Vision-Based Biomechanical Markerless Motion Classification

Yu Liang Liew, Jeng Feng Chin
School of Mechanical Engineering, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia
chinjengfeng@usm.my

Abstract. This study used stick model augmentation on single-camera motion video to create a markerless motion classification model of manual operations. All videos were augmented with a stick model composed of keypoints and lines by using the programming model, which later incorporated the COCO dataset, OpenCV and OpenPose modules to estimate the coordinates and body joints. The stick model data included the initial velocity, cumulative velocity, and acceleration for each body joint. The extracted motion vector data were normalized using three different techniques, and the resulting datasets were subjected to eight classifiers. The experiment involved four distinct motion sequences performed by eight participants. The random forest classifier performed the best in terms of accuracy in recorded data classification in its min-max normalized dataset. This classifier also obtained a score of 81.80% for the dataset before random subsampling and a score of 92.37% for the resampled dataset. Meanwhile, the random subsampling method dramatically improved classification accuracy by removing noise data and replacing them with replicated instances to balance the class. This research advances methodological and applied knowledge on the capture and classification of human motion using a single camera view.

Key words: vision, single camera, markerless, stick model, human motion, motion classification, data mining.

1. Introduction

Human motion analysis entails sensing the human body and extracting static or dynamic data from it in the form of gestures, behaviors, and actions [34]. It emerges as a critical component in operation studies to evaluate performance, such as in sports performance analysis [16], medical rehabilitation [61], video surveillance [18], and virtual reality gaming [28]. In industrial engineering, motion classification aids in verifying the presence of operator action, and the absence of specific actions can lead to process defects and incompletion [1], as well as safety concerns [22].

Fixed-axis and parallel projection are used in vision-based motion classification models to calibrate feature points relative to the previous position of human body parts [53]. The general framework of a vision-based motion classification model includes movement scene capture, human tracking, humans and motion representation, motion recognition, and classification into its respective class [37]. In general, the model processes each frame of the motion video in accordance with its frame sequences. When a human is detected in a video frame, the frame image is segmented to obtain the region of interest [41]. The motion can then be visualized by combining a stick-figure model, a volumetric model, 2D

blobs, and a geometric drawing [2]. Among these methods, the stick-figure model provides a simple but effective solution for estimating a human posture at a specific frame. The stick-figure model is a skeleton-like model composed of several keypoints, each of which represents a coordinate of a body part. These body parts function as moving joints, and their motion vectors are compared with those of the previous frame [10]. The motion is classified by comparing the movement of the person between frames [52].

Most motion capture methods place markers on the body parts of the subject to track the change in motion. However, such a setting necessitates a planned experiment environment with informed subjects, which makes it impractical in a real-life scenario where preparation or interference with the observed activity is not permitted. Several studies used multi-camera recording to reconstruct the 3D view of moving human bodies in the absence of motion capture markers. Nakano et al. [38] used multiple video cameras from different angles to capture frames from various perspectives, which they then merged into 3D visualization using the direct linear transformation method. Meanwhile, Hasler et al. [23] used audio synchronization to conventional video camera recordings and then 3D mesh reconstruction using a feature-based approach.

Kanko et al. [26] used a single 2D camera view to perform gait analysis and movement estimation using a deep learning approach. Tsuji et al. [49] used a single camera to capture video of general movements of infants and identify abnormalities in those movements. Then, they utilized a framework that begins with feature extraction using computer vision and progresses to movement analysis using formula calculations. Finally, they conducted movement classification using a feedforward-type network known as a log-linearized Gaussian mixture network. Zult et al. [63] demonstrated that a conventional video camera could extract the valid keypoints of body parts in the video frame based on the markers using a low-cost 2D camera system. Using a computer vision module such as OpenPose [6], the markers could be replaced by virtual coordinate points [57]. For example, Kim et al. [27] used the OpenPose module to predict knee and hip movement angles in a video captured with a smartphone camera. The validity of this OpenPose-based system with the automated post-processing algorithm has shown early promise, but it may require further verification.

This study investigated the markerless motion classification approach, with motion video captured using a single 2D camera view. The markerless motion classification model classified manual operations extracted from motion video by using the stick model augmentation. This study has two objectives. The first objective is to develop a descriptive model for motion classification based on the overlay of a stick-figure model onto the motion of the operator in video frames. The second objective is to determine the best motion classification strategy by assessing the accuracy of the motion classification model using data mining classifier algorithms. The research advances methodological and applied knowledge on the capture and classification of human motion using a single

camera view. The use of a single camera has cost, configuration, and maintenance benefits. The method can be used in real-world industry applications, such as in analyzing operator performance during a repetitive manufacturing process.

The structure of the manuscript is provided. It begins with an overview of the research context, followed by a brief review of the literature on human motion segmentation, stick-figure models, and motion classification. The following section 3 describes the research methodology, which includes the experiment setup, motion data extraction and computation, and motion classification. Section 4 contains the results and discussion. The final section 5 elaborates the conclusion.

2. Literature review

Motion segmentation is a preprocessing stage of motion analysis that is used to cluster long frame sequences depicting human actions into several shorter, non-overlapping video segments. Subspace clustering and temporal data clustering are two popular clustering methods in the literature. Subspace clustering works by searching a dataset for subspaces and clusters and categorizing data into new distinct spaces based on similar features. For example, Xia et al. [55] combined sparse subspace clustering and a robust kernel low-rank representation method for motion recognition. However, the method ignores the temporal correlation between successive frames. Temporal data clustering divides large amounts of sequential data into non-overlapping chunks. Wang et al. [51] highlighted the importance of temporal information in achieving accurate model performance. However, transfer learning is required to overcome the unpredictability of the results because the temporal clustering method is unsupervised.

Several recent studies used transfer learning to visualize object motion using existing datasets, which is due to that prior knowledge from related source data improves feature identification. Several works partially, such as [62] or fully adapted transfer learning by using deep neural network classifier parameters. They are useful, particularly for detecting multiple people in the same image frame [45].

Rubino et al. [42] proposed semantic motion detection, which uses semantic information to identify object matches between two views. Its underlying principle is similar to that of the convolutional neural network model, which uses patterns from training data to identify features in target data. Simonyan and Zisserman [46] proposed a two-stream convolutional network model with spatial and temporal networks. With prior knowledge of training data from the optical flow model, this model identifies the moving action in the testing video. Meanwhile, Zhou and He [60] used the recurrent network model to estimate the human body region in the image by transforming the image into a pose heatmap. The heatmap would be used to evaluate the coordinates of body joints, and these coordinates are critical for building the stick-figure model.

