Illustration of the feature transforms for wearable sensor signals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
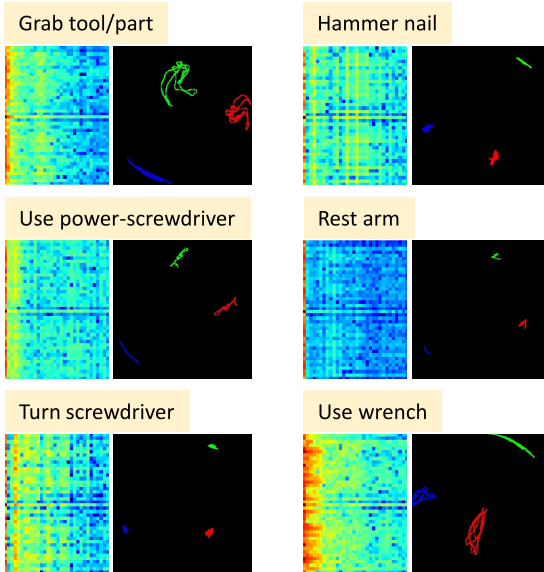

Fig. 6. Examples of IMU image representations by the frequency and spatial feature transforms.

3.3. Visual signal representation

Besides the two mechanisms of feature transforms on the IMU sensor signals, since the recorded video contains rich visual contexts of the worker's activity and visual-based activity recognition also has shown promising results (Ji et al., 2013; Carreira and Zisserman, 2017), we introduce two other mechanisms to represent the video at two levels.

Frame-Level Visual Representation: At the frame level, the middle frame of a video clip is selected as an image representation of an activity, which focuses on the worker's static posture and surrounding environment. The operation is denoted as

$$I_n^{frame} = \mathcal{T}_{frame}(v_n) \tag{10}$$

where $\mathcal{T}_{frame}$ is the operation to extract the middle frame from a video clip v_n.

Video-Level Visual Representation: At the video-clip level, the video clip samples are sampled again to make each video clip have the defined length of frames for a CNN model (to be described in Section 4). The operation is

$$V_n^{clip} = \mathcal{T}_{clip}(v_n) \tag{11}$$

where V_n^{clip} is the resulted fixed-length video-clip from the operation of $\mathcal{T}_{clip}$.

3.4. Kinematics-based data augmentation

For deep learning approaches, a large amount of labeled data are needed to train a valid model with decent performance of generalization. Nevertheless, it is always time-consuming and costly to collect such big data with labels annotated. Data augmentation that synthesizes additional data derived from original ones, is a commonly-used technique to resolve the data shortage problem. Traditionally, image data augmentation techniques include implementing a series of image transformation operations, such as rotating, scaling, shifting, flipping, shearing, etc., on the original images, to generate more image data. The image transformation can introduce more variations and still keep the recognizable contents, and thus it is applied to generate more data for images and video-clips from the visual signal representation. However, the variations introduced by the basic image transformation is not physically-realistic in our sensor signal context.

To include more reasonable variations in the training dataset, we propose a kinematics-based augmentation method to generate more wearable sensor signal samples, rather than implementing image data augmentation on those images resulted from feature transforms. More specifically, the kinematics-based augmentation refers to creating variations by spatial rotation and mirroring on the four channels of orientation signals.

Suppose we have a four-channel orientation signal represented as a quaternion $\mathbf{q}_{n,t}$, a new orientation $\hat{\mathbf{q}}$ can be generated by rotating $\mathbf{q}_{n,t}$ with

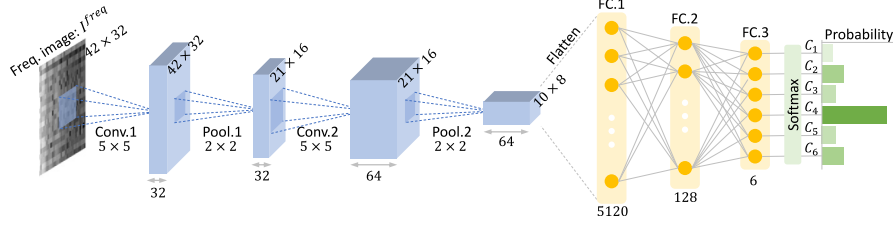$$\hat{\mathbf{q}} = \mathbf{r}_a^\theta \otimes \mathbf{q}_{n,t} \tag{12}$$

