IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 1, January, 2026, pp. 133-144
Journal homepage: http://iieta.org/journals/jesa

Drone Control Using Upper Body Gesture Based on Pose Detection and Random Forest for Monitoring in Smart Agriculture

Muhammad Fuad[1*], Sri Wahyuni[1], Luhur Bayuaji[2], Yuli Panca Asmara[3], Aeri Rachmad[4]

[1] Department of Mechatronics Engineering, Faculty of Engineering, University of Trunodjoyo Madura, Bangkalan 69162, Indonesia
[2] Department of Computer Science, Faculty of Data Science and Information Technology, INTI International University, Nilai 71800, Malaysia
[3] Department of Mechanical Engineering, Faculty of Engineering and Quantity Surveying, INTI International University, Nilai 71800, Malaysia
[4] Department of Information Systems, Faculty of Engineering, University of Trunodjoyo Madura, Bangkalan 69162, Indonesia

Corresponding Author Email: fuad@trunojoyo.ac.id

https://doi.org/10.18280/jesa.590113

Received: 27 November 2025
Revised: 19 January 2026
Accepted: 27 January 2026
Available online: 31 January 2026

Keywords:
gesture-based robot control, Madura corn, random forest, smart agriculture, unmanned aerial vehicle, zero hunger

ABSTRACT

Smart agriculture with intensive monitoring of corn plant diseases is urgently needed to increase crop yields. This study supports Sustainable Development Goals 2, Zero Hunger, by providing an innovative and alternative solution for farmers in identifying corn leaf diseases with a drone that is controlled using human body gestures. This study aimed to develop a new drone monitoring system for detecting corn leaf diseases. This study proposes a gesture recognition system based on pose detection and random forest (RF) algorithm. Pose detection is performed using a machine learning model that identifies key points on the user's upper body. Then, the obtained features are classified using RF. System testing was performed in simulations and experiments in the Madura corn field with several gestures. The average accuracy of gesture recognition without pre-processing is 89.1%. By employing the augmented dataset and utilizing the hip-center normalization, RF demonstrates high classification accuracy 99.70%, computational cost 18.35 ms, real-time capability 26.68 ms, end-to-end latency 56.33 ms, outperforming SVM and 1D-CNN. This study successfully provides an effective gesture-based drone control and has potential applications in the fields of human-robot interaction, gesture-based robot control, and smart agriculture.

1. INTRODUCTION

Sustainable development goals (SDGs) have become a global concern. As the second target of the SDGs [1], zero hunger needs to be addressed by improving performance in sustainable agriculture [2, 3]. Agricultural activities with priority services are rice, corn, soybean. Corn is one of the priority crops with high productivity and various advantages, such as for feed, food, energy, and industrial raw materials [4, 5]. Agricultural sustainability can be achieved by synergizing agriculture and technology through smart agriculture. Smart agriculture involves the use of sensors and actuators, information and communication technologies (ICT), internet of things (IoT), and artificial intelligence robots to support agriculture [6, 7] in soil preparation, seed planting, irrigation [8], crop cultivation, monitoring, harvesting, and logistics [9]. Monitoring is the third ranking agricultural robot function with a value of 13.8% below weeding and tool-carrier [10]. The urgency of this study is to develop the ability to control the drone to monitor the corn plant disease to support smart agriculture for increasing crop yields.

Corn growth was first monitored using radar by Ulaby and Bush in 1976 [11]. The classification of corn leaf disease was investigated by using MobileNetV2 Convolutional Neural Network (CNN) architecture based on images taken manually [12]. A wheeled mobile robot was remotely controlled by using a joystick based on data from the Global Positioning System (GPS) [13]. Some previous researches [14, 15] studied drone's control and guidance that can be used to support smart agriculture. Both papers [14, 15] focus on hardware-centric enhancement and software-driven autonomy, respectively, for drone control. However, neither addresses control paradigms involving direct, intuitive human-machine interaction like gesture-based control. Remote control was used to control an agricultural robot to perform several agricultural duties [16]. An unmanned aerial vehicle (UAV) in the form of quadcopter, controlled by a device with a touch-screen and a graphical user interface (GUI). To increase the transmission range, this UAV was completed with a repeater. Remotely controlled UAVs that follow the desired path and avoid obstacles by exploiting aerial and satellite data were used to modernize natural forest management by supporting monitoring activities [17]. Crop monitoring for plant disease detection was carried out by drone [18], including spraying and assessing fruit color and ripeness.

A combination of global and local information was applied in a drone to detect and count maize tassel in complex agricultural [19]. Remote control can be alternatively substituted by exploiting human body gestures to control the robot intuitively. The study on gesture-based control (GBC) has an impact on drone monitoring in smart agriculture and it also increases the need for more intuitive and efficient human-robot interaction (HRI) systems. A gesture-based control for communicating with a robot using gesture recognition trained by a hand dataset was studied by Peral et al. [20]. The robot did not use joystick or glove as external and wearable devices for remote control. The gesture recognition proposed by Peral et al. [20] was based on deep learning on key frames of pose-detected hand images. The training and testing sets were divided into 74% and 26%, respectively. It resulted in 3117 training and 1101 testing gesture instances. For validation purposes, 10% of all images in the dataset were selected, resulting in 422 validation images. The IVO robot was used in the experiments. It was equipped with a depth sensor set as the onboard camera. However, the gestures were not implemented as commands to make the IVO robot move. The focus was on validating recognition and interaction capability, not gesture-based motion control. In contrast to previous research [20], this research focuses on developing gesture recognition to generate drone movement commands. Body gestures are a type of nonverbal communication that has more variety than hand gestures in expressing a message. It has the potential to deliver information from humans to robots in the form of movement control commands to improve interaction. However, accurately and actually interpreting information from gestures requires a combination of appropriate technological approaches. To address this challenge, this study aims to develop gesture-based control of drone monitoring using human body gestures.

This study proposes a gesture recognition system for human upper body gestures by combining pose detection and random forest (RF) algorithm. The key points of the human upper body are identified using pose detection, which is carried out using a machine learning model. The extracted features are then classified using RF.

This study makes three contributions. First, we developed an RGB image-based human upper body gesture recognition system to control a drone for a monitoring task by tracking 33 human body key points by exploiting the MediaPipe pose detection and gesture classification by utilizing RF algorithm. This approach has the advantage that a human operator controls the drone's movement intuitively without additional or wearable hardware such as a joystick or dedicated gloves with sensors. Second, this study generated a dataset with 6,000 images representing 10 gestures for controlling the drone movement such as up, down, forward, backward, left, right, turn left, turn right, hover, and stop. The proposed gesture recognition method was trained and tested on the dataset by randomly splitting it into 80% and 20%. Third, the proposed method was implemented in real-time gesture recognition for controlling drone movement in simulations and real-world experiments. This proposed gesture-based control system was also tested for controlling the movement of drones to support smart agriculture in a real environment.

2. RELATED WORKS

Gesture is a term used to represent the visible action when it is used to deliver information between entities involved in the interaction [21]. Gesture involves a range of communicative body motions, usually by exploiting the hands and arms. There are several kinds of gestures, such as gesticulation, speech-framed gestures, quotable gestures, pantomime, and signs. Among the various types of gestures, quotable gestures, emblems, or symbolic gestures can convey the meaning of a message without requiring the aid of speech. Our study investigates quotable gestures that can be used to communicate with a drone to control its movement.

