Machine Learning with Applications 12 (2023) 100461

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Machine Learning with Applications

Human activity classification using deep learning based on 3D motion feature

Endang Sri Rahayu [a,b], Eko Mulyanto Yuniarno [a], I. Ketut Eddy Purnama [a], Mauridhi Hery Purnomo [a,c,*]

[a] *Department of Electrical Engineering, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia*
[b] *Department of Electrical Engineering, Jayabaya University, Jakarta, Indonesia*
[c] *University Center of Excellence on Artificial Intelligence for Healthcare and Society (UCE AIHeS), Surabaya, Indonesia*

ARTICLE INFO

Keywords:
Human activity
Motion feature
Deep learning
Classification

ABSTRACT

Human activity classification is needed to support various fields. The health sector, for example, requires the ability to monitor the activities of patients, the elderly, or people with special needs to provide services with fast response as needed. In the traditional classification model, the steps taken to start from the input of data and then proceed with feature extraction, representation, classifier and end with semantic labels. The classification stage uses Convolutional Neural Network (CNN) deep learning to data input, CNN, and semantic labels. This paper proposes a novel method of classifying nine activities based on the movement features of changes in joint distance using Euclidean on the order of frames in each activity segment as input to the CNN model. This study's motion feature extraction technique was tested using various window sizes to obtain the best classification accuracy. The experimental results show that the selection of window size 16 on the motion feature setting will produce an optimal model accuracy of 94.08% in classifying human activities.

1. Introduction

The Human Movement Analysis (HMA) research area is an interdisciplinary research area that attracts great interest from the computer vision, machine learning, multimedia, and medical research communities. The implementation of this research is utilized for human–computer interaction, security (intelligent surveillance), health (assisted clinical studies), information technology (content-based video capture), entertainment (special effects in somatosensory film and game production) for all aspects of our daily life (Seidenari et al., 2013). As an essential research series of HMA, Human Activity Recognition (HAR) forms the basis for all the applications mentioned above. Utilization has been widely used in health care applications such as elderly monitoring, exercise monitoring, and rehabilitation monitoring (Huang et al., 2020). HAR is also developing in applications in the fields of robotics, entertainment, biometrics, and multimedia (Bennamoun et al., 2020). An indoor emergency awareness alarm system was also developed using deep neural networks. The system uses mobile devices for people such as the elderly, people with special needs or children who may need help (Kim & Kim, 2020).

In recent years, the development of deep learning has resulted in significant advances in activity recognition. In various research topics, there are two main methods, framework-based activity recognition and sensor-based activity recognition (Goddard, 2021). One of the main problems of the existing HAR strategy is the relatively low classification accuracy, so it is necessary to increase the accuracy, which requires high computational overhead (Kong et al., 2021). Classification models using deep learning are continuously being developed to improve performance resulting from traditional video classification models.

One of the Deep Learning algorithms to process image or sound data is a Convolutional Neural Network (CNN). CNN in this study was used to classify labelled data using the supervised learning method. Supervised learning work is the presence of target data for data training. Among all types of neural networks, CNN is known as the most successful and is widely used to solve problems of image recognition, object detection/localization, and even text processing (Alpaydin, 2021). CNN (convolutional kernels) combines some local filters with raw input data and generates local translation-invariant features in the convolutional layer. The successive pooling layer extracts fixed-length features via a sliding window from raw input data following some rules like mean, max., etc. (Zhao et al., 2019).

This paper proposes a model for recognizing human activities using deep learning to classify human activities (actions). The preparation of data as input data for the CNN model is the main concern in this paper. Meanwhile, we also pay attention to designing the CNN model that will be trained on the dataset. The source data of the 3D coordinate points of the joints' positions will be processed to detect changes in movement that occur by calculating the distance of the coordinates of the joints

* Corresponding author at: Department of Electrical Engineering, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia.
E-mail addresses: endangsr@jayabaya.ac.id (E.S. Rahayu), ekomulyanto@ee.its.ac.id (E.M. Yuniarno), ketut@te.its.ac.id (I.K.E. Purnama), hery@ee.its.ac.id (M.H. Purnomo).

https://doi.org/10.1016/j.mlwa.2023.100461
Received 25 January 2023; Received in revised form 6 March 2023; Accepted 8 March 2023
Available online 15 March 2023
2666-8270/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

on the adjacent frame. With the motion feature extraction technique, a set of frames representing motion in time series will be taken to form windows that are ready to be input into the CNN model. Each video segment in the joints coordinate data represents one activity. The training and testing data to determine the model's accuracy uses the public dataset of Florence with 3D data from the position of 15 joints consisting of 215 videos.

The rest of this paper is organized as follows. Related work is outlined in Section 2. In Section 3, the proposed method is explained. Section 4 presents an evaluation of the proposed system. Section 5 provides the conclusions and future research directions.

2. Related work

In this section, we present papers related to the research carried out. Attention is paid to various studies using video or RGB image data and sensor detection results to recognize the human activity. Various processing and deep learning methods are based on a series of spatiotemporal frames from joint and skeleton data. Bennamoun et al. (2020) images represent a scene's photometric and geometric information. In addition, low-cost depth cameras, e.g., Intel Realsense - Orbbec Astra - Microsoft Kinect v2, due to their high gain frame rates and the significant nature of public data releases in recent years, can enable real-time applications. Research (Gaglio et al., 2015), recognizing human activity using information from RGB-D cameras, Microsoft Kinect, using the Kinect Activity Recognition Dataset (KARD) with three machine learning techniques, Support Vector Machine, K-means clustering, and Hidden Markov models. Combines these three techniques to detect the postures involved while performing an activity. It aims to classify and model each activity as a spatiotemporal evolution of a known posture.

Human activity recognition (HAR) is carried out using various data sources derived from the detection results of environmental sensors, video, or wearable sensors (Huang et al., 2020; Pham et al., 2020). Research using video or image data sources based on RGB (Chen et al., 2018) or RGB Depth (Bennamoun et al., 2020; Gaglio et al., 2015), as well as information based on skeleton (Ahmad et al., 2020; Li et al., 2020; Liu et al., 2018; Su et al., 2019; Wang & Wang, 2018; Zhang et al., 2020), is used to recognize human movement. 2D images (Liu et al., 2018) research developed a skeleton traversal based on a tree structure derived from the Kinect skeleton image structure. The new Spatio-Temporal Long Short-Term Memory (ST-LSTM) Network is designed for skeleton-based action recognition. Pham et al. (2020) also uses a method that combines LSTM with CNN. Huang et al. (2020) researches a new end-to-end HAR method. The Wang et al. (2021) method uses the Skeleton Edge Motion Network (SEMN) further to explore information on the motion of human body parts. A skeletal edge motion network is proposed by stacking several spatial–temporal blocks to study the effective representation of skeletal data. A two-level hierarchical framework for human action recognition was designed by Su et al. (2019), where a Support Vector Machine is used for coarse-grain classification at the first level. Meanwhile, a combination of Support Vector Machines is used at the second level, and the Hidden Markov model is used for fine-grained classification. The non-intrusive activity recognition method was carried out in Chen et al. (2018) research, spatial and temporal information, based on the distance between two frames in two adjacent frames. Each frame is converted into a feature vector to maintain the order of the frames. This method can be used for incomplete data frames and is considered robust because it can handle noisy activities.

