Construction Robotics (2024) 8:14
https://doi.org/10.1007/s41693-024-00130-0

ORIGINAL PAPER

Automatic recognition of excavator working cycles using supervised learning and motion data obtained from inertial measurement units (IMUs)

Amirmasoud Molaei[1,2] · Antti Kolu[2] · Kalle Lahtinen[2] · Marcus Geimer[1]

Received: 4 October 2023 / Accepted: 4 July 2024

Abstract

This paper proposes an automatic method for excavator working cycle recognition using supervised classification methods and motion information obtained from four inertial measurement units (IMUs) attached to moving parts of an excavator. Monitoring and analyzing tasks that have been performed by heavy-duty mobile machines (HDMMs) are significantly required to assist management teams in productivity and progress monitoring, efficient resource allocation, and scheduling. Nevertheless, traditional methods depend on human observations, which are costly, time-consuming, and error-prone. There is a lack of a method to automatically detect excavator major activities. In this paper, a data-driven method is presented to identify excavator activities, including loading, trenching, grading, and idling, using motion information, such as angular velocities and joint angles, obtained from moving parts, including swing body, boom, arm, and bucket. Firstly, a dataset lasting 3 h is collected using a medium-rated excavator. One experienced and one inexperienced operator performed tasks under different working conditions, such as different types of material, swing angle, digging depth, and weather conditions. Four classification methods, including support vector machine (SVM), k-nearest neighbor (KNN), decision tree (DT), and naive Bayes, are off-line trained. The results show that the proposed method can effectively identify excavator working cycles with a high accuracy of 99%. Finally, the impacts of parameters, such as time window, overlapping configuration, and feature selection methods, on the classification accuracy are comprehensively analyzed.

Keywords Activity recognition · Excavator · Earth-moving operations · Supervised learning · Inertial measurement unit (IMU)

1 Introduction

Heavy-duty mobile machines (HDMMs) are utilized in various industries, such as mining, forestry, and construction, all over the world. These industries, which are growing quickly, have numerous challenges, such as a shortage of skilled workers, extremely harsh environmental conditions, and low productivity and safety (Geimer 2020). In recent years, there has been a pressing need in the construction industry to increase productivity. According to studies, over the past 20 years, the productivity of the construction industry has only increased by 1% (Kassem et al. 2021). The costs of HDMMs also have a significant impact on the total cost of construction projects. Studies show that equipment expenses make up 40% of direct costs in highway construction projects and between 5 and 10% of direct costs in building construction projects (Deshmukh and Mahatme 2016). Moreover, these machines are powered by diesel engines,

✉ Amirmasoud Molaei
amirmasoud.molaei@partner.kit.edu

Antti Kolu
antti.kolu@novatron.fi

Kalle Lahtinen
kalle.lahtinen@novatron.fi

Marcus Geimer
marcus.geimer@kit.edu

1 Institute of Mobile Machines, Karlsruhe Institute of Technology, Rintheimer Querallee 2, Karlsruhe 76131, Baden-Württemberg, Germany

2 Radical Innovation Research Group, Novatron Ltd., Jasperintie 312, Pirkkala 33960, Pirkanmaa, Finland

Published online: 24 July 2024

which emit a great amount of CO_2 when burning fossil fuels (Molaei et al. 2023).

The phrase "If you cannot measure it, you cannot improve it" Lingard et al. (2013) must be recalled to improve the performance of HDMMs in earth-moving projects. It can be highly challenging to precisely track machines' activities and utilization rates on construction sites. The conventional method for measuring and analyzing equipment operations is time-consuming, costly, and error-prone, since it equires the site superintendents to manually observe and record the entire operation for each HDMM. Therefore, an automated technique is highly required. The systematic assessment and analysis of equipment activities can assist managers in optimizing equipment's operation time, improving working efficiency, and making wise project-related decisions, and is a key step toward semi or fully autonomous worksites. Worksite managers and contractors can benefit from it in spotting project problems, accurately pricing and budgeting future projects, and improving management and financial conditions. Additionally, it can assist in determining the appropriate machine size and type for a project, which can ensure that resources are employed properly, maximize equipment usage, and minimize downtime (Chen et al. 2020, 2022; Rasul et al. 2021; Molaei et al. 2022).

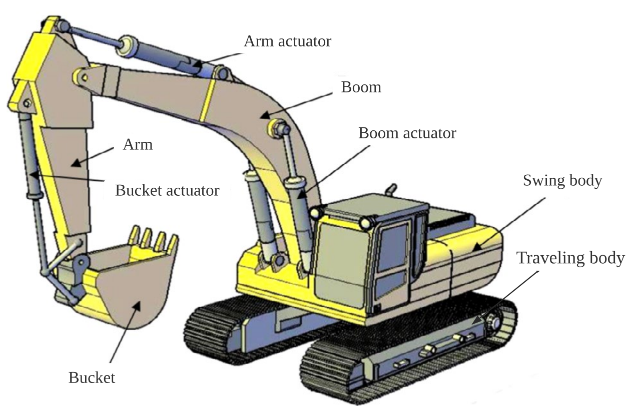
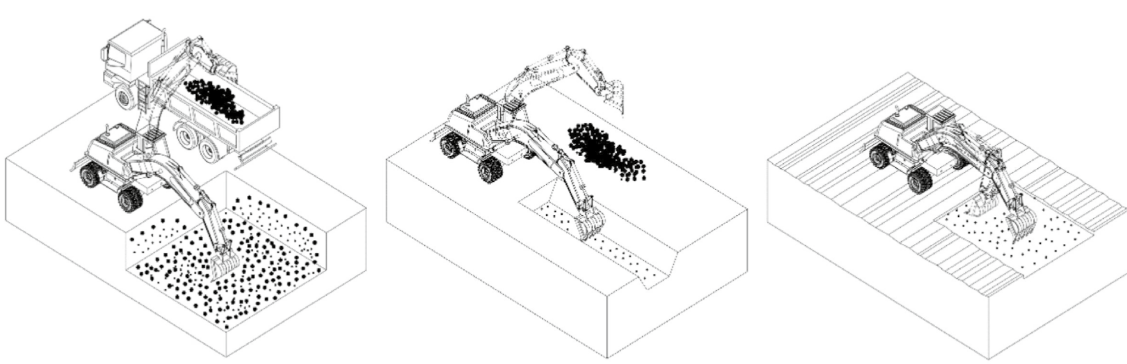


Fig. 1 A typical hydraulic excavator and its different parts (Molaei et al. 2022)

One of the most crucial pieces of equipment in construction projects is a hydraulic excavator. Excavation activities are necessary for almost all construction projects, such as the construction of roads, airports, and industrial and residential buildings (MundaneSagar and KharePranay 2015). An excavator is a versatile piece of machinery used for a variety of tasks, including loading, trenching, and grading operations. An excavator is a machine that is operated by a human operator and is mostly driven by a hydraulic system. Figure 1 shows a typical hydraulic excavator. An excavator's traveling body, swing body, and front digging manipulator are three main parts. The bucket, arm, and boom make up the manipulator. Three revolute joints link the bucket, arm, boom, and swing body (Klanfar et al. 2019).

The productivity of an excavator is defined based on the task and goals of the operation. According to a survey of research in this field, three of the most frequent working cycles performed by an excavator are loading, trenching, and grading (Helmus and Fecke 2015; Holländer 1998; Vukovic et al. 2017). The schematics of these tasks are illustrated in Fig. 2. The loading operation is one of the most significant tasks in construction and mining projects. In this operation, materials are picked up and moved from one place to another utilizing the excavator's manipulator. It could include digging or collecting materials from the ground to prepare a construction site or loading materials onto trucks for transportation. The loading operation is composed of four main steps, including scooping, swinging loaded, dumping, and swinging empty. The excavator productivity in the loading operation is defined based on the quantity of material and cycle time (Molaei et al. 2023). Nonetheless, this definition cannot be employed to determine the productivity of the grading and trenching operations since quality plays the



(a) Loading (b) Trenching (c) Grading

Fig. 2 Typical excavator duty cycles (Vukovic et al. 2017) main role in these operations. In the grading (or leveling) operation, an excavator is utilized to level and smooth the ground's surface. It is usually done for building or landscaping purposes, to prepare a site for construction, or to make a level surface for paving or other activities. The excavator utilizes the bucket to move and spread the material to create a level surface. Compared to other tasks, the grading task needs a higher positioning accuracy of ±5 or ±10 cm. In the trenching operation, an excavator is used to dig trenches in the ground corresponding to the desired width and depth for the installation of underground facilities, such as water and sewer pipes. The productivity definitions of the trenching and grading operations are the length of the trench per unit of time and the graded area per unit of time, respectively. Therefore, task or working cycle recognition of an excavator is one of the essential and primary steps before the productivity analysis since the productivity of a machine is defined based on the task (Molaei et al. 2023).

1.1 Literature review

Numerous research studies have been carried out to recognize excavator activities in different levels of details (LoDs). According to the type of data, the developed methods can be categorized into four main groups:

1. Vision-based methods
2. Audio-based methods.
3. Motion-based methods.
4. Hybrid methods.

Vision-based methods typically use videos or photos captured by cameras. Audio-based approaches use generated sounds from machines to identify activities. Motion-based techniques collect information on the acceleration and orientation of various parts of a machine using different sensors, such as inertial measurement units (IMUs). Hybrid systems incorporate multiple sensor types, such as vision and motion sensors, to identify excavator activities.

