ISSN NO: 9726-001X
Volume 13 Issue 02 2025

Design and Implementation of Head Gesture Recognition System Using 6DOF Inertial Sensors forEnhanced Interaction Control

Mahesh. D[1], Uma Maheshwar Rao. A [2], Gangadhar. T [3], Anusha. CH [4]
[1,2,3] UG Scholar, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India-500100
[4]Assistant Professor, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India-500100
Maheshdevulapally01@gmail.com

Abstract:

Head gesture recognition has emerged as a critical area of research within human-computer interaction, driven by the evolution of sensor technology and the demand for natural, intuitive control systems. Historically, early systems relied on rule-based or thresholdbased methods to interpret head movements from inertial sensor data, but these traditional approaches often suffer from high sensitivity to noise, rigid feature extraction, and limited adaptability to complex or imbalanced datasets. These limitations underscore the problem definition: accurately and robustly recognizing head gestures in real-world scenarios remains challenging, particularly when sensor data are noisy or when gesture classes are unevenly distributed. Motivated by the need for enhanced accuracy and scalability, our proposed system integrates advanced data preprocessing techniques—including cleaning and SMOTE-based balancing—with a multi-model machine learning framework. This framework encompasses a baseline Perceptron classifier, an improved MLP classifier, and a hybrid model combining a deep neural network, i.e., DenseNet for feature extraction with a Random Forest classifier (RFC) for ensemble-based decision making. The system's modular architecture, further enhanced by a user-friendly Tkinter GUI, not only facilitates robust model training and real-time predictions but also demonstrates significant improvements over traditional methods in terms of performance metrics and overall reliability. This comprehensive approach highlights the significance of our work in advancing head gesture recognition, paving the way for practical applications in accessibility, automotive safety, and interactive control systems.

Keywords: Head Gesture Recognition, Human-Computer Interaction, Inertial Sensors, Machine Learning, Deep Learning, Feature Extraction, DenseNet, Random Forest Classifier, Perceptron, Multi-Layer Perceptron, SMOTE, Data Preprocessing, Imbalanced Data, Ensemble Learning, Real-Time Prediction, Tkinter GUI.

1.INTRODUCTION

Head gesture recognition has become an essential aspect of human-computer interaction (HCI). Early systems primarily used cameras and specialized sensors to track movements. However, with the introduction of MEMS technology, compact and cost-effective inertial sensors like accelerometers and gyroscopes enabled more reliable head movement tracking...

As industries embrace smart factories, there is a growing demand for intuitive and adaptive control methods to enhance automation and efficiency. Gesture-based remote control and teleoperation have become significant, requiring high accuracy and responsiveness for precise manipulator control in dynamic environments. Various sensor technologies, enabling real-time interpretation of gestures.

Advancements in machine learning have further enhanced gesture recognition. Techniques like neural networks and feature extraction allow for better interpretation of subtle head movements. SMOTE helps address data imbalances, improving model reliability.

These developments have transitioned gesture recognition from an experimental field to real-world applications. It is now used in assistive technology, virtual reality, and automotive safety systems.