Fig. 7. The architecture of our CNN model for I_n^{freq}. 'Conv.' and 'Pool.' denote the operations of convolution and pooling, respectively.

where $\mathbf{r}_a^\theta$ represents a rotation quaternion of an angle θ about an axis $\mathbf{a} = [a^x, a^y, a^z]$. It can be calculated by

$$\mathbf{r}_a^\theta = [a^x \sin(\theta/2), a^y \sin(\theta/2), a^z \sin(\theta/2), \cos(\theta/2)]. \tag{13}$$

Applying mirroring to the original data is to add variations in some situations, for example, the armband is worn in different dominant arms for different subjects. First, the vector $\mathbf{v}^{mirror}$ mirrored from the current direction vector $\mathbf{v}_{n,t}$ (Eq. (5)) against a certain plane can be calculated with

$$\mathbf{v}^{mirror} = \mathbf{v}_{n,t} - 2\mathbf{v}_{n,t} \cdot \mathbf{n}^T \cdot \mathbf{n} \tag{14}$$

where $\mathbf{n}$ is the normal vector of the given plane.

Then the mirrored quaternion $\bar{\mathbf{q}} = [\bar{q}^x, \bar{q}^y, \bar{q}^z, \bar{q}^w]$, representing the transition between the two vectors v_{ref} and v_{mirror}, can be obtained by

$$[\bar{q}^x, \bar{q}^y, \bar{q}^z] = \mathbf{v_{ref}} \times \mathbf{v^{mirror}}$$
$$\bar{q}^w = 1 + \mathbf{v_{ref}} \cdot \mathbf{v^{mirror}} \tag{15}$$

where $\times$ and $\cdot$ are the cross and dot products, respectively.

For the other six channels of linear acceleration and angular velocity, since their measurements are relative to the sensor's coordinate systems, rotation and mirror operation do not affect the values. Some random noises (uniformly distributed in the range of $\pm 5\%$ of the original signals) are added to simulate the possible fluctuations.

4. Multi-modal recognition

In this section the developed multi-modal approach for worker activity recognition is detailed: four deep learning architectures created for different input modalities are presented; the cost function for training each modality is introduced; and the inference fusion strategies to output the recognition result are described.

4.1. Deep learning architectures of four input modalities

After the preprocessing, signal representation generation and data augmentation described in Section 3, there are N data samples[1] $\{X_1, \dots, X_N\}$, each of which contains four different inputs:

$$X_n = \{I_n^{freq}, I_n^{och}, I_n^{frame}, V_n^{clip}\}, n \in [1, N] \tag{16}$$

where $I_n^{freq}, I_n^{och}, I_n^{frame}$ and V_n^{clip} are the four inputs of frequency feature transform, spatial orientation changing history (och) feature transform, frame-level visual representation and video-level visual representation, respectively.

For the three image inputs, I_n^{freq}, I_n^{och} and I_n^{frame}, 2D convolutional operation (Ji et al., 2013) is applied to extract features layer by layer. The value at position (x, y) in the jth feature map of the ith layer is computed by

$$v_{ij}^{xy} = g\left(b_{ij} + \sum_k \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijk}^{pq} v_{(i-1)k}^{(x+p)(y+q)}\right) \tag{17}$$

---

[1] Here we use the same notation for simplicity but this N is larger than the one in Eq. (1) due to the data augmentation.

where $g(\cdot)$ denotes a non-linear activation function. b_{ij} is the bias for this feature map, k is the index of the feature maps in layer $(i-1)$, w_{ijk}^{pq} is the value at the position (p, q) of the kernel connected to the kth feature map, and P_i and Q_i are the height and width of the two-dimensional kernel, respectively.

For video-clip input V_n^{clip}, 3D convolutional operation (Ji et al., 2013) is applied to deal with the additional temporal dimension. The value at position (x, y, z) in the jth feature map of the ith layer is given by

$$v_{ij}^{xyz} = g\left(b_{ij} + \sum_k \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijk}^{pqr} v_{(i-1)k}^{(x+p)(y+q)(z+r)}\right) \tag{18}$$

where R_i is the size of the 3D kernel along the temporal dimension, w_{ijk}^{pqr} is the (p, q, r)th value of the kernel connected to the kth feature map in the previous layer.