1.1.1 Vision-based methods

Visual recording technologies, including photos and videos, have been extensively utilized to identify excavator activities, monitor progress, and ensure safety on construction sites. In Zou and Kim (2007), an image processing technique is designed to detect only the excavator's idle time. The utilization rate is then calculated by dividing the working time by the entire operation time. The hue–saturation-value (HSV) color space is utilized in the algorithm. In Wang and Olson (2016), a graph-based image segmentation algorithm is presented to detect markers among other features in a natural scene. The method analyzes gradient patterns on the image to precisely estimate lines. A computer vision (CV)-based technique for detecting the actions of excavators (digging, dumping, hauling, and swinging) and dump trucks (filling, moving, and dumping) is provided in Golparvar-Fard et al. (2013). An SVM classifier and the histogram of oriented gradients (HOG) descriptor are used in the proposed method. In Kim et al. (2017), it has been stated explicitly that activity identification is a crucial step in the productivity monitoring of earth-moving activities. A vision-based algorithm is designed based on the tracking–learning–detection (TLD) and bags-of-features (BoF) to recognize excavator actions (work, travel, and idle). The approach is further developed in Kim et al. (2018), where a CV technique is proposed to identify excavator and dump truck activities based on spatio-temporal reasoning and imagine differencing techniques. The suggested approach is divided into four basic steps: (1) equipment detection and tracking, (2) action recognition, (3) interaction analysis, and (4) post-processing. In Bao et al. (2016), a CV-based technique for the activity detection of an excavator in earth-moving operations is developed using highly varying long-sequence videos taken from fixed cameras. At each video frame, the method recognizes excavator activities (swing, dig, dump, idle, and move). In Kim and Chi (2019), using two sequential operating patterns (visual features and operation cycles), a CV algorithm based on a hybrid deep learning algorithm (i.e., convolutional neural networks (CNNs) and long short-term memory (LSTM) network) is provided to detect, track, and identify the activity of an excavator (digging, hauling, dumping, swinging, moving, and stopping) in earthwork operations. The method's extensive computational training time and the requirement for a large amount of training data have been highlighted as two drawbacks. In Chen et al. (2019), a three-dimensional (3D) CNN is developed based on temporal and spatial data to recognize excavator activities (digging, swinging, and dumping). In Roberts and Golparvar-Fard (2019), a CV-based technique is designed for automatically identifying visually distinctive excavator and dump truck actions from individual frames of a video taken at the ground level. The operations of the excavator include idling, loading, swinging, dumping, and moving. In Zhang et al. (2020), a deep learning-based approach is proposed to recognize the actions of an excavator (digging, swinging, and dumping) and a dump truck (moving forward and moving backward) from video frame sequences. In this approach, image and temporal information are extracted using CNN and LSTM, respectively. In Chen et al. (2020), three CNNs are introduced for the identification of excavator activities (digging, swinging, and dumping) and productivity estimation. It has been observed that CV-based approaches face substantial difficulties when the lighting is poor or when several pieces of construction equipment are simultaneously captured. In Kim and Chi (2020), an expensive method is suggested for tracking productivity at a worksite using multiple non-overlapping cameras. The model can recognize excavator and dump truck activities, such as digging, swinging full, dumping, swinging empty, moving, and stopping. In Zhang and Zhang (2021), a safety monitoring system and deep learning-based excavator activity analysis are provided. The systems can recognize excavator actions (digging, swinging, and dumping), identify the surrounding environment, and estimate poses. As the research continues, an algorithm is designed in Zhang and Zhang (2022) for the excavator's activity classification that performs better than the suggested model in Zhang and Zhang (2021). In Chen et al. (2023), a vision-based method is presented to automatically identify activities (digging and loading) of general construction machines (e.g., excavators and loaders) without pre-training or fine-tuning. The proposed method uses zero-shot learning for activity recognition. In Kim et al. (2023), firstly, a pre-trained CNN model is utilized to extract the sequential pattern of visual features from video frames. In the second step, BiLSTM recognizes excavators' activities (dumping, excavation, hauling, and swing) based on the output of the pre-trained CNN. While various CV approaches have been developed to identify the activities of excavators, these methods often focus on sub-tasks or low-level details. However, they face significant limitations and challenges. These challenges include high computational complexity, issues related to camera viewpoints, varying illumination conditions (e.g., excessive brightness or darkness), object occlusions, the presence of multiple pieces of equipment in a scene, background movements, camera shake caused by wind, image blurring due to rain, snow, dust, and fog, the need for extensive storage space for saving image and video data, the installation of multiple cameras to adequately cover large worksites, and a shortage of training dataset, which can significantly impact the performance. Maintaining a direct line of sight to targeted resources is difficult due to the substantial noise present in dynamic construction sites. Furthermore, considerations must be made for the short daylight hours in autumn and winter in certain countries, such as Finland and Norway. Additionally, these methods tend to be relatively expensive, with camera costs ranging from \$1000 to \$10,000 in small-sized worksites and from \$10,000 to \$100,000 in medium-sized worksites (Gong and Caldas 2011; Cheng et al. 2017; Mahamedi et al. 2021; Molaei et al. 2023; Sherafat et al. 2020).

1.1.2 Audio-based methods

Several studies have been published that use audio data to recognize machine activities. These procedures typically involve four essential steps: (1) using a microphone to collect equipment sound data, (2) signal filtering or augmentation, (3) feature extraction, and (4) training classification models to identify equipment actions. In Cheng et al. (2017), an approach is proposed that classifies the activities of construction equipment, including excavators, loaders, and dozers, into two classes: productive activities and non-productive activities, based on the generated sounds. The approach employs an SVM classifier and the short-time Fourier transform (STFT) features. In Sabillon et al. (2020), a method is provided for estimating cycle time and productivity of equipment, such as excavators and dozers, using audio data and a Markov chain filter. The continuous wavelet transform (CWT), STFT, and an SVM classifier are utilized in the activity identification algorithm. In Sherafat et al. (2022), a multi-label multi-level sound classification algorithm is presented to identify excavator activities using STFT and CNN. Although in the paper, mixed construction sound scenario has been studied, the method is still susceptible to some drawbacks, such as the assumption that two types of construction noises always occur at the same time. The suggested audio-based approaches lack the capability to identify excavator working cycles or sub-tasks. Additionally, the precision of these models can be significantly compromised by background noise, and certain equipment may not generate distinctive sound patterns, posing challenges in accurately identifying their activities. Furthermore, these methods face limitations in their applicability to machines like tower cranes, which do not produce identifiable sounds.

1.1.3 Motion-based methods

In motion-based techniques, motion sensors are attached to different moving parts of equipment on construction sites. In Ahn et al. (2015), several supervised classifiers such as Naive Bayes, instance-based learning, KNN, and DT are used to categorize excavator activities (off, idle, working) using acceleration data acquired from accelerometers placed inside the cabins of four excavators. In Mathur et al. (2015), a non-invasive method is presented to calculate the excavator's cycle time based on detected activities, including wheel-base motion, cabin rotation, and arm/bucket movement. Eight classifiers are trained based on the time and frequency domain features of acceleration data, which is collected via a smartphone mounted inside the cabin. In Kim et al. (2018), IMU data is utilized to determine the excavator operation cycle time. The method employs random forests, Naive Bayes, J48, and sequential minimal optimization (SMO) to identify excavator activities (wheel-base motion, anti-clockwise/clockwise cabin rotation, and arm/bucket movement). In Rashid and Louis (2019), synthetic training data is produced utilizing time-series data augmentation techniques on acceleration and orientation data. For the activity classification of excavators and front-end loaders, a recurrent neural network (RNN) is implemented. The excavator activities include engine off, idle, scoop, dump, swing loaded, swing empty, move forward, move backward, and level ground. In Bae et al. (2019), a dynamic time-warping system is presented to identify excavator working cycles (digging, trenching, and leveling) based on joystick measurements. In Rashid and Louis (2020), using motion data (i.e., linear and angular acceleration) from the articulated structural elements of construction equipment, a real-time excavator activity recognition approach is constructed. An SVM, a DT, a KNN, and an artificial neural network (ANN) are machine learning techniques that are trained using the gathered data. The suggested method classifies the activities in different LoDs. In the most detailed level, the activities comprise engine off, idle, scooping, dumping, swinging full, swinging empty, moving forward, moving backward, and leveling. In Slaton et al. (2020), an automatic activity recognition technique is described based on acceleration data and deep learning architectures. An excavator and a roller compactor are employed to implement the suggested procedure. Excavator activities consist of idling, traveling, scooping, dropping, rotation (left), and rotation (right). In Shi et al. (2020), a method is introduced to automatically determine the excavator actions (pre-digging, digging, lifting, unloading, and swinging) based on the main pump pressure waveform. Three classifiers, an SVM, a back propagation neural network (BPNN), and logistic regression (LR), are trained using the dataset. In Langroodi et al. (2021), a method for the activity recognition of construction equipment is suggested. It combines a random forest classifier with a fractional calculus-based feature augmentation method. Three case studies are used to show the performance: (1) two different excavator models, (2) a scaled remotely operated excavator, and (3) a roller. Excavator activities include idling, relocating, swinging, digging, and filling. In Shi et al. (2021), three machine learning algorithms, an LSTM network, an RNN, and an SVM, are utilized to classify the excavator actions (digging, hauling dumping, and swinging) using the control signals of operating handles. In Mahamedi et al. (2021), a deep learning-based technique is described for determining productivity using kinematic data gathered from smartphone sensors mounted on an excavator. The activities of the excavator are categorized into active and inactive classes. While numerous motion-based methods have been proposed to identify excavator activities, they primarily focus on sub-tasks or low-level details. Only one motion-based method has been introduced to recognize the principal tasks or working cycles of an excavator. In Bae et al. (2019), the suggested method employs joystick measurements for task recognition, a strategy that may encounter notable challenges. Joysticks utilized in machines vary across manufacturers, necessitating considerable time and effort for adjustments to interpret joystick output values. Additionally, the precision of joystick measurements differs among various machines, and the method can be highly susceptible to the behaviors and skills of operators. Furthermore, the proposed method in Bae et al. (2019) necessitates several intricate post-processing algorithms to mitigate errors in the primary algorithm.

1.1.4 Hybrid methods

In some studies, hybrid sensors are utilized to acquire more data on equipment and activities. IMUs and microphones are used in Sherafat et al. (2019) to collect vibration and audio data to recognize the excavator's activities, such as stop, scoop, move, and swing. The real-world application of the suggested approach may encounter substantial challenges as it relies on audio data. Additionally, a drawback is its emphasis on sub-tasks or low-level details. In Kim et al. (2021), a hybrid kinematic-visual sensing technique based on deep learning is designed to recognize excavator activities (dig, haul, dump, swing, move, and stop). Kinematic and visual data are collected using built-in sensors of a smartphone (gyroscopes, accelerometers, and camera) that is mounted inside the cabin of an excavator. In Kim and Cho (2020), a method is designed to recognize excavator activities (excavation, leveling, rock excavation, and drive) using multi-modal deep learning models. The suggested fusion network integrates sensor and video-based models. This research continues, and in Kim and Cho (2022), a DNN ensemble called FusionNet is proposed for the identification of excavator activities (slope digging, ditch digging, rock digging, leveling up-down, leveling front-back, leveling left-right, deep digging, drive, and digging). The features are extracted from sensor data and video frames. Only two hybrid methods have been proposed to recognize the tasks of an excavator which have significant challenges (Kim and Cho 2020, 2022). Firstly, they utilize vision sensors. CV-based methods have numerous practical limitations and restrictions that are completely described in Sect. 1.1.1. Secondly, deep-learning models have a high computational complexity and require very large amounts of data.