2. LITERATURE SURVEY

First, we examine research cases using vision sensors, such as RGB cameras and Kinect. According to the study by C Nuzzi et al. The data of 5 classes collected through the RGB camera were classified with an accuracy of 92.6% using the R-CNN algorithm. However, they reported limitations, such as light reflection, boundary extraction of background and hand, and limited working area of camera FOV. To overcome these limitations of RGB cameras, W Fang et al. Proposed a gesture recognition method for 37 hand gestures with CNN and DCGAN. They collected 37 hand gestures under various environmental conditions using artificial light sources. D Jiang et al. Classified 24 hand gestures collected with Kinect with an accuracy of 93.63% through CNN. However, like the RGB sensor, limited FOV and low illuminance are also not allowed in the Kinect. Vision sensor-based gesture recognition methods are frequently employed for remote control of manipulators. One study utilized Kinect V2 and Open Pose to develop a real-time human–robot interaction framework for robot teaching through hand gestures, incorporating a background invariant robust hand gesture detector. The researchers employed a pre-trained state-of-the-art convolutional neural network (CNN), Inception V3, alongside the Open Sign dataset to classify 10 hand gestures. With 98.9% accuracy in hand gesture recognition, they demonstrated gesture-based telemanipulation using an RGB camera. However, this approach requires users to memorize perceivable gestures for the robot, and the vision sensor's depth range constrains its capabilities. In addition, the system has only been tested indoors and may struggle in bright light due to the resulting contrast in RGB images. In cases where Kinect's skeletal information is used, researchers have successfully controlled the speed and steering of a mobile robot and the position of a 5-axis manipulator. Nevertheless, performance and usability issues often arise in research using vision sensors, such as limited field of view (FOV), light reflection, occlusion, and illumination. As a result, this approach is considered nearly infeasible in industrial settings, where operators must stand and perform gestures while facing the monitor screen. EMG sensors limited controllable degrees of freedom (DOFs) and dependency on human kinematic models make them unsuitable for telemanipulation applications when used alone. Vogel combined sEMG with Vicon motion-capture camera systems to record EMG signals from the wrist and pose information to remotely control the DLR LWR-III manipulator and train machine learning models. Furthermore, to minimize occlusion effects in gesture-based telemanipulation using only Kinect, an approach that combined hand posture recognition based on sEMG-derived biofeedback information was introduced. In a study employing IMU and EMG sensors, six static hand motions were recognized and used to control a robot arm by mapping each motion to the corresponding robot arm movement. In the study of IMU-based gesture recognition, six hand gestures were recognized at an average accuracy of 81.6%, and the telemanipulation was achieved only with the predefined motion mapped for each gesture. In the study using the operator's skeletal kinematic model, omnidirectional manipulation was achieved by estimating the hand motion trajectory, even though challenges persisted in the uncertainties of pose estimation and differentiating between unintentional and intentional motions. However, in most cases of IMU-based motion recognition, if the operator's initial body alignment determined just after the sensor calibration does not hold, the accuracy of dynamic gesture recognition will drop drastically. In a study for human motion tracking using a set of wearable IMU sensors, they did not use a body-fixed reference frame, but an earth-fixed frame for calculating the joint position between the body segments with considering the reference method from the biomechanics domain. Thus, in this case, the time-variant body heading direction does not matter because the angle of the human body, an essential feature of the recognition model, is not less affected by the change of the body-heading orientation. To apply this method, the segment axes should be determined segment-by-segment through predefined joint movements, such as pronation–supination for the upper-limb joint and flexion–extension for the lower-limb joint. Moreover, the relation of segments with the global reference frame should be identified after estimating the relative pose of the sensor to the segment. Then, the joint position can be calculated by two connecting segments. It can be said that this method should be time-consuming and inconvenient, as the number of the joints of interest is increased. There has been another approach to securing consistent reference inertial measurement frames in IMU sensor-based human motion analysis of lower-limb and upper-limb

3. PROPOSED METHODOLOGY

The proposed technology for detecting health anomalies in senior citizens uses a smart wearable IoT device powered by an ESP32 microcontroller. The ESP32 is chosen for its low power consumption, Wi-Fi, and Bluetooth connectivity, making it ideal for continuous health monitoring. The device integrates sensors to track heart rate (ECG), blood pressure, temperature, SpO2, and movement (accelerometer for fall detection). The collected data is processed using noise filtering and anomaly detection algorithms. Basic threshold-based detection triggers alerts if values exceed safe limits, while machine learning techniques can identify patterns indicating conditions like arrhythmias or hypertension.

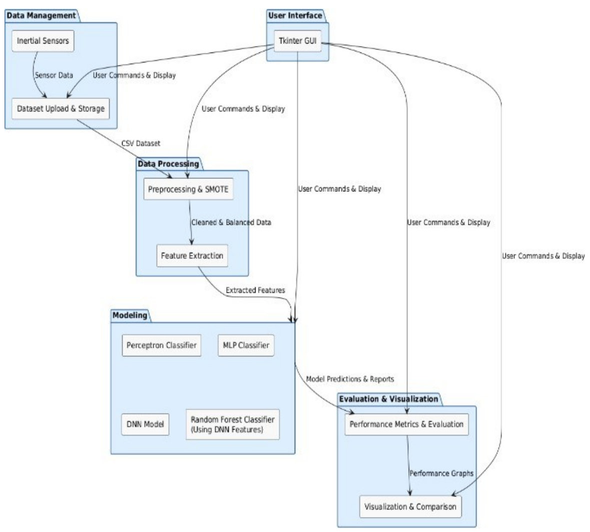

Figure 1: Proposed System

The proposed methodology typically includes the following key features:

- **DNN for Feature Extraction**: A deep neural network (DNN) learns hierarchical patterns from raw input data using multiple dense layers with ReLU activation. The final SoftMax layer is omitted for feature extraction.
- DNN **Architecture**: The network has four hidden layers—128, 64, 32, and 16 neurons—each refining the data. A fifth layer (8 neurons, SoftMax) is used only in standalone classification.
- **Feature Extraction Process**: Instead of using the final SoftMax output, features from the 16-neuron layer (or combined hidden layers) are extracted for further processing.
- **Random Forest Classifier (RFC)**: The extracted features are fed into an RFC, an ensemble learning method that improves prediction accuracy and robustness.
- **RFC Benefits**: By combining multiple decision trees, RFC reduces overfitting and handles noisy or imbalanced data better than using DNN alone.
- **Hybrid Model Advantage**: The combination of DNN's feature extraction and RFC's ensemble learning results in a more accurate and reliable head gesture recognition system.

