HAND GESTURE DRIVEN SMART HOME AUTOMATION LEVERAGING INTERNET OF THINGS

Dhananjay, Kumar[1]; Sowbarnigaa, Kogilavani Shanmugavadivel[1]; Mehal Sakthi, Muthusamy Sivaraja[1]

[1]Department of Information Technology, Anna University, MIT Campus, Chennai, India

ABSTRACT

Smart home automation systems require convenient and efficient user interface to control home appliances. Gesture recognition-based solutions offer flexibility to the users and play a crucial role in advancing human-computer interaction and immersive computing environments. This work proposes a novel solution leveraging deep learning techniques with attention mechanisms including self-attention tailored for processing 3D tensors derived from the gesture images. A set of hand gestures is defined, and the system is trained and optimized to meet the real time requirements in controlling devices. To improve the accuracy, the model is parallelly trained with dynamic learning to adaptively fuse with the classification module. The proposed modular architecture is implemented using Raspberry Pi with IoT devices for a typical home environment. The test result achieves gesture classification accuracy of 98.24% and latency of about 0.2 seconds in real time control. The working model highlights a practical solution under ITU-T Recommendation J.1611 which deals with the functional requirements of a smart home and gateway.

Keywords – Gesture recognition, Smart Home Automation, Internet of Things, Attention mechanism

1. INTRODUCTION

Smart homes blend IoT devices and automation for seamless living. Gestural control represents an intuitive interface, enabling hands-free operation of devices and services within the smart home environment. By interpreting hand movements, gesture recognition systems trigger the operations of appliances. Moreover, IoT technology facilitates seamless communication among devices, creating a cohesive ecosystem where gestures drive home automation.

Gesture recognition offers a compelling solution for controlling devices in smart homes due to its natural and intuitive interface. Unlike traditional methods such as voice or remotes, gesture control allows users to interact with their environment using natural hand movements, eliminating the need for physical touch or voice commands. Additionally, gesture recognition provides a discreet and non-intrusive convenient way to interact with devices, even in noisy environments. The preferences of gesture-based automation over voice-based techniques results from the technical as well as user convenience in controlling home appliances. However, the solution needs to cater to different functional requirements widely used by everyone irrespective of their abilities and disabilities. Overall, the system design goal orients to enhance the user experience in smart homes by offering a user-friendly, customizable, and versatile method of device control.

Existing state-of-art systems like Smartify [1] present home automation by accessing devices via mobile phones and voice capturing technologies. However, voice command-based systems may struggle to distinguish commands accurately amidst background noise / music, leading to errors or misinterpretations. The proposed gesture recognition system focuses solely on hand movements, eliminating the influence of ambient sounds. This ensures precise and reliable control of appliances, even in noisy environments. Other state-of-art systems like the Fibaro [2], primarily utilizes a single gesture called swipe to control devices. However, this necessitates the placement of multiple hardware units across various locations within the same room for comprehensive device control. In contrast, our proposed solution extends beyond single gestures, offering a diverse range of gestures for intuitive device management. Crucially, it eliminates the need for several handheld hardware units by enabling the control of multiple devices from a single location. This enhances user experience and convenience, streamlining smart home interactions without compromising functionality or accessibility.

The conventional approaches to gesture classification often rely on 2D images, limiting their ability to capture the depth and spatial dynamics inherent in human gestures. This limitation underscores the need for a more sophisticated approach, prompting the exploration of 3D tensor representations derived from images. The existing research works [3-5] on gesture recognition utilizes deep learning models with Convolutional Neural Networks. The proposed model uses a deep learning model with attention mechanism and transfer learning model with dynamic learning rate which improves the accuracy and reduces latency to a greater extent.

The ITU-T Recommendation J.1612 [6] outlines technical specifications for efficient smart home device management within IoT ecosystems. The protocols and standards for device discovery, configuration, and maintenance ensure seamless integration and interoperability. Apart from the security measures, the system development needs to address scalability and adaptability, allowing for the integration of new devices and services over time. A standardized framework for smart home device management promotes efficiency, reliability, and security in IoT-driven home automation environments. The proposed work conforms to the ITU-T Recommendations J1611 [7] and J1612 [6], while taking device management and IoT into consideration. Further, in a standardized system development approach, the gateway hardware with a driver and operating system serves as a basic software platform to manage all hardware resources. In proposed solution, a machine learning based method deployed on minicomputer like Raspberry Pi supporting IoT devices facilitates understanding and execution end user's command in real-time.