The development of the ability to recognize the action used for human–robot collaboration was investigated by Terreran et al. [22]. The skeleton with full skeleton of body and hand joint information was used for action classifiers for recognizing human body activities and hand gestures that can be used in collaboration between humans and robots. Skeleton data were obtained using OpenPose to provide 3D joint coordinates. The skeleton was provided by capturing the user body pose that involved 15 joints. These joints were extracted from RGB-D frames using 3D pose estimation. The Shift-Graph Convolutional Network (Shift-GCN) was exploited for action recognition. Our study differed from previous research [22] in that we used MediaPipe to extract 33 body landmarks for gesture recognition using the RF algorithm. This paper use MediaPipe with BlazePose deep learning model for pose estimation because it has best performance compare to OpenPose in real-time and lightweight pose estimation. The performance includes fast frames per second (fps) with very low latency. BlazePose is better than OpenPose in Upper-body gesture control (shoulder, elbow, torso).

HRI research that attempts to recognize emotions from face expressions completed by body actions was explored by Ilyas et al. [23]. It applied upper body images to support facial images to learn what emotions were expressed by the system user. The relationship between upper body action and facial expression was mapped to identify the emotion released. A deep CNN was used to extract facial and bodily features from a dataset of RGB images. It was combined with Long-Short Term Memory (LSTM) to learn from sequential information. This combination was trained on the Bimodal Face and Body Gesture Database (FABO), which demonstrated numerous emotions and related body actions to recognize emotion. Inertial measurement units (IMU) and electromyography (EMG) were employed as wearable sensors to differentiate several personalized gestures [24] for studying gestural input for people with upper-body motor impairments. In contrast with this previous study, our approach uses real-time gesture recognition to determine the meaning of gestures used to move the drone according to the intention of user. A method based on the geometric principle to direct the motion of a robot arm manipulator with 5 degrees of freedom (DoF), SCORBOT ER 9Pro, by exploiting single arm gesture of user [25]. Skeleton images from the Kinect sensor were used to estimate the length and angle of the bone's joints. These data were applied to the gesture recognition system to interpret the gesture as control command. This command provided velocity joints for the manipulator robot to be executed using forward kinematics. In contrast to previous research [25], which used the Kinect sensor to obtain skeletal images, our research uses BlazePose-MediaPipe [26] to extract skeletal data from common webcam RGB images without using specialized devices.

Based on the aforementioned research gaps, this study was designed to address several limitations identified in previous studies. In this study, we developed a control system for

unmanned aerial vehicle (UAV) robot in the form of a drone using upper body gestures detected with the pose detection of MediaPipe and classified using the RF algorithm.

This study proposes to use RF for classification task with rational that in some time-series scenarios, RF matched or exceeded LSTM performance when feature engineering and preprocessing were applied, indicating RF's continued practical utility for spatiotemporal classification [27].

Gestures are inherently spatiotemporal phenomena, characterized by coordinated movements of body joints over time. However, in many practical gesture recognition systems, the temporal evolution of motion is transformed into a fixed-length static feature representation, such as a 132-dimensional vector composed of x, y, z, and visibility of each extracted keypoint. Under this representation, the classification problem shifts from sequence modeling to multivariate pattern recognition, where RF becomes a viable and often advantageous classifier. A key advantage of RF in this context is its ability to split naturally select discriminative features, making RF robust when the 132-dimensional vector contains a mixture of informative and redundant spatiotemporal descriptors. Unlike sequence models such as LSTM or 3D CNNs, RF does not require explicit temporal alignment or large amounts of labeled data to learn temporal dependencies. When the temporal dynamics of gestures are already embedded into static features, RF can effectively separate gesture classes with lower computational cost and faster training. This makes it well suited for real-time or embedded systems, where gesture recognition must be executed with minimal latency. Despite these advantages, the primary limitation of RF in gesture recognition lies in its inability to model temporal dependencies explicitly. RF operates on static input vectors and therefore cannot inherently capture sequential patterns such as motion order, rhythm, or long-range temporal dependencies. As a result, critical temporal cues, such as the directionality of movement or phase transitions within a gesture, may be lost if they are not adequately encoded in the feature vector. Furthermore, RF assumes that the provided features sufficiently summarize the spatiotemporal structure of gestures. If the 132-dimensional vector is derived from overly aggressive temporal aggregation, subtle yet discriminative motion characteristics may be smoothed out, leading to confusion between gestures with similar spatial configurations but different temporal execution. In contrast, models such as LSTM, 3D CNNs, or Graph Neural Networks are better suited for learning these fine-grained temporal patterns directly from sequential data.

The proposed combination involves BlazePose-MediaPipe and RF algorithm is advantageous for agricultural drone control compares with deep learning-based approaches in that RF runs efficiently on standard CPUs and does not require expensive GPUs for training. It is significantly faster to train and tune than deep neural networks. RF is often easier to deploy in low-latency production environments or on edge devices. This system was validated in simulation and implemented in real time on a physical quadcopter robot [28] and tested under various lighting and background conditions to comprehensively evaluate its performance and reliability.

3. GESTURE-BASED DRONE CONTROL USING POSE DETECTION AND RANDOM FOREST ALGORITHM

This section describes the steps in developing a new system for gesture recognition based on the proposed method, which comprises of feature extraction using pose detection and gesture classification using RF algorithm.

3.1 Environment setting and dataset development

The initial stages in developing a gesture recognition system for pose detection-based drone control include setting the environment and compiling the dataset. Setting the environment prepares the working environment for dataset collection, as shown in Figure 1.

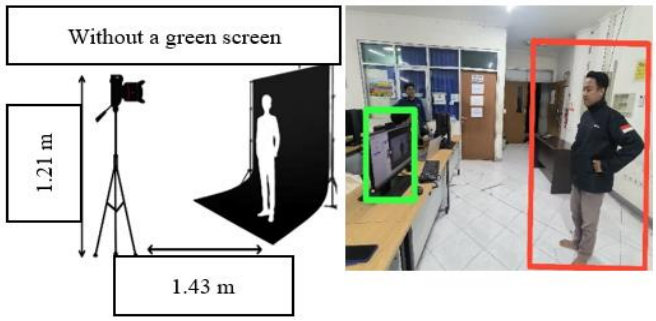

Figure 1. Environment settings for the preparation of gesture data capture

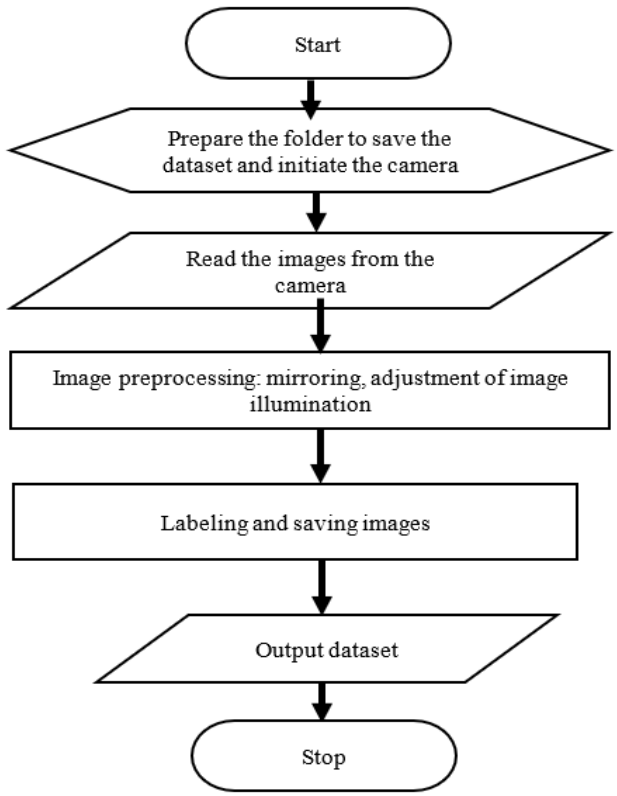

Figure 2. Flowchart of dataset development

Based on the environmental settings, a white wall from the automation and robotics systems laboratory without a green screen was used as the background. The camera and the group members performing the gestures are separated by approximately 1.43 m. The internal webcam camera is located 1.21 m above the floor. The positions of users are marked with red boxes, and the camera is marked with a green box. The dataset development process is shown in Figure 2. The image resolution of the dataset is 640 × 480. The number of datasets

is 600 poses for each gesture. There are 10 folders representing 10 gestures to save this dataset. The dataset contains 6,000 images representing 10 gestures.