The feature maps obtained from a series of convolutional operations are flattened as a feature vector. To solve the classification problem, the vector is further input to a multi-layer neural network. The value of the jth neuron in the ith fully connected layer, denoted as v_{ij}, is given by

$$v_{ij} = g\left(b_{ij} + \sum_{k=0}^{K_{(i-1)}-1} w_{ijk} v_{(i-1)k}\right), \tag{19}$$

where b_{ij} is the bias term, k indexes the set of neurons in the $(i-1)$th layer connected to the current feature vector, w_{ijk} is the weight value in the ith layer connecting the jth neuron to the kth neuron in the previous layer.

In details, the proposed CNN models for the four input modalities are described as follows:

I_n^{freq}: The architecture of our CNN model for I_n^{freq} is illustrated in Fig. 7. It accepts the frequency image as the input, and outputs a probability distribution of the 6 activities. I_n^{freq} has the size of $42 \times 32 \times 1$ (height, width, depth, respectively) and is normalized to the interval $[0, 1]$ before being fed into two 5×5 convolutional layers for feature extraction. Each convolutional layer is down-sampled to a half by implementing a 2×2 max pooling layer. The classification module accepts the $10 \times 8 \times 64$ feature map from the last pooling layer and flattens it as a 5120 feature vector. Then, two fully connected layers are used to densify the feature vector to the dimensions of 128 and C sequentially, where C is the number of worker activity classes. Finally, this C-dimensional score vector S ($[S_1, \dots, S_c, \dots, S_C]$) is transformed to output the predicted probabilities with a softmax function as follows:

$$P(y_n = c | X_n) = \frac{\exp(S_c)}{\sum_{c=1}^C \exp(S_c)} \tag{20}$$

where $P(y_n = c | X_n)$ is the predicted probability of being class c for sample X_n.

I_n^{och} and I_n^{frame}: For these two input modalities, we use transfer learning to solve the image classification problem instead of building and training CNN models from scratch. To extract image features, we use a VGG network (Simonyan and Zisserman, 2014) pretrained on the ImageNet dataset (Deng et al., 2009). For each image input, the feature vector obtained from the fully connected layer FC7 in the VGG model is used to represent the image, then a new classifier is designed on top of it to output the prediction on activity class.

V_n^{clip}: The video-clip input V_n^{clip} contains spatial–temporal information. We use the C3D model pretrained on the Sports-1M dataset (Tran et al., 2015; Jia et al., 2014; Karpathy et al., 2014). The C3D network reads sequential frames and outputs a fixed-length feature vector every 16 frames. We extract activation vectors from the fully connected layer FC6-1, which is then connected to a new classifier to predict the worker activity class.

4.2. Training

Training a deep learning model refers to a process of optimizing the network's weights w to minimize a chosen loss function using training data X. The commonly used regularized cross entropy (Goodfellow et al., 2016) is chosen as the loss function:

$$\mathcal{L}(w) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log[P(y_n = c | X_n)] + \lambda l_2(w) \tag{21}$$

where y_{nc} equals to 0 when the ground truth label of an input image X_n is the cth label, and equals to 1 otherwise. To penalize large weights during training, an l_2 regularization term is applied to the loss function, where λ is its coefficient. To mitigate overfitting issues, the dropout regularization (Srivastava et al., 2014) is used during training, which randomly drops neuron units from the neural network.

4.3. Inference fusion

Just like how human uses five senses to perceive the world, multi-modal approach has the opportunity to integrate all the information and make a comprehensive understanding of the learning problem. Mathematically, each individual model can return a probability distribution on the worker activity prediction, we can design different strategies to fuse the inferences from different models:

Maximum Fusion. This method reports the maximum output within a list of predictions.

$$S_c^{max} = \max_{m \in \{1,2,\ldots,M\}} p_c^m \tag{22}$$

where m is the index of different models and M is the total number of models.