1.2 Contribution

As mentioned earlier, several methods have been proposed to recognize excavators' activities in various earth-moving tasks using different types of sensors. The methods mostly concentrate on sub-tasks or low-level information. However, only three methods (Bae et al. 2019; Kim and Cho 2020, 2022) have been presented to recognize the major activities or tasks of an excavator. These techniques utilize joystick measurements or vision sensors that have many challenges in real-world applications. The challenges have been completely described in the previous section. In this paper, an automatic method is suggested to recognize the task of an excavator, including loading, trenching, grading, and idling, using multiple low-cost IMUs that have been installed on different moving parts of an excavator, including bucket, arm, boom, and swing body. IMU sensors could provide a promising solution for the challenges of the automatic identification of excavators' working cycles since they are affordable, not restricted, can be easily installed, or have been already installed on different machines. The costs of IMUs are within the range [$100–$1000] in small-sized worksites and within the range [$1000–$10,000] in medium-sized worksites. In recent years, in order to estimate the bucket position for automated machine guidance (AMG) or automated machine control (AMC) systems, equipment manufacturers and third-party businesses, such as Novatron, Trimble, Topcon, and Leica, have begun mounting IMUs on the equipment. Moreover, the power consumption of IMUs is satisfactory, and they are robust and resilient in challenging environments, in contrast to CV-based methods. Using four IMU sensors installed on a medium-rated excavator operated by one experienced and one inexperienced operator, a dataset lasting 3 h is collected. Different operating conditions, such as different swing angles, digging depths, types of material, weather conditions, and the skill levels of operators, have been covered in the dataset to increase the robustness of the data-driven method. In the next step, four machine learning techniques, including an SVM, a KNN, a DT, and naive Bayes, are trained using the collected dataset. Then, the effects of different configurations, including time window, overlapping, and feature selection methods, on classification accuracy are extensively investigated. Finally, the results show the presented algorithm has the ability to automatically recognize the major tasks or working cycles of an excavator.

The remainder of this paper is outlined as follows. Section 2 describes the data-driven method for the task recognition of an excavator in earth-moving operations. Firstly, the field data collection is illustrated. Then, data preparation, feature extraction, and classification model training are explained. Results are demonstrated in Sect. 3. Discussion is presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Methodology

In this section, a supervised classification algorithm is introduced to automatically identify the excavator working cycles, including (1) loading, (2) trenching, (3) grading, and (4) idling. These are the most important tasks for an excavator in all construction sites. In the proposed approach, motion sensors such as IMUs are employed to learn about different movements of articulated structural parts of an excavator. Firstly, the data collection procedure is explained, and in the next steps, the classification algorithm is described.

2.1 Field data collection

In the first step, field data were gathered using a single excavator. Figure 3 depicts the crawler excavator utilized in the data collection. The excavator is old, but it has received regular maintenance and inspections every 500 working hours, so it has been kept in good condition. The excavator is a Komatsu ® PC138US with a mass of 13.4 tons and a typical mono boom structure that is equipped with a Novatron Xsite ® machine control system. Quick couplers and a tiltrotator are used to attach the bucket to the arm of the excavator. Throughout the data gathering, the tiltrotator was not moved. The bucket has a heaping capacity of 0.37 m^3 according to the Society of Automotive Engineers (SAE) standard J-296. During the data collection, there was no

Fig. 3 Excavator used in data collection. In the picture, cabin (1), boom (2), arm (3), and bucket (4) are highlighted with red boxes (Molaei et al. 2022)

ongoing construction project on the worksite. The dataset covers a variety of working conditions, such as different swing angles, digging depth, weather conditions, and types of material. The swing angles of operations vary from 60° to 120°, and the digging depths increase up to 2 m. Different types of material, including sand, gravel, clay, and mixed, are utilized in the operations. The studies were conducted in different seasons during 18 months in a private worksite by two operators with different levels of competence. The experienced operator, with over 30 years of expertise, conducted 53% of the experiments, while the remaining data was collected by an inexperienced operator who had recently begun operating the excavator. The experiments represent realistic construction operations, i.e., no directions were provided to the operators on how to perform the tasks to increase the robustness of the proposed algorithm.

An IMU equipped with a three-axis accelerometer and gyroscope is a versatile sensor module widely used in various applications. The accelerometer measures acceleration along three orthogonal axes, providing information about changes in velocity and orientation. Simultaneously, the gyroscope measures angular velocity. Together, these sensors enable the IMU to capture intricate motion dynamics in applications such as orientation tracking, gesture recognition, robotics, and virtual reality. The combination of accelerometer and gyroscope data allows for accurate and real-time monitoring of an object's movement and orientation in three-dimensional space, facilitating precise motion analysis and enhancing the capabilities of devices ranging from smartphones to unmanned aerial vehicles (Slaton et al. 2020). Figure 4 shows the used IMUs in the experiments. The IMUs were manufactured by Novatron® Ltd. and are placed in robust casings. To measure the orientation and angular velocities of the excavator's moving parts, four IMUs were mounted on the machine's bucket, arm, boom, and cabin. Figure 5 depicts the configuration of IMUs on the excavator. Using the Xsite® machine control system, the IMUs were precalibrated. The controller area network (CAN) bus is used to transfer the sensor data. A Kvaser leaf light CAN to USB interface is used to connect the CAN bus to the MathWorks® Simulink model for the data collection, and the data sampling frequency f_s is set to 200 Hz. The duration of the dataset is around 3 h, which means that based on the data sampling frequency, approximately 2,160,000 data points were collected for each channel of the sensor. The amount of data corresponding to each task is shown in Table 1.

Fig. 4 The IMU used in the data collection phase

Fig. 5 The configuration of IMUs on the excavator

Each sensor unit determines the quaternion orientation of the sensor based on measurements from the accelerometer and gyroscope. Then, the joint angles between each moving component of the machine connected by the revolute joints are calculated using the quaternion measurements. The quaternion to Euler angles conversion is expressed in Eq. (2):

$$\begin{aligned} q(t) &= \left[q_w(t)\ q_x(t)\ q_y(t)\ q_z(t)\right]^T, \\ |q|^2 &= q_w^2 + q_x^2 + q_y^2 + q_z^2 = 1, \end{aligned} \tag{1}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_w q_x + q_y q_z)}{1-2\left(q_x^2+q_y^2\right)}\right) \\ -\pi/2 + 2\arctan\sqrt{\frac{1+2(q_w q_y - q_x q_z)}{1-2(q_w q_y - q_x q_z)}} \\ \arctan\left(\frac{2(q_w q_z + q_x q_y)}{1-2\left(q_y^2+q_z^2\right)}\right) \end{bmatrix}, \tag{2}$$

where q represents the unit quaternion, and ϕ, θ, and ψ indicate the roll (rotation around the x-axis), pitch (rotation around the y-axis), and yaw (rotation around the z-axis), respectively (Bernardes and Viollet 2022). The global angular velocities are measured via the gyroscope in the IMU. Global angular velocities are also used to calculate the local angular velocities of each moving part. The local

Table 1 The duration of different tasks in the collected dataset

	Task			
	Loading	Trenching	Grading	Idling
Duration (min)	68.43	41.14	35.26	37.27

angular velocity is the actual angular velocity of the particular body part from which the movements of the other machine elements have been subtracted. The local angular velocity describes the movement of the measured body part as a result of the operator's movement of that particular body part. On the other hand, the global angular velocity includes all movement caused by the machine. Figure 6 shows the local angular velocities and orientation variables. The quaternion data were discarded for further analysis, and the excavator activity identification algorithm uses the joint angles and angular velocities of the machine parts as input data. The input variables consist of the angular velocities of four IMUs (three axes per sensor unit), the local angular velocity of the boom (ω_2), the local angular velocity of the arm (ω_3), the local angular velocity of the bucket (ω_4), the pitch angle of the boom (θ_2), the pitch angle of the arm (θ_3), and the pitch angle of the bucket (θ_4).

2.2 Data windowing

To identify the major activities of an excavator, a data windowing approach is utilized in the proposed algorithm. The position of a moving object is represented by a single data point at a single instant of time, whereas working cycles are composed of sequential motions distributed over a period (for instance, the trenching task does not occur instantly but over a period). In the data windowing process, a defined windowing function is moved along all time-series data dividing a data sequence into numerous smaller, constant-sized pieces of data. A window is a group of consecutive time series data points. In the literature review, most studies chose the time window within [1, 5] s, since they mostly focus on short-term motions and sub-tasks of an excavator. However, this paper mostly concentrates on tasks that are composed of several sub-tasks and take a much longer time compared to an individual sub-task. Based on the Komatsu® performance handbook (Komatsu 2013), the duty cycle of an excavator mostly takes approximately 10 to 20 s. In the presented algorithm, a sliding rectangular windowing function with five different window sizes (10, 12, 15, 18, and 20 s) and with four alternative overlapping configurations (0%, 25%, 50%, and 75% overlap between two consecutive windows) are utilized.

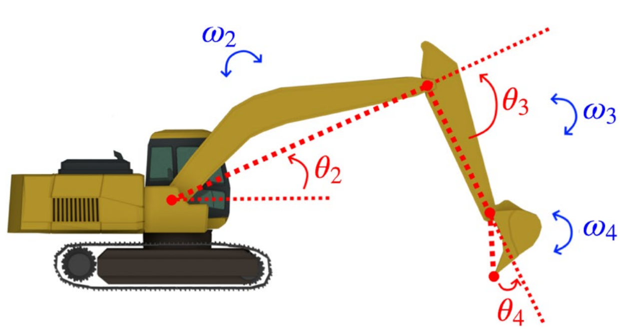


Fig. 6 The local angular velocities and orientation variables are visualized on an excavator's side profile

2.3 Data annotation

The so-called ground truth information must be coupled with the data samples in supervised learning algorithms. An external USB webcam with a frame rate of 20 frames per second was mounted inside the excavator's cabin to record the operations. The webcam is connected to the MathWorks® Simulink model using the Image Acquisition Toolbox provided by MathWorks®. Using this scheme, the recorded video is completely synced with the collected dataset. In the next step, the dataset is manually labeled using MathWorks® MATLAB. If the activity changes, the user informs the program, and the label is changed. Finally, the most frequent label in each window is chosen as the label of that window. It should be taken into account that the recorded videos are used only for data annotation, and the classification models are only dependent on the motion information.

2.4 Feature extraction

After segmenting the time series data into windows, to extract beneficial information from each labeled data window in the dataset, feature extraction is conducted prior to the model training. The basic idea behind feature extraction is to generate variables from the raw data to maximize the amount of information related to the phenomenon that a classifier will be used to model. Ten time-domain statistical features (also called feature vectors), including (1) mean, (2) maximum, (3) minimum, (4) standard deviation, (5) mean absolute deviation, (6) root mean square, (7) peak to peak, (8) interquartile range, (9) skewness, and (10) kurtosis, are retrieved from each window in the gathered dataset.