Applications:

- **Assistive Technology** – Enables communication for individuals with motor disabilities or speech impairments.
- **Gaming and Virtual Reality** – Allows users to control and navigate immersive environments using head gestures.
- **Automotive Systems** – Enhances driver safety by detecting alertness and fatigue.
- **Smart Home and IoT** – Enables hands-free control of appliances, lighting, and other connected devices.
- **Healthcare** – Assists in rehabilitation and physical therapy by tracking head movements.
- **Robotics** – Improves human-robot interaction and enables intuitive control of robots and drone

Advantages:

The Smart Wearable IoT Device for Detecting Health Anomalies in Senior Citizens Using ESP32 offers several advantages, making it a valuable solution for various healthcare applications:

- **Enhanced Accessibility** – Provides an intuitive interface for users with mobility impairments.
- **Improved Human-Computer Interaction (HCI)** – Offers an alternative to traditional input methods like keyboards and touchscreens.
- **Robust Performance** – Uses advanced data preprocessing, SMOTE balancing, and a hybrid machine learning model for high accuracy.
- **Real-Time Processing** – Ensures fast and efficient recognition of head gestures.
- **Scalability and Adaptability** – Modular design allows easy integration into various applications.
- **Noise Resilience** – Combines DenseNet for feature extraction and Random Forest for classification to improve accuracy in noisy conditions
- **Hands-Free Operation** – Allows users to interact with systems without physical contact, making it ideal for environments where hands-free control is necessary, such as healthcare and industrial automation.
- **Energy Efficiency** – The system is designed to run efficiently on embedded platforms, making it suitable for portable and low-power devices

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

JOURNAL OF CURRENT SCIENCE

4. EXPERIMENTAL ANALYSIS

Fig. 2 shows the main graphical user interface (GUI) of the system as it launches. The interface is designed with a modern, user-friendly layout featuring a clear title and organized buttons for various operations such as dataset upload, preprocessing, model training, prediction, and visualization. The GUI provides a central console (text area) where system messages, logs, and outputs are displayed, allowing users to easily follow the workflow and interact with the system.

In Fig.3, the GUI displays the outcome immediately after a dataset has been uploaded. The file path or name is shown in the text console along with the first few rows (a preview) of the CSV dataset. This visual confirmation ensures that the correct file is loaded and that the data (comprising sensor readings and gesture labels) is ready for further preprocessing.

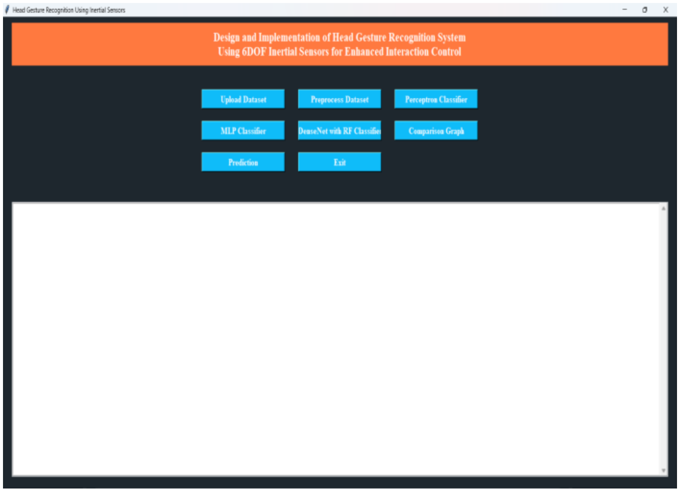

Fig.2: GUI of proposed head gesture recognition system.

Fig.3: GUI of proposed head gesture recognition system after uploading the dataset.

Fig. 4 presents a count plot generated using the dataset's gesture labels (referred to as "Miscare" categories). The plot visually represents the distribution of different gestures within the dataset. Each bar corresponds to a specific gesture (e.g., "MoveRight_2s", "MoveLeft_2s"), with the bar height indicating the frequency of occurrence. This visualization is crucial for identifying any class imbalances that might affect model performance and is typically generated after the preprocessing stage.