These images were recorded from 5 participants demonstrating 10 gestures in front of the camera. The proposed pose detection system uses MediaPipe. MediaPipe is a pose landmark that marks objects with nodes. MediaPipe tracks 33 body landmark features, representing the approximate location of body parts within the human skeleton.

The MediaPipe nodes represent 33 positional landmark features on the human body, with x as the horizontal coordinate, y as the vertical coordinate, z as the relative depth, and visibility. There are 33 nodes (vertices) connected by some edges to form an undirected graph of the full human pose as the result of pose detection in this study exploited MediaPipe BlazePose [26], as shown in Figure 3.

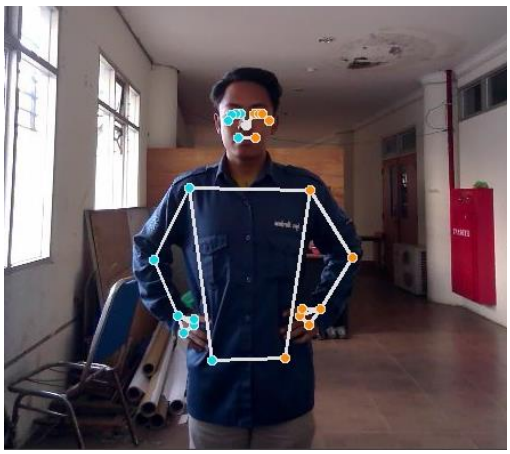

Figure 3. Nodes or landmark features in the human upper body

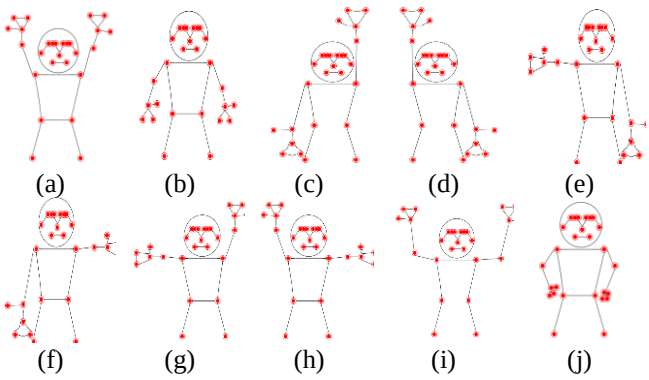

Figure 4. Gesture design to control drone movement represents ten motions consisting of (a) up, (b) down, (c) forward, (d) backward, (e) left, (f) right, (g) turn left, (h) turn right, (i) hover, and (j) stop

The pose detection-based gesture recognition system for controlling drone uses the upper body command gesture with 25 landmark nodes. From the head, hands, and body to the human waist. The planned gesture material is represented as a human skeleton used in the upper body gesture recognition system, as shown in Figure 4. This process produces a set of gesture data called a gesture dataset in the dataset compilation stage.

The gesture data are stored in several folders with label names according to the gesture. Each folder represents one gesture category: up, down, forward, backward, left, right, turn left, turn right, hover, and stop. The storage folder is divided into each gesture category for the next stage of data processing. Figure 5 shows samples of ten gestures of the dataset with 6,000 images. Each of the 10 gestures have 600 images.

Figure 5. Dataset with 6,000 images representing ten gestures

3.2 Feature-extraction using pose detection

The dataset was extracted using pose detection, and the results were saved into a comma separated value (csv) file. This step uses MediaPipe and the csv library. MediaPipe is initialized in the still image mode with a minimum detection confidence level of 30%. Pose detection detects 33 landmark features of the human pose. Each body feature comprises four data: x, y, z, and visibility. The feature extraction process is shown in Figure 6. This extraction results in 132 values (33 points × 4 values per point) per upper body image, as shown in Figure 7.

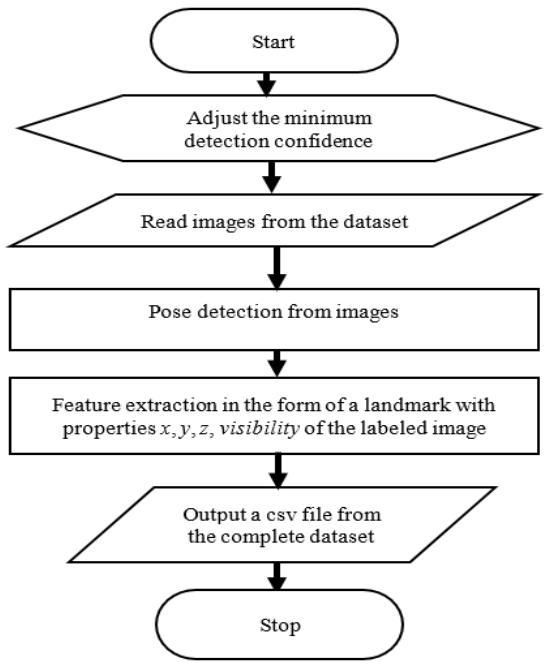

Figure 6. Flowchart of feature extraction using pose detection

3.3 Training and testing based on random forest algorithm

The RF architecture involves a combination of multiple decision trees [27] built using a training dataset of the pose detection dataset organized into bootstrap datasets. The RF architecture is depicted in Figure 8. Each decision tree is trained with one bootstrap dataset, and the classification results from each tree are voted. The majority vote from all

decision trees determines final classification result, which becomes the output of this algorithm.

Figure 7. Pose detection extracts nodes from the user's upper body image

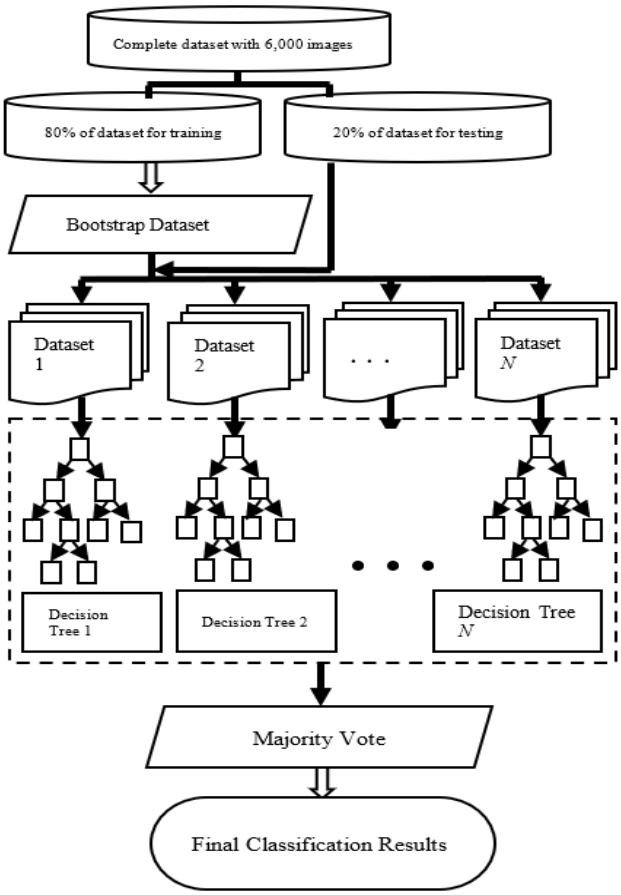

Figure 8. Random forest (RF) architecture

Figure 9 depicts the decision trees in the RF. The dataset is the root node as the decision tree's starting point of the decision tree. The detected dataset will be evaluated, for example, "is the user's hand stretched to the side or in a ready standing position?".