2.5 Feature selection

The next step of data preparation for a classification algorithm is feature selection. The feature selection aims to find the features that contain the most relevant information to the classification problem, reduce the size of the feature space, and provide a faster and more efficient algorithm. It should be noted that some features could not be useful because they do not contain value-adding information and can therefore be discarded for further investigation. Feature selection algorithms are classified into three main groups:

- Filter-type feature selection algorithms assess the importance of features based on their characteristics, such as variance and relevance to the response variable. Important features are chosen during data preprocessing, independently of the training algorithm, and are used to train the model.

- Wrapper-type feature selection algorithms initially train the model with a subset of features and then iteratively add or remove features based on a selection criterion. This criterion directly evaluates the impact on model performance when adding or removing a feature. The process continues until certain stopping criteria are met.
- Embedded-type feature selection algorithms incorporate feature importance evaluation into the model learning process. After training a model, the algorithm identifies the importance of each feature within the trained model. This method selects features that complement the specific learning process.

Filter-type feature selection algorithms have multiple advantages, including computational efficiency, independence from learning algorithms, scalability to high-dimensional datasets, feature ranking capabilities, reduced overfitting concerns, and broad applicability across machine learning tasks (Müller and Guido 2016; Guyon and Elisseeff 2003). In the context of excavator task recognition, employing filter-type feature selection approaches offers several advantages. Filter-type methods allow for efficient preprocessing by selecting relevant features based on their variance and correlation with task labels. By focusing on intrinsic feature characteristics rather than complex model interactions, filter-type selection methods reduce computational complexity and processing time and ensure the robustness and generalizability of the classification model. Moreover, the simplicity and interpretability of the selected features facilitate insights into the underlying mechanisms of the activity recognition algorithm. In this study, three different subsets of features using the three most important and common filter-type feature selection algorithms for classification problems are used to train supervised classification models: (1) selected features using the ReliefF algorithm, (2) selected features using the minimum redundancy maximum relevance (MRMR) algorithm, and (3) selected features using the Chi-squared test.

2.5.1 ReliefF

ReliefF is a popular and effective feature selection algorithm designed to identify and prioritize relevant features in high-dimensional datasets. The primary goal of ReliefF is to evaluate the importance of features based on their ability to distinguish between instances with similar and dissimilar class labels. The algorithm works by iteratively sampling instances from the dataset and updating feature weights according to their relevance. For each instance in the dataset, ReliefF calculates the “hit” and “miss” scores for each feature. The “hit” score is increased if the feature values of the nearest instance with the same class label are similar and decreased if they are dissimilar. Conversely, the “miss” score is increased if the feature values of the nearest instance with a different class label are similar and decreased if they are dissimilar. After sampling a sufficient number of instances, the final feature weights represent their relevance in distinguishing between different classes. Higher weights indicate more relevant features. One advantage of ReliefF is its ability to handle noisy and redundant features, making it robust in real-world scenarios (Müller and Guido 2016).

2.5.2 Minimum redundancy maximum relevance (MRMR)

The minimum redundancy maximum relevance (MRMR) algorithm is a feature selection method designed to identify a subset of features that maximizes the relevance to the target variable while minimizing redundancy among the selected features. The algorithm operates in two main steps: relevance evaluation and redundancy minimization. In the relevance evaluation step, MRMR computes the relevance of each feature to the target variable. Common metrics for measuring relevance include mutual information or correlation coefficients. In the redundancy minimization step, MRMR considers the pairwise redundancy between features. The algorithm aims to select features that are individually relevant to the target variable while maintaining a diverse set of features to minimize redundancy. It achieves this by maximizing the mutual information between each selected feature and the target variable while minimizing the mutual information between the selected features. The final feature subset obtained by MRMR represents a trade-off between high relevance to the target variable and low redundancy among the selected features. MRMR is particularly effective in scenarios with high-dimensional data, where selecting a subset of the most informative features can improve model performance and interpretability (Müller and Guido 2016).

2.5.3 Chi-squared test

The Chi-squared test is a statistical method commonly used for feature selection. It assesses the independence between a feature and the target variable by comparing the observed distribution of values to the expected distribution under the assumption of independence. For a given feature and target variable, the Chi-squared test computes a statistic that quantifies the difference between the observed and expected distributions. The higher the Chi-squared statistic, the more significant is the association between the feature and the target variable. Once the Chi-squared statistic is calculated for each feature, a significance threshold (e.g., determined by a p-value) is used to identify features that are statistically significant in their association with the target variable. Features exceeding this threshold are retained for further analysis, while others may be considered less relevant. It is a simple yet powerful method for identifying features that contribute significantly to the predictive power of a model (Müller and Guido 2016).

2.6 Classification models

Supervised learning is a foundational paradigm in machine learning where a model is trained on a labeled dataset to make predictions or infer patterns in unseen data. In this learning framework, the algorithm is provided with a dataset containing input–output pairs, where the outputs (labels or target values) are known for the corresponding inputs. The goal of supervised learning is to learn a mapping or relationship between the input features and the target variable so that the model can generalize and make accurate predictions on new, unseen data. During training, the algorithm iteratively adjusts its parameters based on the discrepancy between its predictions and the actual outcomes, aiming to minimize the prediction error (Bishop 2006).

Although activity recognition algorithms are proposed using both supervised and unsupervised methods, supervised learning algorithms show better performance for equipment activity recognition (Golparvar-Fard et al. 2013). The characteristics and amount of data will determine which supervised learning algorithm should be utilized. As a result, there is no one best classifier, and each method needs to be assessed independently. Based on the most commonly used supervised classifiers in construction resource activity identification algorithms in the literature review, four classifiers, including a support vector machine (SVM), a k-nearest neighbors (KNN) algorithm, a naive Bayes classifier, and a decision tree (DT), are employed to classify the tasks based on the given dataset.

2.6.1 Support vector machine (SVM)

Support vector machines (SVM) are powerful supervised learning models used for classification and regression tasks. SVMs work by finding the optimal hyperplane that separates different classes in the feature space. Consider a binary classification problem with two classes, labeled as 1 and −1. The SVM aims to find a hyperplane represented by the equation:

$$w \cdot x + b = 0, \tag{3}$$

where w is the weight vector, x is the input feature vector, and b is the bias term. The decision function is given by:

$$f(x) = \text{sign}(w \cdot x + b). \tag{4}$$

The goal is to find the optimal w and b that maximize the margin between the two classes. The margin is the distance between the hyperplane and the nearest data point of either class. Let x_+ and x_- be two support vectors on the positive and negative sides of the hyperplane, respectively. The margin is given by:

$$\text{margin} = \frac{2}{\|w\|}. \tag{5}$$

The SVM optimization problem is to maximize the margin subject to the constraint that all data points are correctly classified:

$$\begin{aligned} &\text{maximize} \quad \frac{2}{\|w\|} \\ &\text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \text{for all } i, \end{aligned} \tag{6}$$

where y_i is the class label of the i-th data point, and (x_i, y_i) are the training samples. This problem can be converted into a minimization problem by introducing a regularization term:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(w \cdot x_i + b)), \tag{7}$$

where C is the regularization parameter, and N is the number of training samples. SVMs are effective in finding the optimal hyperplane for separating classes in the feature space, providing a robust solution for classification problems (Bishop 2006).

2.6.2 K-nearest neighbor (KNN)

The k-nearest neighbors (KNN) algorithm is a simple and intuitive supervised learning method used for classification and regression tasks. In the context of classification, given a new data point, the algorithm assigns it to the majority class among its k-nearest neighbors in the feature space. Let X be the feature space and Y be the corresponding labels. For a new data point x_{new}, the algorithm identifies its k-nearest neighbors by measuring distances, commonly using the Euclidean distance metric:

$$d(x_i, x_{\text{new}}) = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{\text{new}j})^2}, \tag{8}$$

where x_i is a training data point, x_{ij} is the j-th feature of x_i, and n is the number of features. The classification decision for x_{new} is based on the majority class among its k-nearest neighbors. In the case of ties, a common approach is to assign the class based on a distance-weighted vote. The choice of k is a crucial parameter that affects the algorithm's performance. A smaller k leads to more flexible models, but may be sensitive to noise, while a larger k provides smoother decision boundaries but might overlook local patterns. In summary, KNN is a non-parametric, instance-based learning algorithm that makes predictions based on the local similarity of data points in the feature space (Bishop 2006).

2.6.3 Naive Bayes

The naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' theorem, particularly designed for classification tasks. It assumes that features are conditionally independent given the class label, which is a simplifying yet powerful assumption. Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of features and Y be the class label. The goal is to compute the probability of a class label y given the feature vector x, denoted as $P(Y = y|X = x)$. According to Bayes' theorem:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y) \cdot P(Y = y)}{P(X = x)}. \tag{9}$$

The "naive" assumption in naive Bayes is that features are conditionally independent given the class, allowing us to express the likelihood $P(X = x|Y = y)$ as the product of individual feature probabilities:

$$P(X = x|Y = y) = \prod_{i=1}^{n} P(x_i|Y = y). \tag{10}$$

The classifier assigns the class label y that maximizes the posterior probability $P(Y = y|X = x)$. In practice, we often use the logarithm of probabilities to avoid numerical underflow:

$$\hat{y} = \arg\max_{y} \left[\log(P(Y = y)) + \sum_{i=1}^{n} \log(P(x_i|Y = y)) \right]. \tag{11}$$

Training the naive Bayes classifier involves estimating the prior probabilities $P(Y = y)$ and the conditional probabilities $P(x_i|Y = y)$ from the training data. Naive Bayes is computationally efficient and works well for high-dimensional data, although the independence assumption may not always hold in real-world scenarios (Bishop 2006).

2.6.4 Decision tree (DT)

A DT is a versatile and widely used machine learning algorithm for both classification and regression tasks. It builds a tree-like structure by recursively partitioning the feature space based on the values of different features, and each leaf node represents the predicted outcome. The decision-making process in a DT involves evaluating conditions at each node and following the corresponding branch. At each internal node, the tree asks a binary question based on a specific feature, and the data is split into subsets accordingly. This process continues until a stopping criterion is met, such as a predefined tree depth or a minimum number of samples in a node. In a classification task, each leaf node corresponds to a class label, and the majority class of the samples in that node is the predicted class. The construction of a DT involves selecting the best features for splitting at each node. Common metrics for measuring the impurity of a node are Gini impurity and entropy. The algorithm aims to minimize the impurity in the resulting child nodes. DTs are interpretable, and their visual representation provides insights into the decision-making process. However, they may be prone to overfitting, especially when the tree is deep. Strategies like pruning, limiting tree depth, and setting minimum samples per leaf help mitigate overfitting (Bishop 2006).