A stick-figure model is a skeleton-like structure used to represent important body

joints and track body motion patterns [21]. Annotations of keypoints from the body pose estimation are used to accomplish this model. Handcrafted features, such as histogram of oriented gradient, are used in the previous stick model. However, the accuracy of the identified keypoints is below the acceptable range [13]. Single-person or multi-body human body position estimations are used in modern times. The single-person approaches locate body parts through direct regression and heatmap conversion [14]. Chan et al. [9] used a mathematical regression coefficient model to simplify the 2D stick model of human motion for direct regression. Figure 1 depicts the construction of the 2D stick model, which is composed of several points of body parts and lines. The model presents a more straightforward interpretation by using joints as calculation points.

However, the regression-based stick model construction always necessitates additional procedures to accurately map the feature points onto the subject in an image. Carreira et al. [8] added a corrective measure to the neural network model structure by including a simple error feedback connection. The predicted error was fed back into the network in

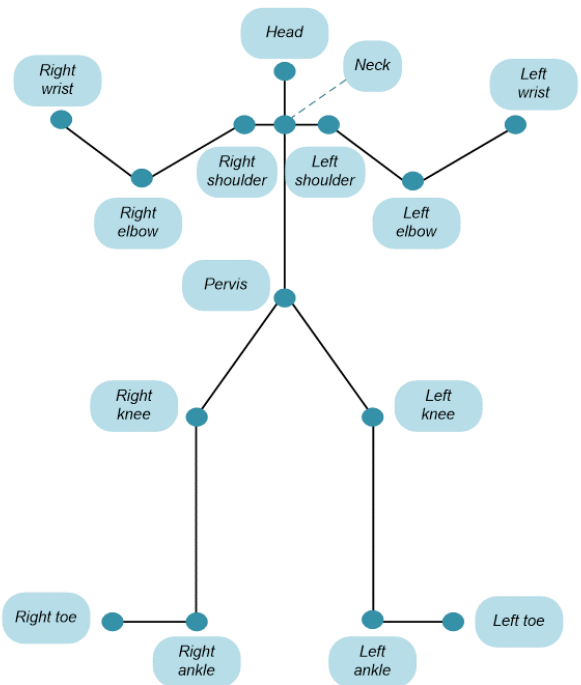


Fig. 1. A simple two-dimensional stick model.

the form of backpropagation to gradually improve keypoint location prediction. Luvizon et al. [35] presented the soft-argmax operation, which is an improved method. After this operation is integrated into the deep convolutional neural network, it can convert the feature maps directly to joint coordinates by finding the maximum values from the target functions. This new method produces results that are comparable to those of the heatmap-based framework. However, unlike the heatmap-based approach, expanding this method into multi-person cases is problematic.

The detection-based framework is typically built on deep learning datasets that have been pre-trained using thousands of human images. Sun et al. [47] used a convolutional neural network with two-stride convolutions to reduce resolution and a main body that outputs feature maps to implement their approach. At the network's end, the regressor estimates the keypoint positions by evaluating the loss function of the heatmap using comparisons between predicted and ground-truth heatmaps.

Various motion classification techniques have been proposed. Switonski et al. [48] investigated data mining for markerless motion extraction in motion capture data. They used dynamic time warping (DTW) technique to classify the human motion data into gait patterns. In time-series data, the model identifies variations in the orientation of motion capture and subject for motion recognition. It calculates the angles in the joint data and the classification probability with the minimum distance classifiers (MDC). MDC is combined with k-nearest neighbor classifiers to maximize the accuracy and consistency of both types of classifiers. Schneider et al. [44] used the DTW approach to evaluate warping distance after annotating the skeleton model using the OpenPose module dataset. Prior to applying the classifier, the image data in coordinates were normalized to condense the data range into a smaller number. Thereafter, nearest neighbor classifiers were used to classify the warping distance of time-series data. The results still have some limitations, such as reliance on the representativeness of the dataset, poor recognition precision when noise reduction is required, and the need for motion capture marker setup.

Qian et al. [40] evaluated multi-class support vector machine (SVM) classifiers by removing the background and extracting the centroids and instantaneous speed of human motion. The frame sequence comparison produces a contour coding of motion energy image with a square-to-circular coordinate transformation, which converts plane coordinates to polar coordinates. SVMs were also used as classifiers in the study by Choi et al. [11] study to classify the gait motion pattern. The joint angle and distances between body parts are among the parameters used. SVM is an excellent option for accurately recognizing motion, but many more classifiers have yet to be tested in motion classification.

Yang and Zhao [58] used decision tree classifiers to determine the motion class of firefighters, but string-type descriptions rather than numbers were utilized as attributes. Zhang et al. [59] employed an interactive system to classify six different motions using three classifiers: naïve Bayes, SVM, and random forest. The results showed that the

Tab. 1. Descriptions for experimental motion activities.

Motion Activity	Description
Moving box	Bend down the body, lift the box with two hands, stand upright, walk a few steps, bend down the body, put down the box, resume to a standing position.
Moving pail	Bend down the body, lift pail by its handle with one hand, stand upright, walk a few steps, bend down the body, put down the pail, resume to a standing position.
Sweeping	Grasp a broom with one hand, move the broom down until its brush touching the floor, pull the broom to sweep the dirt, lift the broom up.
Mopping	Bend down the body, lift the box with two hands, stand upright, walk a few steps, bend down the body, place the box down, and resume standing. Bend down the body, lift the pail by the handle with one hand, stand upright, walk a few steps, bend down the body, set the pail down, and resume standing. Grasp a broom with one hand, lower the broom until the brush touches the floor, pull the broom to sweep the dirt, and then raise the broom. Grasp a mop with two hands, slightly bend the body, move the mop in one direction until it touches the floor, then reverse the mop movement.

random forest classifier has the highest classification accuracy when using position and vector data. Li et al. [31] investigated the motion recognition model using the random forest algorithm and the difference in normalized joint coordinates between keyframes. Fong et al. [17] agreed that the random forest classifier performs the best using position and vector data from the skeleton model. It outperforms the neural network approach and other traditional classifiers in terms of classification accuracy.

3. Methodology

3.1. Experimental motion selection

The experiment was designed to involve activities observable in common full body operations. As described in Table 1, the motion activities featured in the experiment are moving carton box, moving pail, sweeping floor, and mopping floor. Moving carton box and moving pail are highly similar operations, as well as sweeping and mopping floor. The intention is to create complexity in learning when the system is being presented with highly similar datasets.