The proposed work introduces a novel solution by employing a dedicated CNN model with self and other attention mechanisms, specifically tailored for processing 3D tensors derived from the images. Beyond the core classification challenge, the solution addresses additional complexities associated with real-time gesture recognition devices. Integrating the entire pipeline, from real-time image capture to 3D tensor generation and classification, requires careful consideration of computational efficiency and system responsiveness. The inherent complexity of human gestures information [8], allowing for precise 3D spatial structure capture and accurate regression of hand poses, poses a challenge to conventional image-based classification systems. Moreover, issues such as lighting conditions, background noise, and varying user positions [9] add to the work's intricacy. The system development using existing machine learning model like transfer learning model ResNet with a dynamic learning rate tries to enhance the accuracy and robustness of gesture recognition systems, enabling seamless and natural interactions between end users and IoT enabled devices.

2. PROPOSED WORK

The proposed work entails a comprehensive approach to hand gesture recognition for smart home appliance control as shown in Figure 1. Initially, the camera feed undergoes preprocessing to enhance the quality of the frames extracted for analysis. This preprocessing step includes noise reduction, image normalization, and potential background subtraction to isolate the hand region, crucial for gesture recognition. Following preprocessing, the region of interest, typically the hand palm, is segmented from the background. This segmentation step is pivotal for focusing the analysis on relevant features for gesture classification. By isolating the hand region, the subsequent models concentrate on discerning the nuances of hand movements with greater accuracy. The segmented hand palm region is then passed through two distinct modules for gesture classification. The first module employs an attention-based CNN model, which dynamically focuses on salient features within the hand region. This attention mechanism enhances the model's ability to capture subtle variations in hand gestures, improving classification performance.

In the development of our attention mechanism, the Self-Attention Input (SAI) layer plays a crucial role by decomposing the feature representations into Value, Key, and Query components, operating in the format of (batch size, channel number, height, width). This layer employs batch matrix multiplication to compute attention scores, which enables the model to selectively focus on relevant spatial-temporal features. The attention-driven approach allows for more nuanced understanding by emphasizing key features within the spatial-temporal context of the hand gestures, thus improving the accuracy of gesture recognition.

Following the attention score computation, the system delves into meticulous refinement processes. Attention UV (Att UV) and Attention Others (Att Others) strategically process UV and other points extracted from the Self-Attention Output (SAO) layer, ensuring that the system homes in on critical spatial-temporal features. This attention-driven refinement is pivotal in preparing the features for subsequent stages, facilitating a more precise and context-aware understanding of gestures. It establishes a robust foundation for the feature pooling stage, ensuring that the system captures and processes intricate details essential for classification module.

The hand region of interest is simultaneously processed by a transfer learning model with dynamic learning rate. This model adjusts its learning rate dynamically based on the characteristics of the input data, optimizing the training process for improved performance. By incorporating dynamic learning rate mechanisms, the model effectively adapt to variations in gesture dynamics and environmental conditions. Unlike traditional approaches that extract positional parameters before inputting them into learning models, the proposed system directly utilizes the image data. This approach offers several advantages, including simplifying the preprocessing stage and reducing computational complexity. By feeding image data into the learning model, the system preserves the spatial information inherent in gestures, allowing for more accurate classification.

Finally, the results from both the models (attention and transfer learning) are fused using a class probability fusion technique. This fusion process intelligently combines the outputs from the attention-based CNN model and dynamically learning transfer learning model to produce a more robust classification outcome. The merging of predictions from two parallel channels stands out as a critical component in the proposed gesture recognition process. After localizing the gestures, one channel processes the image data directly, while the other extracts positional parameters and employs a CNN neural network for classification. The predictions from these channels integrated using a fusion mechanism allows the system to combine the insights gained from both approaches. Upon successful gesture classification, the system triggers corresponding operations of home appliances via a relay connected to a Raspberry Pi, thereby controlling their functions through hand gestures.

The algorithm for tensor extraction takes the frames of video as input and provides 16 3D tensors as output.

