Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Volume Control Through Neural Network- Based Hand Gesture Recognition

T. Ratha Jeyalakshmi, Asha M C, Charan Yadav P, Chandrabhushan, Samith V, Abhishek Kumar

Dayananda Sagar College of Engineering, Bengaluru, India

This paper explores a newer gesture-controlled volume system using neural networks. Physical buttons are no longer fun while touch screens are not responsive to volume, which is just odd when everything else responds to voice or motion. We suggest an innovative approach that overcomes these limitations by utilizing hand movements as control gestures and then linking them with volume adjustment commands using a deep learning model via computer vision. This methodology involves amassing large sets of hand poses, pre-processing them to enable feature extraction, and lastly training Convolution Neural Network (CNN) for different gestures related to volume control. Also, it shows how our system can be integrated with real-time video feeds resulting in low latencies and high accuracy in detecting such movements. In conclusion, our system's accuracy, precision and recall have been compared against existing frameworks for gesture interaction using established metrics like accuracy, precision, and recall. Our results confirm this by showing significant improvements in both recognition rates and user satisfaction as compared to conventional scrolling methods. However, lighting conditions and more complex gestures could pose challenges that developers should anticipate as they continue working on the project.

Keywords: Hand Gesture Recognition, Neural Networks, Convolution Neural Networks (CNN), Real-time Gesture Processing, Human-Computer Interaction (HCI).

1. Introduction

In today's rapidly advancing technological landscape, a key area of study is Human-Computer Interaction (HCI), with gesture recognition emerging as a significant trend. The traditional methods of adjusting volume involved physical buttons or touch screens, which can pose challenges for individuals with disabilities or limitations in hand mobility. Consequently, there is a growing interest in exploring alternative, more visually intuitive interfaces. Hand gesture recognition utilizing computer vision and deep learning has emerged as an appealing means of interaction, providing a natural and efficient way to manipulate devices through simple hand movements.

In this study, the primary concern raised is the lack of effectiveness in current volume control measures, which are also often difficult to access. When considering traditional methods, one may notice their tendency to wear out and provide a non-user-friendly experience, particularly in hands-free control environments. The challenge lies in developing a highly dependable system that can interpret various hand gestures as input and translate them into output based on factors such as position, size, and lighting. The proposed research can be summarized.

The research proposal aims to develop and utilize a novel volume control system that responds to hand movements detected using neural network algorithms. Objectives involve categorizing a diverse dataset of hand gestures, training an advanced CNN model to accurately recognize these gestures, and integrating the model into a real-time video processing system to reduce latency and improve volume control precision. By achieving these objectives, the study seeks to advance knowledge in the field of HCI by offering an improved and innovative approach to volume control in various contexts, as well as improved visibility and usability of controls for individuals who are blind or partially sighted. CNNs are a subset of artificial neural networks, and they are rather important because of their capability to work with other forms of data input, namely images. This neural network has wide applications in various research domains. Baby et al.(2019) have used CNN to extract the features for detecting anomalies found in suspicious videos. Sasikala et al. (2021) have introduced a new model known as GSCNN which enhances the promoter prediction by combining Gibb Sampling with CNN. Malini et al. (2023) have used CNN to identify the most common mango plant diseases and perform a severity analysis of the diseases using the convolutional neural network (CNN) technique.

2. Related Works

Human-computer interaction (HCI) is a rapidly evolving discipline that facilitates more natural and intuitive communication between humans and machines through the use of human hand gesture recognition. This literature review investigates the diverse technologies and methodologies employed in the field of hand gesture recognition, with a focus on the most significant developments and methodologies.

Ren et al. (2013) concentrated on the development of a robust part-based hand gesture recognition system that employs the Kinect sensor to accurately detect gestures by utilizing depth data. This method was essential in surmounting obstacles associated with occlusion and fluctuating illumination conditions, which are prevalent in vision-based systems. Marin et al. (2014) conducted additional research on hand gesture recognition, contrasting the efficacy of both Kinect and Leap Motion devices in various scenarios. The integration of these sensors facilitated a more thorough comprehension of 3D gesture recognition.