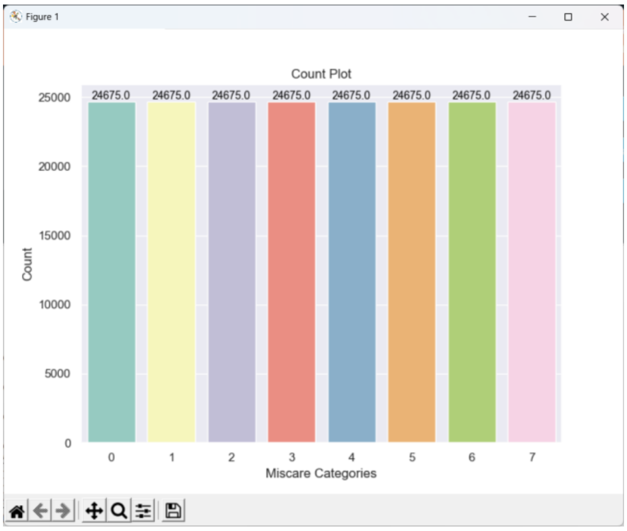

Fig. 4: Count plot vs miscare categories.

Fig. 5: GUI of proposed head gesture recognition system after performing preprocessing operation.

After data preprocessing—which includes cleaning missing values, normalization, and balancing using SMOTE—the GUI updates to show detailed preprocessing logs and information. Fig. 5 illustrates that the dataset has been refined, with key statistics (such as the dimensions of the training and testing sets) and a preview of the transformed data displayed in the text console. It confirms that the data is now better suited for effective feature extraction and model training.

Fig. 6 depicts the GUI after the system has trained the Perceptron classifier. The text console displays messages indicating that the model has either been loaded (if previously saved) or trained from scratch. It also shows training results, including performance metrics like accuracy, precision, recall, and F1-score, providing immediate feedback on the classifier's performance on the training and testing data.

The confusion matrix generated for the Perceptron classifier is shown in Fig.7 The matrix is typically rendered as a heatmap where rows represent the actual gesture classes and columns represent the predicted classes. The numbers in each cell highlight the count of correct and misclassified instances, offering a detailed view of where the classifier may be confusing one gesture for another.

Fig. 8 shows the Receiver Operating Characteristic (ROC) curve for the Perceptron classifier. The ROC curve plots the true positive rate against the false positive rate at various threshold settings. It provides an aggregate measure of performance across all classification thresholds, often accompanied by the area under the curve (AUC) value. This graph is useful for assessing the classifier's discriminative ability.

ISSN NO: 9726-001X

Volume 13 Issue 02 2025

In Fig. 9, the GUI is updated to reflect the results of training an MLP classifier. Similar to the Perceptron training GUI, the text area displays logs and performance metrics for the MLP model. This section highlights improvements in handling non-linear patterns in the data as compared to the simpler Perceptron model, indicating the benefits of the hidden layers in the MLP.

The confusion matrix for the MLP classifier is depicted in Fig.10. As with the Perceptron, this heatmap shows the distribution of correct and incorrect predictions across all gesture classes. A comparison with Fig.7 can illustrate whether the MLP reduces misclassifications and improves overall accuracy in recognizing head gestures.

Fig. 11 illustrates the performance of the MLP classifier. By comparing the ROC curves (and corresponding AUC values) of the MLP with those of the Perceptron (Fig.8), one can evaluate the relative improvement in the model's ability to distinguish between different gesture classes.

Fig. 6: GUI of proposed head gesture recognition system after applying model building and training using perceptron classifier.

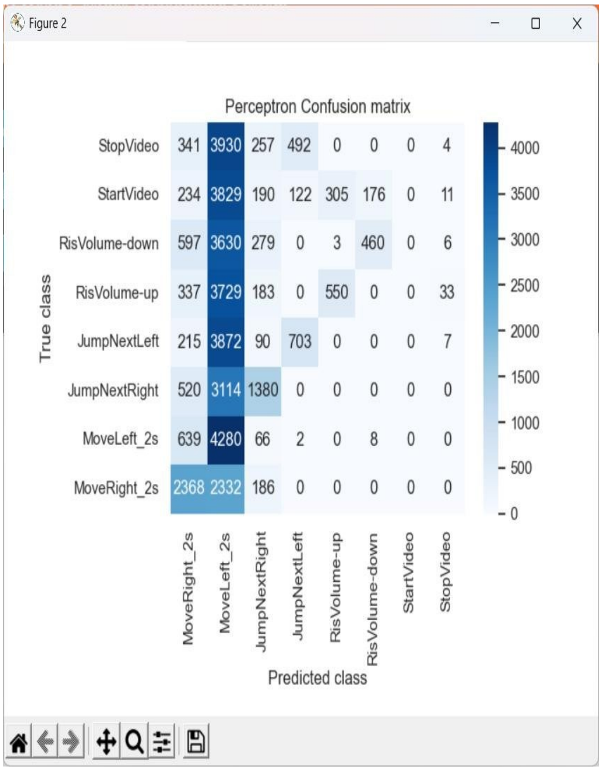

Fig. 7: Confusion matrix of perceptron classifier.

