Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

GANE - Gesture Recognition Cane

Gyanchith Hari[1], Krishna Priya R[2], Vishnupriya H[3], Rishikesh T Nair[4], Dr Manazhy Rashmi[5]

[1, 2, 3, 4] *Department Of Electronics and Communication Engineering,Amrita Vishwa Vidyapeetham,Amritapuri Campus, India*

[5] *Assistant Professor, Department Of Electronics and Communication Engineering,Amrita Vishwa Vidyapeetham,Amritapuri Campus, India*

Abstract:- People who use a cane to get around often have difficulty using other devices, such as smartphones, at the same time. Our goal is to develop an innovative cane system that cloud services and gesture recognition technology to help elderly users leverage gesture recognition to interpret and translate user gestures into meaningful commands and actions in their smartphones and create a user-friendly and reliable cane system that can be seamlessly integrated into daily life.

Keywords*: Gestures, Gane, ANN, AWS, NodeMCU.*

1. Introduction

Our daily lives are now heavily reliant on smartphones. While the advancements in smartphone apps and new technologies have made life better for everyone, there is a significant opportunity for smartphones to improve the lives of elderly citizens who use a cane. AI is also helping people to become more aware and skilled at navigating their surroundings. However, despite these benefits, smartphone accessibility remains a challenge. Studies show that people find it difficult to use mobile app user interfaces. Additionally, there are several situations where elderly individuals who rely on a cane for mobility may find it difficult to quickly locate or access a smartphone. While voice commands are a solution, they may not always work in noisy environments, users may not want to use them for privacy reasons, and they may not be available in all languages. Gestures are promising because they are natural, easy to perform, and do not require users to change how they grip their canes. However, the solution must not interfere with the normal use of the cane and should be applicable to any cane without modification. The main aim of this proposed method is to develop an innovative cane system that uses the power of cloud-based DL services and gesture recognition technology. This system 1 is called GANE, which interprets the user's gestures using AI and translates them into meaningful actions on the user's phone. This simplifies the use of canes for the elderly, improving their daily lives and giving them a greater sense of independence and control. To achieve this, the mpu6050 sensor is used to collect raw data from the cane, which includes motion data in six different directions tracked by a three-axis gyroscope and a three-axis accelerometer. This data is then sent to AWS IoT Core using NodeMCU, where it is processed by a lambda function and our model is deployed in the cloud. Once the model predicts the gesture, the corresponding action on the mobile app (such as emergency calls or weather updates) is triggered. The main objective of GANE is to create a user-friendly and reliable cane system that can be easily integrated into daily life.

2. Hardware

Major Hardware Units are -MPU6050 Sensor- The MPU6050 (fig 1) is a commonly used Inertial Measurement Unit (IMU) that combines a 3-axis gyroscope and a 3-axis accelerometer on a single chip.

(i) Gyroscope: The MPU6050 integrates a 3-axis gyroscope, allowing it to measure rotational motion around the X, Y, and Z axes and it provides angular velocity data in degrees per second.

(ii) Accelerometer: The 3-axis accelerometer in the MPU6050 measures acceleration along the X, Y, and Z axes and is provided in terms of gravitational acceleration(g).

(iii) Digital Motion Processor: The MPU6050 includes a Digital Motion Processor that can offload some sensor fusion calculations, providing quaternion data and reducing the workload on the microcontroller.

(iv) Communication Interface: The MPU6050 communicates with microcontrollers using the I2C (Inter-Integrated Circuit) protocol. It has programmable digital filters to improve the quality of sensor data.

(v) Temperature Sensor: The MPU6050 has an integrated temperature sensor that provides the temperature of the chip.

(vi) Power Consumption: It is designed to operate at low power, suitable for battery-powered devices. The accelerometer and the gyroscope value give the gesture produced by the user. A lightweight power bank providing a 5V output is employed to power the entire system. This setup ensures extended operational time, as the system can be powered off when not in use, effectively conserving energy. This mechanism saves a lot of energy and makes it easy to use.