Input: ICVL Hand Gesture Dataset with depth images
Output: Sixteen 3D tensor points
1. Begin
2. Load the ICVL Hand Gesture Dataset.
3. Apply median filtering as a part of data preprocessing.
4. Ensure uniform sequence lengths through padding or truncation
5. Apply 2D attention along with self-attention to the model to get the feature tensors.
5: Add dense layers to model
6. Train the model with train dataset samples.
7. Finetune the model with error rate and loss in extracting the feature tensors
8. End

The algorithm for gesture classification takes the tensors as input and applies a CNN model for classifying gestures. In addition, transfer learning model is trained with dynamic learning rate and a probabilistic fusion is done to identify the gesture class.

Input: Sixteen 3D tensors
Output: Gesture predicted
1. Begin
2. Initialize a Sequential Model.
3. Add convolutional and pooling layers to model:
3.1: *Conv3D* layer with parameters (filters=f, kernel_size=k, padding=p):
Perform convolution: H_i = activation (Conv($H\_\{i\text{-}1\}$; f, k, p)) where $H\_0 = H$.
3.2: For each *MaxPooling* 3D layer with parameters (pool_size=s, padding=p): Apply pooling: H_i = MaxPool($H\_\{i\text{-}1\}$; s, p).
3.3: Apply the *Conv3D* layer and *MaxPooling3D* layer twice iteratively.
4: Add flatten layer to model
4.1: Flatten the output tensor *H_final* from the last Conv3D or *MaxPooling3D* layer into a 1D tensor.

5.1: For each Dense layer with units u and activation a: Compute dense output: $P(i)$= a(Dense($H\_\{i\text{-}1\}$; u)) where $H\_\{i\text{-}1\} = H\_final$
6: Applying Resnet with dynamic learning rate to provide output: $P(j)$
7. Add the output layer to model and compute the class