The motion features in this study are a series of frames from joint coordinates in a three-dimensional (3D) dataset of the human skeleton that is moved sequentially. Motion feature research by Aldahoul et al. (2022) uses the EfficientDet-D7 method. Movement through dynamic representation and matching framework feature sequences in activity recognition in Li et al. (2018) research using segmentation techniques. Chen et al. (2023) uses a new framework to identify and differentiate between humans by human image processing augmented by colour and depth. For example, from a commercially available depth camera such as the Microsoft Kinect, it is possible to derive specific features from the image that can distinguish between individuals in a 3D frame.

Deep learning is a framework capable of deep learning to obtain a set processing time, especially for training the network very quickly (Phyo et al., 2019). Convolutional Neural Network (CNN) is a deep learning method used to complement human activity recognition. CNN learns millions of internal parameters from a data representation optimized for training data. Research using CNN was carried out by Lee et al. (2017) using data from the accelerator as input. Meanwhile, Gochoo et al. (2019) research uses input data from binary sensors, namely Passive Infrared (PIR) motion sensors and door sensors which are processed using the Deep Convolutional Neural Network (DCNN). Pham et al. (2020) uses wearable sensors (SensCapsNet) to recognize human activity, with an accelerometer and gyroscope sensor data input that generates spatial–temporal data. Liu et al. (2018) is research on spatial and temporal domains to simultaneously analyze hidden sources of action-related information in human skeletal sequences across seven benchmark data sets.

2.1. Human activity recognition

Computer vision technology that leads to the analysis of human motion is developing by the many applications based on human movement. Analysis can be done through body structure analysis, human tracking, human action recognition, etc. Physical behaviour is a concern to be accurately and quickly analyzed for implementation in various fields. The terms "action" and "activity" are commonly used in the activity recognition community for physical behaviour. In some cases, they are used interchangeably, and in other cases, they are used to denote behaviours of different complexity and duration (Chen & Nugent, 2019). We distinguish between action recognition and action detection. Recognition of the activity (action) referred to in this paper recognizes the class/class of action being carried out by the actor in the video. At the same time, action detection is where and when the action is carried out in a series of videos, which will not focus on this research. We also distinguish between action prediction and action recognition. Action prediction is that the action video data arrives sequentially. Meanwhile, action recognition takes full observation as input. The key to obtaining an accurate initial classification is extracting the most discriminatory information from the initial segment in the temporal sequence (Fu, 2016).

The purpose of human action recognition is to predict the label of an individual or group of people's actions from video observations. In addition, the problem to be solved is that different people may show different poses in doing the same activity. All of these factors will result in a sizeable intra-class display and pose variety, which confuses many existing activity recognition algorithms (Fu, 2016). This is what causes various human activity recognition research methods to continue to develop to obtain better results.

3. Our method

This section describes a proposed method for classifying human activities using a deep learning CNN model. The explanation begins with presenting the system model in general; then, it will explain the extraction of motion features and the CNN model used to classify human activities.

The flow diagram of the human activity classification method using the proposed motion feature extraction is shown in Fig. 1. The dataset was prepared using the motion feature extraction method as model input. Furthermore, the extracted data is processed with CNN to generate activity classes. Finally, we evaluate the performance results of the proposed system.

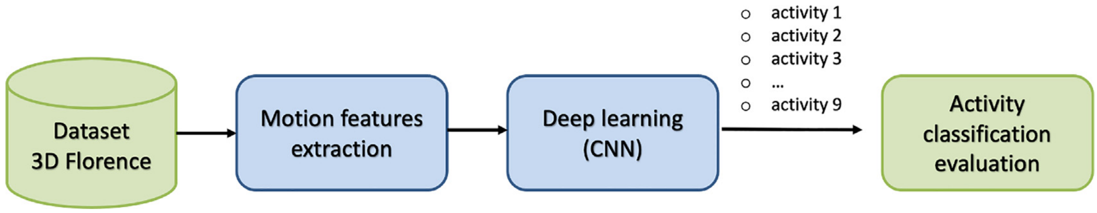

Fig. 1. Flow diagram of human activities classification.

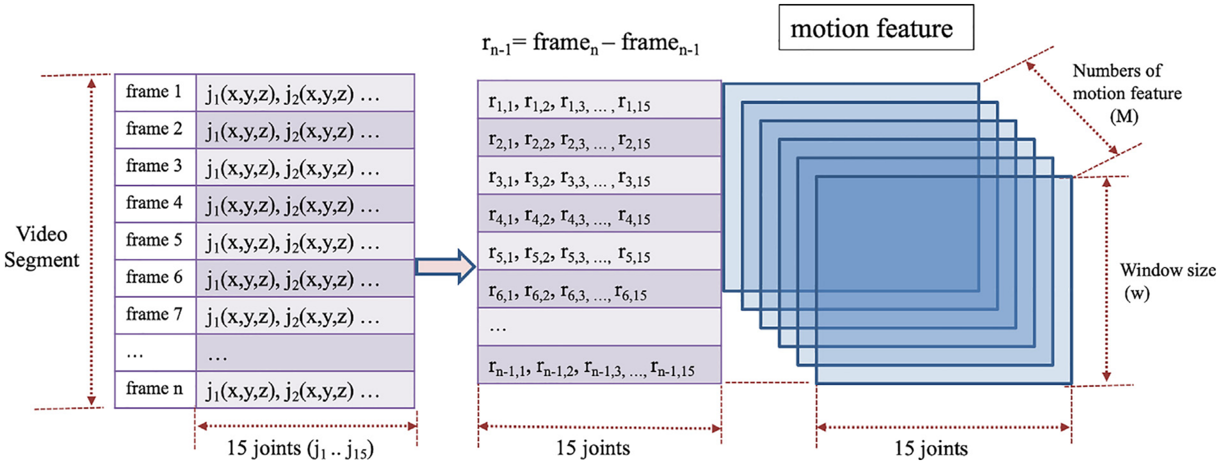

Fig. 2. Method of motion feature extraction.