The variation in human action influences pose recognition. Eight participants between the ages of 23 and 24 volunteered for the motion video collection to account for the abovementioned effect. Each participant was required to complete a series of aforementioned activities in various settings. Different backgrounds (outdoor and indoor) and light conditions were used in the video sample collection given that video backgrounds affected motion recognition using a markerless system [5]. The outdoor used natural light, whereas the indoor light conditions could be bright or dim.

The motion classification samples were collected at the university student hostels. The motion recording was conducted with a digital single-lens reflex (DSLR) camera, specifically a Nikon DSLR D3200 model, with a frame rate of 60 frames per second and a video frame size of 7201080 pixels in three color channels. During video capture, a camera tripod stand supports the camera and fixes its position. Figure 2 depicts and labels the camera setup parallel to the motion activity. During video capture, each participant was required to face the camera parallelly.

A participant repeated each motion activity 10 times, which were recorded all in the same video. All sample videos were manually trimmed into individual activity videos by using video editing software. The first three segmented videos of each sample video were considered motion warm-up and were excluded from the subsequent processing stage. A total of 100 videos from each motion class were chosen at random for further processing. All 400 videos were uploaded to Google Drive in folders named after the motion class to be processed by programming.

The stick model augmentation estimates body part position using the COCO dataset [33]. By associating joint coordinates with individuals, the COCO dataset has been used in multi-person tracking and keypoint annotations [30]. The COCO dataset contains over 200,000 labeled object instances and at least 250,000 human samples. The dataset includes annotations and information for all body part instances, which aid in segmentation and estimation of keypoint coordinates. The COCO dataset was used to train the model for object detection and estimation using transfer learning. As shown in Table 2, 18 points per person had to be recognized from the COCO dataset onto each human image. Python was used to annotate the stick model keypoints and lines onto the human body in the video frames for the stick model overlay. The body joint position was estimated using OpenPose [24], which has been integrated with OpenCV [36].

With 4D blobs, the image was converted into image data. The blobs were fed into the trained neural network, which identified the maximum points in the object area and detected the objects. The architecture of the pre-trained network was divided into two branches: the top branch, which predicts the confidence map, and the bottom branch, which estimates the affinity field. Affinity field refers to the storage of unstructured pairwise relationships between body parts in a field [7].

The code was written in Python and executed in Google Colab [20] using the Python 3

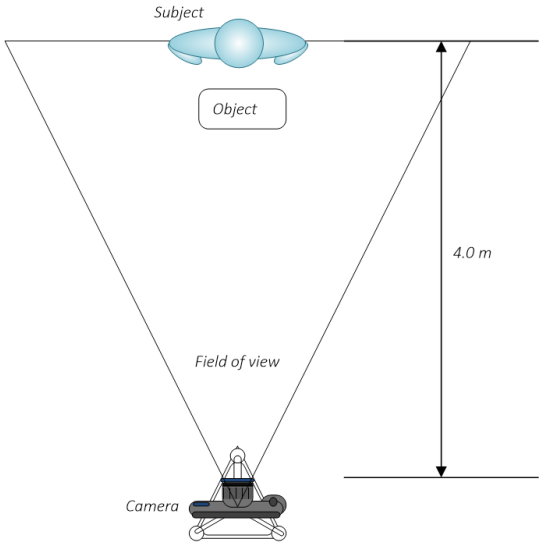


Fig. 2. Motion video capturing scene setup.

Tab. 2. Representation of each number for body joints.

Number	Body Joint	Number	Body Joint
0	Nose	9	Right knee
1	Neck	10	Right ankle
2	Right shoulder	11	Left hip
3	Right elbow	12	Left knee
4	Right wrist	13	Left ankle
5	Left shoulder	14	Right eye
6	Left elbow	15	Left eye
7	Left wrist	16	Right ear
8	Right hip	17	Left ear

Google Compute Engine Tensor processing unit backend [19] with 35.25 GB of high-RAM. All videos were saved in Google Drive folders. The COCO dataset was imported, and the keypoints were sequentially paired (Table 3) with a different number to represent the various body joints identified in Table 2.

The flowchart (Figure 3) summarized the algorithm for overlaying the stick figure. To save computation power, all experimental motion videos were annotated with keypoints and line connections based on specified body part pairings every half a second.

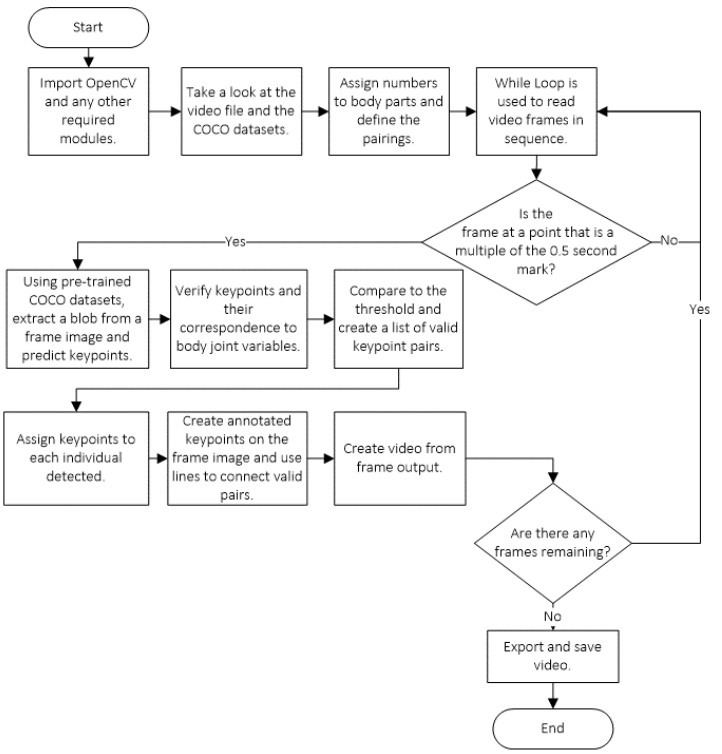


Fig. 3. Flowchart for programming model augmentation using stick figures.