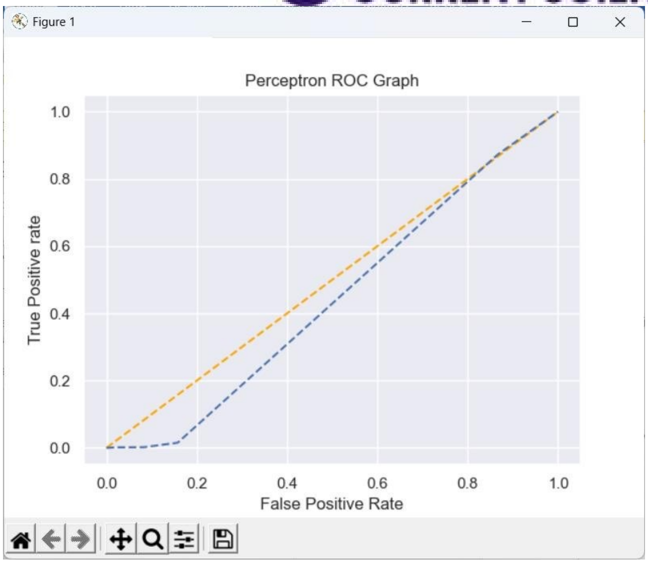

Fig. 8: ROC graph obtained using perceptron classifier.

Fig. 9: GUI of proposed head gesture recognition system after applying model building and training using MLP classifier.

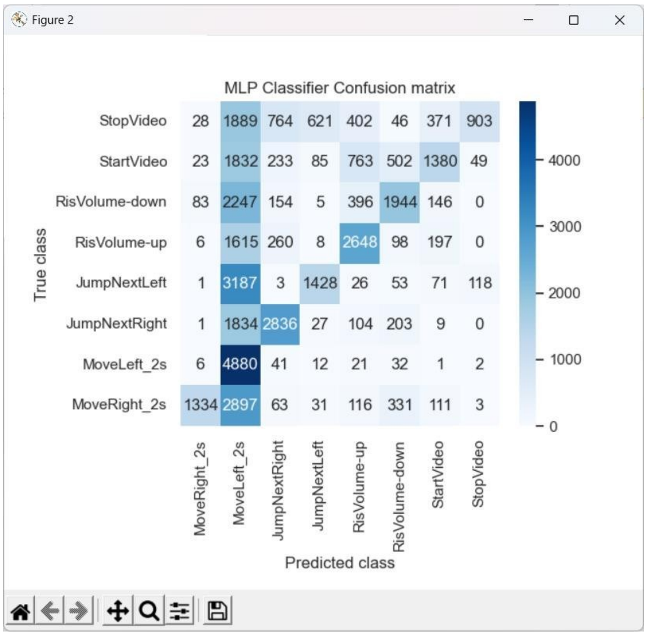

Fig. 10: Confusion matrix obtained using MLP classifier.

ISSN NO: 9726-001X
Volume 13 Issue 02 2025

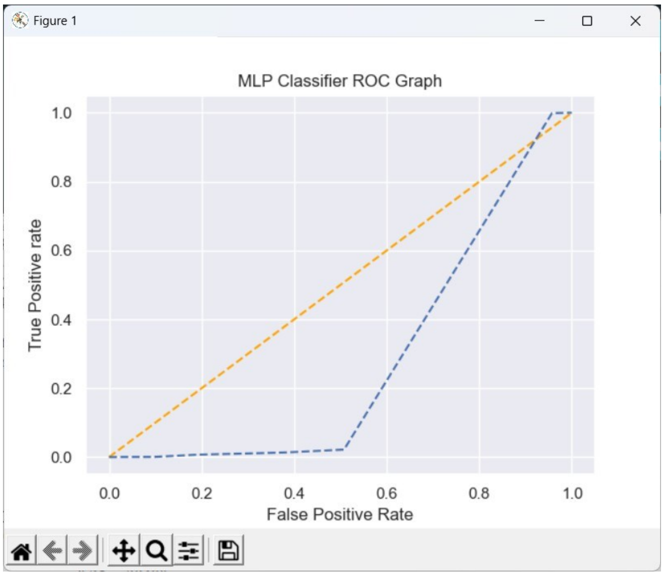

Fig. 11: ROC graph obtained using MLP classifier.

Fig. 12: GUI of proposed head gesture recognition system after applying model building and training using DenseNet with RFC model.