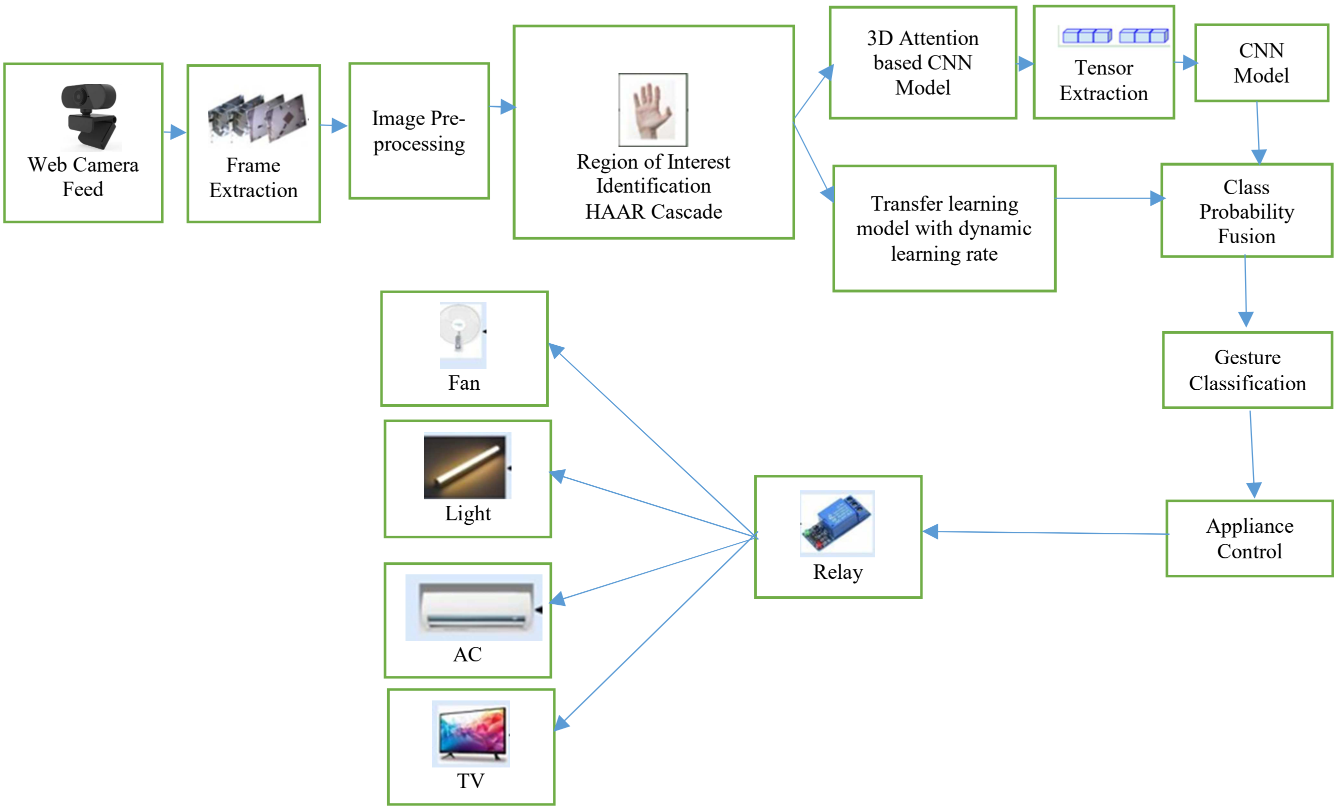

Figure 1 - Workflow

3. IMPLEMENTATION WITH EXPERIMENTAL DETAILS

3.1 Dataset and preprocessing

The ICVL Hand Gesture Dataset, comprising depth images from Intel, with 22069 training frames and 1600 testing frames, focusing on 16 hand joints, was used in training / testing. Another dataset called Hand Gesture Recognition Dataset [10] containing total 24000 images of 20 different gestures was employed. For training purpose, there are 900 images in each directory and for testing purpose there are 300 images in each directory. Table 1 shows sample images from the dataset.

Table 1 – Dataset Sample Images

Class Label	Gesture Image	Class Label	Gesture Image	Class Label	Gesture Image
0		1		2	
3		4		5	
6		7		8	
9		10		11	
12		13		14	

In the preprocessing stage for gestures, the first step involves resizing the input images to a fixed size. This standardization ensures uniformity in the input data fed into the model, reducing computational complexity and memory requirements during both training and inference phases. Following this, normalization techniques are applied to scale the pixel values of the images to a standardized range, typically [0, 1]. This normalization stabilizes the training process and aids convergence by ensuring that the input data has a consistent scale, facilitating effective learning by the model. Next, noise reduction techniques are applied to mitigate the impact of noise and artifacts present in the input images. Median filter method is used to smooth out irregularities and enhance image clarity. In a typical home environment, noise reduction helps in improving the quality of the input data, making it easier for the model to extract relevant features and patterns associated with different hand gestures.

Finally, data augmentation techniques are applied to increase the diversity and robustness of the training dataset. Augmentation methods such as rotation, scaling, translation, and flipping introduce variations to the training data, enabling the model to generalize better to unseen variations in hand gestures and environmental conditions. Additionally, cropping the Region of Interest (ROI), such as the hand palm, from the input images helps in focusing the model's attention on the relevant area for gesture classification. By sequentially applying these preprocessing methods, the input images are effectively prepared for feature extraction and classification by the corresponding modules of the gesture recognition system.

3.2 Attention based CNN model

The model configuration parameters, including learning rate, batch size, and epochs, were established. The data was organized into separate training and testing sets. To ensure uniform sequence lengths, padding or truncation methods were implemented. Following this, a 3D attention-based CNN model is applied to extract the feature tensors. After the 3D attention-based CNN model processed the data, it produced a set of 16 3D vectors that showed how the hand was positioned in space. These contained information about where each part of the hand was, like the fingers and the center of the palm, and how they were moving.

The array of 3D vectors served as the foundational input for the subsequent model, which was specifically designed to discern and classify the intricate hand gestures. By delving into the minute variations and spatial dynamics encoded within these vectors, the model was able to discern complex patterns and correlations unique to each gesture. It comprehensively analyzed the interrelation between different joints and their spatial orientations, allowing for the accurate identification and classification of diverse hand gestures. With its robust analytical framework, the model effectively discerned the subtle differentiations between gestures, considering the relative positioning, movement trajectories, and spatial interactions between the hand joints.