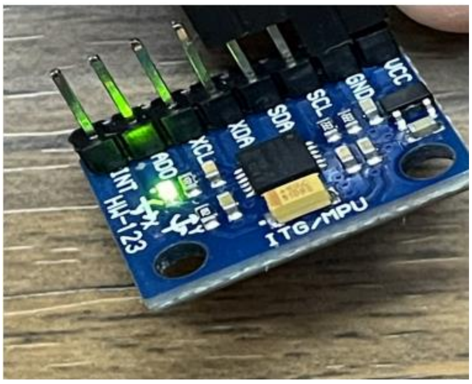

Fig 1: Mpu6050 with connections

NodeMCU - The ESP8266(fig 2) is a popular Wi-Fi module widely used in electronics and IoT projects. It features an integrated microcontroller. This module is known for its low cost, ease of use, and versatility. Key features of the ESP8266 include its built-in Wi-Fi capabilities, supporting both station and access point modes. This provides wireless features to our system by transmitting the raw data into the cloud eliminating the system in the middle when Arduino is used. Thus, this microcontroller is perfect for taking inputs to make a prediction.

Fig 2: NodeMCU

3. Proposed Method and algorithm

For training the collected data to prediction, Arduino UNO R3 was used as a microcontroller and took the input data. The Arduino IDE is the user interface between the system and the microcontroller. This software makes use of Arduino C for coding. Initially, programmed the controller to take serial data input using MPU6050.Then for training connected the hardware and execute the code so that the proposed work get the raw data. With the help of different Python scripts for labelling and normalizing the data.

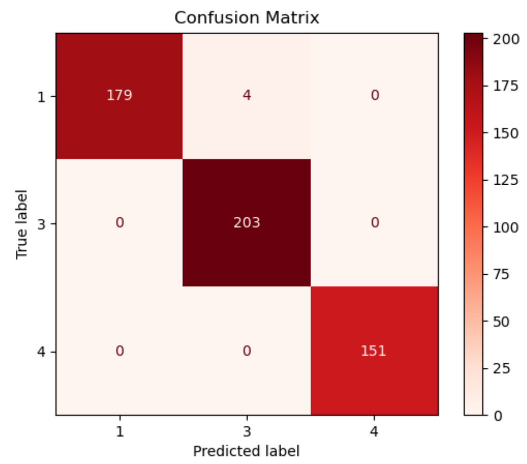

Fig 3: ANN Classification Confusion matrix

The dataset comprises 9399 readings, each consisting of 6 values: 3 from accelerometers (ax, ay, az) and 3 from gyroscopes (gx, gy, gz). These readings are grouped into sequences of 6, resulting in a training dataset with 1342 rows. Each row has six columns representing consecutive sensor readings. This grouping forms a window size of 6, suitable for capturing a complete gesture. Although larger window sizes are possible with different hardware modules, for the ESP8266, 6 was determined as the maximum size feasible within onboard memory constraints before transmitting to AWS. The dataset includes data from 3 different gestures, split into a 60:40 ratio for training and testing, with 537 instances reserved for testing. Using a simple Artificial Neural Network (ANN), we achieved an outstanding accuracy of 99.26%.

Artificial Neural Networks (ANN). ANNs are powerful computational models that can learn complex patterns and relationships in data through iterative training processes.

The workflow of the algorithm in step-by-step. The six main phases that occur in the algorithm.

(1) The ANN model operates similarly to the human brain. It can process a large dataset for training.

(2) The data is transferred from the inside layer to the outside layer through hidden layers.

(3) These hidden layers act as activation functions that help to identify patterns within the dataset

(4) In this model, ReLu was used as the hidden layer, which sets negative values to zero and promotes sparse activation. Softmax was used as the outside layer for multiclass classification.

(5) The model primarily depends on the weights of parameters. During each iteration of training, the weights are updated based on the error calculated in the previous pass. This process continues until the model learns to make accurate predictions.

(6) Once the model is trained, it is tested using a separate dataset to ensure its accuracy and performance. After successful testing, the model is used to make predictions on a new dataset. The efficiency of this model was beneficiary for this proposed work prediction as shown in Figure 4.