3.1. Motion feature

We use the sliding window technique on time series data to construct motion features (Chen & Nugent, 2019). Each video segment in the dataset consists of several frames. We will calculate the change in the distance at each joint position coordinate in adjacent frames. The identification of joint motion is carried out by calculating the change in the 3D coordinate distance of each joint in each video segment. The Euclidean Distance formula $\mathrm{r}_i^{\mathrm{t}}$ (1) is used to calculate the change between 3D coordinates (x, y, z) from each ith joint.

$$\Delta r_i^t = \sqrt{(x_i^{t+1} - x_i^t)^2 + (y_i^{t+1} - y_i^t)^2 + (z_i^{t+1} - z_i^t)^2} \tag{1}$$

The results of the calculation of the adjacent distance show that the movement change in frame t changes to frame $t + 1$. Thus, let the combined distance $r_{\mathrm{n,i}}$ with i is the joint position from 1 to 15, the feature matrix R (2) is defined as follows:

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,15} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,15} \\ r_{3,1} & r_{3,2} & r_{3,3} & \cdots & r_{3,15} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{n-1,1} & r_{n-1,2} & r_{n-1,3} & \cdots & r_{n-1,15} \end{bmatrix} \tag{2}$$

Next, several times series frames will be taken by performing motion features. Fig. 2 illustrates the steps of extracting motion features from 3D coordinate data from 15 joints. Several w samples are taken, where w is the window size to form a series of windows. We refer to this series of sliding windows as motion features. Given a video segment that contains one activity, consisting of several frames. First of all, we divide into segments with window size w. Window size is the number of frames taken in each video segment to form a motion feature. Each video segment be M motion features, i.e. $F_1, F_2, \ldots, F_{\mathrm{M}}$ and the sub-feature $F_{1:\mathrm{i}}$ is defined as

$$F_{1:i} = \{F_1, F_2, \ldots ., F_i\} \tag{3}$$

where i is the number of joints.

The number of motion features formed in each video segment will differ from one video segment to another. It depends on the number of frames in each video segment.

Fig. 3 illustrates how motion features are created on windows size w. The motion feature extraction stage will produce a three-dimensional matrix used as input data in the CNN deep learning model.

3.2. Convolutional neural network

Convolutional Neural Network (CNN) has a significant impact on the computer vision field. Scale, rotation and noise are challenges in computer vision AI research. CNN's ability can solve this problem (Mukhopadhyay, 2018). Experiments using CNN conducted in this study include convolutional layers and density layers.

Convolutional Layer: First, we have to define the hyper-parameters for the convolutional layer in most neural network frameworks, i,e.:

- number of filters
- filer size
- stride
- padding
- activation function

The primary purpose of a convolutional layer is to detect features such as edges, lines, colour groups and other visual elements. The more filters we create on the convolutional layer, the more features it can detect.

Dense Layer: The number of neurons indicates the number of neurons that make up this layer. The activation function indicates the type of activation used. Dense layers can implement activation functions such as ReLU, sigmoid or hyperbolic tangent. For example, if the network distinguishes between several activities, such as standing, sitting, and waving, then there are three output neurons. We can apply the Softmax function to the final layer as a probability function. Softmax makes each neuron provide possible image representation for each group.

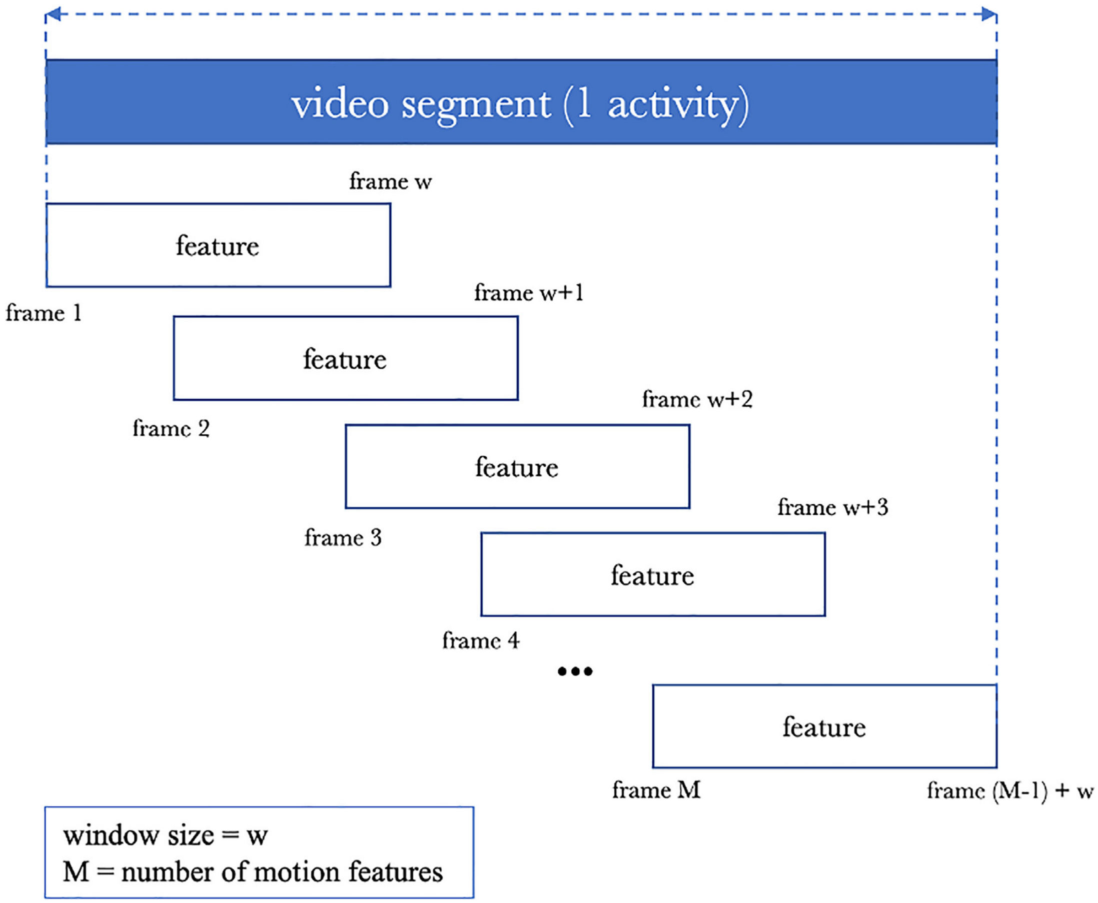

Fig. 3. Ilustration of feature motion extraction in each video segment, window size = 10.

CNN parameters are organized into three-dimensional structural units, known as filters or kernels. Filters are usually square in terms of their spatial dimensions, which are usually much smaller than the layer to which the filter is applied. On the other hand, the depth of the filter is always the same as the applied layer (Alpaydin, 2021).