3.2. Data collection and calculation for motion

Keypoint estimation in the stick model overlay was used to calculate the position coordinates for each body joint in a frame. Positions, velocity, and acceleration were among the data extracted from the stick-figure model. Eyes and ears were irrelevant to motion evaluation. Thus, only the first 14 body joints were counted in the extracted data. Owing to the single 2D view of the video, the initial velocities of body parts were calculated for the x- and y-axes only in the motion classification model. Meanwhile, the cumulative velocity and acceleration were used to account for the time-series effect of the video. Table 4 lists the extracted motion variables with n, which indicates the representation number of body joint plus one. The representation number of body joint can be found in Table 2. The initial velocities of body joints were calculated using Equations (1) and (2) for x- and y axes, respectively.

$$u_{x_n} = \frac{x_{1_n} - x_{0_n}}{t}, \tag{1}$$

$$u_{y_n} = \frac{y_{1_n} - y_{0_n}}{t}, \tag{2}$$

where:
x_{1_n} – x-axis coordinate at the first instance for body joint n,
x_{0_n} – x-axis coordinate at the start for body joint n,
y_{1_n} – y-axis coordinate at the first instance for body joint n,
y_{0_n} – y-axis coordinate at the start for body joint n,
t – time interval, here equal to 0.5.

Equations (3) and (4) were used to calculate the cumulative velocity of a body part in the x and y directions, respectively. Meanwhile, (5) and (6) defined the equations for calculating the cumulative acceleration of body parts in the x and y-axes, respectively.

Tab. 3. Body joints pairing with the number indication.

Number Pair	Body Joints Pairing	Number Pair	Body Joints Pairing
1,2	Neck – Right shoulder	11,12	Left hip – Left knee
1,5	Neck – Left shoulder	12,13	Left knee – Left ankle
2,3	Right shoulder – Right elbow	1,0	Neck – Nose
3,4	Right elbow – Right wrist	0,14	Nose – Right eye
5,6	Left shoulder – Left elbow	14,16	Right eye – Right ear
6,7	Left elbow – Left wrist	0,15	Nose – Left eye
1,8	Neck – Right hip	15,17	Left eye – Left ear
8,9	Right hip – Right knee	2,17	Right shoulder – Left ear
9,10	Right knee – Right ankle	5,16	Left shoulder – Right ear
1,11	Neck – Left hip		

Tab. 4. Initial variables and vector variables for motion data extraction.

Initial Velocity Variables	Description	Vector Variables	Description
u_{x_n}	Initial velocity of nth body part at x-axis.	v_{x_n}	Cumulative velocity in the x-direction of nth body part.
u_{y_n}	Initial velocity of nth body part at y-axis.	v_{y_n}	Cumulative velocity in the y-direction of nth body part.
		a_{x_n}	Cumulative acceleration in the x-direction of nth body part.
		a_{y_n}	Cumulative acceleration in the y-direction of nth body part.

$$v_{x_n} = \sum_{i=1}^{m} \frac{x_{i_n} - x_{(i-1)_n}}{t_i - t_{i-1}} \, t_i \,, \tag{3}$$

$$v_{y_n} = \sum_{i=1}^{m} \frac{y_{i_n} - y_{(i-1)_n}}{t_i - t_{i-1}} \, t_i \,, \tag{4}$$

$$a_{x_n} = \sum_{i=1}^{m} \frac{x_{i_n} - x_{(i-1)_n}}{(t_i - t_{i-1})^2} \, t_i \,, \tag{5}$$

$$a_{y_n} = \sum_{i=1}^{m} \frac{y_{i_n} - y_{(i-1)_n}}{(t_i - t_{i-1})^2} \, t_i \,, \tag{6}$$

where:
i – the instance index,
m – total number of frames in the video divided by 30,
t – time interval, here equal to 0.5,
n – body joint number (0 to 13) + 1,
x_{i_n} – x-axis coordinate at i^{th} instance for body joint n,
$x_{(i-1)_n}$ – x-axis coordinate at the previous instance for body joint n,
y_{i_n} – y-axis coordinate at i^{th} instance for body joint n,
$y_{(i-1)_n}$ – y-axis coordinate at the previous instance for body joint n.

These formulas were then incorporated into the programming algorithm. Each position and vector variable had 14 attributes to represent different body joints. Thus, the total number of attributes was 84, and a motion type class attribute was added at the end. All attributes were extracted and saved in a comma-separated values (CSV) file for use in data preprocessing and mining.

Several issues contribute to value errors from the annotation of the stick model, and they would be addressed differently. One of the issues in estimating coordinates is the inability to detect body parts due to a blocked view. Motion videos feature human subjects interacting with objects to perform the required activity. As a result, the interacted object is likely to become an impediment to viewing body parts from the motion video. An example is undetected feet by the programming algorithm due to the carton box obscuring its view. Aside from being blocked by objects, some body parts for keypoint detection are kept out of camera view by the pother body parts of the participant. The most notable occurrences involve sweeping and mopping, in which some participants choose to turn their bodies in different directions while performing the action. In both cases, missing coordinate data were replaced with estimated vector data. The COCO keypoint dataset uses a large amount of image data to detect human body part positions and estimate missing keypoints by comparison with other body parts [32]. This estimation method is valid only for common gestures like parallel standing and

lifting objects. The reason is that the dataset only has a few images for each pose. The issues were resolved by assuming that body part movement momentum continued from the previous frame to the current frame. The position of an undetected body part was estimated using the coordinates of the previous frame plus the instantaneous velocity of that body part.

Another error is mistaking unrelated objects for body part keypoints. These objects are identified as human body parts by Setjo et al. [45]. Using the multi-person dataset for keypoint estimation, multiple sets of keypoints are detected. However, separating humans from false positives is required. Thus, the bottom-up approach [6] of associating joints to people was used to reduce misidentification.

3.3. Data preprocessing

Data preprocessing steps are critical for preparing data for an effective data mining process. Several techniques were used to normalize the data extracted from the stick model. Then, the outliers and extreme values of normalized data were calculated before reacting. Duplicate instances were also identified in the preprocessing stage.

Motion data are normalized to standardize the range of different units or scales in the attributes. It simplifies large-number numeric attributes and improves data quality without affecting the final data classification result [25]. Three popular normalization techniques were used in this study: min-max normalization (MMN), Z-score normalization (ZSN), and decimal scaling normalization (DSN). MMN reduces the un-normalized data to a specific lower and upper boundary, which is typically 0 to 1 or -1 to 1. The formula in Equation (7) was used to calculate MMN [43].

$$v'_{i,n} = \frac{v_{i,n} - \min(v_n)}{\max(v_n) - \min(v_n)}(\max_{\text{new}} - \min_{\text{new}}) + \min_{\text{new}}\,, \tag{7}$$

where:
$v'_{i,n}$ – new normalized variable data at i^{th} instance,
$v_{i,n}$ – original variable data at i^{th} instance,
$\min(v_n)$ – minimum value of variable data in the n^{th} attribute,
$\max(v_n)$ – maximum value of variable data in the n^{th} attribute,
$\min_{\text{new}}$ – new minimum value, usually -1 or 0,
$\max_{\text{new}}$ – new maximum value, usually 1.