There were several advantages of choosing the ANN algorithm like tasks that involve serialized data often require continuous outputs, such as predicting the next value in a sequence or estimating a specific value based on the data. ANN is capable of providing continuous outputs with ease. Serialized data can sometimes contain noise or outliers. ANN is more resilient to this type of noise because they have a built-in capacity to learn underlying patterns and smooth out inconsistencies. ANN can be adjusted to various tasks by modifying their structure and training parameters. This flexibility enables them to be utilized for a wide range of tasks that require sequential data analysis, such as classification, regression, and anomaly detection. Scalability is an important factor when it comes to choosing an algorithm ANN can be effectively trained even on large datasets with appropriate optimization techniques.

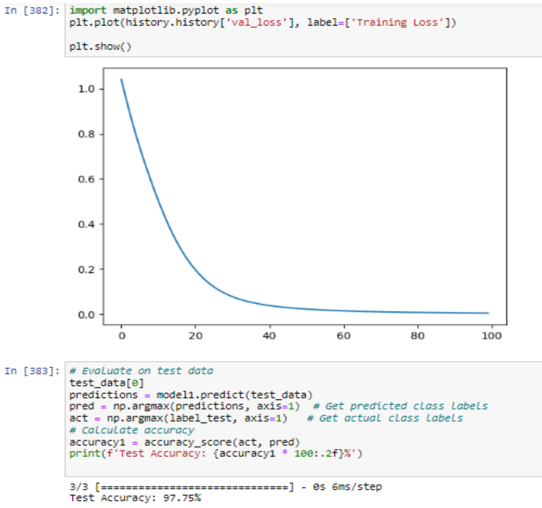

Fig4: Accuracy of ANN model

4. Software

AWS IoT Core

AWS IoT core is an IOT service by AWS that provides proper interaction with the Web and connected devices. NodeMCU communicates with AWS IoT Core via MQTT, ensuring easy and secure data transfer. This two-way communication allows the NodeMCU to send sensor data to the cloud while receiving updates or instructions. Each NodeMCU is registered in the AWS IoT Core device registry, providing a centralized management system. Security is ensured through mechanisms such as X.509 certificates, which ensure authenticated and encrypted communication.

AWS LAMBDA

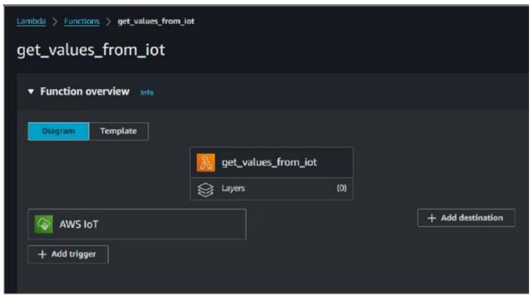

Fig5: Lambda function

AWS lambda acts as an intermediary between IOT Core and the deployed model. An IOT Core rule is created such that every time a value is received it triggers the lambda function as shown in figure 5. Then named it as, ”get values from iot” This lambda function will send the values to the render where the already deployed trained ANN Model exists. Once the prediction is done in Render it will send the result back to lambda which is then stored in a database. These gestures predicted will then be sent to the mobile app from which the action is triggered.

APP Details

The application, developed using React Native, provides various functionalities based on user gestures. React Native ensures cross-platform compatibility, making the app accessible on both iOS and Android devices. The choice of DynamoDB as the database is driven by its excellent scalability features and fast data querying capabilities.

Key functionalities offered by the app include displaying the current time, providing weather updates, and enabling emergency calls. Each of these functionalities is linked to specific gestures recognized by the app. Upon detecting a gesture, the app assigns a unique identifier to it. Subsequent predictions of new gestures trigger the corresponding functionality. Importantly, these newly detected gestures are then stored in DynamoDB for future reference.

Additionally, the app incorporates a fallback mechanism for scenarios where the app doesn't receive gesture prediction due to some internal server issue. The app retrieves information from DynamoDB regarding the most recent gesture recorded. It then compares the stored ID with the one cached in memory. A match indicates that the gesture has been previously executed.

Block Diagram

As shown in figure 6, the motion data of the CANE is collected using an MPU6050, IMU sensor through a NodeMCU. This Nodemcu sends these values to AWS IoT core. i.e, the value reached the web. The values that reach IOT Core invoke the lambda function that sends these values to a where the model is deployed. The final gesture prediction done by the model will be sent back to lambda and then stored it in DB and send it to the mobile application. When a particular gesture is predicted, the particular action on mobile (like emergency call, announce weather update) mapped against it will get triggered.