4. Experiments

We experimented with dataset extraction algorithms to generate motion features and then input them into deep learning models. In classifying human activities through motion feature extraction, we observe several window sizes to prove the effect of window size. The optimal length of the window size must be considered to avoid the risk of identification failure (Ahad et al., 2020). The results will be evaluated to prove the accuracy of the proposed method. Through the confusion matrix, we discuss and analyze the reasons for misclassification or try to decipher the rationality of each error in accuracy per class.

4.1. Dataset

The public data used in this research is from Florence 3D Action (MICC, 2018). The dataset collected at the University of Florence during 2012 has been captured using a Kinect camera. It includes nine activities. Ten subjects were asked to perform the above actions 2/4 times during acquisition. Table 1 shows the features of the dataset used in this study. A-frame consists of video identity, actor identity, activity label and 3D coordinates of 15 joints. Several successive frames make up a video segment about an activity. The dataset contains 215 video segments, with the number of frames per video segment ranging from 7 to 34 for a total of 4016 frames.

An actor as a subject is represented by 15 joints in Kinect as shown in Fig. 4, namely: j_1: Head, j_2: Neck, j_3: Spine, j_4: Left shoulder, j_5: Left elbow, j_6: Left wrist, j_7: Right shoulder, j_8: Right elbow, j_9: Right wrist, j_{10}: Left hip, j_{11}: Left knee, j_{12}: Left ankle, j_{13}: Right hip, j_{14}: Right knee, j_{15}: Right ankle. While the list of activities is shown in Fig. 5, which consists of nine activities: “wave”, “drink from a bottle”, “answer phone”, “clap”, “tight lace”, “sit down”, “stand up”, “read watch”, and “bow”.

Table 1
Dataset features.

Field name	Annotation
video id	1..215
actor id	1..10
activity label	1..9
j_1	(x_head,y_head,z_head)
j_2	(x_neck,y_neck,z_neck)
j_3	(x_spine,y_spine,z_spine)
j_4	(x_leftshoulder,y_leftshoulder,z_leftshoulder)
j_5	(x_leftelbow,y_leftelbow,z_leftelbow)
j_6	(x_leftwrist,y_leftwrist,z_leftwrist)
j_7	(x_rightshoulder,y_rightshoulder,z_rightshoulder)
j_8	(x_rightelbow,y_rightelbow,z_rightelbow)
j_9	(x_rightwrist,y_rightwrist,z_rightwrist)
j_{10}	(x_lefthip,y_lefthip,z_lefthip)
j_{11}	(x_leftknee,y_leftknee,z_leftknee)
j_{12}	(x_leftankle,y_leftankle,z_leftankle)
j_{13}	(x_righthip,y_righthip,z_righthip)
j_{14}	(x_rightknee,y_rightknee,z_rightknee)
j_{15}	(x_rightankle,y_rightankle,z_rightankle)

4.2. Motion features extraction

The number of frames in each video segment in the dataset varies from 7 to 34. In each frame count group, we show the number of video

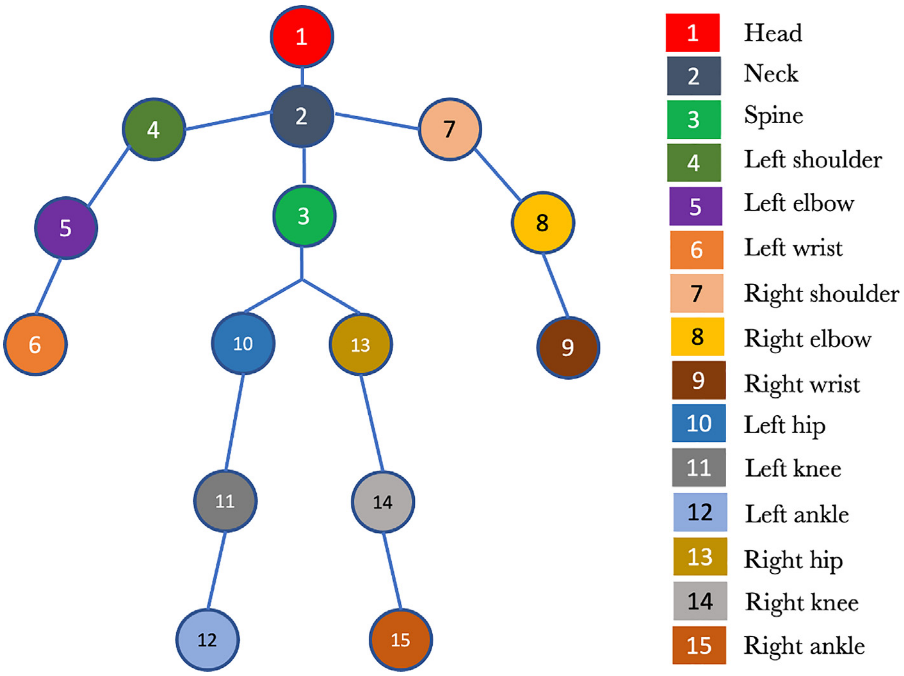

Fig. 4. Fifteen joints (j) detected by means of the Kinect.

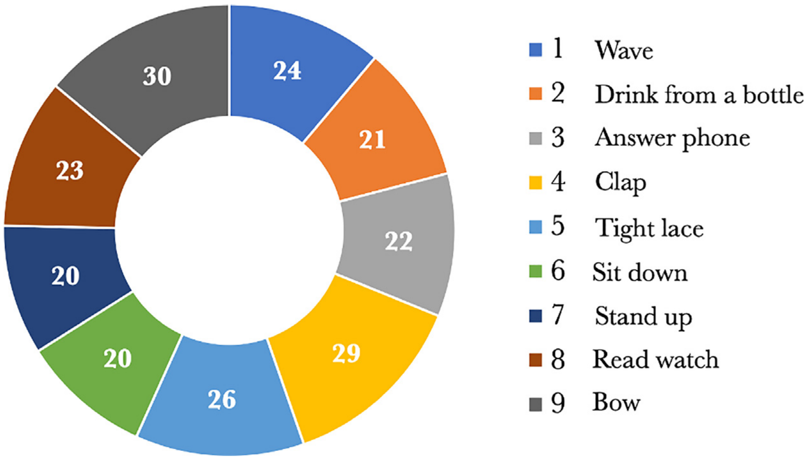

Fig. 5. Activity Frequency on the dataset in 215 video segments.

segments in Table 2. By considering the results of the calculation of the distance between adjacent frames, each video segment will have $n-1$ frames where n is the number of frames for each video segment of the dataset. Let S be the total number of frames in the w window size that will be processed to create the motion feature, according to the (4) formula as follows:

$$S = \sum_{n=7}^{34} n \times vs_n \tag{4}$$

when vs is a number of video segments, each count of frames, n is the value of the number of frames for each video (7 to 34 frames).