The results of the branching will be evaluated in the next stage, for example, "is the user doing a straight hand pose to the left?". The final result or decision is determined by the last node. For example, the result of the gesture is "turning left" as the command output to control the robot's movement. The training flowchart is depicted in Figure 10.

The CSV files containing body pose data were combined into one large CSV file for RF training. CSV of all images in the dataset is read and separated into feature and label data. The feature and label data are randomly partitioned into training and testing data with ratios 80% and 20%, respectively.

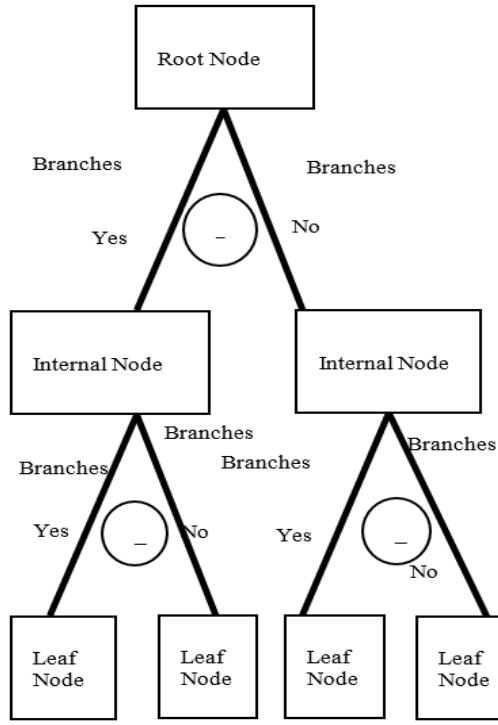

Figure 9. Decision tree in random forest architecture

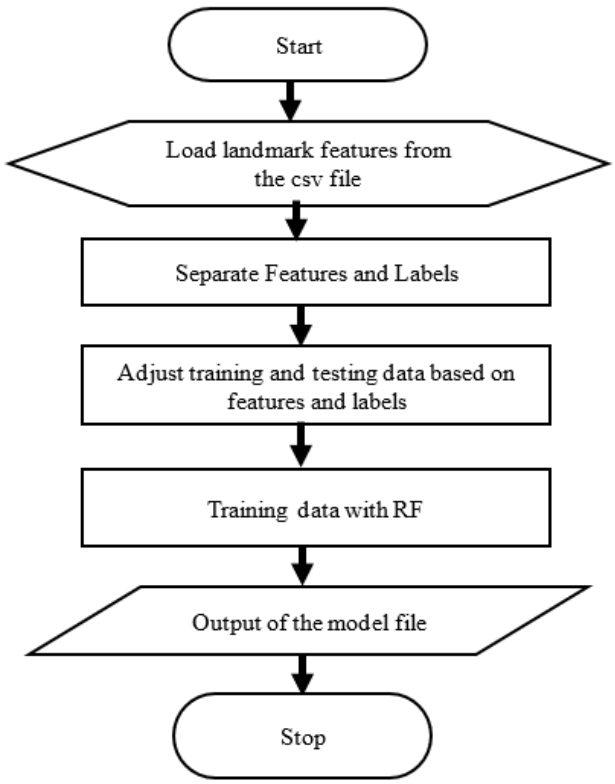

Figure 10. Flowchart of training data using random forest (RF)

An RF is prepared by defining the number of decision trees in the forest. In this study, the default value is 100. The bootstrap is created by maintaining the randomness of the samples used when building trees using the value of 42. The RF algorithm is trained using feature data paired with target labels to produce a training model. The model file is used in the testing process, as shown in Figure 11.

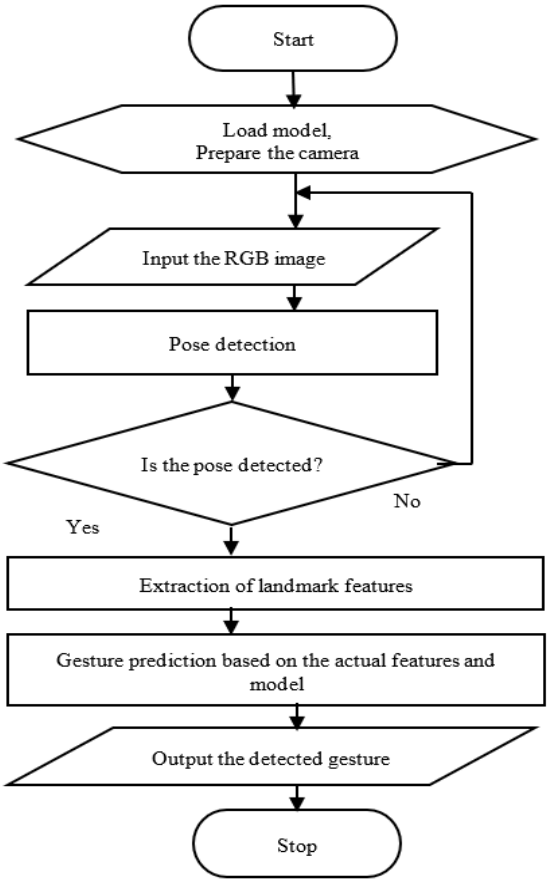

Figure 11. Flowchart of test for the gesture-based control (GBC) system

The testing process involves loading the model and preparing the camera to capture the user's actual RGB image. RGB images are used in the pose detection process. The extracted landmark features are compared with the training model in the gesture prediction process to obtain the detected gesture output. The visualization of landmark pose detection was successfully performed using the MediaPipe library, which identified 33 body points and displayed the gesture classification results directly on the screen, as shown in Figure 12. The system detected the "hover" gesture by labeling it "Pose: hover" and mapped the body position with a skeleton line based on the detected landmarks.

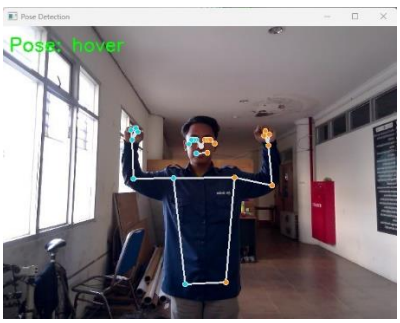

Figure 12. Gesture recognition testing using pose detection and random forest (RF) algorithm

3.4 Drone control system

Figure 13 shows the gesture-based drone control system. The RF model FL_{xy}^{m} resulted from training process is employed as the ground truth that acts as a reference for robot movement. The detected landmark features F_{xy}^{a} from the user who demonstrate their gestures in front of the camera are used to predict the gesture command G_c based on the model FL_{xy}^{m}.

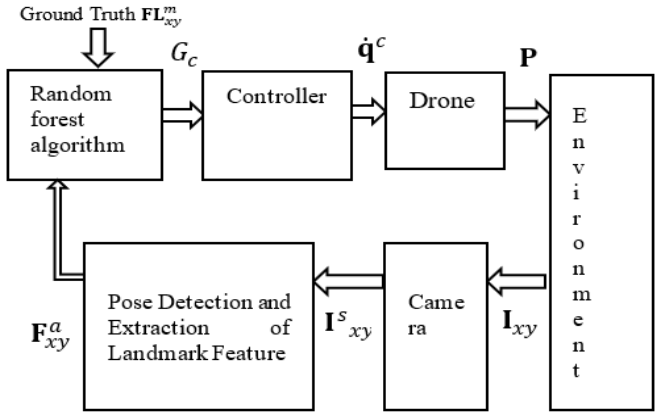

Figure 13. Block diagram of the gesture-based drone control system

The drone motion is described as follows [28]:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} k_t & k_t & k_t & k_t \\ 0 & -lk_t & 0 & lk_t \\ k_t & 0 & k_t & 0 \\ k_d & -k_d & k_d & -k_d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m}\left(\begin{bmatrix} (U_1)(c\psi s\theta c\phi + s\psi s\phi) \\ (U_1)(s\psi s\theta c\phi - c\psi s\phi) \\ (U_1)(c\phi c\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \right) \tag{2}$$

where, U_1 describes the total thrust of all motors, U_2 expresses roll control input, U_3 represents pitch control input, U_4 presents yaw control input, k_t symbolizes thrust coefficient, k_d symbolized drag torque, l is length of arm, and ω represents angular velocity of each motor, m describes the mass, g expresses the gravity acceleration. While s and c are abbreviation of sine and cosine of roll ϕ, pitch θ, and yaw ψ angle, respectively. The output of the equation of motion is $[\ddot{x} \quad \ddot{y} \quad \ddot{z}]^T$ the drone acceleration along the X, Y, and Z axes.