At the core of the enhanced architecture lies an intricate feature extraction layer with attention mechanism, seamlessly integrating Convolutional 2D, Batch Normalization, ReLU, Residual, and MaxPooling 2D operations. This amalgamation is meticulously crafted to capture nuanced spatial-temporal patterns, providing a robust foundation for subsequent processing. This advanced layer synergistically leverages the power of Convolutional 2D operations to detect hierarchical features, Batch Normalization for stabilizing and accelerating training, ReLU for introducing non-linearity, Residual connections for overcoming vanishing gradient issues, and MaxPooling 2D for down-sampling and preserving essential information. The collaborative effect of these operations enhances the model's capacity to discern intricate gesture nuances.

Recognizing the need for a more comprehensive dataset, both images and corresponding 16-point hand representations, the work seamlessly transitioned to the Leap Gesture Dataset. This dataset enriches the training data with crucial visual information, forming the cornerstone for a robust hand gesture recognition model. The decision to integrate the Leap Gesture Dataset was guided by the understanding that a holistic approach, combining image data and key hand points, is vital for training the CNN model to get the class labels as intermediate output.

In the context of our gesture classification module, the system design goal lies in constructing and training a Convolutional Neural Network (CNN) model for effective hand gesture recognition. The implementation process commences by loading preprocessed data encompassing 3D pose information and their corresponding coarse labels for various gestures. The model consists of convolutional layers, max-pooling layers, fully connected layers, ReLU activation functions, and L2 regularization to prevent overfitting. The model is subsequently compiled with the Adam optimizer and a sparse categorical cross entropy loss function, making it ready for training. The training process ensues, involving 30 epochs and monitoring performance against a validation set to ensure generalization. Early stopping is used to stop the model training when overfitting occurs.

3.3 Transfer learning with dynamic learning rate

In the proposed system, once the region of interest is identified, it is passed to the next module, which employs a dynamic learning rate adjustable ResNet-based transfer learning model for gesture classification. The ResNet (Residual Network) architectures are renowned for their ability to effectively train deep neural networks, even with a large number of layers. By leveraging transfer learning, the system capitalizes on pre-trained ResNet models, fine-tuning them to recognize hand gestures specific to the application. This approach significantly reduces the training time and computational resources required to achieve high classification accuracy.

One key aspect of the model is its dynamic learning rate adjustment mechanism. The system employs a piecewise learn rate schedule with a learn rate drop factor of 0.2. This factor allows for the systematic reduction of the learning rate during training, thereby enabling the model to converge more effectively. The learn rate drop period is set to 1 epoch, ensuring that the learning rate is updated at the end of each training epoch. Additionally, the initial learn rate is set to 1×10^{-4}, providing an appropriate starting point for the training process.

The dynamic adjustment of the learning rate is crucial for optimizing the training process and improving the model's performance over time. By gradually decreasing the learning rate as training progresses, the system prevents the model from getting stuck in local minima and facilitates smoother convergence towards the global optimum. This dynamic learning rate strategy ensures that the model can effectively adapt to the complexities of the gesture recognition task, ultimately leading to more accurate classification results.

Finally, the output of the ResNet-based transfer learning model is used for gesture classification, where gestures are categorized into n classes based on their visual representations. This classification process is essential for enabling the system to accurately interpret and respond to user gestures, thereby facilitating intuitive and seamless interaction within the smart home environment. Through the integration of advanced learning techniques and dynamic learning rate adjustment, the system achieves robust and efficient gesture recognition capabilities, enhancing user experience and system performance.

3.4 Class Probability Fusion and Integration with IoT devices

Following the fusion process, the system evaluates the probabilities associated with each class to determine the final predicted gesture. This strategic fusion of predictions ensures that the system maximizes its capability to capture diverse aspects of gestures, thereby enhancing overall accuracy and reliability in classification tasks. By effectively leveraging the strengths of both channels, the system enables seamless and intuitive interaction within smart home environments, enhancing user experience.

The technique behind class probability fusion is mathematically defined as follows:

Let $P(i)$ represent the probability of class i predicted by Attention based CNN Model (Model 1) and $P(j)$ represent the probability of class j predicted by CNN Model with Dynamic learning (Model 2) where $i, j = 1,2,3,4,5,\ldots n$ where n is the number of classes. The final predicted gesture is determined by selecting the class with the highest probability among the predictions from both models. This is expressed as:

$$Final\ gesture = ArgMax(max(P(i), P(j))) \quad (1)$$

Equation (1) computes the maximum probability among the predictions from both models for each class and selects the class with the highest maximum probability as the final predicted gesture.