Average Fusion. In this method, we adopt the average to fuse the outputs of different modalities, i.e.,

$$S_c^{avg} = \frac{1}{M} \sum_{m=1}^{M} p_c^m \tag{23}$$

Weighted Fusion (Weighted Maximum Fusion and Weighted Average Fusion). We introduce the informativity value γ^m to evaluate the prediction confidence of each modality m. γ^m is calculated with Eq. (24), which is modified from the Shannon entropy of a discrete probability distribution to vary in the interval of $[0,1]$.

$$\gamma^m = \frac{\sum_{k=1}^{K} p_k^m \log p_k^m}{\log K} + 1 \tag{24}$$

where m is the index of modalities and k is the index of top-K candidates. p_k^m represents the probability of the kth class candidate at the mth model. I^m will be close to 0 if all the top-K candidates have similar probabilities (i.e., $p_k^m \approx 1/K$), and 1 if the probability of top-1 class candidate is about reaching 1 (i.e., $p_1^m \approx 1$).

Then every predicted probability p_k^m of the mth model is weighted by γ^m of this model and the weighted maximum fusion and the weighted average fusion scores are

$$S_c^{max-w} = \max_{m \in \{1,2,\ldots,M\}} \gamma^m p_c^m \tag{25}$$

$$S_c^{avg-w} = \frac{1}{M} \sum_{m=1}^{M} \gamma^m p_c^m \tag{26}$$

For the above four fusion strategies, the final predicted label is chosen as the one that maximizes the fusion score (e.g., for weighted average fusion, $c^* = \arg\max_c S_n^{avg-w}$).

5. Experiments and evaluation metrics

5.1. Implementation details

The CNN architectures of the four input modalities described in the previous sections are constructed using TensorFlow (Abadi et al., 2015) and Keras libraries. They are trained individually so that each of them can make its own inference for further decision fusion. The SGD optimizer is used in training, with the momentum of 0.9, the learning rate of 0.001 and the regularizer coefficient of 1e−5. The batch size for each of the four models is 512, 64, 64 and 512, respectively, which is limited by the computation memory. The number of training epochs is 1000 and 100 for the first modality I_n^{freq} and the other modalities, respectively. We use a workstation with one 12-core Intel Xeon processor, 64 GB of RAM and two Nvidia Geforce 1080 Ti graphic cards for the training jobs. It takes approximately 30 min to train each model for a leave-one-out experiment.

5.2. Evaluation metric

Two evaluation policies are conducted, i.e., half-half and leave-one-out policies. In the half-half evaluation, after randomly shuffling, one half of the dataset is used for training and the other half is kept for testing. In the leave-one-out evaluation, the samples from 7 out of 8 subjects are used for training, and the samples of the left one subject are reserved for testing. We employ several commonly used metrics (Goodfellow et al., 2016) to evaluate the classification performance, which are listed as follows:

- Accuracy

$$Accuracy = \frac{\sum_n^N 1(\hat{y}_n = y_n)}{N} \tag{27}$$

- Precision and Recall

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN} \tag{28}$$

- F_1 score

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{29}$$

where $1()$ is an indicator function. For a certain class y_i, True Positive (TP) is defined as a sample of class y_i that is correctly classified as y_i; False Positive (FP) means a sample from a class other than y_i is misclassified as y_i; False Negative (FN) means a sample from the class y_i is misclassified as another 'not y_i' class. F_1 score is the harmonic mean of Precision and Recall, which ranges in the interval $[0,1]$.

6. Results and discussion

In this section, we first perform evaluations of the data augmentation methods. Then, we compare the performance of different fusion methods. After that, we explore various modalities and their combinations for an ablation study. The performance of our approach on some public dataset is also reported. Then, we conduct visualizations for a better understanding of the CNN model. Finally, future research needs are discussed.

6.1. Evaluation of the data augmentation methods

To evaluate the effectiveness of our proposed kinematics-based augmentation (KA) method, we compare it to the jittering augmentation (JA) method (Sermanet and LeCun, 2011), which has been proved to be an effective method and is commonly used in CNN-based image classification tasks.

Table 3
Comparison (%) of accuracy regarding to data augmentation: None (without data augmentation), JA (jittering augmentation), KA (kinematics augmentation) and JA+KA, for the leave-one-out experiments.

Modalities	Data augmentation methods			
	None	JA	KA	JA+KA
I_n^{freq}	88.01	88.71	90.18	**90.51**
I_n^{och}	63.59	65.01	**77.27**	75.33

Table 4
Comparison (%) of different fusion methods for the leave-one-out experiments.