Based on (1) and (2), the controller generates velocity commands $\dot{\mathrm{q}}^c$ based on the detected gesture command G_c. The drone moves by using $\dot{\mathrm{q}}^c$ to reach new position P in environment. The following are the mapping between gesture commands G_c to drone control inputs as position commands.

$$U_1 = k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{3}$$

$$\ddot{z} = \frac{1}{m}((U_1)(c\phi c\theta) - mg) \tag{4}$$

$$\ddot{x} = \frac{1}{m}\big((U_1)(c\psi s\theta c\phi + s\psi s\phi)\big) \tag{5}$$

$$\ddot{y} = \frac{1}{m}((U_1)(s\psi s\theta c\phi - c\psi s\phi)) \tag{6}$$

$$U_4 = k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{7}$$

$$\ddot{\psi} = \frac{U_4}{I_Z} \quad (8)$$

The gesture commands of upward, downward, and hover are related with (3) and (4). To move the quadcopter to upward direction, the angular velocity of all four motors should be increased equally to overcome gravity. To move the quadcopter to downward direction, the angular velocity of all four motors should be decreased equally. To hover, all four motors should spin at an equal speed such that total upward thrust equals the force gravity.

The gesture commands of forward and backward are mapped to (3) and (5). The forward motion is achieved by speed up rear motors and slow down front motors. On the contrary, the backward motion is realized by speed up front motors and slow down rear motors.

The gesture commands of left and right motion are applied by utilizing (3) and (6). The left motion is reached by speed up right-side motors and slow down left-side motors. Whereas, the right motion is attained by speed up left-side motors and slow down right-side motors.

The gesture commands of rotate left and rotate right motion are implemented by using (7) and (8). Rotate left is achieved by increasing speed of clock wise (CW) motors (motor 1 and 3) and decreasing speed of counter clock wise (CCW) motors (motor 2 and 4). On the other hand, rotate right is reached by increasing speed of CCW motors and decreasing speed of CW motors. Table 1 shows the specifications of some of the software used for the proposed system implementation.

Table 1. Specification of the software used in this study

Library	Detail
python	python == 3.9
cv2	opencv-python == 4.11.0.86
mediapipe	mediapipe == 0.10.21
numpy	numpy == 1.26.4
joblib	joblib == 1.4.2
matplotlib	matplotlib == 3.9.4
scikit-learn	scikit-learn == 1.6.1
pandas	pandas == 2.2.3

4. SIMULATION AND EXPERIMENTAL RESULTS

4.1 Simulation results

The gesture-based drone control was simulated in a 3D environment with dimension 8 × 8 × 5 m^3 by exploiting multithreading to simultaneously run the frame of gesture recognition and the robot's movement animation. In this simulation, the simulated drone in the animation window moved in the direction that corresponded to the detected gesture, as shown in Figure 14. The validation flowchart using computer simulation is depicted in Figure 15.

4.2 Experimental results

The validation process using real world experiment in the corn field is described in Figure 16. This process is crucial to ensure that the drone can translate each user gesture into a corresponding action. Figure 17 shows the validation of the proposed gesture-based drone control in a real-world experiment. The landmark features of the demonstrated gestures performed by the five participants were detected in real-time by MediaPipe pose detection based on the poses performed by the user. The proposed gesture recognition system identified gestures using a pretrained RF model. Once a gesture was recognized, the system sent commands to the drone using wireless communication to drive the motors according to the specified action.

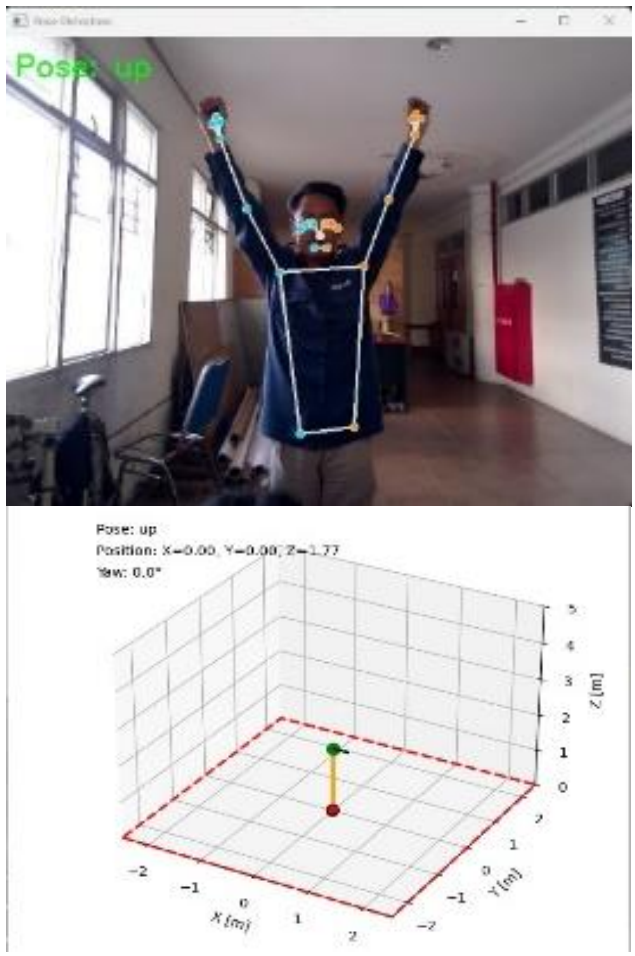

Figure 14. The testing of gesture-based control using simulated drone in 3D environment

Table 2. Experimental results of the proposed system

Gestures	Accuracy (%)					
	Par#1	Par#2	Par#3	Par#4	Par#5	Average of Each Gesture
Up	94.6	94.6	91.7	91.9	91.7	92.9
Down	75.0	90.0	82.5	71.8	87.5	81.4
Forward	81.1	75.9	79.6	80.8	83.6	80.2
Backward	91.2	91.2	94.1	88.6	94.1	91.8
Left	93.3	93.3	93.3	93.3	93.3	93.3
Right	93.3	90.0	90.0	86.7	93.3	90.7

Turn Left	86.4	88.3	81.7	88.1	84.5	85.8
Turn Right	93.1	93.1	93.1	90.0	93.1	92.5
Hover	90.0	93.8	89.8	90.0	92.0	91.1
Stop	71.4	97.1	97.1	97.1	94.3	91.4
Average of each participant	87.0	90.7	89.3	87.8	90.7	89.1

Each gesture was tested multiple times by five participants to ensure control accuracy, and the system demonstrated consistent responses to the labeled gestures, as described in Table 2. Based on the experimental results, the proposed gesture-based drone control using pose detection and RF has an average accuracy of 89.1%.