The entire frame used in the experiment is 3801. We conducted experiments using different window sizes (w = 7, 10, 13, 16, 19, 22, 25). The smallest window size corresponds to the smallest number of frames, 7, as shown in Table 2. A window size of 7 will cover all frames in the entire video segment. The number of frames used in each tested window size can be seen in Table 3. The table also provides information on the number of motion features created by capturing several frames in each activity in the video segment. In window size 7, the dimensions of the motion features are (2511, 7, 15), which means there are 2511 motion features; the window size is seven at 15 joints. This data dimension is used as input data for deep learning. It is the maximum number of motion features input to the CNN model. In a window of size 10, a dimensional motion feature will be formed (1871, 10, 15). Video segments that have fewer than ten frames will be skipped. Similarly, when we take a window size of 13, the dimensions of the motion feature formed are (1273, 13, 15); this means that some video segments with several frames less than 13 will be skipped. As the window size increases, the number of motion features decreases, which means that the amount of input data to the CNN model also decreases.

Table 3 shows the number of motion features at several window sizes (w) that will be used as CNN model input.

We experiment with seven window sizes represented by w, i.e., 7, 10, 13, 16, 19, 22, and 25, which will form a number of motion features M with a matrix size of $M \times w \times 15$. Our experiment used seven window sizes to prove the effect of window size on the accuracy of the human activity classification model. The larger the window size, the fewer motion features are formed. It will reduce the number of motion features as input data in the CNN model.

Table 2
Total for each group of frames on 215 video segments.

Number of frames	Number of video segments	Total frames
7	2	14
8	1	8
9	5	45
10	8	80
11	7	77
12	7	84
13	12	156
14	18	252
15	19	285
16	18	288
17	15	255
18	13	234
19	14	266
20	17	340
21	10	210
22	12	264
23	15	345
24	4	96
25	1	25
26	5	130
27	5	135
28	1	28
29	2	58
30	1	30
31	2	62
32	0	0
33	0	0
34	1	34
Total	**215**	**3801**

4.3. Random split validation

After the motion feature extraction, we perform the random split validation before the resulting data series becomes input data to the CNN model. All data is involved randomly and split into 80% training and 20% testing data in modelling to provide representative validation results.

4.4. CNN model for classifying human activity

The CNN model in this paper predicts nine activities performed by ten actors who repeat their activities two to four times in 215 video segments. A CNN consists of convoluted layers and a fully connected neural network. CNN architectural design model, as shown in Table 4.

The operation between the filter and the spatial regions in the layer is performed at each position that allows defining the next layer, where the activations retain their spatial relationship from the previous layer (Zhao et al., 2019). The convolution layer consists of neurons arranged to form a filter. The size of the first layer is 32 × 32 × 3, 3 is the number of channels. These three filters will be shifted throughout the image. Each shift will perform a "point" operation between the input and the filter value to produce output. The activation function used is ReLu, where the output value of the neuron can be expressed as 0 if the input is negative. If the input value of the activation function is positive, then the output of the neuron is the value of the activation input itself. The process at the convolution layer produces a feature map with a multidimensional array, so it is necessary to flatten or reshape the feature map to form a vector so that it can be used as input in a fully connected layer. This layer consists of the ReLu activation function, and in the final stage of the model, the SoftMax activation function is used. The input layer on a fully connected network comes from the flattened layer of neurons resulting from 2D convolution. Furthermore, several neurons used are generated from Dense 256. Dense 256 is designed for three layers, then continued with Dense 64. Finally, dense nine layers show the number of categories of 9 activities used so that the parameter obtained is 585, which comes from the calculation of $9 \times 64 + 9$.

The activation function used in this proposed method is the rectified linear unit or ReLU. The ReLu (ρ) of x is just a max value of 0 and x, which means it will return 0 if the input is negative or raw input otherwise, as follows (5) (Indrakumari et al., 2021):

$$\rho(x) = max(x, 0) \tag{5}$$

As the last layer of the neural network, the softmax activation function is used to convert the score into a normalized probability distribution. The SoftMax activation function (6) is defined as

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{6}$$

where i is an index of the output vector. It basically transforms (σ) a vector z with arbitrary real values to a vector with values ranging from 0 to 1, and they are such that they all add up to 1 (Indrakumari et al., 2021). e is the constant math e (2.71828...), and z_i represents the ith element of the input vector z. The softmax function applies an exponential function to each input vector element, then normalizes the resulting value by dividing it by the sum of all the exponential values.

From all the stages described, we develop a model algorithm 1 that is applied to classify the dataset used.

Algorithm 1 ClassActivity ⇐ VideoImage3D

```
Require: $class = 9$
Ensure: class(9) ⇐ Dataset(id, activity, joints(x,y,x))
$frame(activity, joints) \Leftarrow read(ArrayOfDataset)$
if $joint \leq 15$ then
    $ListData \Leftarrow EuclideanDistance(frame.joints_{n+1} - frame.joints_n)$
    while $WindowSize = 7, 10, 13, 16, 19, 22, 24$ do
        $data \Leftarrow array(ListData(WindowSize))$
        $label \Leftarrow array(ListData(frame.activity))$
    end while
end if
procedure TrainTestSplit(0.8,0.2)
    X.Train, y.Train, X.Test, y.Test ⇐ (data,label)
end procedure
procedure ModelCNN(X.Train, y.Train, X.Test, y.Test)
    model(accuracy, loss) ⇐ sequential
    Conv2D(32,32)
    Dropout(rate ⇐ 0.1)
    Conv2D(32)
    Flatten
    Dense(256, 256, 256, 64, 64, 9; activation ⇐ ReLu)
    model.compile
    model.fit
end procedure
Plot Figure
ConfusionMatrix(actual, predicted)
```

4.5. Results and discussion

For validation, we used 1.6 GHz Intel Core i5, 8 GB RAM, 2133 MHz LPDDR3 with a macOS Mojave version 10.14.5. The programming language Python 3.8.1. We used the Florence 3D public dataset (MICC, 2018).

The experiment was carried out with 50 epochs, experimenting on window sizes 7, 10, 13, 16, 19, 22, and 25. Window size 7, the smallest, will cause all video segments to be covered so that the number of frames that will make up the motion feature will be maximized. A small sampling is proven not to produce the best accuracy value. The sequence of movement patterns that the system will detect is also getting smaller. The following experiment tested the addition of window size. It turned out that the larger the window size, the better the recognition made by the system so that at a window size of 16, the accuracy value reached 94.08%. The experimental results are shown in

Table 3
Number of motion features, window size and number of joints as matrix size for CNN model input.