Fusion methods	Accuracy	Precision	Recall	F score
Maximum	93.68	92.48	92.50	91.09
Average	**97.17**	**97.04**	**96.82**	**96.81**
Weighted Max.	93.68	92.45	92.49	91.07
Weighted Avg.	96.79	96.38	96.28	96.04

For the KA method, four rotation angles $\{\pm\pi/8, \pm\pi/4\}$ are selected for rotation augmentation, i.e., new samples are generated by implementing rotation on the original signal samples, and two mirroring planes yz-plane and xz-plane are chosen for mirroring augmentation, overall yielding 6 augmented samples for each actual sample. Then the amount of the augmented training dataset is 6 times more than the original one. Note that the augmentation is applied directly on each original signal sample s_n before the feature transforms.

As for the JA method, to have a fair comparison with the KA method, 6 augmented samples are generated by randomly translating in the range of $\pm10\%$ of the image width/height, scaling in the range of $[0.9, 1.1]$ ratio, and rotating in the range of $[-5, +5]$ degrees. In the JA method, the augmentation is applied to each image I_n^{freq} and I_n^{och} after the feature transforms.

We also evaluate the performance of the JA+KA method, in which the augmented data from the JA and KA methods are integrated. The leave-one-out evaluations of the two modalities I_n^{freq} and I_n^{och} on our activity dataset with the different augmentation methods are shown in Table 3 (the half-half accuracies are not considered for the comparison purpose because they are about reaching 100%). All the three augmentation methods have accuracy improvements compared with the models without using data augmentation. For I_n^{freq}, the JA method improves the accuracy from 88.0% to 88.7%, and the KA method outperforms the JA method, whose accuracy is 90.2%. By combining the JA and KA methods, the accuracy is slightly further improved to 90.5%. For I_n^{och}, the accuracy is improved from 63.6% to 65.0% with the JA method, and is further improved by using the KA method, which is 77.3%, 12 percentage points higher than the JA method. However, the JA+KA method does not further improve the accuracy and its accuracy 75.3% is lower than the KA method.

Overall, the data augmentation techniques, JA and KA, demonstrate the effectiveness in improving the model performance, because the augmentation process introduces more variations to the training dataset to simulate the potential variations in the unseen samples, which pushes the deep learning model to learn the most discriminative features and enables the trained model with more generalization capabilities. Meanwhile, the KA method outperforms the JA method. It is because rather than introducing variations to the image, like what JA method does, KA method directly generates some physically-realistic variations to the original signal sample, which is more effective to augment the dataset to be more comprehensive. Although JA+KA method improves the performance of I_n^{freq} slightly compared with KA method, it does not for I_n^{och}. Because JA+KA method has a larger amount (i.e., 2 times) of training data and I_n^{och} has a more complex architecture than I_n^{freq}, which makes the training less efficient, we choose KA method for both of the modalities in the following study as a compromise between performance and training efficiency.

6.2. Evaluation of different fusion methods

Each of the four input modalities generates a vector output before the fusion step. Then, these vector outputs are fused to have only one score vector as the final output. To study the effect of the four different fusion methods: (1) maximum fusion, (2) average fusion, (3) weighted maximum fusion and (4) weighted average fusion, a set of experiments are conducted on our activity dataset.

The comparisons of the fusion performance, in terms of accuracy, precision, recall and F score, are listed in Table 4. The average fusion method performs better than the maximum fusion method for all the metric items. The weighted maximum method has the same accuracy as the maximum method but lower precision, recall and F score. The weighted average method has lower performance than the average method. The two weighted methods do not contribute additional improvement as they did in Tao et al. (2018b). Therefore, the average method is chosen as the fusion strategy for our following experiments.

6.3. Evaluation of different input modalities

A central idea of our approach is that the reasoning based on multiple modalities can significantly improve the inference performance based on single modality. To validate this idea, we perform a comprehensive ablation study where we progressively increase the number of modalities and try different modality combinations. The performance of these cases in terms of accuracy, precision, recall and F score with two evaluation policies (half-half and leave-one-out) is summarized in Table 5.