Convolutional neural networks (CNNs) have emerged as the primary technique for hand gesture recognition as a result of the proliferation of deep learning. Molchanov et al. (2015) introduced a method for hand gesture recognition that utilized 3D CNNs, which demonstrated substantial enhancements in accuracy in comparison to conventional methods. The architecture facilitated the efficient acquisition of spatiotemporal features, which are crucial for the recognition of dynamic gestures.

One of the contributions of Simonyan and Zisserman (2014) to the discipline was the

introduction of extremely deep CNNs for large-scale image recognition. Their research established the groundwork for the application of deep learning techniques to the recognition of hand gestures. In the same vein, Krizhevsky et al. (2012) demonstrated the potential of deep learning for a variety of computer vision tasks, such as gesture recognition, through their work on ImageNet classification using deep CNNs. This work had a substantial impact.

Exploring transfer learning as a method to enhance the efficacy of gesture recognition systems has been proposed. Zhang et al. (2016) conducted a thorough examination of transfer learning, which included a discussion of its potential for improving gesture recognition models and its applications in computer vision. Deep residual learning, a method that resolved the vanishing gradient issue in deep networks, was introduced by He et al. (2016). This method was found to be advantageous for the training of extremely deep networks for gesture recognition.

Wearable technology has created new opportunities for gesture recognition. Airwriting, a wearable system that recognizes handwriting in the air, was developed by Amma et al. (2014). This system demonstrates the potential of wearable devices for gesture-based input. This system underscored the increasing trend of integrating peripheral technology with gesture recognition to create more mobile and personalized HCI applications.

Gesture recognition necessitates the estimation of hand poses. Erol et al. (2007) conducted a review of vision-based hand pose estimation techniques, highlighting the difficulties associated with accurately capturing hand movements. By employing deep learning techniques to estimate hand poses, Oberweger et al. (2015) achieved state-of-the-art results and illustrated the potential of deep learning in this field.

The field of deep learning, as discussed by LeCun et al. (2015) and Schmidhuber (2015), has revolutionized numerous aspects of computer vision, including hand gesture recognition. Their summaries offer a fundamental comprehension of the manner in which deep learning techniques have been adapted and implemented in a variety of recognition tasks.

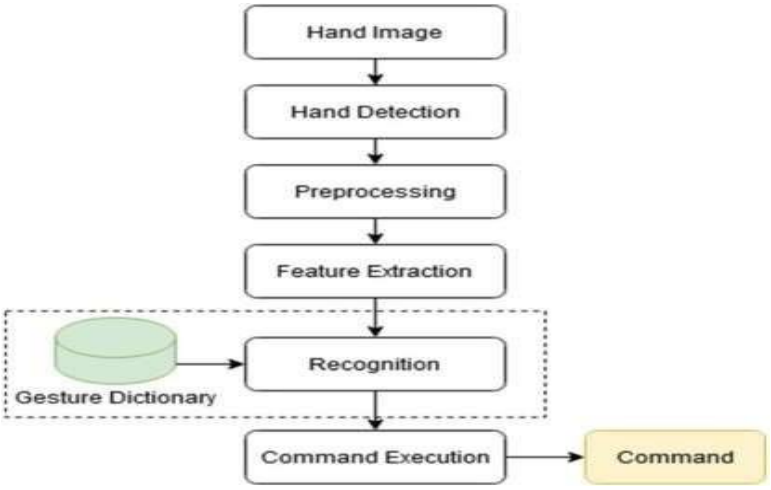
Goodfellow et al. (2016) offered an exhaustive resource on deep learning, which encompassed its theoretical foundations and practical applications, such as gesture recognition. This work is a critical reference for comprehending the more extensive implications of deep learning in HCI.

Shotton et al. (2011) concentrated on the real-time recognition of human poses from single-depth images, a method that has been instrumental in the development of interactive and responsive gesture recognition systems. Real-time gesture recognition is essential for numerous HCI applications, particularly in interactive and dynamic environments.