Window size	Number of frames used	Percentage of frames used	Number of motion features formed	Matrix size
7	3801	100%	2511	(2511, 7, 15)
10	3734	98%	1871	(1871, 10, 15)
13	3493	92%	1273	(1273, 13, 15)
16	2800	74%	760	(760, 16,15)
19	2023	53%	403	(403, 19,15)
22	1207	32%	178	(178, 22, 15)
25	502	13%	70	(70, 25, 15)

Table 4
Architecture of implemented CNN Model (window size = 16)

Layer	Filter number	Ouput shape	Activation	Param #
Convolution 2D	32, kernel size = 3	(None, 16, 15, 32)	ReLu	320
Convolution 2D	32, kernel size = 3	(None, 16, 15, 32)	ReLu	9248
Dropout rate = 0.01	–	(None, 16, 15, 32)	–	0
Convolution 2D	32, kernel size = 3	(None, 16, 15, 32)	ReLu	9248
Flatten	–	(none, 7680)	–	0
Dense	–	(none, 256)	ReLu	1966336
Dense	–	(none, 256)	ReLu	65792
Dense	–	(none, 256)	ReLu	65792
Dense	–	(none, 64)	ReLu	16448
Dense	–	(none, 64)	ReLu	4160
Dense	–	(none, 9)	SoftMax	585

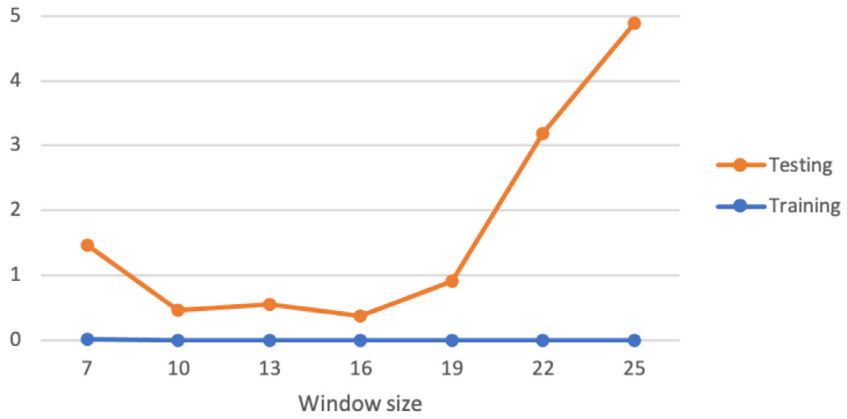

Fig. 6. Experimental results: Training and validation loss for window size 7, 10, 13, 16, 19, 22 and 25.

Table 5
Experimental Result: Loss and accuracy on training and testing data.

Window size	Training (80%)		Testing (20%)	
	Loss	Accuracy	Loss	Accuracy
7	2.22E–05	100%	1.4669	82.31%
10	0.0041	100%	0.4549	90.93%
13	1.83E–05	100%	0.54	93.33%
16	1.51E–05	100%	0.364	**94.08%**
19	2.11E–05	100%	0.8995	90.12%
22	1.18E–05	100%	3.1818	86.11%
25	2.60E–06	100%	4.8933	64.29%

Fig. 6 which shows the loss of training data and testing data, while the accuracy values for training data and test data are shown in Fig. 7.

Loss and accuracy values for some of the tested window sizes are shown in Table 5. The experimental results show that when the window size gets bigger, it will result in fewer input data for the model; this is proven to reduce the recognition ability of the detected activity. It appears that at the window sizes 19, 22, and 25, the accuracy values are decreasing.

The results of the confusion matrix in Tabel 6 show the predicted results of 9 actual activities classification. There appears to be a correlation between the confusion matrix display and the value that indicates the number of activities predicted to be correct from the actual activity information detected as incorrect activity. In the experiment using a window size of 16, which has the highest accuracy value, the prediction results based on the confusion matrix are obtained. Table 6 shows the 9 activity classes mentioned. The top 95% value of the row gives the recognition result or accuracy for the ‘answer phone’ class. However, the ‘answer phone’ class is confused with the ‘clap’ class by 5%. The values in the first row are summed up as 100%. From this Table, we can predict that the ’drink from a bottle’ class is highly confused or misclassified by the ’read a watch’ activity class (20%). Similarly, we can see that the ‘read watch’ class is also confused with the ‘drink from a bottle’ class (10%). So, both the ’drink from a bottle and ‘read watch’ classes get confused with each other. So accuracy can be improved either way if some preprocessing or enhancement can distinguish these two confusing classes of activity. Besides that, the ‘read watch’ class is confused with ’sit down’ class by 3% and ’stand up’ class by 3%. Using the Python programming language in this research, it is shown in Fig. 8 visualization of different confusion matrices for window size 16. A confusion matrix with various colour combinations is demonstrated, and numbers indicate the number of detected activities. The diagonal value is the level of accuracy for the number of activity classes that have been recognized. The slight difference in joint movement between one activity causes prediction errors, but choosing the optimal window size can reduce prediction errors.

Table 6
Normalized confusion matrix for 9 activities classification with a window size of 16. Rows represent basic truths and columns represent predictions.

	1	2	3	4	5	6	7	8	9
1 - answer phone	**0.95**	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00
2 - bow	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3 - clap	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
4 - drink from a bottle	0.00	0.00	0.00	**0.80**	0.20	0.00	0.00	0.00	0.00
5 - read watch	0.00	0.00	0.00	0.10	**0.83**	0.03	0.03	0.00	0.00
6 - sit down	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00
7 - stand up	0.00	0.00	0.00	0.17	0.00	0.00	**0.83**	0.00	0.00
8 - tight lace	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.97**	0.03
9 - wave	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**

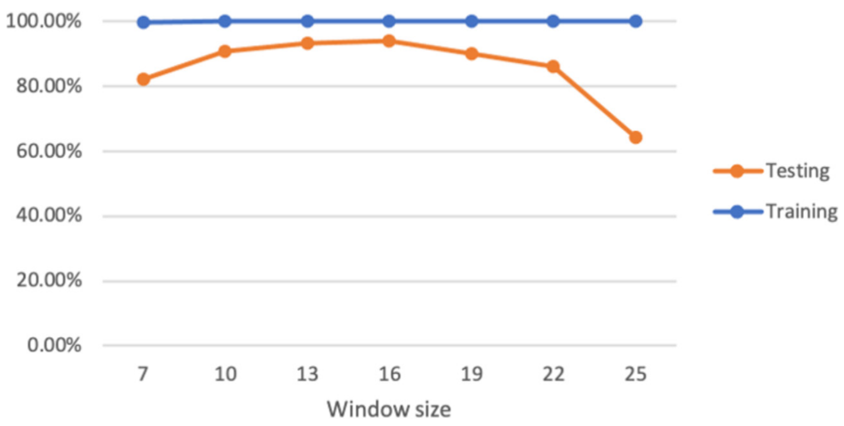

Fig. 7. Experimental results: training and validation accuracy for window size 7, 10, 13, 16, 19, 22 and 25.