To simplify the abbreviation, we use $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ and $\mathcal{M}_4$ to represent the four input modalities, I_n^{freq}, I_n^{och}, I_n^{frame} and V_n^{clip}, respectively. For the single-modal cases, although $\mathcal{M}_1$ and $\mathcal{M}_2$ are both based on the IMU signals, $\mathcal{M}_2$ shows lower performance because it only uses the 4 orientation channels out of the 10 channels. Also, it demonstrates that the frequency feature transform provides more discriminative features for activity recognition. $\mathcal{M}_3$ performs better than $\mathcal{M}_4$, which shows that the current pretrained VGG model can extract more discriminative features than the C3D model. Overall, $\mathcal{M}_1$ achieves the highest performance in the single-modal cases, whose metric items are accuracy (90.2%), precision (90.7%), recall (89.5%) and F score (87.6%), respectively.

For the dual-modal cases, all the 6 combinations are evaluated. All the cases have better results compared with their related single-modal cases, e.g., $\mathcal{M}_{\{2,3\}}$ performs better than both $\mathcal{M}_2$ and $\mathcal{M}_3$. $\mathcal{M}_{\{1,3\}}$ has the highest accuracy as their individual modalities are also the highest two for the single-modal cases.

For the triple-modal cases, 4 combinations are tested. The fusion of more modalities further improve the performance than the duel-modal cases. $\mathcal{M}_{\{1,3,4\}}$ has the highest accuracy as their individual modalities are also the highest three for the single-modal cases.

Finally, a quad-modal case $\mathcal{M}_{\{1,2,3,4\}}$ including all the four modalities is experimented, which achieves the highest performance. Therefore, we choose the quad-modal architecture for our model.

For the half-half experiments, almost all of the testing samples are correctly recognized. It is higher than the leave-one-out experiments. This is because all the testing subjects are seen in the half-half experiment, while the testing subject in the leave-one-out experiment is unseen.

6.4. Performance comparison on the public dataset

To validate the generalization of our method, a commonly-used public dataset for human activity recognition, PAMAP2 dataset (Reiss and Stricker, 2012), is also chosen for comparison. This dataset has 12 human activities (lying, sitting, standing, walking, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning,

Fig. 8. Examples of Class Activation Map (CAM) visualization.

Table 5
Overall performance (%) of the half-half (hh) and leave-one-out (loo) experiments.

Methods	Accuracy		Precision		Recall		F Score	
	hh	loo	hh	loo	hh	loo	hh	loo
Previous (Tao et al., 2018a)	97.6	87.4	97.8	89.0	97.5	89.5	97.7	87.6
$\mathcal{M}_1$	99.5	90.2	99.5	90.7	99.6	90.9	99.5	90.3
$\mathcal{M}_2$	93.0	77.3	92.3	77.5	92.5	78.3	92.4	75.0
$\mathcal{M}_3$	100	86.8	100	83.0	100	83.2	100	81.3
$\mathcal{M}_4$	100	80.8	100	79.1	100	77.7	100	74.3
$\mathcal{M}_{\{1,2\}}$	99.6	91.1	99.6	91.5	99.6	92.1	99.6	91.4
$\mathcal{M}_{\{1,3\}}$	100	94.8	100	94.9	100	94.6	100	94.3
$\mathcal{M}_{\{1,4\}}$	100	92.2	100	93.1	100	91.3	100	90.1
$\mathcal{M}_{\{2,3\}}$	100	90.3	100	90.9	100	87.8	100	87.0
$\mathcal{M}_{\{2,4\}}$	100	85.0	100	84.0	100	82.8	100	80.2
$\mathcal{M}_{\{3,4\}}$	100	89.5	100	86.3	100	86.0	100	84.2
$\mathcal{M}_{\{1,2,3\}}$	100	95.3	100	95.4	100	95.4	100	95.2
$\mathcal{M}_{\{1,2,4\}}$	100	93.9	100	93.5	100	94.1	100	93.0
$\mathcal{M}_{\{1,3,4\}}$	100	95.9	100	95.2	100	94.8	100	94.2
$\mathcal{M}_{\{2,3,4\}}$	100	92.6	100	90.4	100	90.7	100	89.5
$\mathcal{M}_{\{1,2,3,4\}}$	**100**	**97.2**	**100**	**97.0**	**100**	**96.8**	**100**	**96.8**

$\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ and $\mathcal{M}_4$ represent the four input modalities, I_n^{freq}, I_n^{och}, I_n^{frame} and V_n^{clip}, respectively. $\mathcal{M}_{\{\cdot\}}$ denotes a multi-modality model, e.g., $\mathcal{M}_{\{2,3\}}$ represents the fusion of the $\mathcal{M}_2$ and $\mathcal{M}_3$ modalities. We use average fusion method to fuse multiple modalities together.