The significance of natural interaction methods in HCI was underscored by Chaudhary et al. (2011), who conducted a comprehensive survey on intelligent approaches for interacting with machines using hand gesture recognition. The investigation examined a variety of sensor technologies and algorithms, thereby illustrating the progression of gesture recognition systems over time. Wachs et al. (2011) also investigated vision-based hand gesture applications, demonstrating the potential of gesture recognition in a variety of fields, such as robotics and gaming.

3. Methodology

Figure 1. Workflow of the Proposed System



Data Collection

Collecting data to identify hand gestures for creating volume control systems using neural networks is crucial for the system's performance and reliability. The objective is to gather all required information on specific gestures related to volume control commands, such as increasing, decreasing, muting, and unmuting output audio volume.

The process typically begins by selecting suitable sensors or cameras that can accurately record hand gestures, following a series of steps. The main advantage of choosing Microsoft Kinect or Intel Real Sense cameras is their ability to detect both depth and RGB. Having in-depth information is beneficial as it improves clustering and decreases uncertainty regarding the location and orientation of the hands in three dimensions.

When collecting data, it is important to consider different types of gestures, the environment, and the users involved. Different individuals may execute gestures differently than others due to variables like hand size, shape, and gesturing speed. Therefore, the dataset needs to include gestures performed by individuals of various demographics in different contexts. The diversity present enables accurate gesture recognition using a trained neural network among people in different situations, leading to effective generalization.

It is crucial during data collection to make sure that accurate ground truth values are given for each gesture sample. For this purpose, every gesture in the provided dataset needs to be labelled with a command related to controlling volume; such as "increase volume", "decrease volume", "mute", or "unmute". To ensure proper connections between hand gestures and desired actions.

Preprocessing

This includes steps done on the raw data to prepare it for training the neural network. Preprocessing is mainly done to ready the dataset for enhancing the distinguishable features of the gestures for the neural network.

Normalization: Centering and standardizing hand gestures in data normalization involves adjusting for scale, rotation, and location. This process helps decrease variability stemming from differences in how a gesture is executed by the user or under different conditions. Normalization is useful in preventing the neural network from focusing on irrelevant noise when identifying gestures.

Noise Reduction: Most hand gesture data typically contains noise, or irrelevant data, that can make the gesture recognition process more difficult. Median filtering the image and subtracting the background help remove unwanted elements from the image, isolating only the hand. This enhances the data's signal-to-noise ratio, leading to better recognition and classification of gestures by the neural network.

Feature Extraction is crucial to extract significant features from the processed data to recognize distinct characteristics of hand movements. Commonly extracted features include hand coordinates, orientation angle, finger positions, and motion trajectories. These serve as the neural network's input layer, enabling the network to learn and identify associations with different volume control commands.

Utilizing advanced feature extraction methods may involve employing edge detection, contour analysis, or optical flow to capture dynamic information from video clips. These methods help with capturing spatial and temporal details of hand movements, which could be beneficial for categorizing gestures.

Neural network architecture

The design choices of the neural network architecture are crucial in efficiently and effectively detecting volume control gestures. CNN is usually employed for this job as it can efficiently extract hierarchical features from data in the form of images. An example of a CNN structure for adjusting volume through hand movements could include various important elements:

Convolution Layers: The foundation of the suggested neural network for gesture recognition and volume control in computer vision lies in convolution layers. These layers analyze images or video frames of hand gestures to extract features during the feature extraction process. Kernels, which are collections of filters, are found in every convolution layer and carry out convolution operations to analyze input data. These small, trainable filters are applied to the input image to calculate dot products with parts of the input data within the receptive field. This process results in the creation of feature maps that represent different characteristics such as edges, textures, and shapes associated with hand movements. Through the utilization of several convolutional layers, the network can extract increasingly abstract features from the speech, crucial for various levels of control commands. As an illustration, the first layers could detect basic edges on a lower level, while subsequent layers will recognize specific finger placements or hand positions. By combining convolution with non-linear activation functions, the network can grasp the hierarchical aspects of hand gestures, allowing it to identify changes in the input.