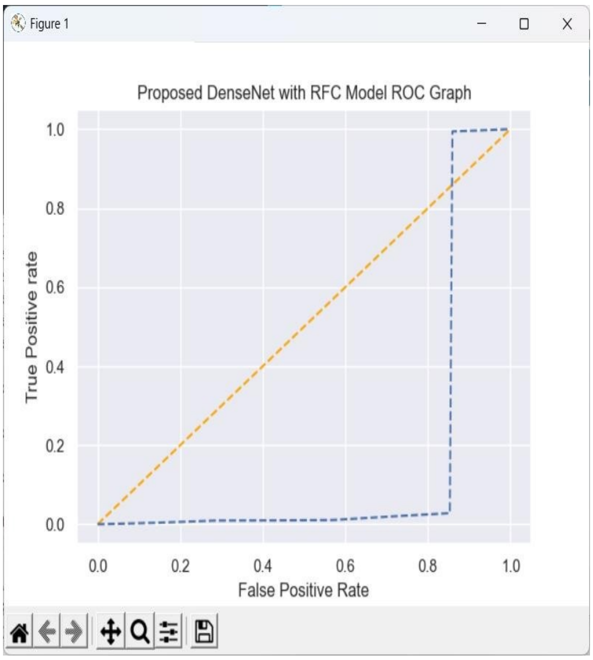

Fig. 13: ROC graph obtained using proposed DenseNet with RFC model.

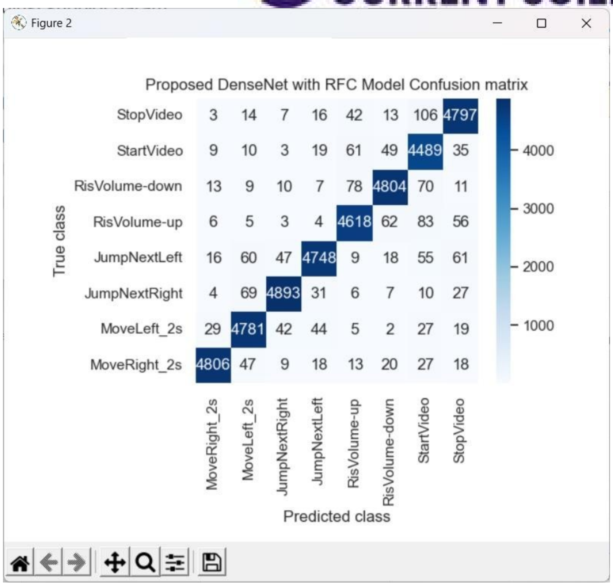

Fig. 14: Confusion matrix obtained using proposed DenseNEt with RFC model.

Fig. 15: Performance comparison graph of existing perceptron, MLP, and proposed DenseNet with RFC models.

Fig. 16: Sample predictions on test data using proposed DenseNet with RFC model.

Fig. 12 presents the GUI after the more advanced DenseNet with Random Forest Classifier (RFC) model has been trained. The GUI shows logs related to the deep neural network's training process and the subsequent extraction of features. It then displays performance metrics obtained after integrating the RFC, indicating that the hybrid model has been executed successfully and is ready to generate predictions.

Fig. 13 depicts ROC graph, the performance of the proposed DenseNet with RFC model is visualized. The curve and the corresponding AUC value reflect the model's enhanced capability in classifying head gestures by leveraging both deep feature extraction and ensemble learning. The graph typically shows a steeper curve and higher AUC, demonstrating improved discriminative power.

Fig. 14 demonstrate the confusion matrix, presented as a heatmap, details the performance of the DenseNet with RFC model. It provides insights into the model's prediction accuracy across different gesture categories, highlighting areas of strength and potential confusion. The matrix is expected to show fewer misclassifications compared to the simpler models, emphasizing the benefits of the hybrid approach.

Fig.15 consolidates the performance metrics of all three models into a single comparative graph. It typically includes bar charts or line graphs that display metrics such as accuracy, precision, recall, and F1-score for each model. This visual comparison clearly shows the incremental improvements gained by using more complex architectures, with the DenseNet with RFC model usually outperforming the Perceptron and MLP classifiers.

Fig. 16 shows example predictions made on test data by the DenseNet with RFC model. This may include a table or a visual overlay on the test dataset, where the actual gesture labels are compared with the model's predictions. It serves as a practical demonstration of the model's real-world applicability, validating the system's ability to correctly classify new, unseen head gesture data.

