Comparing feature usage in IMU-based gesture control for omnidirectional robot via wearable glove

Dahnial Syauqy, Eko Setiawan, Edita Rosana Widasari
Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia

Article Info

Article history:

Received Dec 5, 2023
Revised Mar 21, 2024
Accepted Apr 19, 2024

Keywords:

Euler
Hand gestures
Omnidirectional
Quaternion
Random forest
Wearable system

ABSTRACT

To improve the intuitiveness of maneuver control on omniwheeled mobile robot, many hand gesture-based robot controls have been developed. The focus of this research is to develop a wearable system for data acquisition from inertial measurement unit (IMU) sensors and compare its features to be used as gesture recognition using the random forest algorithm. With the need of resource constrained device for wearable system based on microcontrollers, we compared the use of Euler and quaternion-based orientation data as input features. As additional comparison, dimension reduction was also carried out using the principal component analysis (PCA) method. Hand gestures are recognized using data obtained by the IMU sensor embedded in the wearable glove. This study compared the accuracy and size of library files embedded in microcontrollers in several feature usage scenarios. The test evaluation results of all scenarios show that the use of all features provides a balance between high accuracy but small file sizes, respectively 99% and 9.2 KB. However, the use of other fewer features, such as by only using 3 Euler data, 4 quaternion data, or by using PCA algorithm (PC=3) can also be used since the accuracy is still above 90%, with a relatively larger file size.

Corresponding Author:

Dahnial Syauqy
Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya
Veteran St. No.10-11, Ketawanggede, Lowokwaru, Malang, East Java 65145, Indonesia
Email: dahnial87@ub.ac.id

1. INTRODUCTION

Omnidirectional robots are one type of special wheeled robot that allows the robot to move freely in various directions [1]. This type of robot uses omni wheels that allow it to execute movements maneuver with a higher degree of freedom (DOF) than conventional wheeled robots [2]. With holonomic motion capabilities, the robot can freely move in all directions without changing the orientation of the robot [3]. In general, for regular wheeled robots, the types of maneuvers that can be done are limited to forward, backward and rotation movement. With the omnidirectional type, the robot can also perform side motion maneuver, diagonal motion, and in-place rotation.

In some non-autonomous omnidirectional robots that have been developed, they are controlled directly using a physical remote [4] or using a smartphone application [5]. In both ways, some maneuvers are represented by pressing the directional buttons on the controller. To improve the intuitiveness of maneuver control, many hand gesture-based robotic controls have been developed [6]–[8]. To be able to acquire hand gestures, it is generally done in two ways, externally using a camera [9] or internally using sensors embedded in wearables glove or bracelets [10]. The weakness of camera-based gesture acquisition is certainly related to processing power on computing devices [11]. Therefore, many sensor-based gesture acquisition studies have been carried out, for example to manage smart home devices [12], sign language recognition [13], and to control wheeled robot maneuvers.

With the need for several maneuvering gestures to be performed, the focus in this study is to acquire sensor data from the inertial measurement unit (IMU) in the form of Euler and quaternion-based orientation data and perform gesture recognition using random forest algorithms. Principal component analysis (PCA) is a method of deriving the dimensions of features while retaining most of the information in the dataset [14], [15]. Generally, this algorithm is used to overcome the curse of dimensionality problem that occurs when the number of data dimensions is large enough compared to the sample size. Besides being used for data dimension reduction, PCA is also often used as a pre-processing technique before performing other statistical analyses such as classification or regression [16]. PCA can also be used to perform data compression on data storage and transfer [17]. With the need for a wearable system that is based on microcontrollers and has resource constraints, this study compares the use of both features and analyzes the application of the PCA algorithm and its impact on the accuracy and use of microcontroller memory.

The main objective of this paper is to compare the use of Euler and quaternion-based orientation data as input features of recognizing hand gesture using data obtained by the IMU sensor embedded in the wearable glove. As additional comparison, dimension reduction was also carried out using the PCA method. To provide structured explanation, the rest of the article is formatted as follows: chapter 2 provides related research and literature, chapter 3 describes the methodology used in the research, chapter 4 explains the experimental testing result and the discussion, and finally closed by chapter 5 conclusions.

2. RELATED LITERATURE

2.1. Related research

Research proposed by Jain *et al.* [18] in 2019 presenting gesture control of four-wheel mobile robot. Accelerometer was used to obtain and control the arm of robot by using human hand. The gesture control was only used to pick and place of single object. In 2018, another research related to hand gestures to control home appliances was proposed by Verdadero *et al.* [19]. The author used android based hand gesture interface system and mainly used its camera to detect static hand gesture to be processed. Later, the detected gesture will be sent using infrared to the controlled appliances. The main limitation of the system is that it requires correct static gesture with proper light illuminance for accurate recognition. Schade *et al.* [20] in 2023 proposed hand gesture recognition system using gloves for gaming purposes. The author used gloves with 3-axis 9 DOF IMU sensors on the palm and each finger. To compact the orientation representation, the author also used quaternion that consist of 4 numbers, thus each sensor provides 13 values. To collect sensor data from the gloves, they used microcontroller unit but then the data were sent to PC wirelessly for processing and classification.