2.7 Performance measures

In this research, the performance of classifiers is assessed using four standard performance metrics: *accuracy*, *precision*, *recall*, and F_1*Score*. The *accuracy* metric is calculated as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \tag{12}$$

where TP denotes true positives, FP denotes false positives, FN denotes false negatives, and TN denotes true negatives. The *accuracy* is a fundamental metric that provides an overall measure of correct predictions. It calculates the ratio of correctly predicted instances to the total instances. While accuracy is informative, it may not be sufficient on its own, especially in imbalanced datasets, where the class distribution is skewed (Bishop 2006). The cost of misclassification is taken into consideration when calculating the *precision* and *recall* of the model. The *recall* is the percentage of true instances (i.e., true positive + false negative) that are accurately predicted as positive (i.e., true positive), whereas the *precision* is the percentage of predicted positive instances (i.e., true positive + false positive) that are truly positive (i.e., true positive). The *precision* and *recall* metrics are computed using Eqs. (13) and (14), respectively.

$$\text{precision} = \frac{TP}{TP + FP} \times 100\%, \tag{13}$$

$$\text{recall} = \frac{TP}{TP + FN} \times 100\%. \tag{14}$$

The *precision* is crucial when the cost of false positives is high. The *precision* is particularly relevant when misclassifying positive instances has significant consequences. The *recall*, also known as sensitivity or true positive rate, focuses on the ability to capture all positive instances. The *recall* is important when missing positive instances has severe implications, such as in anomaly detection (Bishop 2006). High *precision* and *recall* values are desirable, but it might be difficult to maximize both metrics for a classification model. The $F_1 Score$, which is the harmonic mean of *precision* and *recall*, is computed using Eq. (15).

$$F_1\text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{15}$$

The F_1Score provides a balanced metric that considers both false positives and false negatives. It is especially useful in situations where there is an uneven class distribution. The F_1Score becomes valuable when striving for a trade-off between precision and recall (Bishop 2006).

3 Results

In this section, the results of the proposed method are illustrated. Firstly, a small portion of the dataset is visualized to show the difference between experienced and inexperienced operators' behavior. Then, the dataset is divided into train and test subsets. The most important features are obtained using feature selection algorithms. In the next step, the classification methods are trained using selected features. The effects of time windows and overlapping configurations are evaluated. Finally, *k*-fold cross-validation is performed to show the robustness of the suggested approach.

3.1 Data visualization

The difference between data collected from experienced and inexperienced operators is presented. Figure 7 shows the pitch angles of the boom, arm, and bucket in two loading operations that are performed by experienced and inexperienced operators. In these operations, working conditions, including swing angle (around 60°) and type of material (sand), are the same. As shown, the experienced operator can easily control the manipulator of the excavator, and the pitch angles show cyclic behaviors. However, the inexperienced operator is unable to effectively control the manipulator, and there are different movements.

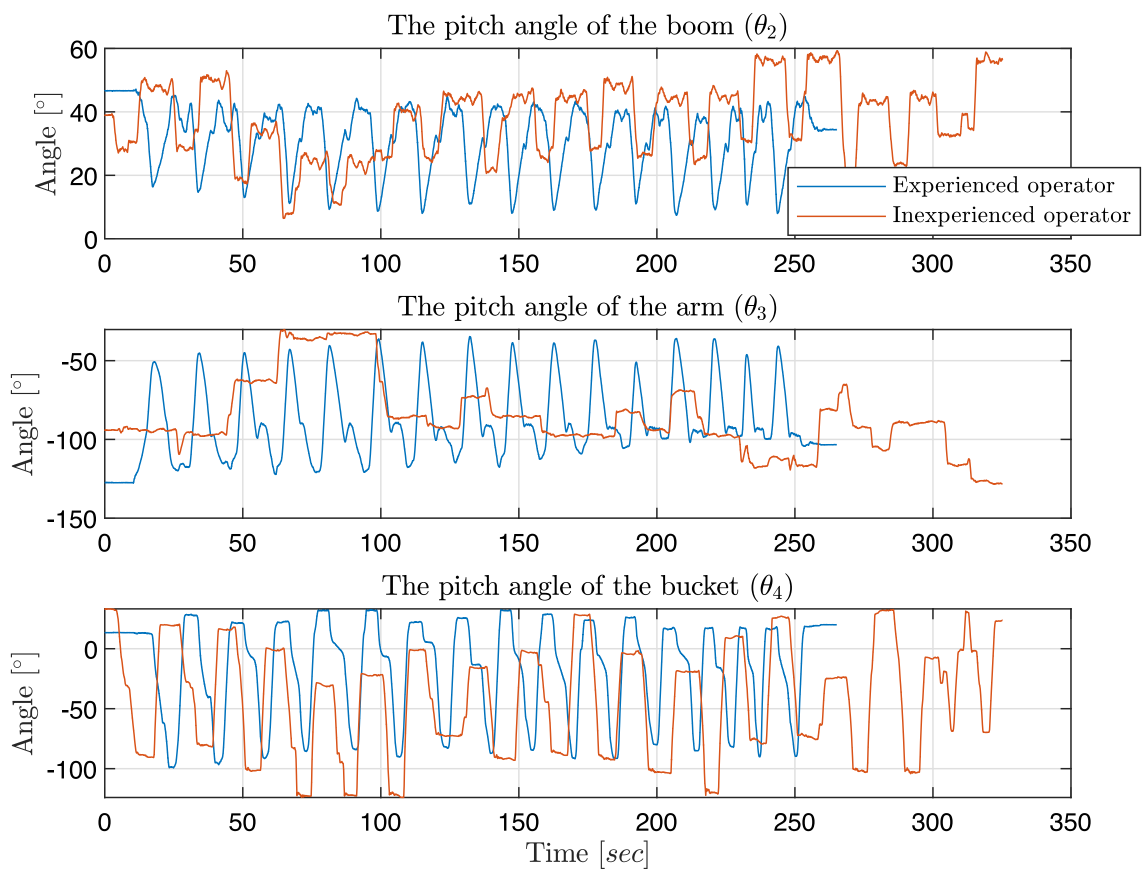


Fig. 7 The pitch angles of the boom, arm, and bucket in two loading experiments operated by experienced and inexperienced operators

3.2 Classification model training and evaluation

To analyze data-driven modeling approaches, different subsets of the dataset must be utilized for model training and testing. In our research, to show the robustness of the classification method, the dataset is divided into training and testing datasets, with 50% of the data used for training and 50% used for testing. The random splitting involves randomly selecting instances from the dataset to populate each subset, ensuring that the data is representative of the overall distribution. This randomness helps prevent bias in the model evaluation process and ensures that the model's performance is assessed on unseen data. Additionally, random splitting allows for repeatability and reproducibility, as the process can be easily replicated to validate the consistency of the results across multiple iterations. The proposed approach has been implemented using Statistics and Machine Learning Toolbox in MathWorks® MATLAB R2021a on a laptop with a 1.8 GHz Intel Core i7 CPU and 16 GB of RAM running on a Windows 10 operating system. This toolbox provides functions and apps to describe, analyze, and model data. It includes supervised, semi-supervised, and unsupervised machine learning algorithms, multidimensional data analysis, and feature selection methods.

Firstly, the three feature selection algorithms introduced in Sect. 2.5 are applied to the training dataset to select the most important features. In the dataset, there are 180 different feature vectors (18 measurements × 10 features). The 35 most important features of each feature selection algorithm are chosen as the main features for training the classification methods. For the sake of briefness, only the ten most important features obtained using the MRMR algorithm are presented in Table 2.

In the next step, the *accuracy*, *precision*, *recall*, and $F_1 Score$ of different classification models utilizing different feature selection algorithms with associated time window and overlapping configurations are presented in Table 3. The time window and overlapping configurations show the highest *accuracy* for the corresponding classification model and feature selection algorithm. Also, for each classification model, the best performance (highest *accuracy*) is highlighted in bold. The results show that the proposed classification algorithm can automatically recognize the tasks of an excavator with an *accuracy* of more than 99%. Also, it can be concluded that the IMU sensors, their placement on the machine, and the motion variables are chosen correctly. In the next step, the confusion matrices of the twelve classification algorithms (introduced in Table 3) are presented in Table 4. Two classification algorithms, including the SVM with MRMR feature selection algorithm, a time window of 20 sec, and 0% overlapping, and the KNN classifier with MRMR feature selection algorithm, a time window of 20 s, and 50% overlapping, have the highest *accuracy* of 99.62%.

Table 2 The ten most important features obtained using the MRMR algorithm

Number	Measurement	Features
1	Angular velocity of bucket (ω_x)	Mean
2	Angular velocity of boom (ω_x)	Kurtosis
3	Angular velocity of boom (ω_x)	Mean
4	Pitch angle of arm (θ_3)	Root mean square
5	Angular velocity of frame (ω_y)	Skewness
6	Angular velocity of boom (ω_y)	Skewness
7	Angular velocity of bucket (ω_z)	Kurtosis
8	Angular velocity of boom (ω_z)	Mean
9	Angular velocity of bucket (ω_z)	Mean absolute deviation
10	Angular velocity of frame (ω_y)	Mean absolute deviation

Table 3 The performance measures for different classifiers with different configurations

Classification models	Feature selection	Time window (sec)	Overlapping (%)	Metrics			
				Accuracy (%)	Precision (%)	Recall (%)	F_1 *Score* (%)
SVM	ReliefF	18	50	99.31	99.33	99.36	99.34
	MRMR	**20**	**0**	**99.62**	**99.59**	**99.50**	**99.55**
	Chi-squared	20	0	99.24	99.10	99.50	99.30
KNN	ReliefF	20	50	99.23	99.18	99.12	99.15
	MRMR	**20**	**50**	**99.62**	**99.54**	**99.62**	**99.58**
	Chi-squared	18	50	99.31	99.36	99.28	99.32
NB	**ReliefF**	**20**	**75**	**98.93**	**98.76**	**98.91**	**98.83**
	MRMR	20	0	98.86	98.94	98.75	98.85
	Chi-squared	20	50	98.85	98.93	98.65	98.79
DT	ReliefF	20	75	98.25	98.01	98.41	98.21
	MRMR	**15**	**75**	**98.71**	**98.85**	**98.59**	**98.72**
	Chi-squared	18	50	98.11	98.05	98.37	98.21