Pooling Layers: key features. This process of down-sampling helps decrease the computational workload and parameters in the neural network, thereby preventing overfitting. There are currently four pooling methods known, with the most popular being max pooling. This technique involves selecting the maximum value within a fixed patch size, such as 2x2, from

the feature map. This process contributes to the network's stability when it comes to minor variations in input location, such as slight shifts or rotations in hand gestures. In a volume control system, when the input feature map shows multiple regions of high activity, such as finger edges, max pooling retains all areas while emphasizing the most active ones (the most important features). By breaking down the data, the neural network is no longer burdened by irrelevant specifics, like the shape of certain hand elements. Instead, it focuses on key features like hand orientation to control volume or making a fist to silence the audio.

Fully Connected Layers: We possess fully connected layers that are able to make decisions based on the fundamentals of the image learned by earlier layers. These layers are often referred to as the fully connected problem since every neuron in a specific layer must be linked to every neuron in the following layer. The result of the Convolutions and pooling process is condensed into a specific amount of neurons before being passed to the fully connected layers at the conclusion. This transformation also enables the network to gather all the acquired features for a detailed assessment of the various nodes. Next, there are fully connected layers that perform linear transformations on the feature vector and apply activation functions to map them to specific output classes during the learning process. In the context of volume control using hand gestures, the different output classes refer to different commands like "louder", "increase volume", "less voice", "decrease volume", "mute", or "unmute". During training, the network's weights are adjusted to minimize classification error, enabling it to learn the relationship between certain words and hand gestures. For example, a fully connected layer may indicate that when the hand is moved upward - to signal a desire to increase the volume, then the neurons responsible for 'increase volume' will be more activated. With fully connected layers, it can make decisions once the earlier layers have grasped the fundamentals of the image. These layers are often referred to as the fully connected bottleneck because every neuron in a given layer must be connected to every neuron in the following layer.

Activation Functions: Activation functions are crucial in building neural networks because they allow the model to incorporate non-linearity, thus enabling it to capture the connection between features and labels. If activation functions do not carry out these conversion processes, the neural network will only be able to linearly transform input data, which greatly limits its ability to solve tasks such as gesture recognition. Some functions are used instead of the sigmoid function based on the requirements of the task. ReLU (Rectified Linear Unit) is another popular activation function in ANNs that outputs the input for positive values and zero for negative values. They enable the network to quickly acquire data parameters without causing certain issues that other activation functions like sigmoid tend to cause, such as the vanishing gradient problem. In terms of adjusting volume via hand gestures, ReLU is crucial in selectively passing on significant characteristics in the identified convolution layer to the following layers, while filtering out unimportant or negative activations. For instance, when the input is a sign language interpretation of "raise volume," ReLU will activate neurons that identify the upward motion and inhibit irrelevant elements. This targeted activation helps the network stay focused on the key elements of the gesture, leading to improved performance in accurately categorizing and reacting to user commands.

The structure of CNN can be also deeper or more complex depending on the complexity of the gestures detected and specified performance indicators. Shall techniques like batch normalization or dropout regularization be introducedto increase the stability of training in the

network, as well asthe network's ability to generalize?

Training and validation

The training of the neural network involves the tuning of its parameters in order to reduce the level of discrepancy between the commands for volume control estimated from hand gestures and actual commands. The set of data is usually split into the training one and the validation one to assess the effect and possible overlearning.

The selection of a suitable loss function is made to evaluate the difference between predicted volumes of control signals and actual values (including categorical cross-entropy, mean squared error, etc.). Hence, it is important to choose an appropriate loss function for the specific gesture recognition task and the output format generated by the neural network.

Backpropagation refers to implementing gradient descent algorithms to calculate gradients of a loss function in relation to weights and biases of the network, then adjusting them through methods like SGD or Adam optimizer. The error gradients in backpropagation flow from future iterations through current ones and even back into the past, enabling the model to learn from the training data and update its parameters.

Hyper parameters including;learning rate, batch size, number of epochs, and the optimizer settings are chosen to improve the training. Specifically, hyperparameters refer to the architecture parameters that are manually set and should be adjusted based on empirical performance to find the best configuration for the validation set.