Another research by Tsai *et al.* [21] in 2018 proposed the use of FPGA to process hand gesture recognition system based on dual camera with depth-map. The main reason of using FPGA is that because of the complexity and high computational time of running dual-camera based recognition algorithm. In 2019 Sabuj *et al.* [22] proposed another simple approach for hand gesture-based robot control for assisting people with paralysis disability. Instead of using orientation-based sensors, they used 4 infrared sensors mounted on glove. The combination of touching on the sensors determines the movement of the robot. Based on that approach, they limit up to 4 directional movements added with one idle. Even though it provides very fast directional determination, the number of gestures is tied to the physical number of sensors used and its combination.

2.2. Euler and quaternion orientation system

Euler rotation refers to the use of three rotation angles to represent rotation (commonly known as roll, pitch, and yaw). These three angles measure rotation on three orthogonal axes (e.g., X, Y, and Z axes). Although simple, Euler's rotation can have problems such as gimbal lock [23], in which some angular configurations result in a loss of rotational freedom.

Quaternion is a more complex mathematical representation for rotation than Euler. The quaternion uses four numbers (x, y, z, w) for the representation of rotation. Some of the advantages of quaternion are the absence of gimbal lock problems; can be used on smooth rotational interpolation; and are generally suitable for use in physics calculations, computer graphics, and robotics.

2.3. Random forest

Random forest is one of the popular algorithms in machine learning for classifying, regression, and prediction. Random forest is an ensemble form, which means it combines predictions from several basic models to achieve better results than individual models [24]. Random forest is based on the concept of a decision tree in the form of a hierarchical structure that makes decisions using a series of questions and conditions. Random forest works by creating multiple decision trees where each tree is trained on a subset of training data that is randomly retrieved by bootstrap sampling. The advantage of the random forest algorithm is that it can handle large datasets with many features; and can be used to assess the importance of features in the model. However, tuning hyperparameters is crucial to get the best performance from the random forest and avoid overfitting [25].

Some of the main hyperparameters to note include the number of trees (*n_estimators*), the number of features captured in each split (*max_features*), the maximum depth of trees (*max_depth*), the minimum number of samples required to divide the nodes (*min_samples_split*). In addition, cross-validation methods to evaluate model performance with various combinations of hyperparameters should be used. There are two general methods for finding the optimal combination of hyperparameters: grid search and random search. In random search, random values are selected from a predetermined range. When random search is complete, an evaluation of model performance with relevant metrics such as accuracy, is performed in each combination of hyperparameters and the combination that gives the best results is selected.

2.4. Principal component analysis

PCA is a multivariate statistical technique used to reduce the dimensions of a data set by preserving most of the information contained in it [26]. PCA is used to find patterns and structures in data by identifying the most correlated variables and subtracting the dimensions of those variables. PCA is used to reduce dimensionality from data by eliminating less significant variables and keeping more important variables. PCA can also be used to compress data by reducing dimensions, making it easier to store and transmit data.

The eigenvalue is a measure of how much variance is described by a component (principal component) in the data. Each component has a different eigenvalue, and the component with the highest eigenvalue is the most significant major component in the data. Therefore, PCA is done by selecting the components with the highest eigenvalues and ignoring those components with lower eigenvalues. The PCA process involves transforming data into a new space consisting of major components sorted by eigenvalues. By sorting components by eigenvalue, PCA makes it possible to identify the most significant major components in the data and eliminate the less significant components. In this new space, data can be represented using a smaller number of components, making it easier to analyze and interpret data.

3. RESEARCH METHOD

3.1. System design

In general, the system block diagram consists of two parts, the wearable glove subsystem, and the omnidirectional robot subsystem. The block diagram is shown in Figure 1. In the wearable glove sub-system, we use one microcontroller unit, 1 IMU sensor unit, and is equipped with a battery embedded in a glove which is designed to be worn on the right wrist. Furthermore, the robot pilot can make several gestures to be recognized and sent to the omni-wheel robot sub-system. The microcontroller used is ESP32 which is equipped with classic Bluetooth connectivity. The IMU sensor used is the IMU 9 DOF type with the BNO055 type. After the robot pilot performs a gesture, the IMU sensor acquires orientation data in the form of 3-dimensional Euler data and 4-dimensional quaternion data. Following that, the microcontroller processes the data using a random forest algorithm and sends the results to the mobile robot sub-system via a classic Bluetooth connection. In this study will also make comparisons with the application of the PCA algorithm to reduce features while maintaining the highest possible variance.