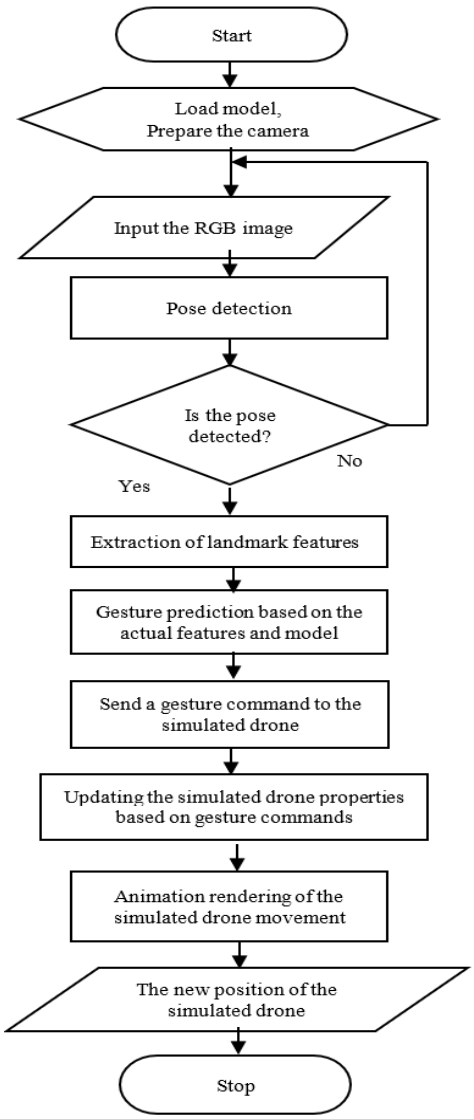

Figure 15. Testing flow using simulation

Table 3. Augmentation's parameter

Augmentation	Parameter
Gausian Noise (Node level)	0.003
Small Scalling	0.98-1.02
Small Rotation (roll and yaw)	±10 degree
Number of augmentations	3 per image

The low accuracy of this experiment prompted us to make improvements by repeating the gesture data recording process, training, and testing. With 5 participants, 100 images were recorded. Three augmentation process was performed on these 500 data, resulting in 4 variants. The augmentation's parameters are described in Table 3. This resulted in 2,000 data for each gesture. The total data for the 10 gesture classes were 20,000 images.

Each image was extracted to resulting keypoints of upper body. From total 33 keypoints, this study that proposed to use the upper body only used 25 keypoints from index 0 to 24. Each keypoint had 3 data involves x, y, and z. A pre-processing step was proposed in this study by implementing normalization based on *hip_center* (9) to make skeletal data invariant to a person's position and body size. The new joint position is obtained based on the hip_center data by utilizing (10).

$$hip\_center = (pts[23] + pts[24])/2.0 \quad (9)$$

$$joint\_new = joint\_original - hip\_center \quad (10)$$

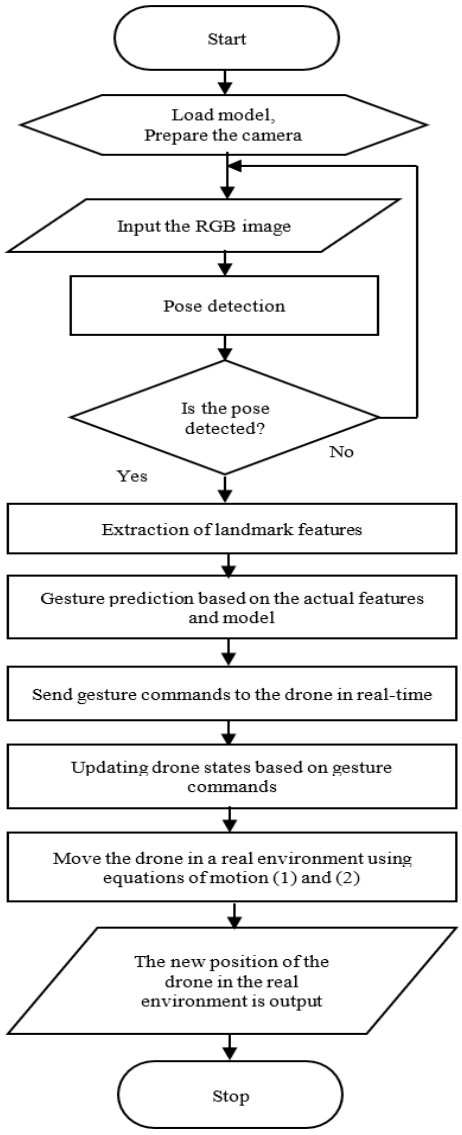

Figure 16. Testing flow using experiment

Figure 17. Experiments of the proposed gesture-based drone control in a real corn field

Table 4. Setting parameter of 1D-CNN

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 79, 64)	256
batch_normalization (BatchNormalization)	(None, 79, 64)	256
max_pooling1d (MaxPooling1D)	(None, 39, 64)	0
conv1d_1 (Conv1D)	(None, 37, 128)	24,704
batch_normalization_1 (BatchNormalization)	(None, 37, 128)	512
max_pooling1d_1 (MaxPooling1D)	(None, 18, 128)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 128)	295,040
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

To demonstrate its effectiveness, this study compared the proposed method (Random Forest) with Support Vector Machine (SVM) and 1 dimentional Convolutional Neural Network (1D-CNN). Table 4 shows the 1D-CNN setting parameters. The confusion matrix and training curve of 1D-CNN are shown in Figure 18.

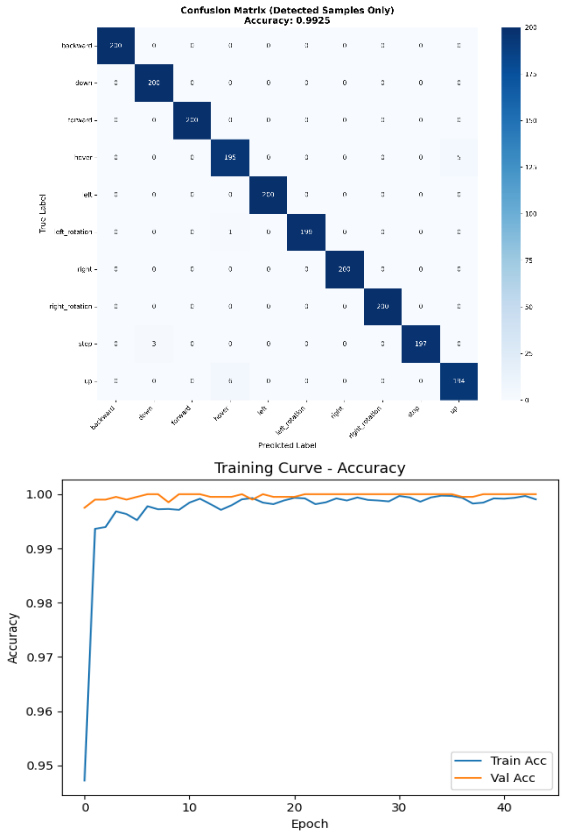

Figure 18. Confusion matrix and training curve of 1D-CNN

Table 5 presents the SVM setting parameters. The confusion matrix and training curve of SVM are shown in Figure 19.