Table 4 The confusion matrices of different classification algorithms

Classification models	True	ReliefF Predicted L	T	G	I	MRMR L	T	G	I	Chi-squared L	T	G	I
SVM	L[a]	217	2	0	0	100	0	0	0	98	1	1	0
	T[b]	1	131	0	0	0	60	0	0	0	60	0	0
	G[c]	0	1	111	0	0	1	49	0	0	0	50	0
	I[d]	0	0	0	118	0	0	0	54	0	0	0	54
KNN	L	196	0	0	0	196	0	1	0	218	0	0	1
	T	0	118	0	0	0	118	0	0	2	130	0	0
	G	0	3	97	0	0	1	99	0	0	0	111	1
	I	0	0	0	106	0	0	0	106	0	0	0	118
NB	L	385	2	1	0	99	1	0	0	196	1	0	0
	T	1	230	4	0	0	60	0	0	0	118	0	0
	G	0	1	197	0	1	1	48	0	2	1	97	0
	I	0	0	2	208	0	0	0	54	0	2	0	54
DT	L	378	2	8	0	521	3	1	0	213	5	1	0
	T	1	232	1	1	4	311	2	0	3	127	2	0
	G	1	4	193	0	6	2	260	0	0	0	112	0
	I	0	0	0	210	0	0	0	284	0	0	0	118

Feature selection algorithms: ReliefF, MRMR, Chi-squared (Predicted)

The time window and overlapping configurations of the classification algorithm are shown in Table 3

[a] L stands for loading operation

[b] T stands for trenching operation

[c] G stands for grading operation

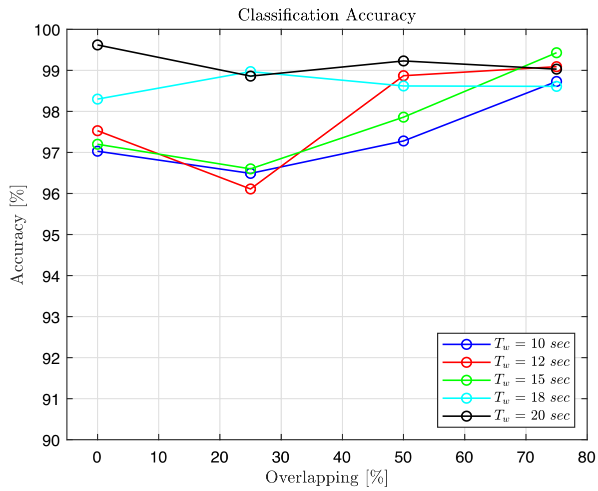
[d] I stands for idling

3.3 Time window analysis

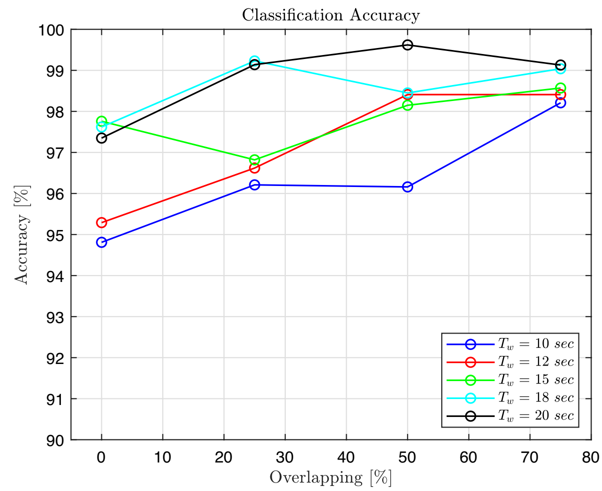
Secondly, the impacts of the time window on the classification algorithms are analyzed. Figure 8 a shows the classification accuracy of the SVM classifier with the MRMR feature selection algorithm using different configurations. On average, the time windows of 20 s and 10 s show the highest and lowest *accuracy* in different overlapping configurations, respectively. The classification accuracy of the KNN classifier with the MRMR feature selection algorithm using different configurations is presented in Fig. 8b. In total, the time windows of 20 s and 10 s show the highest and lowest *accuracy*, respectively. The classification accuracy of the Naive Bayes classifier with the ReliefF feature selection algorithm and the DT with the MRMR feature selection algorithm are illustrated in Fig. 8c and d, respectively. The time window of 20 and 18 s shows higher performance compared to the time window of 10, 12, and 15 s. The average classification *accuracy* of different classification algorithms for different time windows is presented in Table 5.

3.4 Overlapping analysis

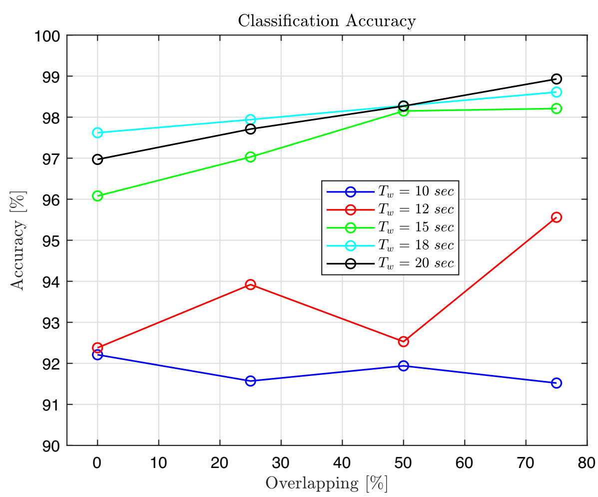
Thirdly, the impacts of overlapping configuration on the classification algorithm are assessed. The classification *accuracy* of the SVM classifier with MRMR feature selection algorithm using different overlapping configurations is demonstrated in Fig. 9a. On average, the overlaps of 75% and 25% illustrate the highest and lowest classification *accuracy*, respectively. Then, the classification *accuracy* of the KNN classifier with the MRMR feature selection algorithm is shown in Fig. 9b. Generally, the overlaps of 75% and 0% show the highest and lowest *accuracy*, respectively. The classification *accuracy* of the naive Bayes classifier with the RelieF feature selection algorithm and the DT classifier with the MRMR feature selection algorithm is presented in Fig. 9c and d, respectively. On average, the overlaps of 75% and 50% show the highest classification *accuracy*. The average classification *accuracy* of different classification algorithms for different overlapping configurations is illustrated in Table 6.



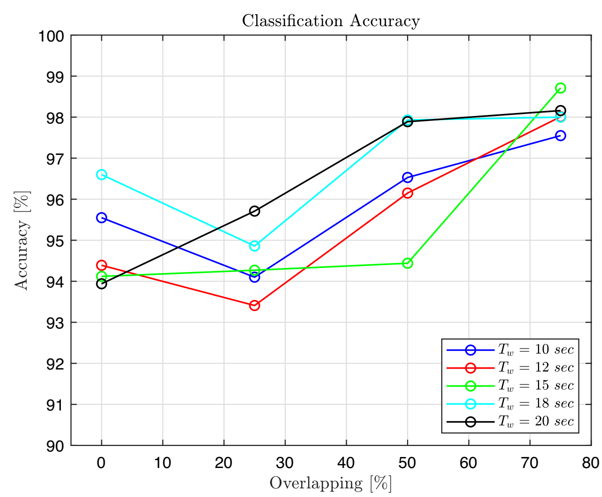
(a) The *accuracy* of the SVM classifier with the MRMR feature selection algorithm.



(b) The *accuracy* of the KNN classifier with the MRMR feature selection algorithm.



(c) The *accuracy* of the Naive Bayes classifier with the ReliefF feature selection algorithm.

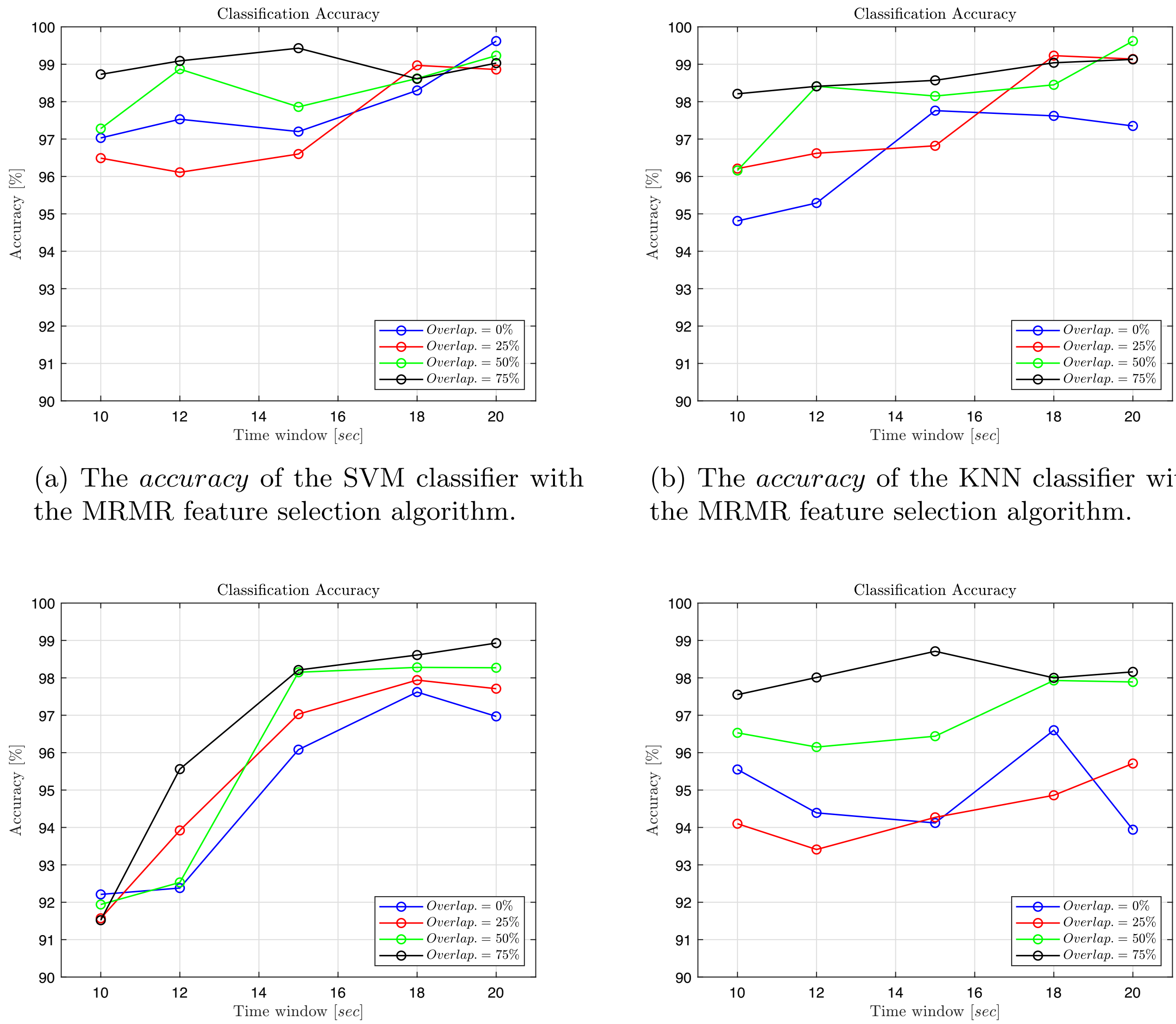


(d) The *accuracy* of the DT with the MRMR feature selection algorithm.