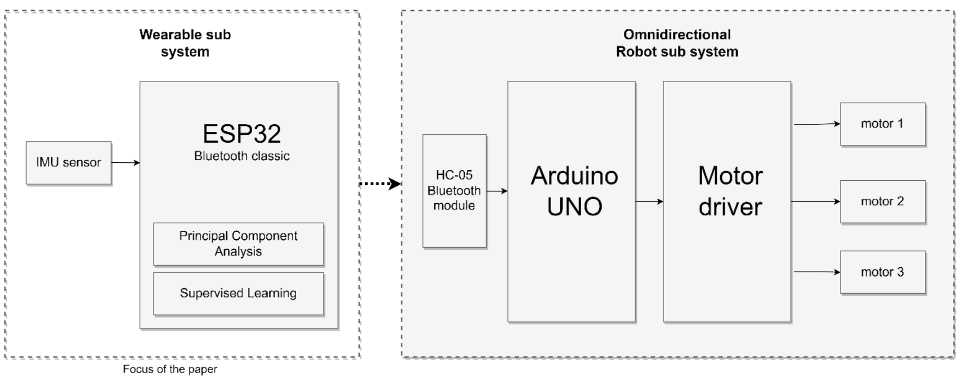

Figure 1. Hardware block diagram

In the omni-wheel mobile robot sub-system, there is a microcontroller (Arduino UNO) as well as a receiver for sending data (Bluetooth module), 3 units of omni-wheeled wheels along with their respective motors and drivers. Continuously, the mobile robot listens to data transmission in the form of instructions for the robot's direction of motion, as well as executing movements by giving commands to each motor connected to the omni wheel. The focus of this paper research is the gesture acquisition on the wearable glove sub-system; thus, this experiment is limited to the wearable system only and limited to 5 types of robot maneuvers listed in Table 1.

Table 1. Overview of collected datasets

Types of maneuvers	Label in dataset	Amount of dataset
Forward	1	252
Backward	2	242
Right side	3	215
Left side	4	252
Neutral	0	201
Total		1162

3.2. Hardware implementation

Before carrying out the training data collection stage, wearable gloves are built based on the design that has been made before. The glove system was made of right-handed gloves, then all electronic components were put into a box made using a 3D printer and pinned to the glove. We used battery as power supply, put it in the box and configured in such a way that makes replacement easy. After the physical form of the wearable glove has been completed, then we validated the sensor readings and communication to the mobile robot sub-system through classic Bluetooth connectivity. The results of the hardware installation on the wearable glove are illustrated in Figure 2.

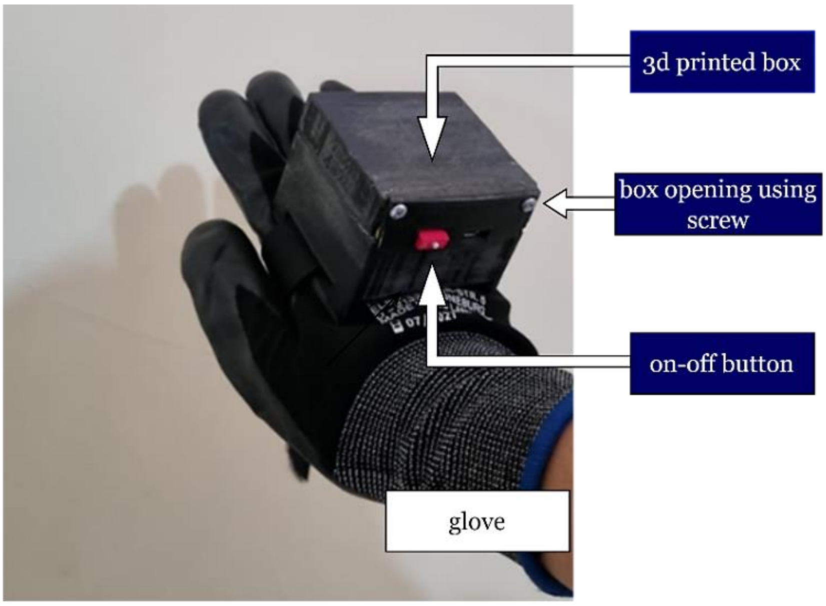

Figure 2. Hardware installation on wearable glove

3.3. Dataset acquisition

After validating components embedded in the glove and verifying that it was able to acquire IMU sensor data, the training data retrieval stage is then carried out. The process of retrieving training data was carried out by one person who performs several maneuvers of robot movements repeatedly in all directions. The data retrieval process was carried out on the condition that the wearable glove was connected via a USB cable to the laptop. Then, recorded data were sent through a serial port connected to the ESP32 microcontroller and then saved it in csv format. Table 1 shows an overview and the amount of data recorded as a dataset in each maneuver class.

3.4. Flow design of wearable glove subsystem

After the dataset has been collected, the process of developing pattern recognition algorithms using random forests is carried out using scikit learn in Python. Initially, the preprocessing stage was done by splitting datasets into training data and data testing with a proportion of 70:30 with data stratification. Next, the main dimension was selected as the random forest input feature. In our test cases, there were several scenarios of input features used in research, i.e:

- All IMU sensor reading data (7 dimensions consisting of 3 Euler data and 4 quaternion data) as input features.
- Based on only 3 Euler data as input feature.
- Based on only 4 quaternion data as input feature.
- Using all IMU sensor reading data but transformed using PCA algorithm for dimension reduction.