The ZSN method uses mean and standard deviation to normalize data into a scaled value ranging from -1 to 1, with zero mean and unit variance. ZSN is expressed by (8).

$$v'_{i,n} = \frac{v_{i,n} - \mu_n}{\sigma_n}\,, \tag{8}$$

where:
$v'_{i,n}$ – new normalized variable data at i^{th} instance,
$v_{i,n}$ – original variable data at i^{th} instance,
μ_n – mean of all data in the n^{th} attribute,
σ_n – standard deviation of all data in the n^{th} attribute.

DSN measures the maximum values of an attribute and rescales them by moving the decimal point of instance values. This method of normalization is useful for data with logarithmic variation in the attribute. In (9), the DSN formula is written as follows.

$$v'_{i,n} = \frac{v_{i,n}}{10^j} \tag{9}$$

where:
$v'_{i,n}$ – new normalized variable data at i^{th} instance,
$v_{i,n}$ – original variable data at i^{th} instance,
$j = \log_{10}(\max(v_n))$.

Google Colab was loaded with the CSV file containing the extracted motion data from the stick-figure model. The three normalization methods were applied using the Python Scikit-learn (Sklearn) module [12], which resulted in three different normalized CSV dataset files.

The normalized datasets were then resampled in the WEKA interface [54] (version 3.8.5) under the supervised instance filter section using a random subsampling method. Its goal was to improve the instances by removing noise from the motion data. The imbalanced result from different subsets created during cross-validation could be due to noise instances. With or without replacement, the random subsampling method generated a random subsample of a dataset. The data were balanced with replicated instances from the remaining data to maintain the same class bias as the original unprocessed dataset without compromising the total sampling number for the experiment. The three datasets were preprocessed and saved to new CSV files before the motion classification experiment was started.

3.4. Motion classification

The WEKA Experimenter was used to run the motion classification experiment, which included all three normalized datasets and eight different classifiers (Table 5). The default WEKA settings were used except for the options in brackets that required manual input. Each classifier was run 10 times. The 10-fold cross-validation option was used to divide the training and validation data into 10 sets, with each set serving as the testing set iteratively in 10 rounds of validation. For each data preprocessing technique, the experiment was repeated with resampled datasets. A total of 4800 experimental trials were conducted.

Tab. 5. Classifiers used in the experiment.

Classifier	Description
ZeroR	The most basic rule-based classifiers predict the majority class while ignoring all predictors or attributes [15].
OneR	Selects the single most informative attribute and classifies instances solely on the basis of this attribute's criteria [39].
J48 Decision Tree (pruned)	Produces pruned trees that begin at the root node and classify instances into branches by sorting them according to attribute values [29].
Random forest	Building many individual decision trees with each random forest tree results in a class prediction, and the class with the most votes becomes the final model's prediction.
Random tree	The decision tree and Random Forest approaches are combined to predict the class by fitting several decision tree classifiers on different sub-samples of the dataset and averaging to improve prediction accuracy and avoid over-fitting.
k-Nearest neighbors ($k = 5$)	A lazy learner method that classifies instances based on evaluated Euclidean distances that define the closeness to each class, where k represents the number of neighbours considered to find the majority of a class label [29].
Naïve Bayes	Calculates the conditional probability of the classes based on the assumption that each attribute is independent of the others [56].
Multilayer perceptron	It is made up of neural network layers, which include input, hidden, and output layers. Back-propagation is used to train neurons to process data and recognize patterns [50].

4. Results and discussion

4.1. Stick model overlay

The stick model was used to annotate all 400 videos, and body part keypoints and lines indicated the connection between body joints. Figure 4 depicts video frames with human body parts augmented by the stick model when moving a carton box, moving a pail, sweeping, and mopping, in that order.

A set of 100 videos from the same motion class took an average of 27 min to complete the stick model overlay process. A motion video contains 8 to 10 frames that are designated for stick model processing and data extraction. As a result, a single frame took between 1.62 and 2.03 s to complete the stick model augmentation process.

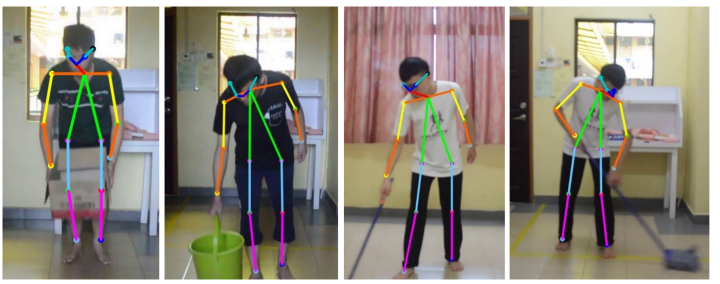


Fig. 4. Sample video frame with stick model overlay.

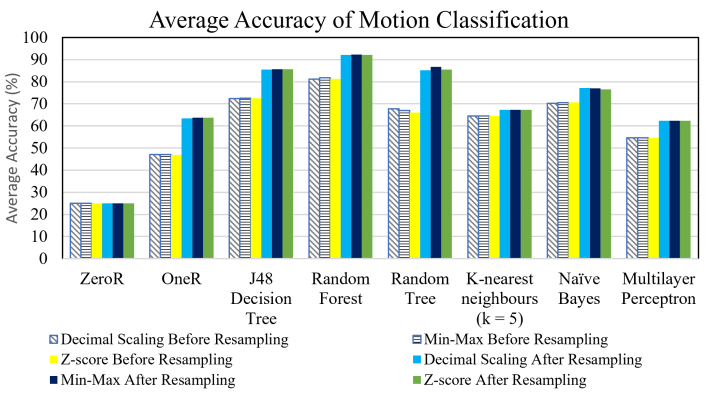


Fig. 5. Graph of average accuracy for all motion classification experimental trials.

4.2. Motion classification

A total of 4800 data mining experimental trials were conducted, which involved variable permutations of eight classifiers, three normalization techniques, and the use of resampling prior to classification. The average accuracies for each classifier and normalization technique permutation, with or without resampling, were evaluated. They are recorded in Table 6 and plotted as a graph in Figure 5.