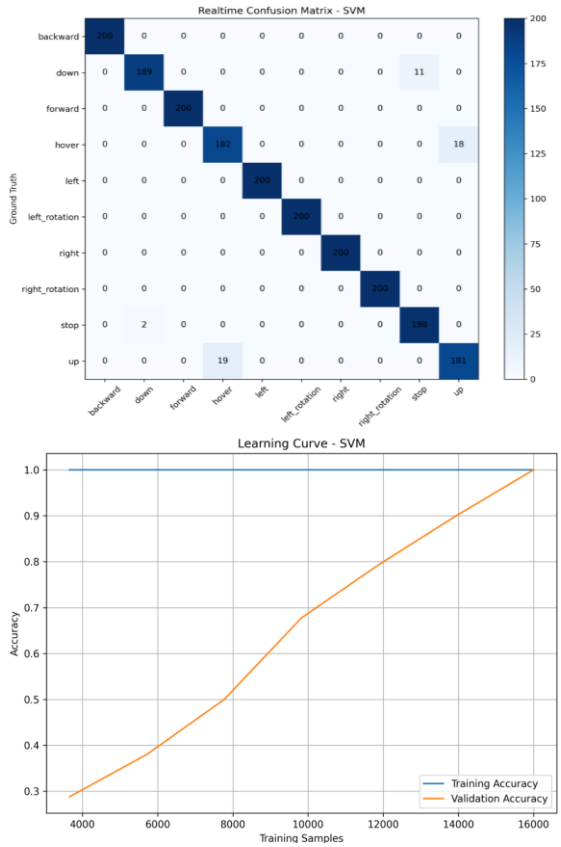

Figure 19. Confusion matrix and training curve of SVM

Table 5. Setting parameter of SVM

Category	Components	Parameters
Preprocessing	StandardScaler	with_mean = True, with_std = True
Model	Pipeline	(scaler + SVM)
	SVC (Support Vector Classifier)	-
	kernel	"rbf"
	C	10.0
	gamma	"scale"
	probability	True
	decision_function_shape	"ovr"
	random_state	42
	class_weight	Default (None)
	cache_size	Default (200 MB)
Data Splitting	Train / Test	Separated (dataset_train.csv and dataset_test.csv)
Cross-Validation	Learning Curve	StratifiedKFold, n_splits = min(5, smallest)
Evaluation	Metrics	Accuracy, Classification Report, Confusion Matrix
	Learning Curve	8 points of training (0.1–1.0)

Table 6 presents the RF setting parameters. The confusion matrix and training curve of RF are shown in Figure 20.

Table 6. Setting parameter of random forest (RF)

Category	Components	Parameters
Preprocessing	Scaling	None
Model	RandomForestClassifier	Base model with random_state=42, n_jobs=-1
Hyperparameter Tuning	GridSearchCV	cv=5 (StratifiedKFold), scoring="accuracy", n_jobs=-1
	param_grid → n_estimators	[200, 300, 500]
	param_grid → max_depth	[None, 20, 40]
	param_grid → min_samples_split	[2, 5]
	param_grid → min_samples_leaf	[1, 2]
	param_grid → criterion	["gini"]
	random_state (model)	42
	n_jobs	-1
Data Splitting	train_test_split	test_size=0.2, stratify=True, random_state=42
Cross-Validation	StratifiedKFold (for GridSearch)	n_splits=5, shuffle=True, random_state=42
Evaluation	Metrics	Accuracy, Classification Report, Confusion Matrix
	Learning Curve	train_sizes = 0.1 – 1.0 (8 points)

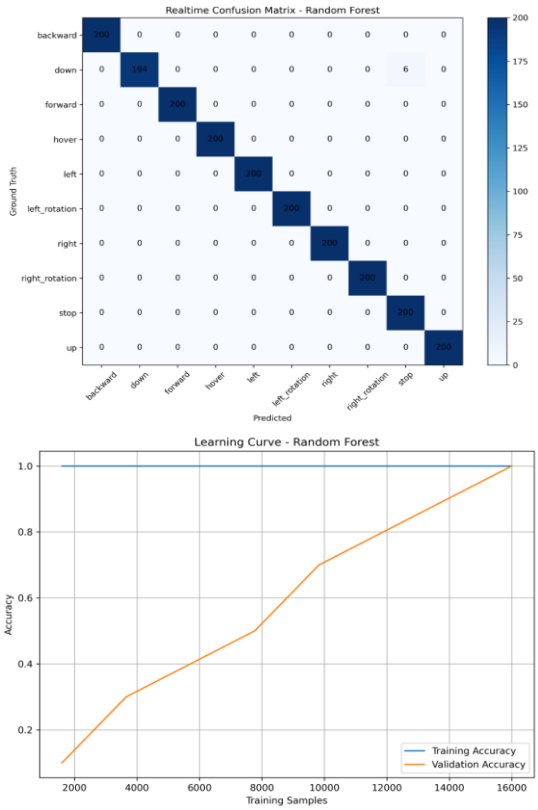

Figure 20. Confusion matrix and training curve of random forest (RF)

After going through the training process, real-time testing was carried out on 1D-CNN, SVM, and RF by the help of 5 other participants that different from the participants involved in dataset development. The results of comparison between these methods are presented in Table 7. Table 7 compares the performance of three classification algorithms, SVM, 1D-CNN, and the proposed RF, in terms of classification accuracy, preprocessing time, frame rate, and inference time. From the accuracy perspective, the proposed RF method achieves the highest classification accuracy of 99.70%, outperforming 1D-CNN (99.25%) and SVM (97.50%). Although the improvement over 1D-CNN appears marginal (0.45%), it indicates that RF can effectively exploit the discriminative power of the engineered feature set without requiring deep hierarchical feature learning.

The significantly higher accuracy compared to SVM suggests that ensemble-based nonlinear decision boundaries better capture the complex relationships among the extracted gesture features. In terms of computational efficiency, SVM exhibits the lowest inference time (0.43 ms) and the highest frame rate (31.21 FPS), indicating very fast decision making once features are available. However, this comes at the cost of notably lower classification accuracy. In contrast, the 1D-CNN shows the highest preprocessing time (62.55 ms) and the lowest frame rate (11.16 FPS), reflecting the overhead of convolutional operations and deep feature extraction, which limits its suitability for real-time systems on resource-constrained platforms.

The proposed RF method demonstrates a balanced performance profile. Its preprocessing time (18.35 ms) is substantially lower than that of 1D-CNN and only slightly higher than SVM, while its frame rate (22.21 FPS) remains within real-time operational requirements. Although its inference time (26.68 ms) is comparable to that of the 1D-

CNN, RF achieves this with significantly lower preprocessing overhead and without requiring GPU acceleration or extensive model training.

The proposed method (RF) results the average of end-to-end latency from image capture to drone command generation of 56.33 ms. It is faster than produced by SVM and 1D-CNN of 61.24 ms and 56.54 ms respectively.

Table 7. The results of comparison experiments

Algorithm	Accuracy (%)	Pre-processing (ms)	Frame (FPS)	Inference Time (ms)	Avg End-to-End Latency (ms)
SVM	97.50	16.88	31.21	0.43	61.24
1D-CNN	99.25	62.55	11.16	27.07	56.54
RF (The Proposed Method)	99.70	18.35	22.21	26.68	56.33

5. CONCLUSION

Drone control using upper body gestures based on pose landmark detection and the RF algorithm was successfully implemented. The program could recognize upper body gestures in real-time through the camera and convert them into control commands sent to the robot. MediaPipe pose detection was effective in detecting and extracting body landmark points, which were then processed into numerical features for the classification.

The average accuracy of gesture recognition on ten gesture commands performed without any pre-processing is 89.1%. By employing the augmented dataset and utilizing the hip-center normalization, the proposed RF-based method demonstrates a strong balance between high classification accuracy 99.70%, moderate computational cost 18.35 ms, real-time capability 26.68 ms, and end-to-end latency 56.33, outperforming both SVM and 1D-CNN in terms of overall suitability for practical gesture recognition systems.