The four scenarios were analyzed and compared based on the results of the accuracy and size of the results of the model formed. Random forest is a supervised algorithm that requires hyperparameter tuning to produce the best accuracy. Therefore, in each scenario also simultaneously perform the hyperparameter tuning process with the random search feature with several parameters using 10-fold cross validation. Hyperparameters that are tuned include *n_estimator, max_features, max_depth, min_samples_split, min_samples_leaf,* and *bootstrap*. Finally, the model created by scikit is then ported into C format so that it can be embedded into a microcontroller using micromlgen. Micromlgen is a python library that can export scikit models into C microcontroller format [27].

Prior running these 4 scenarios, for the sake of efficiency in the number of tests, only the most efficient *max_depth* and *n_estimator* will be used both in terms of accuracy and file size. Thus, the initial test is to compare the parameters and the impact of changes in the two variables on the accuracy and size of files successfully generated by the micromlgen library. Figure 3 shows a software flow chart on the wearable glove subsystem.

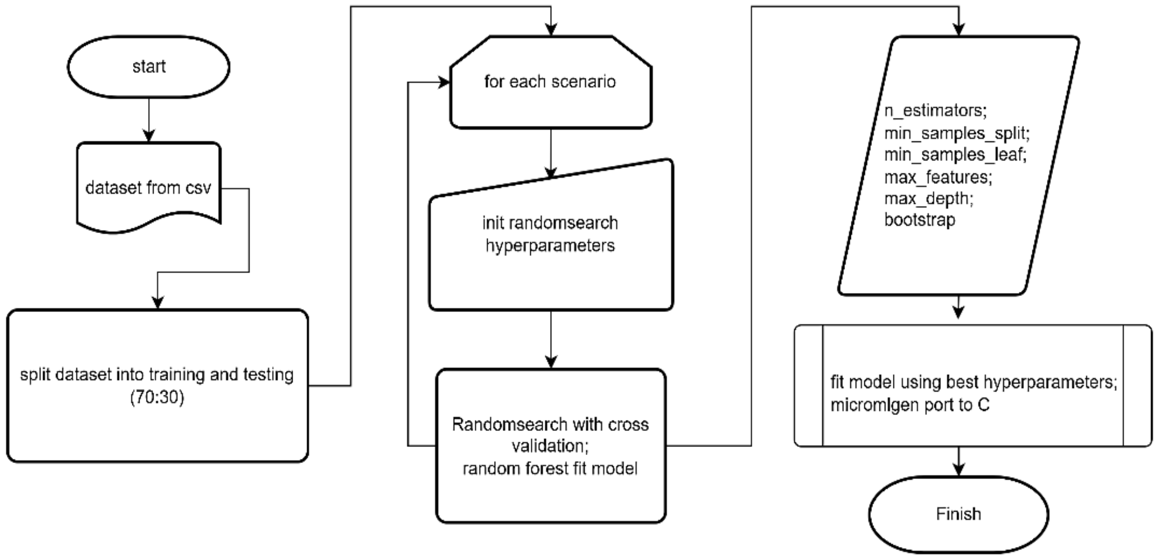

Figure 3. Flow chart on the wearable glove subsystem

4. RESULT AND DISCUSSION

4.1. Testing of *max_depth* and *n_estimator* parameter selection

In this test, all IMU sensor reading features are used as algorithm input (7 dimensions: 3 Euler data and 4 quaternion data). Several tests were carried out with the distinction of grid search ranges to find the best hyperparameters on the random forest algorithm. There are 4 sampling grid searches that will be used, in the range of 1 to 3; 1 to 5; 6 to 10; and 10 to 50. The expected result is to find the most efficient *max_depth* and *n_estimator* hyperparameter values in terms of accuracy and file size at the same time. This is because the algorithm will be embedded in devices with limited resources, so it is desirable that the use of storage size is suppressed as optimally as possible so that it can be used for other functions outside of pattern recognition computing.

In the results shown in Table 2, it appears that the higher the grid search range applied, the accuracy will increase. However, the other side is that the size of files generated by micromlgen libraries in .h format and will be embedded in microcontrollers is also increasing. In order to get a balance between accuracy and file size created, grid search 1 to 5, with an accuracy of 99% and a size of 9.02 KB was chosen as the basis of the hyperparameters used in the next section of testing.