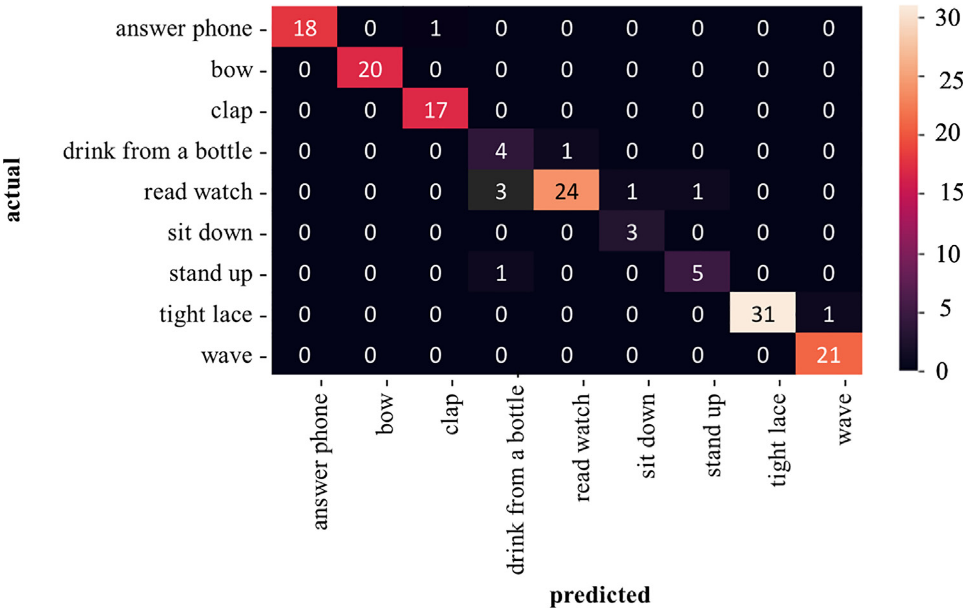

Fig. 8. Confusion matrix for 9 activities classification with a window size of 16. Shows the number of true and false predictions.

Based on the confusion matrix on Fig. 8, assuming the positive activity is ‘answer phone’, we compiled Table 7 to measure the number of data correctly classified (True Positives (TP), True Negatives (TN)) and incorrectly classified (False Positives (FP), False Negatives (FN)) by the model. Then we calculate accuracy, precision, recall, and F1-score as performance evaluation metrics of a classification model using the following formula (Bramer, 2016):

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 - score = \frac{2 \times (precision \times recall)}{precision + recall} \tag{10}$$

By comparing the predicted labels of a model with the true labels of the data and counting the number of correct and incorrect predictions according to formulas (7), (8), (9), and (10), we obtain accuracy = 99.3421%, precision = 100%, recall = 0.94736842, and F1-score = 0.97297297.

Table 7
Confusion matrix for evaluating the performance of a classification model.

	Predicted positive	Predicted negative
Actual Positive	TP = 18	FN = 1
Actual Negative	FP = 0	TN = 133

Table 8
The results of comparing the model implementation with window size = 16 using the UTKinect-Action3D dataset.

Dataset	Activity	Joints	Frames	Accuracy
Florence-3D	9	15	760	94.08%
UTKinect-Action3D	10	20	220	93.18%

Table 8 shows the experimental results using the UTKinect-Action3D dataset (joints_s01_e02.txt) as part of a study on action recognition from deep sequences (Xia et al., 2012), compared to the Florence 3D dataset as experimental data in this paper. From UTKinect-Action3D, we use movement data of a person performing ten activities ('walk', 'sitDown', 'standUp', 'pickUp', 'carry','throw', 'push', 'pull', 'waveHands', 'clapHands'). The video capture results using a single stationary Kinect camera produce a series of frames containing the frame number and coordinates (x, y, z) of the 20 joints. The experimental results show a loss value of 0.1964 and an accuracy of 93.18%. So it is proven that our model can accurately classify different datasets.

5. Conclusion

This paper proposes a method using 3D motion feature extraction as part of the data preparation stage for a CNN model that will classify nine human activities. The extraction step begins by calculating the Euclidean distance of each joint in adjacent frames in a video segment. Furthermore, several frames are taken to form a time series sequence in one activity segment to form a motion feature validated using 80/20 random split validation, which is then used as input for the CNN model. The design of the CNN model consists of three convolution layers using ReLu activation and three layers on a fully connected network. The activation function used in five fully connected network layers uses the ReLu activation function, while the last layer uses the SoftMax activation function with nine parameters with the number of activities to be classified. The experimental results using seven variations of window sizes show that a window size that is too small provides activity sequence data that the model does not sufficiently recognize. At the same time, a window size value that is too large will reduce the accuracy value in recognizing activities because fewer motion features are formed. This experiment confirmed that choosing a window size is very important for the best accuracy. In the experiment using seven window sizes, the selection of window size 16 resulted in the best accuracy of 94.08% in recognizing human activities. Using a confusion matrix to measure the classifier's performance by classifying an activity correctly or misclassified, we obtain accuracy = 99.3421%, assuming a positive value on the 'answer phone' activity. The future work is to develop activity recognition research to identify the activities of the elderly living alone by identifying the duration of activity obtained from sensors to detect abnormal activity in the elderly. This research is the basis for developing an intelligent assistance system for the elderly.

CRediT authorship contribution statement

Endang Sri Rahayu: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Visualization, Project administration. **Eko Mulyanto Yuniarno:** Conceptualization, Methodology, Visualization, Supervision, Software, Formal analysis, Investigation, Data curation. **I. Ketut Eddy Purnama:** Formal analysis, Visualization, Supervision, Funding acquisition. **Mauridhi Hery Purnomo:** Conceptualization, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mauridhi Hery Purnomo reports financial support, administrative support, article publishing charges, and writing assistance were provided by Ministry of Education, Culture, Research, and Technology (953/PKS/ITS/2021).

Data availability

Data will be made available on request.

References

Ahad, A. R., Antar, A. D., & Ahmed, M. (2020). *IoT sensor-based activity recognition: Human activity recognition* (p. 214). http://dx.doi.org/10.1007/978-3-030-51379-5.

Ahmad, T., Mao, H., Lin, L., & Tang, G. (2020). Action recognition using attention-joints graph convolutional neural networks. *IEEE Access*, *8*, 305–313. http://dx.doi.org/10.1109/ACCESS.2019.2961770.