Fig. 8 The analysis of the impacts of the time window on different classification algorithms and feature selection algorithms. The combinations of classification methods and feature selection techniques are chosen based on the highest *accuracy* in Table 3

Table 5 The average classification *accuracy* of different classification algorithms in different time windows (the best performance is highlighted in bold)

Classification models	Feature selection	Overlapping [%]	Time window [sec]				
			10	12	15	18	20
SVM	MRMR	[0, 25, 50, 75]	97.38%	97.90%	97.77%	98.62%	**99.18%**
KNN	MRMR	[0, 25, 50, 75]	96.35%	97.18%	97.82%	98.58%	**98.81%**
NB	ReliefF	[0, 25, 50, 75]	91.81%	93.60%	97.37%	**98.11%**	97.97%
DT	MRMR	[0, 25, 50, 75]	95.93%	95.49%	95.38%	**96.84%**	96.42%



(a) The *accuracy* of the SVM classifier with the MRMR feature selection algorithm.

(b) The *accuracy* of the KNN classifier with the MRMR feature selection algorithm.

(c) The *accuracy* of the Naive Bayes classifier with the ReliefF feature selection algorithm.

(d) The *accuracy* of the DT with the MRMR feature selection algorithm.

Fig. 9 The analysis of the impacts of the overlapping configuration on different classification algorithms and feature selection algorithms

Table 6 The average classification *accuracy* of different classification algorithms in different overlapping configurations (the best performance is highlighted in bold)

Classification models	Feature selection	Time window [sec]	Overlapping [%]			
			0	25	50	75
SVM	MRMR	[10,12,15,18,20]	97.94%	97.40%	98.37%	**98.98%**
KNN	MRMR	[10,12,15,18,20]	96.57%	97.60%	98.16%	**98.67%**
NB	ReliefF	[10,12,15,18,20]	95.05%	95.63%	95.83%	**96.57%**
DT	MRMR	[10,12,15,18,20]	94.92%	94.47%	96.99%	**98.09%**

3.5 K-fold cross-validation

To demonstrate the robustness of the suggested classification algorithm, the k-fold cross-validation is also carried out. Cross-validation is frequently employed in applied machine learning to assess how well a model performs on data that has not been observed. A dataset is randomly partitioned into k groups, or folds, of the same size. One fold serves as a holdout set, while the other $k-1$ folds are utilized to fit the model. After this procedure has been conducted k times, the

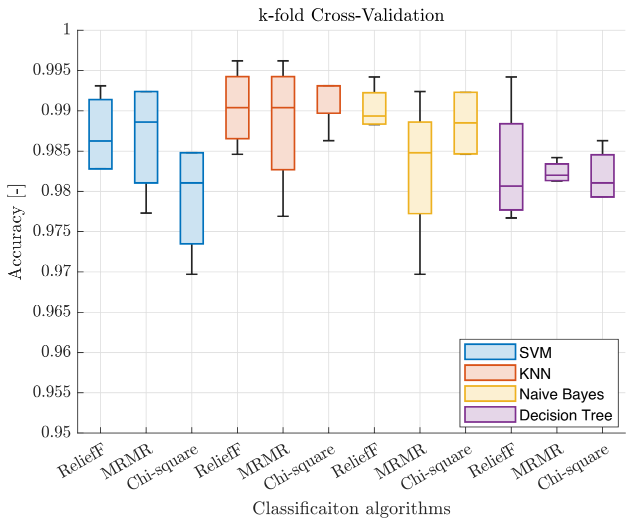


Fig. 10 Analysis of *k*-fold cross-validation. Each box chart displays the following information: median, lower and upper quartiles, and minimum and maximum values

final estimate is obtained by averaging the outcomes of the *k* holdout sets. In this paper, the value of *k* is assumed to be 4. Figure 10 displays the results of *k*-fold cross-validation for the 12 classification algorithms presented in Table 3. The *accuracy* of classification algorithms is comparable to the outcomes obtained in Table 3.

4 Discussion

The presented method can be utilized to automatically recognize the major activities of excavators. Integrating the task recognition method with other systems or applications can significantly improve the overall efficiency, safety, and control. By linking task recognition with productivity monitoring systems, real-time dashboards can be generated to display task-specific metrics, including productivity definitions, completion times for specific tasks, and equipment utilization. This enables managers and operators to closely monitor progress and pinpoint areas for enhancement. Identifying behavioral patterns that might pose safety risks allows for their incorporation into training programs, ensuring a proactive approach to safety. For instance, if an operator engages in unsafe task execution, the system can promptly trigger an alert for corrective action. Task recognition data also proves instrumental in optimizing collaboration between human operators and autonomous elements. Predictive maintenance for equipment becomes more precise by leveraging task recognition data to analyze usage patterns, facilitating proactive scheduling, and minimizing downtime. Historical analysis of task recognition data further aids in identifying trends and patterns, offering valuable insights for informed decision-making on resource allocation, equipment upgrades, and process enhancements.

In the future, the method should be extended to other HDMMs, including front-end loaders and compactors. Motion sensors such as IMUs should be installed on different moving parts of a machine to be able to track different types of activities. For example, IMUs can be installed on the bucket, boom, and cabin of front-end loaders to recognize activities.

There are several limitations and challenges in the proposed method. Firstly, the duration of the dataset is around 3 h, and test and train datasets are collected using the same machine. The dataset should be enlarged by gathering data from various excavators of different sizes. Also, the resulting model should be tested with an operator whose data was not involved in the training dataset since human operators use machines in different ways. To ensure that the algorithm is robust, various operational conditions, such as different swing angles, digging depths, material types, and weather conditions should be taken into account throughout the data collection phase. Nonetheless, expanding the dataset poses challenges due to the significant costs associated with renting excavators and hiring operators, underscoring the need for judicious resource allocation and careful planning. The labeling of the dataset is another drawback of the suggested approach. Labeling is a key and time-consuming step in supervised learning techniques. Moreover, other classification models and feature selection methods can be tested on the collected dataset.

Beyond dataset-related considerations, our method also grapples with inherent challenges associated with IMUs. These challenges encompass sensor noise, calibration discrepancies, synchronization issues, sensor placement intricacies, and maintenance concerns. Addressing these factors is critical to ensure the reliability and accuracy of motion data captured by IMUs. Furthermore, it is essential to discuss how our proposed method navigates through variances in excavator types, operator proficiencies, task complexities, and environmental conditions, all of which can significantly influence motion data and classification outcomes. Addressing these challenges can enhance the robustness and applicability of our proposed method in real-world excavator task recognition scenarios.

5 Conclusion

In this paper, a data-driven method is proposed to identify the main working cycles of an excavator, including loading, grading, trenching, and idling. Firstly, a dataset spanning 3 h, consisting of orientation variables and angular velocities, is collected using a medium-rated excavator equipped with four IMUs attached to different moving parts, including bucket, arm, boom, and swing body. The operations were performed by both a skilled and an unskilled operator under varying conditions, including different material types, swing angles, digging depths, and weather conditions. Four classification techniques, namely support vector machine (SVM), k-nearest neighbor (KNN), decision tree (DT), and Naive Bayes, along with three feature selection approaches, including the ReliefF algorithm, MRMR algorithm, and Chi-squared test, are utilized for training a classification model. The proposed approach achieves a remarkable 99% *accuracy* using SVM with the MRMR algorithm. Then, the impacts of different time windows and overlapping configurations on the classification *accuracy* are analyzed. On average, the time window of 20 s and overlapping of 75% demonstrate high accuracy. Comprehensive analyses attest to the algorithm's resilience and adaptability in real-world scenarios. This research significantly contributes to the field, presenting a robust solution for automating excavator task recognition, pivotal for enhancing productivity and operational efficiency in heavy-duty mobile machine operations.

Acknowledgements The MORE project is an innovative European Industrial Doctorate (EID) research and training programme that tries to answer the challenges of HDMMs. The MORE project contributes to the development of state-of-the-art HDMMs by proposing effective solutions driven by digitalization and artificial intelligence (AI). There are eight researchers in the MORE project that work on three main work packages: (1) Process, (2) Machine, and (3) Control. In the process work package, we investigate a set of tasks that are performed by a machine or machines to achieve the purpose of work.

The authors would like to acknowledge and express their sincere gratitude for the continued support of Janne Koivumäki, Jari Valtonen, and Johanna Ylisaari during the data collection and analysis steps.

Funding Open Access funding enabled and organized by Projekt DEAL. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 858101.

Data availability The data presented in this study is confidential and intended solely for this research. Access to the data is restricted and should not be shared, reproduced, or used for any other purpose without explicit permission from the authors. Any requests for data access should be directed to the corresponding author, who will evaluate the legitimacy of the request in accordance with our data-sharing policies and ethical guidelines.