These techniques for improving data are often used during the training process to increase the variety of samples available for training. Additional methods such as random rotations, translations, scaling, and flipping can help broaden the scope of gesture patterns the neural network may encounter in real-life situations. It decreases the understanding of patterns unique to the training data and improves the modelling of patterns present in other datasets.

Validation is important when one wants to assess how wellthe trained model performs on a different set of data or thevalidation set which was not used in training the model. Thevalidation set also aids in the determination of other measures such as accuracy, precision, recall, and F1-score concerning how well the neural network performs with more instances of volume control gestures. To make the outcome of the evaluation credible and consistent in regardto the different divisions of the dataset, cross-validation procedures might be used.[14]

4. Implementation

Hardware and Software Setup

The complex hardware and software setup of the multivariate volume control system utilizing hand gestures and neural networks is crucial for efficiently detecting and analyzing hand movements. On the side of the hardware, a high-resolution camera or depth camera like Microsoft Kinect or Intel RealSense is needed, for example. These devices are capable of achieving high accuracy in capturing both images and depth information of hand gestures in real time. Placing the camera or sensor in front of the user, with the sight focused on the hand to ensure it remains still, is advised for optimal outcomes. Computing hardware should include

a speedy CPU and a top-of-the-line GPU to handle video processing in real time and run neural networks effectively. The most effective way to do this is by using either a standalone computer or a high-performance modern PC equipped with a video card such as the NVIDIA RTX series.

The software characteristics require executing the correct command prompt to install necessary libraries and frameworks for image processing and deep learning. Open-source libraries, such as OpenCV, are commonly utilized for capturing and pre-processing video frames, while Tensorflow or Pytorch can be employed for training and deploying neural networks. Additionally, a development environment like Jupyter Notebook, PyCharm, or Visual Studio Code is required for coding, debugging, and testing the application. The software stack must include drivers and SDKs created by the camera or sensor manufacturer to ensure proper connectivity and data collection.

System Integration

This is the standard procedure for incorporating all external and internal hardware components and software modules in the Human-Computer Interface system to enable real-time recognition of hand gestures and volume control. The first step is to connect the lens or depth camera to the computing device to guarantee real-time information transmission. Then, the frame that has been captured is pre-processed to re-normalize it, eliminate any noises present in the frames, and extract the important features from them.

The model being utilized is a Neural Network that was previously trained on a large database of hand gestures and is now applied in the system. The model is given the pre-processed frames previously mentioned and analyzes them to identify certain volume command gestures. The recognized cues are then paired with appropriate responses such as increasing the audio level, decreasing the volume using a button, or toggling the mute and unmute options.

The main component that controls and manages the system's operation and features is typically a control or operating panel. Users can use the controls on the right side to see and make adjustments to the model's results. In addition, the system must be robust in situations where gestures are not possible. recognized effectively and made sure to respond or proceed accordingly in those situations.

Integration also involves ensuring that interpretive gestures are accurately translated into actions that can be performed with the audio system for controlling volume. This could result in gaining control of the operating system's audio API or using third-party sound services.