Algorithm Name	Accuracy	Precision	Recall	f1-score
Perceptron Classifier	24.683384 %	38.479845 %	24.595878 %	20.841534 %
MLP Classifier	43.953901 %	63.451410 %	43.873357 %	44.213627 %
DenseNet with RFC Model	96.134752 %	96.128814 %	96.131997 %	96.122536 %

Table 1: Summarizing the performance metrics for the three models.

- **Accuracy:** This metric indicates the overall percentage of correct predictions out of all predictions made by the model. For example, the Proposed DenseNet+RFC Model achieved an accuracy of approximately 96.13%, meaning it correctly classified 96.13% of the input samples. In contrast, the simpler Perceptron classifier had a much lower accuracy of around 24.68%, suggesting it struggles with the complexity of the head gesture data.

- **Precision:** Precision measures the proportion of positive identifications that were actually correct. A high precision value (such as the 96.13% for the DenseNet+RFC model) indicates that when the model predicts a certain gesture, it is highly likely to be correct. The Perceptron, with a precision of about 38.48%, shows a higher rate of false positives compared to the more sophisticated models.

- **Recall:** Recall (or sensitivity) quantifies the proportion of actual positives that were correctly identified. The DenseNet+RFC model's recall of 96.13% means that it successfully captures nearly all instances of each gesture class, while the lower recall values of the Perceptron and MLP models indicate they are missing a significant number of true gesture instances.

- **f1-score:** The f1-score is the harmonic mean of precision and recall, providing a balance between the two. It is particularly useful when dealing with imbalanced classes. The DenseNet+RFC model's f1-score of 96.12% reflects its superior balance between correctly predicting the gesture classes and minimizing false negatives and positives. In contrast, the Perceptron's f1-score of around 20.84% highlights its overall poor performance in this regard.

Overall, the table and corresponding explanation demonstrate how the proposed DenseNet+RFC model significantly outperforms both the Perceptron and MLP classifiers, offering much higher accuracy, precision, recall, and f1-score. This underscores the advantage of using a hybrid deep learning and ensemble approach for head gesture recognition.

5. CONCLUSION

In conclusion, the proposed head gesture recognition system has demonstrated significant improvements over traditional methods by effectively integrating advanced data preprocessing, feature extraction, and a multi-model machine learning framework within a user-friendly GUI. The system processes inertial sensor data through rigorous cleaning and balancing steps, enabling robust feature extraction and model training. Comparative experiments reveal that while a simple Perceptron classifier achieved an accuracy of around 24.68% and an MLP classifier reached approximately 43.95%, the hybrid DenseNet with Random Forest Classifier outperformed both by attaining an accuracy of about 96.13%, with corresponding high precision, recall, and f1-scores near 96%. These results underscore the hybrid model's ability to capture complex, non-linear patterns in the data and significantly reduce misclassifications, thereby proving its efficacy in accurately recognizing head gestures. This work not only addresses the limitations of traditional approaches by enhancing overall system robustness and reliability but also paves the way for practical applications in interactive control, accessibility, and automotive safety, making it a promising solution for real-world deployment.

The future scope includes integrating the system with real-time embedded platforms and exploring more advanced deep learning models to further enhance accuracy and robustness. Additionally, adapting the approach to accommodate a broader range of gestures and multi-modal sensor data could open new applications in interactive and assistive technologies.

REFERENCES

[1] Kumar, N.; Lee, S.C. Human-machine interface in smart factory: A systematic literature review. *Technol. Forecast. Soc. Chang.* 2022, *174*, 121284.

[2] Nuzzi, C.; Pasinetti, S. Lancini, M.; Docchio, F.; Sansoni, G. Deep learning-based hand gesture recognition for collaborative robots. *IEEE Instrum. Meas. Mag.* 2019, *22*, 44–51.

[3] Fang, W.; Ding, Y.; Zhang, F.; Sheng, J. Gesture recognition based on CNN and DCGAN for calculation and text output. *IEEE Access* 2019, *7*, 28230–28237.

[4] Jiang, D.; Li, G.; Sun, Y.; Kong, J.; Tao, B. Gesture recognition based on skeletonization algorithm and CNN with ASL database. *Multimedia Tools Appl.* 2019, *78*, 29953–29970.

[5] Suarez, J.; Murphy, R.R. Hand gesture recognition with depth images: A review. In Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, Paris, France, 13 September 2012; pp. 411–417.