Declarations

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

References

Ahn CR, Lee S, Peña-Mora F (2015) Application of low-cost accelerometers for measuring the operational efficiency of a construction equipment fleet. J Comput Civ Eng 29(2):04014042. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000337

Bae J, Kim K, Hong D (2019) Automatic identification of excavator activities using joystick signals. Int J Precis Eng Manuf 20(12):2101–2107. https://doi.org/10.1007/s12541-019-00219-5

Bao R, Sadeghi MA, Golparvar-Fard M (2016) Characterizing construction equipment activities in long video sequences of earthmoving operations via kinematic features. In: Construction Research Congress, pp 849–858. https://doi.org/10.1061/9780784479827.086

Bernardes E, Viollet S (2022) Quaternion to Euler angles conversion: a direct, general and computationally efficient method. PLoS One 17(11):1–13. https://doi.org/10.1371/journal.pone.0276302

Bishop CM (2006) Pattern recognition and machine learning. Springer, Singapore. https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

Chen C, Zhu Z, Hammad A (2020) Automated excavators activity recognition and productivity analysis from construction site surveillance videos. Autom Constr 110:103045. https://doi.org/10.1016/j.autcon.2019.103045

Chen C, Zhu Z, Hammad A (2022) Critical review and road map of automated methods for earthmoving equipment productivity monitoring. J Comput Civ Eng 36(3):03122001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001017

Chen C, Xiao B, Zhang Y, Zhu Z (2023) Automatic vision-based calculation of excavator earthmoving productivity using zero-shot learning activity recognition. Autom Constr 146:104702. https://doi.org/10.1016/j.autcon.2022.104702

Cheng C-F, Rashidi A, Davenport MA, Anderson DV (2017) Activity analysis of construction equipment using audio signals and support vector machines. Autom Constr 81:240–253. https://doi.org/10.1016/j.autcon.2017.06.005

Chen C, Zhu Z, Hammad A, Ahmed W (2019) Vision-based excavator activity recognition and productivity analysis in construction. In: Computing in civil engineering, pp 241–248. https://doi.org/10.1061/9780784482438.031

Deshmukh DA, Mahatme PS (2016) Factors affecting performance of excavating equipment: an overview. Int J Sci Res. https://doi.org/10.21275/v5i1.nov153044

Geimer M (2020) Mobile working machines. SAE International, Warrendale, Pennsylvania. https://doi.org/10.4271/9780768094329

Golparvar-Fard M, Heydarian A, Niebles JC (2013) Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers. Adv Eng Inform 27(4):652–663. https://doi.org/10.1016/j.aei.2013.09.001

Gong J, Caldas CH (2011) An object recognition, tracking, and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations. Autom Constr 20(8):1211–1226. https://doi.org/10.1016/j.autcon.2011.05.005

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182. https://doi.org/10.5555/944919.944968

Helmus M, Fecke M (2015) Standardisierung definierter Lastzyklen und Messmethoden zur Energieverbrauchsermittlung von Baumaschinen: Schlussbericht zum Forschungsvorhaben. Bergische Universität Wuppertal, Interdisziplinäres Zentrum III; Wuppertal. https://doi.org/10.2314/GBV:86993631X

Holländer C (1998) Untersuchungen zur beurteilung und optimierung von baggerhydrauliksystemen. Phd thesis, TU Braunschweig, Braunschweig, Germany

Kassem M, Mahamedi E, Rogage K, Duffy K, Huntingdon J (2021) Measuring and benchmarking the productivity of excavators in infrastructure projects: a deep neural network approach. Autom Constr 124:103532. https://doi.org/10.1016/j.autcon.2020.103532

Kim J-Y, Cho S-B (2020) Classifying excavator operations with fusion network of multi-modal deep learning models. In: 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019), pp 25–34. Springer, Cham. https://doi.org/10.1007/978-3-030-20055-8_3

Kim J, Chi S (2019) Action recognition of earthmoving excavators based on sequential pattern analysis of visual features and operation cycles. Autom Constr 104:255–264. https://doi.org/10.1016/j.autcon.2019.03.025

Kim J, Chi S (2020) Multi-camera vision-based productivity monitoring of earthmoving operations. Autom Constr 112:103121. https://doi.org/10.1016/j.autcon.2020.103121

Kim J-Y, Cho S-B (2022) A deep neural network ensemble of multimodal signals for classifying excavator operations. Neurocomputing 470:290–299. https://doi.org/10.1016/j.neucom.2020.01.127

Kim J, Chi S, Hwang B-G (2017) Vision-based activity analysis framework considering interactive operation of construction equipment. Comput Civ Eng. https://doi.org/10.1061/9780784480830.021

Kim J, Chi S, Seo J (2018) Interaction analysis for vision-based activity identification of earthmoving excavators and dump trucks. Autom Constr 87:297–308. https://doi.org/10.1016/j.autcon.2017.12.016

Kim H, Ahn CR, Engelhaupt D, Lee S (2018) Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement. Autom Constr 87:225–234. https://doi.org/10.1016/j.autcon.2017.12.014

Kim J, Chi S, Ahn CR (2021) Hybrid kinematic-visual sensing approach for activity recognition of construction equipment. J Build Eng 44:102709. https://doi.org/10.1016/j.jobe.2021.102709

Kim I-S, Latif K, Kim J, Sharafat A, Lee D-E, Seo J (2023) Vision-based activity classification of excavators by Bidirectional LSTM. Appl Sci. https://doi.org/10.3390/app13010272

Klanfar M, Herceg V, Kuhinek D, Sekulić K (2019) Construction and testing of the measurement system for excavator productivity. Rudarsko-geološko-naftni zbornik (The Mining-Geological-Petroleum Bulletin). https://doi.org/10.17794/rgn.2019.2.6

Komatsu: specification and application handbook, 31st edn. Komatsu, Tokyo, Japan (2013). Komatsu. https://www.directminingservices.com/wp-content/uploads/2011/05/Edition31.pdf

Langroodi AK, Vahdatikhaki F, Doree A (2021) Activity recognition of construction equipment using fractional random forest. Autom Constr 122:103465. https://doi.org/10.1016/j.autcon.2020.103465

Lingard H, Wakefield R, Blismas N (2013) If you cannot measure it, you cannot improve it: measuring health and safety performance in the construction industry. In: 19th Triennial CIB World Building Congress, Queensland University of Technology, Brisbane, Queensland, Australia, pp 1–12. https://researchrepository.rmit.edu.au/esploro/outputs/9921859226601341

Mahamedi E, Rogage K, Doukari O, Kassem M (2021) Automating excavator productivity measurement using deep learning. Proc Inst Civ Eng 40(14):121–133. https://doi.org/10.1680/jsmic.21.00031

Mathur N, Aria S, Adams T, Ahn C, Lee S (2015) Automated cycle time measurement and analysis of excavator's loading operation using smart phone-embedded imu sensors. Comput Civ Eng. https://doi.org/10.1061/9780784479247.027

Molaei A, Kolu A, Lahtinen K, Geimer M (2023) Automatic estimation of excavator actual and relative cycle times in loading operations. Autom Constr 156:105080. https://doi.org/10.1016/j.autcon.2023.105080

Molaei A, Kolu A, Haaraniemi N, Geimer M (2023) Automatic estimation of excavator's actual productivity in trenching and grading operations using Building Information Modeling (BIM). Actuators. https://doi.org/10.3390/act12110423

Molaei A, Geimer M, Kolu A (2022) An approach for estimation of swing angle and digging depth during excavation operation. In: Proceedings of the 39th International Symposium on Automation and Robotics in Construction, pp 622–629. International Association for Automation and Robotics in Construction, Bogota, Columbia. https://doi.org/10.22260/ISARC2022/0087

Müller AC, Guido S (2016) Introduction to machine learning with python: a guide for data scientists, pp 394. O'Reilly Media, Inc., United States of America. https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/

MundaneSagar R, KharePranay R (2015) Comparative study of factors affecting productivity and cycle time of different excavators and their bucket size. Int J Recent Innov Trends Comput Commun 3:6518–6520. https://doi.org/10.17762/ijritcc.v3i12.5087

Rashid KM, Louis J (2019) Times-series data augmentation and deep learning for construction equipment activity recognition. Adv Eng Inform 42:100944. https://doi.org/10.1016/j.aei.2019.100944

Rashid KM, Louis J (2020) Automated activity identification for construction equipment using motion data from articulated members. Front Built Environ. https://doi.org/10.3389/fbuil.2019.00144

Rasul A, Seo J, Khajepour A (2021) Development of integrative methodologies for effective excavation progress monitoring. Sensors 21(2):364. https://doi.org/10.3390/s21020364

Roberts D, Golparvar-Fard M (2019) End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level. Autom Constr 105:102811. https://doi.org/10.1016/j.autcon.2019.04.006

Sabillon C, Rashidi A, Samanta B, Davenport MA, Anderson DV (2020) Audio-based bayesian model for productivity estimation of cyclic construction activities. J Comput Civ Eng 34(1):04019048. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000863

Sherafat B, Rashidi A, Lee Y-C, Ahn CR (2019) A hybrid kinematic-acoustic system for automated activity detection of construction equipment. Sensors 19(19):4286. https://doi.org/10.3390/s19194286

Sherafat B, Ahn CR, Akhavian R, Behzadan AH, Golparvar-Fard M, Kim H, Lee Y-C, Rashidi A, Azar ER (2020) Automated methods for activity recognition of construction workers and equipment: state-of-the-art review. J Constr Eng Manag 146(6):03120002. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001843

Sherafat B, Rashidi A, Asgari S (2022) Sound-based multiple-equipment activity recognition using convolutional neural networks. Autom Constr 135:104104. https://doi.org/10.1016/j.autcon.2021.104104

Shi Y, Xia Y, Zhang Y, Yao Z (2020) Intelligent identification for working-cycle stages of excavator based on main pump pressure. Autom Constr 109:102991. https://doi.org/10.1016/j.autcon.2019.102991

Shi Y, Xia Y, Luo L, Xiong Z, Wang C, Lin L (2021) Working stage identification of excavators based on control signals of operating handles. Autom Constr 130:103873. https://doi.org/10.1016/j.autcon.2021.103873

Slaton T, Hernandez C, Akhavian R (2020) Construction activity recognition with convolutional recurrent networks. Autom Constr 113:103138. https://doi.org/10.1016/j.autcon.2020.103138

Vukovic M, Leifeld R, Murrenhoff H (2017) Reducing fuel consumption in hydraulic excavators–a comprehensive analysis. Energies. https://doi.org/10.3390/en10050687

Wang J, Olson E (2016) AprilTag 2: efficient and robust fiducial detection. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4193–4198. https://doi.org/10.1109/IROS.2016.7759617

Zhang S, Zhang L (2022) Construction site safety monitoring and excavator activity analysis system. Construct Robot 6:151–161. https://doi.org/10.1007/s41693-022-00077-0

Zhang J, Zi L, Hou Y, Wang M, Jiang W, Deng D (2020) A deep learning-based approach to enable action recognition for construction equipment. Adv Civ Eng 2020:8812928. https://doi.org/10.1155/2020/8812928

Zhang S, Zhang L (2021) Vision-based excavator activity analysis and safety monitoring system. In: Proceedings of the 38th International Symposium on Automation and Robotics in Construction, pp 49–56. International Association for Automation and Robotics in Construction, Dubai, UAE. https://doi.org/10.22260/ISARC2021/0009

Zou J, Kim H (2007) Using hue, saturation, and value color space for hydraulic excavator idle time analysis. J Comput Civ Eng 21(4):238–246. https://doi.org/10.1061/(ASCE)0887-3801(2007)21:4(238)