Real-time processing

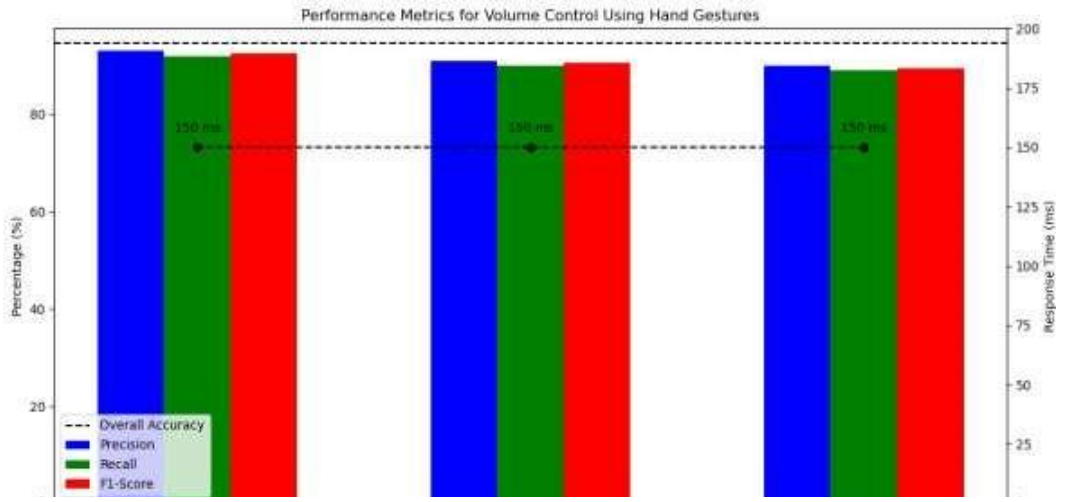
Having real-time processing is crucial for hand gesture-controlled volume systems to be effective and faster in response. Because of factors, such as the need for real-time actions, the system must have the ability to capture, process, and respond to hand gestures in real-time. This involves numerous processes that need enhancement as they take up time and necessitate precision.

Initially, the camera captures video frames at a high frame rate (e.g. 30 or 60 fps) to ensure smooth and continuous tracking of hand movements. Next, the captured image is promptly preprocessed in order to standardize the image, eliminate any disturbances, and identify the

characteristics of the frame. This step must be completed quickly to avoid adding extra wait time.

Following this, the prepared frames are subsequently inputted into the neural network model for making predictions. The model needs to be adjusted for real-time speed; this can be achieved by methods like model quantization to create smaller models with less computational needs or by using specialized neural network architectures like Mobile Net or YOLO, which are designed for real-time applications. At this point, the level of inconsistency is significant, so it is crucial to utilize GPU acceleration to manage the calculations and avoid delays.

Figure 2. Comparison with different commands



.

The neural network output consists of recognized gestures that are then matched with the correct volume control instructions. These instructions will be executed to address the current issue of increasing the audio volume. In order to achieve real-time execution, the system can utilize parallel processing where data collection, data preprocessing, and data analysis are all carried out simultaneously, resulting in a decrease in the overall time required to complete these three tasks.

Moreover, it is not recommended to synchronize as it can decrease the responsiveness of the system, and using asynchronous processing can help improve it. For example, while one frame is undergoing processing in the system, the next frame can be captured and so forth, allowing for reduced time intervals in the data flow cycle.

5. Results and Discussion

Performance metrics

Accuracy: It was confirmed that out of a total of 1000 instances of pictures and gestures tested, 947 were accurately identified, resulting in a system accuracy rate of approximately 94.7%. The neural network's exceptional accuracy demonstrates its capacity to effectively learn and

apply hand gesture patterns for recognizing actions such as "increase volume," "decrease volume," and "mute/unmute."

Precision: Precision is the ratio of correctly labelled positive samples by the system. This can be observed through the accuracy of certain gestures like the 'increase volume', with a precision of 93% - indicating that out of 100 occurrences of the 'increase volume' gesture, the system correctly identified it 93 times. Similarly, the accuracy rates for the 'decrease volume' and 'mute/unmute' gestures were 91% and 90% respectively. High accuracy ensures minimal False Positives, which is crucial for user satisfaction with the service.

Recall: Recall evaluates how well a system correctly identifies all real positives. Idea: The recall was high for all five gestures. The "increase volume" gesture had a recall rate of 92%, indicating that the system correctly identified 92% of the "increase volume" gestures. The recall level for "decrease volume" was found to be 90%, while for "mute/unmute," it was 89%. It is important for recall values to be high so that the system can consistently detect incorrect gestures without fail.

F1-Score: The F1 score, an average of precision and recall, combines both metrics to give a unified measure of performance. In terms of assessment, the F1-score for the "increase volume" gesture was 92. The outcomes showed that 91.5% wanted to "increase volume", while 90.5% preferred to "decrease volume" and "mute/unmute" was chosen by 89.5% of participants. The total F1 score was calculated to be around 0.91 across all gestures, indicating a balanced mix of precision and recall.