Table 2. Results of grid search range testing for accuracy and file size

Range of grid search used	1 to 3	1 to 5	6 to 10	10 to 50
Max depth obtained	3	4	6	10
n estimators obtained	2	3	6	11
Accuracy	95%	99%	100%	100%
File size	4.15 KB	9.02 KB	29.8 KB	68.1 KB

4.2. Testing 4 scenarios with different types of input data

After determining the hyperparameters, using a search grid in range of 1 to 5 for *max_depth* and *n_estimator*, then it was tested alternately with 4 input scenarios as described in the test configuration. Each scenario is explained as follows, and later, overall result discussion is presented:

4.2.1. Scenario 1: using all IMU sensor reading data (7 dimensions consisting of 3 Euler data and 4 quaternion data) into the input feature

In scenario 1, the input feature used as vector input random forest is 7-dimension orientation data consisting of 3 Euler data (eX, eY, eZ) and 4 quaternion data (qW, qX, qY, qZ). Hyperparameter tuning results are obtained using random grid search. Figure 4 shows the confusion matrix using data testing of the built model. The hyperparameters and the accuracy result, as well as the code size is presented in Table 3.

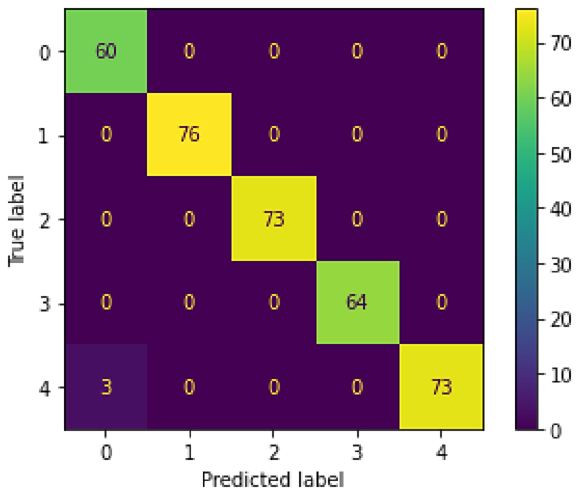

Figure 4. Confusion matrix all features

Table 3. Overall results comparison of accuracy and size of ported models produced

	Euler	Quaternion	ALL	PCA 3	PCA 2
Max depth	5	5	4	5	5
N estimator	5	4	3	5	4
Accuracy	94.5%	98.8%	99%	95.1%	71.60%
Size	18.6 KB	23.0 KB	9.02 KB	17.2 KB+1.78 KB=18.98 KB	17.3 KB +1.78 KB=19.08 KB

4.2.2. Scenario 2: using 3 data Euler as input feature

In scenario 1, the input feature used as a vector input random forest is 3-dimensional Euler orientation data (eX, eY, eZ). As previous steps, random grid search was used to perform hyperparameter tuning. Figure 5 shows the confusion matrix using data testing of the built model. The hyperparameters and the accuracy result, as well as the code size is presented in Table 3.

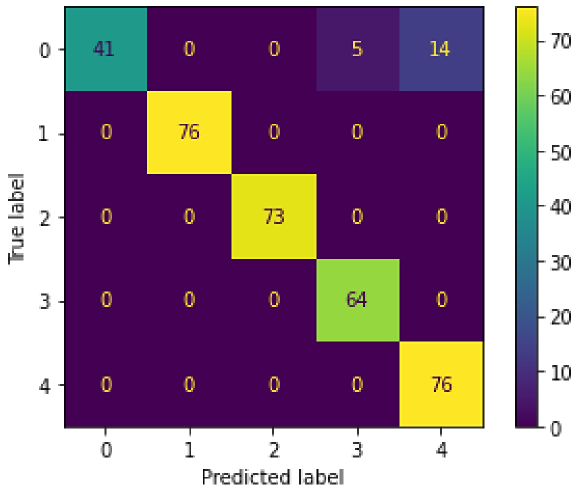

Figure 5. Confusion matrix only Euler

4.2.3. Scenario 3: using 4 data quaternion as input feature

In scenario 3, the input feature used as a vector input random forest is 4 dimensions of quaternion orientation data (qW, qX, qY, qZ). As previous steps, random grid search was used to perform hyperparameter tuning. Figure 6 shows the confusion matrix using testing data testing of the built model. The hyperparameters and the accuracy result, as well as the code size is presented in Table 3.

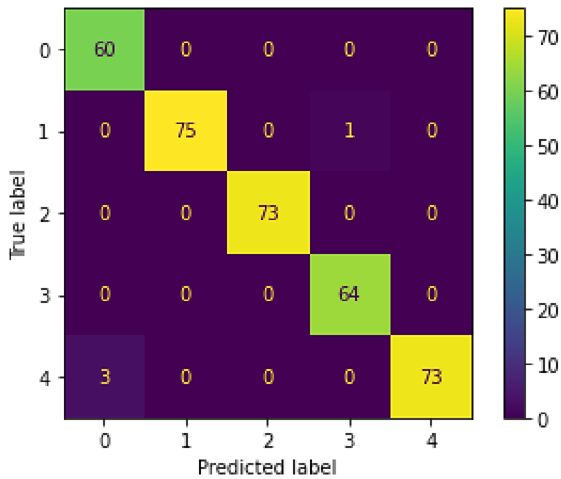

Figure 6. Confusion matrix only quaternion

4.2.4. Scenario 4: using the entire IMU sensor reading data with the application of the PCA algorithm for dimension reduction