Once the gesture is successfully classified using the trained model on the Raspberry Pi, the resulting class label serves as a command to operate appliances via a relay system. This relay system acts as an intermediary between the Raspberry Pi and the appliances, enabling seamless integration of gesture-based control into the smart home environment. With this setup, a wide range of appliances including fans, lights, televisions, and air-conditioners can be controlled using hand gestures recognized by the proposed model. Each gesture class corresponds to a specific appliance operation, allowing users to intuitively interact with their smart home ecosystem without the need for physical switches or remote controls. This integration of gesture recognition system with IoT based appliance control enhances user experience and facilitates a more interactive and responsive home environment. The hardware experimental set up is shown in Figure 2.

Figure 2 – Experimental setup

4. RESULTS AND DISCUSSIONS

4.1 Comparative analysis of model performance

The model referred in [11] which is CNN-based, achieves a training accuracy of 82.36% and a testing accuracy of 66.18%. In contrast, the baseline model for gesture classification [12] demonstrates higher accuracies, with a training accuracy of 98.6% and a testing accuracy of 72.62%. Moreover, the gesture classification model utilizing Tensor extraction and attention mechanisms achieves even greater accuracy, boasting a training accuracy of 99.53% and a testing accuracy of 83.2%. Lastly, the transfer learning model surpasses all others in accuracy, achieving an impressive training accuracy of 99.16% and an outstanding testing accuracy of 98.24%. These results underscore the effectiveness of gesture classification through different model architectures, with the dynamic learning approach particularly excelling in both training and testing accuracies. Table 2 lists the comparative analytical performance measure of the attempted models.

Table 2 – Comparison with existing models

Sl. no	Model Name	Training Accuracy	Testing Accuracy
1	CNN model [12]	82.36%	66.18%
2	Baseline CNN Model for Gesture Classification [13]	98.6%	72.62%
4	Gesture Classification using Tensor extraction – Attention based	99.53%	83.2%
5	CNN Model with dynamic learning	99.16%	98.24%

4.2 Realtime testing and latency analysis

The latency is observed in predicting a gesture from accepting the input till updating the state of the appliance. Latency calculation is carried using python's inbuilt time package. The observed latency is 0.195 seconds, which is the average delay for 20 consecutive predictions. The existing model stated in [14] has a latency of 0.312 seconds. A comparative analysis of latency is presented in Table 3.

Table 3 – Gesture recognition latency

Sl. No	Model Name	Accuracy	Latency
1	Long-term Memory Augmented Network [14]	97.3%	0.312 seconds
2	Proposed model	99.16%	0.195 seconds

The test accuracy, denoting the proportion of correctly classified instances in our model when assessed on previously unseen data, stands impressively at 98.24%. Conversely, the training accuracy, gauging the proportion of correctly classified instances within the training dataset, registers at 99.16%. The training loss of 2.63%, indicates the extent of error or deviation between the model's predictions and actual target values during the training process, guiding the optimization of our machine learning model. Validation loss, another critical metric, quantifying the discordance between the model's predictions and the actual target values on validation data, is 5.9%, thus serving as a vital gauge of model generalization and performance on unseen data.

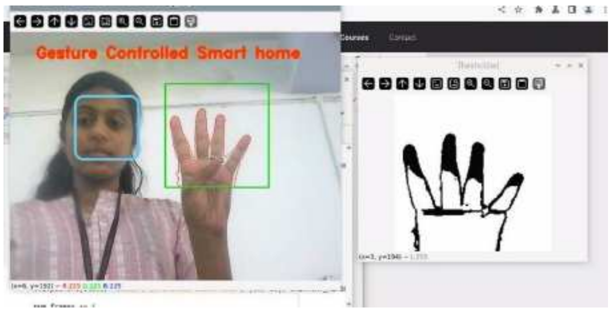

Figure 3(a) – Realtime Implementation to on the appliance

Figure 3(b) – Realtime Implementation to off the appliance

A sample result of real-time experimentation of system output depicted in Figure 3, shows gesture 4 being classified. It invokes the light to be in ON state whereas gesture 5 invokes the light to be in OFF state. The appliance control with gestures is shown in Table 4.