These results validate the effectiveness of the proposed approach for real-time applications, particularly in scenarios where computational resources are limited and reliable gesture recognition is critical. The experiment results show that the robot can move appropriately according to gesture-based commands such as up, down, forward, backward, left, right, turn left, turn right, hover, and stop. Based on the simulation and experimental results, it was concluded that the proposed system can be used as an alternative interactive robot control method. In future research, we will investigate gesture recognition to control multiple drones to support smart agriculture.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude and acknowledge the Institute for Research and Community Service (Lembaga Penelitian dan Pengabdian kepada Masyarakat, LPPM) at the University of Trunojoyo Madura (UTM) and INTI International University, Malaysia for supporting the international collaboration research (Penelitian Kolaborasi Internasional) in 2025 (Grant 346/UN46.4.1/PT.01.03/RISMAN/2025).

REFERENCES

[1] Iqbal, B.A. (2025). Sustainable Development Goals: Performance and Challanges, 1st ed. Network for Theoretical and Empirical Research in Multidisciplinary Studies Pvt. Ltd.

[2] Herdiansyah, H., Majesty, K.I. (2024). Conflict mitigation strategies for sustainable agriculture in palm oil expansion. International Journal of Sustainable Development and Planning, 19(5): 1893-1902. https://doi.org/10.18280/ijsdp.190527

[3] Espolov, T., Espolov, A., Satanbekov, N., Tireuov, K., Mukash, J., Suleimenov, Z. (2023). Economic trend in developing sustainable agriculture and organic farming. International Journal of Sustainable Development and Planning, 18(6): 1885-1891. https://doi.org/10.18280/ijsdp.180624

[4] Asfaw, D.M., Asnakew, Y.W., Sendkie, F.B., Abdulkadr, A.A., Mekonnen, B.A., Tiruneh, H.D., Ebad, A.M. (2024). Analysis of constraints and opportunities in maize production and marketing in Ethiopia. Heliyon, 10(20): e39606. https://doi.org/10.1016/j.heliyon.2024.e39606

[5] Walanda, D.K., Anshary, A., Napitupulu, M., Walanda, R.M. (2022). The utilization of corn stalks as biochar to adsorb BOD and COD in hospital wastewater. International Journal of Design & Nature and Ecodynamics, 17(1): 113-118. https://doi.org/10.18280/ijdne.170114

[6] Xu, Y., Batumalay, M., Chan, C.K., Wider, W., Fu, K., Yang, L.M., Peng, J.S. (2025). Hot topics and frontier evolution of formation control research in multiple robots. Journal of Robotics, 2025(1): 3827954. https://doi.org/10.1155/joro/3827954

[7] Sharma, K., Shivandu, S. K. (2024). Integrating artificial intelligence and Internet of Things (IoT) for enhanced crop monitoring and management in precision agriculture. Sensors International, 5(2024): 100292. https://doi.org/10.1016/j.sintl.2024.100292

[8] Miller, T., Mikiciuk, G., Durlik, I., Mikiciuk, M., Łobodzińska, A., Śnieg, M. (2025). The IoT and AI in agriculture: The time is now—A systematic review of smart sensing technologies. Sensors, 25(12): 3583. https://doi.org/10.3390/s25123583

[9] Huda, SSM S., Akhtar, A., Ahmed, E., Hoq K. Md. S., Islam, Md. N. (2026). Artificial intelligence in agriculture across south Asia: Technology adoption, improvements, and sustainability outcomes. Sustainable Futures, 11(2026): 101620. https://doi.org/10.1016/j.sftr.2025.101620

[10] Spagnualo, M., Todde, G., Carria, M., Furnitto, N., Schillaci, G., Failla, S. (2025). Agricultural robotics: A technical review addressing challenges in sustainable crop production. Robotics, 14(2): 9. https://doi.org/10.3390/robotics14020009

[11] Ulaby, F., Bush, T. (1976). Corn growth as monitored by radar. IEEE Transaction on Antennas and Propagation 24(6): 819-828. https://doi.org/10.1109/TAP.1976.1141452
[12] Putro, S.S., Ansori, N., Fuad, M., Rochman, E.M.S., Asmara, Y.P., Rachmad, A. (2025). Corn leaf disease classification using Convolutional Neural Network based on MobileNetV2 with RMSProp optimization. Mathematical Modelling of Engineering Problems, 12(2): 465-474. https://doi.org/10.18280/mmep.120211
[13] Ulnar, I., Topakci, M. (2015). Design of a remote-controlled and GPS-guided autonomous robot for precision farming. International Journal of Advanced Robotic Systems 12(12): 1-10. https://doi.org/10.5772/62059
[14] Shams, O.A., Alturaihi, M.H., Mustafa, M.A.S., Majdi, H.S. (2023). Enhancement of drones' control and guidance systems channels: A review. Journal Européen des Systèmes Automatisés, 56(2): 201-212. https://doi.org/10.18280/jesa.560204
[15] Alshbatat, A.I.N., Awawdeh, M. (2024). Vision-based autonomous landing and charging system for a Hexacopter Drone. Journal Européen des Systèmes Automatisés, 57(1): 225-237. https://doi.org/10.18280/jesa.570122
[16] Akolkar, S.M., Jejurkar, D., Ahire, P., Jore, K., Gawale, P. (2025). Remote control based multipurpose AgroRobot. International Journal for Multidisciplinary Research, 7(1): 1-8. https://doi.org/10.36948/ijfmr.2025.v07i01.36559
[17] Mohan, M., Richardson, G., Gopan, G., Aghai, M.M., et al. (2021). UAV-supported forest regeneration: Current trends, challenges, and implications. Remote Sensing, 13(13): 2596. https://doi.org/10.3390/rs13132596
[18] Hernandez, H.A., Mondragon, I.F., Gonzales, S.R., Pedraza, L.F. (2025). Reconfigurable agricultural robotics: Control strategies, communication, and applications. Computers and Electronics in Agriculture, 234: 110161. https://doi.org/10.1016/j.compag.2025.110161
[19] Ye, J.X., Yu, Z.H. (2024). Fusing global and local information network for tassel detection in UAV imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 17: 4100-4108. https://doi.org/10.1109/JSTARS.2024.3356520
[20] Peral, M., Sanfeliu, A., Garrell, A. (2022). Efficient hand gesture recognition for human-robot interaction. IEEE Robotics and Automation Letters, 7(4): 10272-10279. https://doi.org/10.1109/LRA.2022.3193251
[21] McNeill, D. (2014). The emblem as metaphor. In From Gesture in Conversation to Visible Action as Utterance: Essays in honor of Adam Kendon, pp. 75-94. https://doi.org/10.1075/z.188.05nei
[22] Terreran, M., Barcellona, L., Ghidoni, S. (2023). A general skeleton-based action and gesture recognition framework for human-robot collaboration. Robotics and Autonomous Systems, 170: 104523. https://doi.org/10.1016/j.robot.2023.104523
[23] Ilyas, C.M.A., Nunes, R., Nasrollahi, K., Rehm, M., Moeslund, T.B. (2021). Deep emotion recognition through upper body movements and facial expression. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021), pp. 669-679. https://doi.org/10.5220/0010359506690679
[24] Yamagami, M., Portnova-Fahreeva, A.A., Kong, J., Wobbrock, J.O., Mankoff, J. (2023). How do people with limited movement personalize upper-body gestures? Considerations for the design of personalized and accessible gesture interfaces. In Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23). Association for Computing Machinery, New York, NY, USA, pp. 1-15. https://doi.org/10.1145/3597638.3608430
[25] Fuad, M. (2015). Skeleton based gesture to control manipulator. In 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, pp. 96-101. https://doi.org/10.1109/ICAMIMIA.2015.7508010
[26] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M. (2020). BlazePose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204. https://doi.org/10.48550/arXiv.2006.10204
[27] Santoro, D., Ciano, T., Ferrara, M. (2024). A comparison between machine and deep learning models on high stationarity data. Scientific Reports, 14: 19409. https://doi.org/10.1038/s41598-024-70341-6
[28] Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. IEEE Access, 9: 120247-120260. https://doi.org/10.1109/ACCESS.2021.3106864