In scenario 4, the input feature used as a vector input random forest is all dimensions of Euler orientation data (eX, eY, eZ) and quaternion (qW, qX, qY, qZ) but before processing using random forest, the PCA method is applied to the dataset to reduce the number of dimensions. As previous steps, random grid search was used to perform hyperparameter tuning. Figure 7 shows the confusion matrix using data testing of the built model where Figure 7(a) using PC=3 and Figure 7(b) using PC=2. To find out the right number of principal components and explained variance ratio produced, PCA analysis was previously carried out on a PC of at least 1 to a maximum number of 7 which means using all data dimensions. From Figure 8 it can be seen that at least the number of principal components 3 and 2 gives 100% and 90% explained variance ratios respectively. Thus, in this scenario, two tests were carried out, which consist of using PC=3 and PC=2. The hyperparameters and the accuracy result, as well as the code size is presented in Table 3.

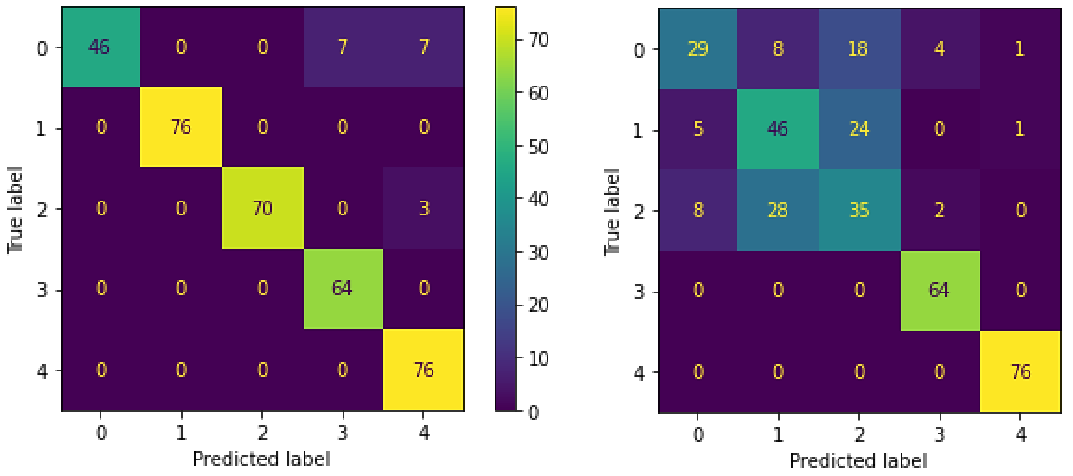

Figure 7. Confusion matrix all features with PCA (a) PC=3 and (b) PC=2

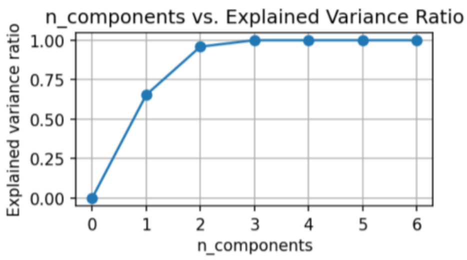

Figure 8. Number of principal components and explained ratios

4.3. Overall result and discussion

Based on the four scenarios carried out, the overall results are presented to make comparisons both in terms of accuracy and size of the results of ported Scikit models formed using the MicroMLGen library. Table 3 shows a comparison of all scenarios. The results of testing the input feature with all scenarios above show that by using random grid search, the best *max_depth* and *n_estimator* is in the range of 1 to 5. It appears that the highest accuracy of 99% was achieved using all 7-dimensional features in the form of 3 Euler data plus 4 quaternion data for the classification of all gestures. Meanwhile, the lowest accuracy is obtained by using only 2 features of PCA transformation results, which resulted 71.6%. The use of 3 Euler data, 4 quaternion data and 3 PCA transformation data as input features results in more than 90% accuracy which means that it can also be used as an alternative to gesture recognition. However, with limitations on computing resources in the microcontrollers used, the size of the embedded file is also one of the considerations in choosing the features used.

The result of porting using MicroMLGen is a file with the extension .h which is then later called in the microcontroller code. The smaller the library used will provide more space for coding development on the microcontroller. Thus, the size preference is the library with the smallest size but still has good accuracy. Therefore, in the test case of this study, it was decided to use all features (7 data) as input features with an accuracy of 99% and a library size of only 9.02 KB.

5. CONCLUSION

The design of using IMU sensors embedded in gloves, with the aim of acquiring hand gestures has been implemented in this study. The microcontroller used is ESP32 which is equipped with classic Bluetooth connectivity. The IMU sensor used 9 DOF IMU BNO055 embedded in the wearable glove to obtain hand gestures. This study compares the accuracy and size of library files embedded in microcontrollers from several feature scenarios, consisting of 3-dimensional Euler data and 4-dimensional quaternion data. Next, the microcontroller processes the data using the random forest algorithm and sends the results to the mobile robot's sub-system via a classic Bluetooth connection. The test evaluation results of all scenarios show that the use of all features provides a balance between high accuracy but small file sizes, respectively are 99% and 9.2 KB. However, the use of fewer features, for example only 3 Euler data, or 4 quaternion data, or using the PCA algorithm with 3 PCs transformation features can also be used because the accuracy is still above 90%, but with a relatively larger file size so that other functional adjustments can be made by the microcontroller.