[6] Mazhar, O.; Navarro, B.; Ramdani, S.; Passama, R.; Cherubini, A. A real-time human-robot interaction framework with robust

JOURNAL OF CURRENT SCIENCE

background invariant hand gesture detection. *Robot. Comput. Manuf.* 2019, *60*, 34–48.
[7] CMU-Perceptual-Computing-Lab/Openpose. GitHub. Available online: https://github.com/CMU-Perceptual-Computing-Lab/openpose (accessed on 13 Jan 2025).
[8] Zhou, D.; Shi, M.; Chao, F.; Lin, C.M.; Yang, L.; Shang, C.; Zhou, C. Use of human gestures for controlling a mobile robot via adaptive cmac network and fuzzy logic controller. *Neurocomputing* 2018, *282*, 218–231.
[9] Bouteraa, Y.; Ben Abdallah, I.; Ghommam, J. Task-space region-reaching control for medical robot manipulator. *Comput. Electr. Eng.* 2018, *67*, 629–645.
[10] Vogel, J.; Castellini, C.; van der Smagt, P. EMG-based teleoperation and manipulation with the DLR LWR-III. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011.
[11] Bouteraa, Y.; Ben Abdallah, I. A gesture-based telemanipulation control for a robotic arm with biofeed-back-based grasp. *Ind. Robot. Int. J.* 2017, *44*, 575–587.
[12] Chico, A.; Cruz, P.J.; Vasconez, J.P.; Benalcazar, M.E.; Alvarez, R.; Barona, L.; Valdivieso, A.L. Hand Gesture Recognition and Tracking Control for a Virtual UR5 Robot Manipulator. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 12–15 October 2021.
[13] Kulkarni, P.V.; Illing, B.; Gaspers, B.; Brüggemann, B.; Schulz, D. Mobile manipulator control through gesture recognition using IMUs and Online Lazy Neighborhood Graph search. *Acta IMEKO* 2019, *8*, 3–8.
[14] Shintemirov, A.; Taunyazov, T.; Omarali, B.; Nurbayeva, A.; Kim, A.; Bukeyev, A.; Rubagotti, M. An open-source 7-DOF wireless human arm motion-tracking system for use in robotics research. *Sensors* 2020, *20*, 3082.
[15] Roetenberg, D.; Luinge, H.; Slycke, P. *Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors*; Technical Report; Xsens Motion Technologies B.V.: Enschede, The Netherlands, 2013; pp. 1–7.
[16] Luinge, H.; Veltink, P.; Baten, C. Ambulatory measurement of arm orientation. *J. Biomech.* 2007, *40*, 78–85.
[17] Van der Helm, F.C.T.; Pronk, G.M. Three-dimensional recording and description of motions of the shoulder mechanism. *J. Biomech. Eng.* 1995, *117*, 27–40.
[18] Zhang, J.-T.; Novak, A.C.; Brouwer, B.; Li, Q. Concurrent validation of Xsens MVN measurement of lower limb joint angular kinematics. *Physiol. Meas.* 2013, *34*, N63–N69.
[19] Kim, M.; Lee, D. Wearable inertial sensor based parametric calibration of lower-limb kinematics. *Sens. Actuators A Phys.* 2017, *265*, 280–296.
[20] Jeon, H.; Choi, H.; Noh, D.; Kim, T.; Lee, D. Wearable Inertial Sensor-Based Hand-Guiding Gestures Recognition Method Robust to Significant Changes in the Body-Alignment of Subject. *Mathematics* 2022, *10*, 4753.
[21] Yuan, Q.; Chen, I.-M. Human velocity and dynamic behavior tracking method for inertial capture system. *Sens. Actuators A Phys.* 2012, *183*, 123–131.
[22] Lynch, K.M.; Park, F.C. *Modern Robotics*; Cambridge University Press: Cambridge, UK, 2017.
[23] Yoo, M.; Na, Y.; Song, H.; Kim, G.; Yun, J.; Kim, S.; Moon, C.; Jo, K. Motion estimation and hand gesture recognition-based human–UAV interaction approach in real time. *Sensors* 2022, *22*, 2513.
[24] Chamorro, S.; Jack, C.; François, G. Neural network based lidar gesture recognition for real-time robot teleoperation. In Proceedings of the 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), New York City, NY, USA, 25–27 October 2021.
[25] Se-Yun, J.; Eun-Su Kim, E.-S.; Park, B.Y. CNN-based hand gesture recognition method for teleoperation control of industrial robot. *IEMEK J. Embed. Syst. Appl.* 2021, *16*, 65–72.
[26] Kim, E.; Shin, J.; Kwon, Y.; Park, B. EMG-Based Dynamic Hand Gesture Recognition Using Edge A.I. for Human–Robot Interac-tion. *Electronics* 2023, *12*, 1541.
[27] Cruz, P.J.; Vásconez, J.P.; Romero, R.; Chico, A.; Benalcázar, M.E.; Álvarez, R.; López, L.I.B.; Caraguay, L.V. A Deep Q-Network based hand gesture recognition system for control of robotic platforms. *Sci. Rep.* 2023, *13*, 7956.