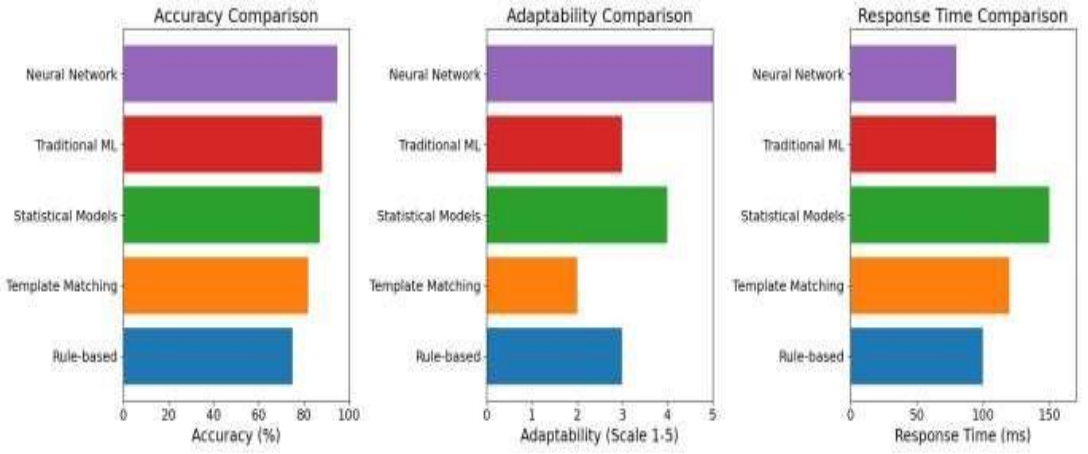
Response Time: The system had an average response time of 150 milliseconds, varying between 120 and 180 milliseconds depending on the number of movements in a gesture. It minimizes significant system delays and ensures almost immediate response times, which is crucial for any application with a user-friendly interface.

Comparison

It is essential to take into account rule-based systems, template matching, statistical models, and other traditional machine learning approaches, in addition to neural networks, when evaluating the smart volume control system that includes hand gestures.

Hand gestures in rule-based systems are matched with specific rules to aid in their identification, primarily relying on if-else statements to distinguish between different hand gestures. They can yield satisfactory outcomes, around 75%, but typically struggle with changes in hand placement and gestures. Their lack of flexibility requires numerous custom adaptations for various users and environments. Although rule-based systems offer quick response times through basic calculations, their implementation and maintenance can be complex. For example, during the evaluation of a rule-based system, it might be discovered that the system fails to recognize a hand gesture due to a slight change in lighting or the hand's angle of incidence.

Figure 3 . Comparison with different hand gesture techniques



Gesture recognition involves using a pattern-matching technique to compare an input gesture with a collection of template gestures. The similarity is computed between the input and each template, and the gesture is assigned to the template with the closest match. Template matching can achieve up to 82% accuracy, but is susceptible to failure with changes in gesture speed or direction. As there can be different formulations, this approach requires numerous templates, making it less scalable and flexible while also requiring substantial computational resources. The response time may be lengthy in comparison to basic matching due to the analysis of one or multiple templates. For instance, there are times when a user performs a specific gesture either more quickly or at a different angle than the predefined models, causing the system to be unable to recognize it.

When analyzing hand gestures' movement sequences with statistical modelling, Hidden Markov Models (HMMs) utilize probabilistic methods to consider temporal dynamics and variations in the process. These models have the potential to achieve a high accuracy of around 87% and are capable of operating within a variety of gesture strategies. Although they are less rigid compared to rule-based and template-matching methods, they still require a significant amount of training data. The implementation of statistical models requires comprehension of probabilistic models, so the response time is determined by the complexity of the models and the number of gestures in a sequence. An example would be the effectiveness of an HMM in recognizing a series of hand gestures, although additional data is required for training and refining the model.