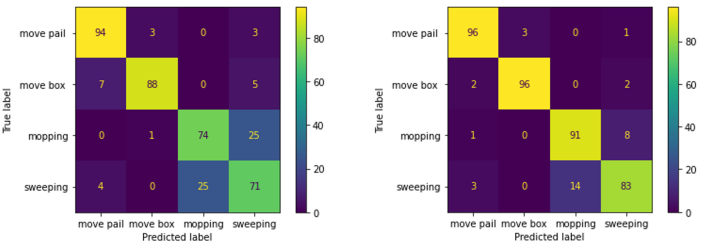
Except for the ZeroR classifier, the average classification accuracy for the datasets after resampling is higher than that for the datasets before resampling. ZeroR classifier maintains an accuracy of 25% regardless of normalization methods or resampling. The reason is that the ZeroR classifier frequently identifies the majority class. However, the majority class does not exist in these datasets because of the motion data distribution. As a result, the accuracy for all normalized datasets with a ZeroR classifier is the same.

Tab. 6. Classification accuracy of different classifiers and normalization technique used before and after the resampling.

Classifier	Normalization Technique	Average Accuracy without Resampling (%)	Average Accuracy with Resampling (%)
ZeroR	Decimal scaling	25.00	25.00
	Min-Max	25.00	25.00
	Z-score	25.00	25.00
OneR	Decimal scaling	47.15	63.45
	Min-Max	47.13	63.70
	Z-score	46.97	63.70
J48 Decision Tree (pruned)	Decimal scaling	72.38	85.52
	Min-Max	72.47	85.62
	Z-score	72.57	85.65
Random forest	Decimal scaling	81.25	92.10
	Min-Max	81.80	92.37
	Z-score	81.40	92.15
Random tree	Decimal scaling	67.78	85.20
	Min-Max	67.00	86.80
	Z-score	66.08	85.55
k-Nearest neighbors ($k = 5$)	Decimal scaling	64.53	67.30
	Min-Max	64.53	67.30
	Z-score	64.53	67.30
Naïve Bayes	Decimal scaling	70.30	77.20
	Min-Max	70.52	77.05
	Z-score	70.62	76.55
Multilayer perceptron	Decimal scaling	54.58	62.32
	Min-Max	54.58	62.32
	Z-score	54.58	62.32

Figure 5 shows that the random forest classifier method achieves the highest accuracy in the datasets before and after resampling categories. The random forest classifier with MMN and resampling has the best performance of the knowledge discovery method combination, with an average accuracy of 92.37%. The finding echoes previous studies on movement or gait analysis [17, 59]. The random forest classifier avoids overfitting in large datasets like the motion dataset. The motion dataset has 84 attributes, which can easily cause overfitting using other classifiers. The normalization techniques produce insignificant differences in classification accuracy while using the same classifiers. Thus, the normalization scale difference insignificantly affects the classification result. Nevertheless, the random forest classifier performs best with the min-max normalized dataset.

The resampling method increases classification accuracy by removing noise or misclassified data and replicating the remaining data to fill the void [4]. Confusion matrices



(a) without resampling (b) with resampling

Fig. 6. Confusion matrix for classification result of the min-max normalized dataset using Random Forest classifier.

for classification results of datasets with and without resampling confirm this explanation. As shown in Figure 6a, the random forest classifier and MMN produce a confusing confusion matrix for the dataset. Figure 6b shows the classification result of the same data mining technique for resampled data, which has a higher accuracy.

According to confusion matrices, sweeping and mopping are more likely to be misclassified due to their high similarity. The experimental motion capture does not impede movement execution. It affects the classification accuracy, especially for motions with similar characteristics. Resampling increases correctly classified instances in mopping and sweeping. The resampling method replaces incorrectly classified instances with replicated instances from correctly classified instances. Arbelaitz et al. [3] agreed that random subsampling improves accuracy. However, they recommended using synthetic minority oversampling technique (SMOTE) to obtain significant statistical differences between class instances. Future research should examine the effect of the SMOTE technique on the dataset.

5. Conclusion

This study develops a descriptive model for markerless motion classification using a single camera view. The stick model overlay uses OpenCV and OpenPose modules as well as COCO datasets. In motion classification, the best data mining strategy is determined by classifier and normalization accuracy. The best classifier is the random forest classifier, which achieves an accuracy of 81%-82% without resampling and an accuracy of 92%-93% with resampling. Using the same classifier, normalization techniques have little to no effect on classification accuracy. The developed algorithm of stick-figure model augmentation and data mining strategy complete the markerless motion classification model. This study can be extended to more complex and variable motion activities in manufacturing, such as manual operations.

Acknowledgement

This work was supported by The Malaysia Ministry of Higher Education (Kementerian Pengajian Tinggi) under Fundamental Research Grant Scheme (FRGS) grant no: FRGS/1/2021/TK0/USM/02/25.

References

[1] M. Aehnelt, E. Gutzeit, and B. Urban. Using activity recognition for the tracking of assembly processes: Challenges and requirements. In *Workshop on Sensor-Based Activity Recognition (WOAR)*, page 12–21, Rostock, Germany, Mar 2014. `https://publica.fraunhofer.de/entities/publication/45148487-5ac9-49c4-8ae8-e07e33aa87ef/details`.

[2] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999. doi:10.1006/cviu.1998.0744.

[3] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, and J. M. Pérez. Applying resampling methods for imbalanced datasets to not so imbalanced datasets. In *Conference of the Spanish Association for Artificial Intelligence*, page 111–120, 2013. doi:10.1007/978-3-642-40643-0_12.

[4] R. De Bin, S. Janitzaa, W. Sauerbrei, and A. L. Boulesteix. Subsampling versus bootstrapping in resampling-based model selection for multivariable regression. *Biometrics*, 72(1):272–280, 2016. doi:10.1111/biom.12381.

[5] M. Bosch, F. Zhu, and E. J. Delp. Video coding using motion classification. In *15th IEEE International Conference on Image Processing*, page 1588–1591, 2008. doi:10.1109/ICIP.2008.4712073.

[6] Z. Cao, G. Hidalgo, T. Simon, et al. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021. doi:10.1109/TPAMI.2019.2929257.

[7] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition*, page 1302–1310, Jan 2017. doi:10.1109/CVPR.2017.143.

[8] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 4733–4742, Dec 2016. doi:10.1109/CVPR.2016.512.

[9] C. K. Chan, W. P. Loh, and I. A. Rahim. Human motion classification using 2d stick-model matching regression coefficients. *Applied Mathematics and Computation*, 283:70–89, 2016. doi:10.1016/j.amc.2016.02.032.