For further research and development, more gestures can be added that accommodate all possible movements on the omni-wheel robot. In addition, IMU sensors can also be added at different locations, for example on the wrist so that it can measure the degree of difference between the orientation of the back of the hand and the wrist. Another thing that can be done is a comparison with other pattern recognition algorithms that are more efficient in terms of accuracy and final size that will be embedded in the microcontroller.

REFERENCES

[1] M. J. A. Safar, "Holonomic and omnidirectional locomotion systems for wheeled mobile robots: a review," *Jurnal Teknologi*, vol. 77, no. 28, Dec. 2015, doi: 10.11113/jt.v77.6799.

[2] R. T. Yunardi, D. Arifianto, F. Bachtiar, and J. I. Prananingrum, "Holonomic implementation of three wheels omnidirectional mobile robot using dc motors," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, 2021, doi: 10.18196/jrc.2254.

[3] M. Wada, "Omnidirectional control of 4wd robotic base," in *2006 SICE-ICASE International Joint Conference*, IEEE, 2006, pp. 2064–2067, doi: 10.1109/SICE.2006.315493.

[4] L. Pacurar, R. Donca, A. Pleşa, V. Muresan, and R. Balan, "Rapid prototyping on an omnidirectional platform using lego education components," *Romanian Review Precision Mechanics, Optics and Mechatronics*, no. 39, pp. 145–148, 2011.

[5] M. Pena-Cabrera, J. Antonio-Gomez M., R. Osorio, H. Gomez, V. Lomas, and I. Lopez-Juarez, "Fuzzy logic for omnidirectional mobile platform control based in fpga and bluetooth communication," in *2014 13th Mexican International Conference on Artificial Intelligence*, IEEE, Nov. 2014, pp. 135–140, doi: 10.1109/MICAI.2014.27.

[6] C. H. Chou and S. W. Lin, "Natural user interface design for control of robots by hand gestures," *Journal of Computers*, vol. 30, no. 1, pp. 174–182, 2019.

[7] P. A. Jawalekar, "Robot control by using human hand gestures," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 02, pp. 389–391, 2018.

[8] M. Wang, W.-Y. Chen, and X. D. Li, "Hand gesture recognition using valley circle feature and hu's moments technique for robot movement control," *Measurement*, vol. 94, pp. 734–744, Dec. 2016, doi: 10.1016/j.measurement.2016.09.018.

[9] T. Kapuscinski, M. Oszust, M. Wysocki, and D. Warchol, "Recognition of hand gestures observed by depth cameras," *International Journal of Advanced Robotic Systems*, vol. 12, no. 4, p. 36, Apr. 2015, doi: 10.5772/60091.

[10] C. Mummadi *et al.*, "Real-time and embedded detection of hand gestures with an imu-based glove," *Informatics*, vol. 5, no. 2, p. 28, Jun. 2018, doi: 10.3390/informatics5020028.

[11] X. Meng, C. M. Cheung, K. L. Ho, K. S. Lui, E. Y. Lam, and V. Tam, "Building smart cameras on mobile tablets for hand gesture recognition," *2012 6th International Conference on Distributed Smart Cameras, ICDSC 2012*, 2012.

[12] Y. Vasylkiv, A. Neshati, Y. Sakamoto, R. Gomez, K. Nakamura, and P. Irani, "Smart home interactions for people with reduced hand mobility using subtle emg-signal gestures," *Studies in Health Technology and Informatics*, vol. 257, pp. 436–443, 2019, doi: 10.3233/978-1-61499-951-5-436.

[13] C.-M. Chiu, S.-W. Chen, Y.-P. Pao, M.-Z. Huang, S.-W. Chan, and Z.-H. Lin, "A smart glove with integrated triboelectric nanogenerator for self-powered gesture recognition and language expression," *Science and Technology of Advanced Materials*, vol. 20, no. 1, pp. 964–971, Dec. 2019, doi: 10.1080/14686996.2019.1665458.

[14] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, Apr. 2016, doi: 10.1098/rsta.2015.0202.

[15] J. Zheng and C. Rakovski, "On the application of principal component analysis to classification problems," *Data Science Journal*, vol. 20, no. 1, Aug. 2021, doi: 10.5334/dsj-2021-026.

[16] T. Sarsodia, U. R. Bhatt, R. Upadhyay, and V. Bhat, "PCA based components selection criteria for computationally efficient physical layer key generation (PLKG) system," *International Journal of Electronics and Telecommunications*, vol. 69, no. 2, pp. 361–369, Dec. 2023, doi: 10.24425/ijet.2023.144372.