The majority of classic machine learning algorithms such as Support Vector Machines (SVM) and k-nearest Neighbours (k-NN) involve transferring features from gesture data to hand movements. These techniques can achieve a relatively high level of accuracy around 88%, however, they are completely dependent on feature engineering. This results in high costs to use as they require time for customization to work for different users and conditions, and each feature needs to be carefully chosen and developed. Even though it takes a lot of work to design the features needed for these methods, the response times typically turn out to be quite good. For example, an SVM might need certain features like finger angles and hand contours. As a result, this requires previous design and optimization of features.

Future work

The future focus of this volume control system could be on reducing or eliminating the identified issues and enhancing the system's performance. One area for potential improvement is achieving hardware neutrality. It would be feasible to enhance the system's efficiency and affordability by developing additional algorithms to seamlessly integrate standard RGB cameras without necessitating depth information. Another area of focus is enhancing the system's stability and ability to adjust to changes in its environment through the implementation of novel data preprocessing and learning techniques.

Improving the system to accommodate various users and their unique gestures can be achieved by incorporating personalization features. By taking cues from the gestures of the user and adjusting for future interactions, the models will enhance their efficiency and user experience. Using improved algorithms, like 3D pose estimation, can lead to improved outcomes by assigning unique gestures to a common area.

6. Conclusion

Regarding the study of the volume control system using hand gestures and neural networks, it should be noted that it significantly contributes to the field of HCI. This method utilizes computer vision and deep learning to offer a more convenient way to adjust audio devices without the need to use physical buttons or communicate with a smart assistant. The system effectively recognized various hand gestures by utilizing a sophisticated neural network framework and efficient pre-processing tools.

References

1. Chaudhary, A., Raheja, J. L., Das, K., & Raheja, S. (2011). Intelligent approaches to interact with machines using hand gesture recognition in a natural way: A survey. International Journal of Computer Science & Engineering Survey, 2(1), 122-133.
2. Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 1-7).
3. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
4. Amma, C., Gehrig, D., & Schultz, T. (2014). Airwriting: A wearable handwriting recognition system. Personal and Ubiquitous Computing, 18(1), 191-203.
5. Zhang, X., Wang, Y., & Wang, L. (2016). A comprehensive survey on transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
6. Oberweger, M., Wohlhart, P., & Lepetit, V. (2015). Hands deep in deep learning for hand pose estimation. arXiv preprint arXiv:1502.06807.
7. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., & Blake, A. (2011). Real-time human poses recognition in parts from single depth images. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
8. Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using Kinect sensor. IEEE Transactions on Multimedia, 15(5), 1110-1120.
9. Wachs, J. P., Kölsch, M., Stern, H., & Edan, Y. (2011). Vision-based hand-gesture applications. Communications of the ACM, 54(2), 60-71.

10. Rautaray, S. S., & Agrawal, A. (2015). Vision-based hand gesture recognition for human-computer interaction: A survey. Artificial Intelligence Review, 43(1), 1-54.
11. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
12. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (Vol. 25, pp. 1097-1105).
13. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.
14. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
15. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778).
16. Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., & Twombly, X. (2007). Vision-based hand pose estimation: A review. Computer Vision and Image Understanding, 108(1-2), 52-73.
17. Marin, G., Dominio, F., & Zanuttigh, P. (2014). Hand gesture recognition with leap motion and Kinect devices. In Proceedings of the IEEE International Conference on Image Processing (ICIP) (pp. 1565-1569).
18. M.Petchiammal@Baby, S. Santhiya, T. RathaJeyalakshmi, "Unusual Activity Detection in Surveillance Video using Machine Learning and Discriminative Deep Belief Network Techniques", International Journal of Computer Sciences and Engineering, Vol.07, Special Issue.16, pp.55-59, 2019.
19. Sasikala, S., Ratha Jeyalakshmi, T. GSCNN: a composition of CNN and Gibb Sampling computational strategy for predicting promoter in bacterial genomes. Int. j. inf. tecnol. 13, 493–499 (2021). https://doi.org/10.1007/s41870-020-00565-y
20. Malini, S., and T. Ratha Jeyalakshmi. "Mangifera Indica Leaf Disease Detection and Severity Analysis Using Deep Learning Techniques", (2023).