[10] M. G. Choi, K. Yang, T. Igarashi, et al. Retrieval and visualization of human motion data via stick figures. *Computer Graphics Forum*, 31(7):2057–2065, 2012. doi:10.1111/j.1467-8659.2012.03198.x.

[11] W. Choi, L. Li, H. Sekiguchi, and K. Hachimura. Recognition of gait motion by using data mining. In *International Conference on Control, Automation and Systems*, page 1213–1216, 2013. doi:10.1109/ICCAS.2013.6704173.

[12] D. Cournapeau, M. Brucher, F. Pedregosa, et al. scikit-learn. Machine Learning in Python, 2023. `https://scikit-learn.org`. [Accessed 15 Jan 2022].

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 886–893, 2005. doi:10.1109/CVPR.2005.177.

[14] Q. Dang, J. Yin, B. Wang, and W. Zheng. Deep learning based 2D human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019. doi:10.26599/TST.2018.9010100.

[15] L. Devasena C. Effectiveness analysis of ZeroR, RIDOR and PART classifiers for credit risk appraisal. *International Journal of Advances in Computer Science and Technology (IJACST)*, 3(11):6–11, 2014. Special issue of ICCAAC 2014. `https://www.warse.org/IJACST/static/pdf/file/iccaac2014sp02.pdf`.

[16] R. Ferdinands. Advanced applications of motion analysis in sports biomechanics. In *Proc. XXVIII International Symposium of Biomechanics in Sports*, page 70–73, Jul 2010. `https://ojs.ub.uni-konstanz.de/cpa/article/view/4383`.

[17] S. Fong, J. Liang, I. Fister, and S. Mohammed. Gesture recognition from data streams of human motion sensor using accelerated PSO swarm search feature selection algorithm. *Journal of Sensors*, 2015:205707, 2015. doi:10.1155/2015/205707.

[18] G. B. Garibotto. 3-D computer vision modeling in video surveillance applications. In C. H. Chen, editor, *Handbook of Pattern Recognition and Computer Vision*, page 747–765. World Scientific, 2009. doi:10.1142/9789814273398_0033.

[19] Google. Cloud Tensor Processing Units (TPUs), 2022. `https://cloud.google.com/tpu`. [Accessed 15 Jan 2022].

[20] Google. Colaboratory, 2022. `https://colab.research.google.com`. [Accessed 15 Jan 2022].

[21] Y. Guo, G. Xu, and S. Tsuji. Tracking human body motion based on a stick figure model. *Journal of Visual Communication and Image Representation*, 5(1):1–9, 1994. doi:10.1006/jvci.1994.1001.

[22] S. U. Han, S. H. Lee, and F. Peña-Mora. Vision-based motion detection for safety behavior analysis in construction. In *Construction Research Congress 2012: Construction Challenges in a Flat World, Proc. 2012 Construction Research Congress*, page 1032–1041, 2012. doi:10.1061/9780784412329.104.

[23] N. Hasler, B. Rosenhahn, T. Thormahlen, et al. Markerless motion capture with unsynchronized moving cameras. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, page 224–231, 2009. doi:10.1109/CVPR.2009.5206859.

[24] G. Hidalgo, Z. Cao, T. Simon, et al. OpenPose, 2022. `https://github.com/CMU-Perceptual-Computing-Lab/openpose`. [Accessed 15 Jan 2022].

[25] A. Jahan and K. L. Edwards. A state-of-the-art survey on the influence of normalization techniques in ranking: Improving the materials selection process in engineering design. *Materials and Design*, 65(1):335–342, 2015. doi:10.1016/j.matdes.2014.09.022.

[26] R. M. Kanko, E. K. Laende, G. Strutzenberger, et al. Assessment of spatiotemporal gait parameters using a deep learning algorithm-based markerless motion capture system. *Journal of Biomechanics*, 122(110414), 2021. doi:10.1016/j.jbiomech.2021.110414.

[27] J. S. Kim, Y. W. Kim, Y. K. Woo, and K. N. Park. Validity of an artificial intelligence-assisted motion-analysis system using a smartphone for evaluating weight-bearing activities in individuals with patellofemoral pain syndrome. *Journal of Musculoskeletal Science and Technology*, 5(1):34–40, 2021. doi:10.29273/jmst.2021.5.1.34.

[28] S. Kloiber, V. Settgast, C. Schinko, et al. Immersive analysis of user motion in VR applications. *Visual Computer*, 36(10-12):1937–1949, 2020. doi:10.1007/s00371-020-01942-1.

[29] S. W. Knox. *Survey of Classification Techniques*. Wiley Series in Probability and Statistics, 2018. doi:10.1002/9781119439868.ch4.

[30] N. Le, A. Heili, and J. Odobez. Long-term time-sensitive costs for CRF-based tracking by detection.

In *European Conference on Computer Vision Workshops, Lecture Notes in Computer Science*, volume 9914, pages 43–51, 2016. doi:10.1007/978-3-319-48881-3_4.

[31] B. Li, B. Bai, and C. Han. Upper body motion recognition based on key frame and random forest regression. *Multimedia Tools and Applications*, 79(7-8):5197–5212, 2020. doi:10.1007/s11042-018-6357-y.

[32] T. Y. Lin, M. Maire, S. Belongie, et al. Research on face recognition based on CNN. In *Microsoft COCO: Common objects in context*, volume 8693 of *Lecture Notes in Computer Science*, page 740–755, 2014. doi:10.1007/978-3-319-10602-1_48.

[33] T.-Y. Lin, G. Patterson, M. R. Ronchi, et al. COCO. Common Objects in Context, 2020. `https://cocodataset.org`. [Accessed 6 Jan 2022].

[34] H. Liu, Z. Ju, X. Ji, C. S. Chan, and M. Khoury. *Human Motion Sensing and Recognition*. Springer, Berlin Heidelberg, 2017. `https://link.springer.com/book/10.1007/978-3-662-53692-6`.

[35] D. C. Luvizon, H. Tabia, and D. Picard. Human pose regression by combining indirect part detection and contextual information. *Computers and Graphics*, 85:15–22, 2019. doi:10.1016/j.cag.2019.09.002.

[36] OpenCV Team. OpenCV, 2022. `https://opencv.org`. [Accessed 15 Jan 2022].

[37] A. N. Mohamed and M. M. Ali. Human motion analysis, recognition and understanding in computer vision: A review. *Journal of Engineering Sciences*, 41(5):1928–1946, 2013. doi:10.21608/jesaun.2013.114925.