Aldahoul, N., Karim, H. A., Sabri, A. Q. M., Tan, M. J. T., Momo, M. A., & Fermin, J. L. (2022). A comparison between various human detectors and CNN-based feature extractors for human activity recognition via aerial captured video sequences. *IEEE Access*, *10*, 63532–63553. http://dx.doi.org/10.1109/ACCESS.2022.3182315.

Alpaydin, E. (2021). *Neural networks and deep learning*. http://dx.doi.org/10.7551/mitpress/13811.003.0007.

Bennamoun, M., Guo, Y., Tombari, F., Youcef-Toumi, K., & Nishino, K. (2020). Guest Editors' introduction to the special issue on RGB-D vision: Methods and applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*(10), 2329–2332. http://dx.doi.org/10.1109/TPAMI.2020.2976227.

Bramer, M. (2016). *Introduction to data mining* (pp. 1–8). http://dx.doi.org/10.1007/978-1-4471-7307-6_1.

Chen, Z., Czarnuch, S., Dove, E., & Astell, A. (2023). Automated recognition of individual performers from de-identified video sequences. *Machine Learning with Applications*, *11*(January), Article 100450. http://dx.doi.org/10.1016/j.mlwa.2023.100450.

Chen, L., & Nugent, C. D. (2019). *Human activity recognition and behaviour analysis*. Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-19408-6.

Chen, Y., Yu, L., Ota, K., & Dong, M. (2018). Robust activity recognition for aging society. *IEEE Journal of Biomedical and Health Informatics*, *22*(6), 1754–1764. http://dx.doi.org/10.1109/JBHI.2018.2819182.

Fu, Y. (2016). *Human activity recognition and the activities holter* (p. VII, 174). Switzerland: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-27004-3.

Gaglio, S., Re, G. L., & Morana, M. (2015). Human activity recognition process using 3-D posture data. *IEEE Transactions on Human-Machine Systems*, *45*(5), 586–597. http://dx.doi.org/10.1109/THMS.2014.2377111.

Gochoo, M., Tan, T. H., Liu, S. H., Jean, F. R., Alnajjar, F. S., & Huang, S. C. (2019). Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN. *IEEE Journal of Biomedical and Health Informatics*, *23*(2), 693–702. http://dx.doi.org/10.1109/JBHI.2018.2833618.

Goddard, N. H. (2021). *Human activity recognition challenge, Vol. 199* (pp. 147–170). Singapore: Springer International Publishing, http://dx.doi.org/10.1007/978-981-15-8269-1.

Huang, J., Lin, S., Wang, N., Dai, G., Xie, Y., & Zhou, J. (2020). TSE-CNN: A two-stage end-to-end CNN for human activity recognition. *IEEE Journal of Biomedical and Health Informatics*, *24*(1), 292–299. http://dx.doi.org/10.1109/JBHI.2019.2909688.

Indrakumari, R., Poongodi, T., & Singh, K. (2021). *Introduction to deep learning* (pp. 1–22). http://dx.doi.org/10.1007/978-3-030-66519-7_1, arXiv:2003.03253.

Kim, H. G., & Kim, G. Y. (2020). Deep neural network-based indoor emergency awareness using contextual information from sound, human activity, and indoor position on mobile device. *IEEE Transactions on Consumer Electronics*, *66*(4), 271–278. http://dx.doi.org/10.1109/TCE.2020.3015197.

Kong, W., He, L., & Wang, H. (2021). Exploratory data analysis of human activity recognition based on smart phone. *IEEE Access*, *9*, 73355–73364. http://dx.doi.org/10.1109/ACCESS.2021.3079434.

Lee, S. M., Yoon, S. M., & Cho, H. (2017). Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE international conference on big data and smart computing* (pp. 131–134). IEEE, http://dx.doi.org/10.1109/BIGCOMP.2017.7881728.

Li, Q., Lin, W., & Li, J. (2018). Human activity recognition using dynamic representation and matching of skeleton feature sequences from RGB-D images. *Signal Processing: Image Communication*, *68*(January), 265–272. http://dx.doi.org/10.1016/j.image.2018.06.013.

Li, G., Yang, S., & Li, J. (2020). Edge and Node Graph Convolutional Neural Network for Human Action Recognition. In *Proceedings of the 32nd Chinese control and decision conference* (pp. 4630–4635). http://dx.doi.org/10.1109/CCDC49329.2020.9163951.

Liu, J., Shahroudy, A., Xu, D., Kot, A. C., & Wang, G. (2018). Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 3007–3021. http://dx.doi.org/10.1109/TPAMI.2017.2771306, arXiv:1706.08276.

MICC (2018). Florence 3D actions dataset - actions from depth cameras. URL: https://www.micc.unifi.it/resources/datasets/florence-3d-actions-dataset/.

Mukhopadhyay, S. (2018). *Deep learning and neural networks* (pp. 99–119). http://dx.doi.org/10.1007/978-1-4842-3450-1_5.

Pham, C., Nguyen-Thai, S., Tran-Quang, H., Tran, S., Vu, H., Tran, T. H., & Le, T. L. (2020). SensCapsNet: Deep neural network for non-obtrusive sensing based human activity recognition. *IEEE Access*, *8*, 86934–86946. http://dx.doi.org/10.1109/ACCESS.2020.2991731.

Phyo, C. N., Zin, T. T., & Tin, P. (2019). Deep learning for recognizing human activities using motions of skeletal joints. *IEEE Transactions on Consumer Electronics*, *65*(2), 243–252. http://dx.doi.org/10.1109/TCE.2019.2908986.

Seidenari, L., Varano, V., Berretti, S., Del Bimbo, A., & Pala, P. (2013). Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 479–485). http://dx.doi.org/10.1109/CVPRW.2013.77.

Su, B., Wu, H., Sheng, M., & Shen, C. (2019). Accurate hierarchical human actions recognition from kinect skeleton data. *IEEE Access*, *7*, 52532–52541. http://dx.doi.org/10.1109/ACCESS.2019.2911705.

Wang, H., & Wang, L. (2018). Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection. *IEEE Transactions on Image Processing*, *27*(9), 4382–4394. http://dx.doi.org/10.1109/TIP.2018.2837386.

Wang, H., Yu, B., Xia, K., Li, J., & Zuo, X. (2021). Skeleton edge motion networks for human action recognition. *Neurocomputing*, *423*, 1–12. http://dx.doi.org/10.1016/j.neucom.2020.10.037.

Xia, L., Chen, C., & Aggarwal, J. (2012). View invariant human action recognition using histograms of 3D joints. In *Computer vision and pattern recognition workshops* (pp. 20–27). IEEE Computer Society Conference.

Zhang, X., Xu, C., & Tao, D. (2020). Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 14321–14330). http://dx.doi.org/10.1109/CVPR42600.2020.01434.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237. http://dx.doi.org/10.1016/j.ymssp.2018.05.050.