Table 6
Performance (%) comparison of existing deep models on the PAMAP2 activity dataset.

Method	Accuracy
Hammerla et al. (2016)	93.70
Murahari and Plötz (2018)	87.50
Zeng et al. (2018)	89.96
Xi et al. (2018)	93.50
Xu et al. (2019)	93.50
Our model	**94.16**

class, computed for every location of an input image, representing how strong the connection is between each location and a specific activity class. A set of CAM visualizations are shown in Fig. 8, where the generated heatmaps are overlaid onto the input images. We can see that the model is able to focus on the hand and tool regions, where exactly the interaction happens.

6.6. Future research needs

At present, we conduct the multi-modal recognition of 6 basic activities. To further push the current approach to the practical application, some directions for future work are considered, such as recruiting more subjects to learn more working styles, optimizing data augmentation techniques to add more variations to the collected data, and exploring different methods of signal preprocessing and feature extraction to fully exploit the recorded signals. In addition, more fusion methods can be explored and every modality can be further improved to reach their optimal performance.

7. Conclusion

Worker behavior awareness is crucial towards human-centered intelligent manufacturing. In this paper, we proposed a multi-modal approach for worker activity recognition. Two sensors (wearable device and camera) were adopted to perceive the worker, and four modalities were built to recognize the activity independently. Then, inference fusion was implemented to achieve an optimal understanding of the worker's behavior. A worker activity dataset has been established, which currently involves 8 subjects and contains 6 common activities in assembly tasks (i.e., grab a tool/part, hammer a nail, use a power-screwdriver, rest arms, turn a screwdriver and use a wrench). A kinematics-based data augmentation method was developed to generate more physically-realistic variations in the training dataset. This performs better than the traditional data augmentation method. The multi-modal approach is evaluated on the dataset and achieves 100% and 97% recognition accuracy in the half-half and leave-one-out experiments, respectively. Our approach can be further generalized to other sensors, modalities, and working contexts.

ironing and rope jumping) captured by three IMU sensors (worn on the wrist, chest and ankle, respectively), and the activities are performed by 9 different subjects. Since the PAMAP2 dataset does not include video recordings, we evaluate the performance of our CNN models of the first two modalities on it. The performance comparison of several existing deep learning models on the PAMAP2 dataset is listed in Table 6. Using the same evaluation protocol, our model achieves the best recognition accuracy, 94.2%, compared with other methods in the literature. The reason is that our proposed feature transforms convert the time-series signals to images in the frequency and spatial domains, which allows CNNs to learn the most discriminative features in a training process. In addition, the inference fusion mechanism augments the recognition of each individual modality.

6.5. Visualizing the class activation map of $\mathcal{M}_3$

It is known that a deep learning model driven by a large amount of data can achieve superior performance for various tasks, such as the image classification task for our third modality $\mathcal{M}_3$. However, it is usually treated as a black box and criticized for weak interpretability, due to the high model complexity and tremendous hidden weights. To have an intuitive interpretation of the connections between image contents and the predicted activity class, we visualize the class activation map (CAM), which is a score heatmap associated with a specific activity

CRediT authorship contribution statement

Wenjin Tao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Ming C. Leu:** Writing - reviewing & editing, Supervision, Project administration, Funding acquisition. **Zhaozheng Yin:** Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. URL https://www.tensorflow.org/.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L., 2013. A public domain dataset for human activity recognition using smartphones. In: ESANN.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, pp. 4724–4733.

Chang, W., Dai, L., Sheng, S., Tan, J.T.C., Zhu, C., Duan, F., 2015. A hierarchical hand motions recognition method based on IMU and sEMG sensors. In: Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on. IEEE, pp. 1024–1029.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: CVPR09.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Hammerla, N.Y., Halloran, S., Plötz, T., 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880.

Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., Eschert, T., 2017. Industrial internet of things and cyber manufacturing systems. In: Industrial Internet of Things. Springer, pp. 3–19.

Ji, S., Xu, W., Yang, M., Yu, K., 2013. 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 35 (1), 221–231.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia. ACM, pp. 675–678.

Jiang, W., Yin, Z., 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In: Proceedings of the 23rd ACM International Conference on Multimedia. ACM, pp. 1307–1310.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732.

Kober, J., Bagnell, J., Peters, J., 2013. Reinforcement learning in robotics: A survey. Int. J. Robot. Res. 32 (11), 1238–1274.

Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., Roning, J., 2009. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In: Control and Automation, 2009. MED'09. 17th Mediterranean Conference on. IEEE, pp. 401–405.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, J., Ardakani, H.D., Yang, S., Bagheri, B., 2015. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Procedia CIRP 38, 3–7.

Lee, J., Davari, H., Singh, J., Pandhare, V., 2018. Industrial artificial intelligence for industry 4.0-based manufacturing systems. Manuf. Lett. 18, 20–23.

Maekawa, T., Nakai, D., Ohara, K., Namioka, Y., 2016. Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, pp. 1088–1099.

Murahari, V.S., Plötz, T., 2018. On attention models for human activity recognition. In: Proceedings of the 2018 ACM International Symposium on Wearable Computers. ACM, pp. 100–103.

Nagorny, K., Lima-Monteiro, P., Barata, J., Colombo, A.W., 2017. Big data analysis in smart manufacturing: A review. Int. J. Commun. Netw. Syst. Sci. 10 (03), 31.

Peterek, T., Penhaker, M., Gajdoš, P., Dohnálek, P., 2014. Comparison of classification algorithms for physical activity recognition. In: Innovations in Bio-Inspired Computing and Applications. Springer, pp. 123–131.

Petruck, H., Mertens, A., 2018. Using convolutional neural networks for assembly activity recognition in robot assisted manual production. In: International Conference on Human-Computer Interaction. Springer, pp. 381–397.

Reiss, A., Stricker, D., 2012. Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers. IEEE, pp. 108–109.

Ronao, C.A., Cho, S.-B., 2014. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In: Natural Computation (ICNC), 2014 10th International Conference on. IEEE, pp. 681–686.

Sermanet, P., LeCun, Y., 2011. Traffic sign recognition with multi-scale convolutional networks. In: Neural Networks (IJCNN), the 2011 International Joint Conference on. IEEE, pp. 2809–2813.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting.. J. Mach. Learn. Res. 15 (1), 1929–1958.

Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., Troster, G., 2006. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In: Wearable Computers, 2006 10th IEEE International Symposium on. IEEE, pp. 97–104.

Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., Tröster, G., 2008. Wearable activity tracking in car manufacturing. IEEE Pervasive Comput. 7 (2).

Stiefmeier, T., Roggen, D., Troster, G., 2007. Fusion of string-matched templates for continuous activity recognition. In: Wearable Computers, 2007 11th IEEE International Symposium on. IEEE, pp. 41–44.

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

Tao, W., Leu, M.C., Yin, Z., 2018b. American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. Eng. Appl. Artif. Intell. 76, 202–213.

Thalmic Labs Inc., 2019. Myo armband. [Online; accessed 17-August-2019]. URL https://support.getmyo.com.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4489–4497.

Wang, P., Liu, H., Wang, L., Gao, R.X., 2018. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann. 67 (1), 17–20.

Xi, R., Li, M., Hou, M., Fu, M., Qu, H., Liu, D., Haruna, C.R., 2018. Deep dilation on multimodality time series for human activity recognition. IEEE Access 6, 53381–53396.

Xu, C., Chai, D., He, J., Zhang, X., Duan, S., 2019. Innohar: A deep neural network for complex human activity recognition. IEEE Access 7, 9893–9902.

Zeng, M., Gao, H., Yu, T., Mengshoel, O.J., Langseth, H., Lane, I., Liu, X., 2018. Understanding and improving recurrent networks for human activity recognition by continuous attention. In: Proceedings of the 2018 ACM International Symposium on Wearable Computers. ACM, pp. 56–63.