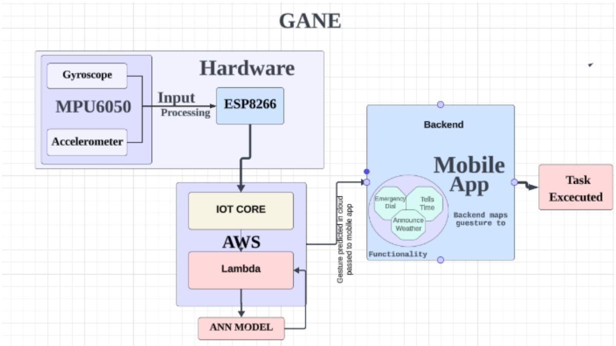

Fig6: Block Diagram

5. Results

The entire pipeline of GANE (Fig 8) was completed. The gesture number predicted was mapped as described in Table (Table: 1) below and was able to map actions like announcing time, weather, and emergency calls as shown in Figure 7 on the mobile phone through the app that was developed. For example, a twist was mapped with the announcement of time.

Table 1: Gesture Prediction Map

Number	Gesture
1	Double tap
3	Swing
4	Twist

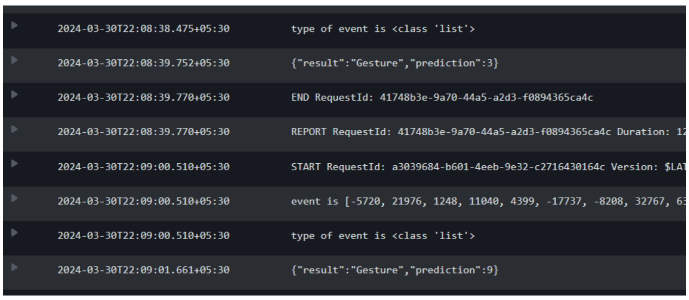

Fig 7: Results

Fig8: GANE-physical model

6. Conclusion

The latest study demonstrates better performance in recognizing hand gestures compared to previous research methods. By using accelerometers and gyroscopes, the study achieved an accuracy of 99.26% with an Artificial Neural Network (ANN) classifier. The dataset comprised 9399 readings grouped into sequences of six, representing three different gestures. This high accuracy contrasts with earlier studies: Andrés & Marco (2017) used sEMG sensors and ANN classifiers for sign language, but faced challenges with overfitting and noisy EMG signals. Chun-Jen et al. (2017) combined computer-generated 3D hand images with real camera images for training, achieving a 77.08% accuracy for 24 hand gestures, while Deepali & Milind (2016) focused on alphabets and attained a 96.15% accuracy. Stefano et al. (2018) used a Leave One Subject Out (LOSO) cross-validation technique to avoid dataset correlation, achieving 92.87% accuracy. Compared to these studies, the current research's simpler gesture set and effective data handling using accelerometer and gyroscope data contributed to its higher accuracy, highlighting the importance of appropriate sensor choice, classifier efficiency, and robust data management techniques in overcoming common issues like overfitting and environmental noise

As technology has evolved, gesture recognition has become an important part of people's interaction with systems. This technology enables devices to capture and interpret gestures, which can be used to execute commands. The proposed device is called GANE which utilizes Cloud-based DL. This device recognizes gestures made with the cane through a cloud-based pipeline. GANE is a module that can be attached to a standard cane, and it uses a combination of sensors and DL to recognize gestures made with the cane. These gestures can be linked to specific actions on the user's smartphone, such as making phone calls, sending text messages, or checking notifications. Technical Implementation in this project - At the core of GANE is a deep-learning pipeline that analyzes sensor data captured from the cane to identify the user's gesture. The development of GANE has been an enriching journey that provided valuable insights into the challenges and opportunities of designing technology for elderly smartphone users. Throughout this project, there were several key lessons that are worth noting:

1. Gesture Recognition is a Viable Interaction Method: For elderly users, gesture recognition provides a promising alternative to traditional touchscreen interfaces, allowing them to interact with smartphones more intuitively and naturally.
2. Cloud-Based Deep Learning Enhances Performance: Cloud-based deep learning provides a powerful and scalable approach to gesture recognition, which offers continuous model updates and improved accuracy over time.