[38] N. Nakano, T. Sakura, K. Ueda, et al. Evaluation of 3D markerless motion capture accuracy using OpenPose with multiple video cameras. *Frontiers in Sports and Active Living*, 2(50):1–9, 2020. doi:10.3389/fspor.2020.00050.

[39] C. G. Nevill-Manning, G. Holmes, and I. H. Witten. The development of Holte's 1R classifier. In *Proc. 1995 2nd New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, page 239–242, Jan 1995. doi:10.1109/ANNES.1995.499480.

[40] H. Qian, Y. Mao, W. Xiang, and Z. Wang. Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters*, 31(2):100–111, 2010. doi:10.1016/j.patrec.2009.09.019.

[41] J. Rittscher and A. Blake. Classification of human body motion. In *Proc. IEEE International Conference on Computer Vision*, volume 1, page 634–639, 1999. doi:10.1109/iccv.1999.791284.

[42] C. Rubino, M. Crocco, V. Murino, and A. Del Bue. Semantic multi-body motion segmentation. In *Proc. 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, page 1145–1152, 2015. doi:10.1109/WACV.2015.157.

[43] C. Saranya and G. Manikandan. A study on normalization techniques for privacy preserving data mining. *International Journal of Engineering and Technology*, 5(3):2701–2704, 2013. `http://www.enggjournals.com/ijet/docs/IJET13-05-03-273.pdf`.

[44] P. Schneider, R. Memmesheimer, I. Kramer, and D. Paulus. Gesture recognition in RGB videos using human body keypoints and dynamic time warping. In *Lecture Notes in Computer Science*, volume 11531, page 281–293, 2019. doi:10.1007/978-3-030-35699-6_22.

[45] C. H. Setjo, B. Achmad, and Faridah. Thermal image human detection using Haar-cascade classifier. In *Proc. 2017 7th International Annual Engineering Seminar*, pages 1–6, 2017. doi:10.1109/INAES.2017.8068554.

[46] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. 27th International Conference on Neural Information Processing Systems*, volume 27 of *NIPS Proceedings*, page 568–576, 2014. `https://ora.ox.ac.uk/objects/uuid:1dd0bcd0-39ca-48a1-9c20-5341d6c49251`.

[47] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 5686–5696, June 2019. doi:10.1109/CVPR.2019.00584.

[48] A. Switonski, H. Josinski, and K. Wojciechowski. Dynamic time warping in classification and selection of motion capture data. *Multidimensional Systems and Signal Processing*, 30(3):1437–1468, 2019. doi:10.1007/s11045-018-0611-3.

[49] T. Tsuji, S. Nakashima, H. Hayashi, et al. Markerless measurement and evaluation of general movements in infants. *Scientific Reports*, 10(1):1–13, 2020. doi:10.1038/s41598-020-57580-z.

[50] J. Wang and Z. Li. Research on face recognition based on CNN. In *IOP Conference Series: Earth and Environmental Science*, volume 170, page 032110, 2018. doi:10.1088/1755-1315/170/3/032110.

[51] L. Wang, Z. Ding, and Y. Fu. Low-rank transfer human motion segmentation. *IEEE Transactions on Image Processing*, 28(2):1023–1034, 2019. doi:10.1109/TIP.2018.2870945.

[52] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003. doi:10.1016/S0031-3203(02)00100-0.

[53] J. A. Webb and J. K. Aggarwal. Visually interpreting the motion of objects in space. *Computer*, 14(8):40–46, 1981. doi:10.1109/C-M.1981.220561.

[54] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, 2016. [Accessed 21 Sep 2021]. `https://www.cs.waikato.ac.nz/ml/weka/book.html`.

[55] G. Xia, H. Sun, L. Feng, et al. Human motion segmentation via robust kernel sparse subspace clustering. *IEEE Transactions on Image Processing*, 27(1):135–150, 2018. doi:10.1109/TIP.2017.2738562.

[56] X. Xie, J. W. K. Ho, C. Murphy, et al. Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software*, 84(4):544–558, 2011. doi:10.1016/j.jss.2010.11.920.

[57] Q. Xu, G. Huang, M. Yu, and Y. Guo. Fall prediction based on key points of human bones. *Physica A: Statistical Mechanics and Its Applications*, 540:123205, 2020. doi:10.1016/j.physa.2019.123205.

[58] L. Yang and T. Zhao. Data mining and ergonomic evaluation of firefighter's motion based on decision tree classification model. In *Advanced Research on Computer Science and Information Engineering: International Conference: Proceedings*, volume Part 2, page 212–217, 2011. doi:10.1007/978-3-642-21411-0_35.

[59] H. Zhang, W. Du, and H. Li. *Kinect Gesture Recognition for Interactive System*. Stanford University Term Paper for CS 299, 2012. `https://cs229.stanford.edu/proj2012/ZhangDuLi-KinectGestureRecognitionforInteractiveSystem.pdf`.

[60] D. Zhou and Q. He. PoSeg: Pose-aware refinement network for human instance segmentation. *IEEE Access*, 8:15007–15016, 2020. doi:10.1109/aCCESS.2020.2967147.

[61] H. Zhou and H. Hu. Human motion tracking for rehabilitation—A survey. *Biomedical Signal Processing and Control*, 3(1):1–18, 2008. doi:10.1016/j.bspc.2007.09.001.

[62] T. Zhou, H. Fu, C. Gong, et al. Multi-mutual consistency induced transfer subspace learning for human motion segmentation. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 10274–10283, 2020. doi:10.1109/CVPR42600.2020.01029.

[63] T. Zult, J. Allsop, J. Tabernero, and S. Pardhan. A low-cost 2-D video system can accurately and reliably assess adaptive gait kinematics in healthy and low vision subjects. *Scientific Reports*, 9(1):1–11, 2019. doi:10.1038/s41598-019-54913-5.

Yu Liang Liew currently works as a mechanical design engineer since 2021. He received his Bachelor of Engineering in Manufacturing Engineering with Management from the School of Mechanical Engineering at Universiti Sains Malaysia (USM). His interests include computer vision, artificial intelligence, big data and statistical analysis.

Jeng Feng Chin received his Ph.D. degree in Manufacturing Engineering from The University of Birmingham, United Kingdom. He has been an associate professor in the School of Mechanical Engineering at Universiti Sains Malaysia (USM) in Penang, Malaysia, since 2006. His research interests comprise computer-integrated manufacturing, lean manufacturing, production management, machine learning, artificial intelligence and optimization.