[17] T. Zhu, X. Cheng, W. Cheng, Z. Tian, and Y. Li, "Principal component analysis based data collection for sustainable internet of things enabled cyber–physical systems," *Microprocessors and Microsystems*, vol. 88, p. 104032, Feb. 2022, doi: 10.1016/j.micpro.2021.104032.

[18] M. Jain *et al.*, "Object detection and gesture control of four-wheel mobile robot," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jul. 2019, pp. 303–308, doi: 10.1109/ICCES45898.2019.9002323.

[19] M. S. Verdadero, C. O. Martinez-Ojeda, and J. C. D. Cruz, "Hand gesture recognition system as an alternative interface for remote controlled home appliances," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, IEEE, Nov. 2018, pp. 1–5, doi: 10.1109/HNICEM.2018.8666291.

[20] A. Schade *et al.*, "On the advantages of hand gesture recognition with data gloves for gaming applications," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, IEEE, Mar. 2023, pp. 313–315, doi: 10.1109/PerComWorkshops56833.2023.10150283.

[21] T.-H. Tsai, Y.-C. Ho, and Y.-R. Tsai, "Implementation of 3d hand gesture recognition system using FPGA," in *2018 International SoC Design Conference (ISOCC)*, IEEE, Nov. 2018, pp. 131–132, doi: 10.1109/ISOCC.2018.8649939.

[22] B. Sabuj, M. J. Islam, and M. A. Rahaman, "Human robot interaction using sensor based hand gestures for assisting disable people," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, IEEE, Dec. 2019, pp. 1–5, doi: 10.1109/STI47673.2019.9068087.

[23] Q. Bai, M. Shehata, and A. Nada, "Review study of using euler angles and euler parameters in multibody modeling of spatial holonomic and non-holonomic systems," *International Journal of Dynamics and Control*, vol. 10, no. 5, pp. 1707–1725, Oct. 2022, doi: 10.1007/s40435-022-00913-9.

[24] S. Han and H. Kim, "On the optimal size of candidate feature set in random forest," *Applied Sciences (Switzerland)*, vol. 9, no. 5, p. 898, Mar. 2019, doi: 10.3390/app9050898.
[25] V. Saxena and A. Aggarwal, "Comparative study of select non parametric and ensemble machine learning classification techniques," in *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, IEEE, Dec. 2020, pp. 110–115, doi: 10.1109/ICACCCN51052.2020.9362798.
[26] M. D. Wood, L. E. R. Simmatis, J. A. Jacobson, S. P. Dukelow, J. G. Boyd, and S. H. Scott, "Principal components analysis using data collected from healthy individuals on two robotic assessment platforms yields similar behavioral patterns," *Frontiers in Human Neuroscience*, vol. 15, May 2021, doi: 10.3389/fnhum.2021.652201.
[27] "MicroML." github, Accessed: Dec. 05, 2023. [Online]. Available: https://github.com/eloquentarduino/micromlgen.

BIOGRAPHIES OF AUTHORS

Dahnial Syauqy received his undergraduate degree in electrical engineering at Universitas Brawijaya, and then graduated in 2014 with double master's degree program coordinated between Universitas Brawijaya, Indonesia and National Central University, Taiwan in biomedic electrical engineering focusing on signal processing in speech. Currently he is a full-time lecturer at Computer Engineering Undergraduate Program, Department of Informatics Engineering, Universitas Brawijaya. His research interests are embedded system, pattern recognition, and signal processing. He can be contacted at email: dahnial87@ub.ac.id.

Eko Setiawan graduated from Universitas Brawijaya, Indonesia, in 2008. He received the M.Eng. degree from University of Miyazaki, Japan in 2012 and from Universitas Brawijaya, Indonesia in 2013. He received the Ph.D. degree from University of Miyazaki, Japan, in 2018. He worked as a full-time lecturer with Universitas Brawijaya, Indonesia in Faculty of Computer Science. He was a staff member on the quality assurance unit in Universitas Brawijaya. He also served as editor section in Indonesian National Journal JTIIK. He received research grant from ASEAN-IVO. His current research interests include embedded system, robotics, and control system. He can be contacted at email: ekosetiawan@ub.ac.id.

Edita Rosana Widasari completed her doctoral studies at the University of Miyazaki (UoM) in Japan and was honored with the Female Researcher Encouragement Award from UoM in 2021. She also graduated with a master's degree through a double-degree program between the Department of Electrical Engineering at Universitas Brawijaya and UoM-Japan and obtained her undergraduate degree from the Department of Electrical Engineering at Universitas Brawijaya. She has been serving as a lecturer in Department of Informatics Engineering at Universitas Brawijaya since 2016, teaching courses that include digital signal computing, computer-based medical systems, computer and network systems analysis, electronic circuits, and intelligent systems. Beginning in 2022, she has taken on the role of head of the Robotics and Embedded Systems Laboratory. Her research interests primarily lie in the field of biomedical engineering, with a focus on signal processing and its application to embedded systems. She can be contacted at email: editarosanaw@ub.ac.id.