3. User-Centered Design is Crucial for Accessibility: Iterative design and user feedback are essential for developing accessible technologies that cater to the specific needs and preferences of elderly users. The successful development of GANE demonstrates the potential of technology to bridge the digital divide and empower the elderly to fully participate in the modern world.

7. Limitations

There are few potential limitations associated with utilizing cloud-based deep learning approaches. One significant constraint is the reliance on a stable network connection, given that computations are performed through the cloud. Therefore, in an external environment, it is imperative to possess a reliable internet connection. Additionally, concerns about privacy arise, particularly in relation to IoT devices. For this project, AWS IoT Core is employed to facilitate the transfer of data from the hardware device to the IoT core. It is noteworthy that data in transit via AWS IoT Core is encrypted using Transport Layer Security (TLS), ensuring secure communication. Furthermore, data at rest within AWS is encrypted using AWS-owned encryption keys, enhancing security measures.

8. Future Works

In the pipeline for future development, the app aims to:

1. Expand its dataset collection efforts, broadening the range of gestures recognized by the system. This will involve gathering diverse datasets to encompass a wider spectrum of gestures, ensuring the app remains adaptable to various user behaviors and preferences.

2. Introduce a dynamic mapping feature that allows users to interchange gestures with specific app functionalities. This customization option will empower users to tailor the app's interaction patterns according to their individual needs and preferences, enhancing user engagement and satisfaction.

3. Focus on enhancing the model's accuracy through iterative refinement processes and advanced machine learning techniques. By continuously optimizing the model's algorithms and fine-tuning its parameters, the app aims to achieve higher levels of accuracy in gesture prediction, thereby delivering a more seamless and intuitive user experience.

References

[1] Rung-Ching Chen, Christine Dewi, Wei-Wei Zhang, Jia-Ming Liu, and Su-Wen Huang. Integrating gesture control board and image recognition for gesture recognition based on deep learning. *International Journal of Applied Science and Engineering*, 17:237–248, September 2020.

[2] Pete Warden, Matthew Stewart, Brian Plancher, Colby Banbury, Shvetank Prakash, Emma Chen, Zain Asgar, Sachin Katti, and Vijay Janapa Reddi. Machine learning sensors. *Cornell University Archive*, 2022.

[3] Franz Pernkopf, Senior Member, Wolfgang Rothand Matthias Zohrer IEEE,Lukas Pfeifenberger, G unther Schindler, Holger Froning, Sebastian Tschiatschek, Robert Peharz, Matthew Mattina, and Zoubin Ghahramani.Efficient and robust machine learning for real-world systems.*Cornell University Archive*, 2018.

[4] Hui Han and Julien Siebert.Tinyml: A systematic review and synthesis of existing research.In *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 269–274, 2022.

[5] Shishir G. Patil, Don Kurian Dennis, Chirag Pabbaraju, Nadeem Shaheer, Harsha Vardhan Simhadri, Vivek Seshadri, Manik Varma, and Prateek Jain. Gesturepod: Enabling on-device gesture-based interaction for white cane users. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST ’19, page 403–415. Association for Computing Machinery, 2019.

[6] Aman Kumar Singh, Abdullah Ahmed Arifi, R Harikrishnan, Bhuvi Datta, and Shivali Amit Wagle. Iot based home automation using app aws. In *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pages 1–9, 2022.

[7] Sudip Chakraborty and Sreeramana Aithal. Let us create a lambda function for our iot device in the aws cloud using c. *International Journal of Management Technology and Social Sciences*, 8:145–155, 06 2023.

[8] Saloni Gandhi, Anuja Gore, Sakshi Nimbarte, and Jibi Abraham. Implementation and analysis of a serverless shared drive with aws lambda. In *2018 4th International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2018.

[9] Sudarshan S. Chawathe. Data modeling for a nosql database service. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON),* pages 0234–0240, 2019.

[10] Mais Yasen and Shaidah Jusoh. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 09 2019.