Table 4 – Appliance Control

Class Label	Gesture	Appliance Control
4		Light ON
5		Light OFF

5. CONCLUSION

The proposed system culminates in a comprehensive approach to gesture-driven smart home automation leveraging IoT. Through the integration of advanced machine learning techniques such as deep learning, transfer learning, and attention mechanism, the system is capable of accurately recognizing and responding to hand gestures in real-time. The proposed system architecture, modular in design, seamlessly integrates continuous camera feed, gesture recognition modules, and IoT, demonstrating enhanced computational capabilities and efficiency. The attention mechanism with dynamic learning rate enhances the system's adaptability and performance, making it a potent tool for real-world gesture recognition systems. Achieving impressive testing accuracy of 98.24%, and 99.16% on train accuracy, the system demonstrates its efficacy in real-world applications. The low latency of 0.195 seconds for 20 consecutive predictions further emphasizes the system's real-time responsiveness and practical utility. Overall, the proposed system represents a significant advancement in gesture-driven smart home automation, showcasing the transformative potential of leveraging IoT and advanced computational techniques for enhancing human-computer interaction in smart environments. The working model offers a practical solution for functional requirements for a smart home gateway under ITU-T Recommendation J.1611.

REFERENCES

[1] "Smartify-India's Leading Home Automation Store." Smartify . https://smartify.in/ (accessed Feb 23,2024)

[2] "Smart Home Use Case – FIBARO" Fibaro. Smart Home Use Case - FIBARO Home | FIBARO https://www.fibaro.com/en/smart-home-in-use/ (accessed March 11,2024)

[3] J. -H. Song and S. -J. Kang, "3D Hand Pose Estimation via Graph-Based Reasoning," in IEEE Access, vol. 9, pp. 35824-35833, 2021.

[4] Aggarwal, A., Bhutani, N., Kapur, R. et al. Real-time hand gesture recognition using multiple deep learning architectures. SIViP 17, 3963–3971, 2023.

[5] Ge. Le, Liang. H, Yuan. J and Thalmann. D, "Real-Time 3D Hand Pose Estimation with 3D Convolutional Neural Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 4, pp. 956-970, 2019.

[6] ITU-T J1612 "The architecture for a smart home gateway" International Telecommunication Union Recommendation https://www.itu.int/rec/T-REC-J.1612-202307-I/en.

[7] ITU-T J1611 "Functional requirements for a smart home gateway" International Telecommunication Union Recommendation https://www.itu.int/rec/T-REC-J.1611-202210-I.

[8] Alnuaim A, Zakariah M, Hatamleh WA, Tarazi H, Tripathi V, Amoatey ET. "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet," Comput. Intell. Neurosci., Mar 26, 2022.

[9] Nogales, R.E.; Benalcázar, M.E., "Hand Gesture Recognition Using Automatic Feature Extraction and Deep Learning Algorithms with Memory," Big Data and Cognitive Computing. 2023; 7(2):102.

[10] Hand gesture Recognition Dataset https://www.kaggle.com/datasets/aryarishabh/hand-gesture-recognition-dataset (accessed March 11,2024)

[11] J. S. Peixoto, A. R. Cukla, M. A. de Souza Leite Cuadros, D. Welfer and D. F. Tello Gamarra, "Gesture Recognition using Fast DTW and Deep Learning Methods in the MSRC-12 and the NTU RGB+D Databases," in IEEE Latin America Transactions, vol. 20, no. 9, pp. 2189-2195, Sept. 2022.

[12] Ganji. N, Gandreti. S and Krishnaiah. T. R, "Home Automation Using Voice and Gesture Control," 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 394-400, 2022.

[13] Kurian. B, Regi. J, John. D, P. H and Mahesh. T. Y, "Visual Gesture- Based Home Automation," 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), Kalady, Ernakulam, India, pp.286-290, 2023.

[14] L. Zhao, X. Lu, Q. Bao and M. Wang, "In-Place Gestures Classification via Long-term Memory Augmented Network," IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Singapore, Singapore, 2022, pp. 224-233, 2022.