

Wireless Hand Gesture Robot Control by Voice Recognition using Machine Algorithm

Shashidhar R¹, Nishchal G Nayaka², Chethan S³ and Roopa M⁴

¹⁻³Department of Electronics and Communication Engineering, JSS Science and Technology University Mysuru, India
Email: shashidhar.r@jssstuniv.in, nischalnayaka23@gmail.com, chethansudhas098@gmail.com

⁴Department of Electronics and Communication Engineering, Dayananda Sagar College of Engineering Bangalore, India
Email: surajroopa@gmail.com

Abstract—This research focuses on developing a hand gesture-controlled robot using machine learning (ML) algorithms and an accelerometer, with the goal of creating a more intuitive human-robot interaction. Initially centered on signal processing, the project evolved to integrate both gesture recognition and voice command identification for robot control. The system utilizes an ADXL335 accelerometer connected to an Arduino Mega and an L293D motor driver to detect hand movements and translate them into directional commands (left, right, forward). A machine learning model was developed to classify spoken commands ("front," "back," "right," "left") using the Mel-frequency cepstral coefficients (MFCCs) for feature abstraction of the data and support vector machines (SVM) for training along with testing, achieving a 92.59% accuracy. Despite the success of the ML model, time constraints prevented full integration with the robot's physical control system. This project lays the groundwork for future research aimed at combining voice and gesture control, contributing to more seamless human-robot interaction.

Keywords— MFCC, SVM, gesture recognition, voice control, human-robot interact.

I. INTRODUCTION

Human-Robot interaction (HRI) is evolving in different sectors, establishing its foundations within close work with intelligent robotic systems and the users[16]. For a human to be able naturally communicate with such robots, important is an accurate recognition of hand gestures that has been in the past predominantly addressed by vision-based methods requiring cameras or depth sensors. Still, these approaches have to deal with variations in hand shapes, movement speed and lighting situation. To address these limitations, surface electromyography (sEMG) sensors have been employed to identify the electrical impulses related with muscle contractions that accompany hand actions which can provide a better understanding about of the particular muscle's activities. To classify hand gestures based on sEMG data, machine learning with deep learning methods is used. One prominent idea uses Root Mean Square (RMS) Support Vector Machines, controlling 15 and 12 hand gestures for able-bodied users and amputees respectively to achieve similar accuracies. But it has been difficult to design human feature sets and choose a good classifier. Precise gesture interpretation has recently been enabled Recurrent Neural Networks (RNN). In general, CNNs to detect hand shapes and finger poses as well as motions based on image or video frames and RNNs such LSTM/GRUs for temporal dynamics which is a key component for gesture detection.

In a current project, an accelerometer captures hand motion, a camera detects motion, and an RF module facilitates communication. The robotic system uses accessible components like servo motors, DC motors, a robot chassis, and Arduino Uno, integrating these elements for effective gesture-based control.

II. RELATED WORKS

HGS-SCNN detects hand gesture with a higher accuracy score than that of ptml, and it can be generalized to different applications. It also offers a robust gesture control system, improving human-machine interaction. Efficient gesture communication in Robotics makes use of posture detection leads to optimization corrections in gesture communication and functions increasing the level of automation and efficiency in the tasks [1]. The transmitter consists of an accelerometer connected to an Arduino and the receiver consists of radio frequency modules, which capture signals. This system enables the medical assist motors to work in different directions relative to the hand movements. It can be employed in military settings, industrial uses, and assistive devices for people requiring hands-free operation [2]. Using DC motors controlled by an Arduino Nano, the paper outlines a car chassis. Compares the accelerometer data against reference values, to create movement commands — forward, backwards, right and left — and defines the basis for advanced commands It can be used in handling hazardous materials and in assistive technology applications for people who disabilities [3].

They developed a system that tracks wrist gestures using an accelerometer that sends signals to a receiver which then executes certain functions. This data is transmitted wirelessly to one RF module, and displays over a GUI where the required movements are being made by the robotic vehicle motors. This inexpensive technology skillfully connects real and digital interaction.[4] This hand-operated robot control system utilizes an accelerometer to detect arm motion, with an Arduino microcontroller processing the signals to send directional commands. The Arduino converts analog accelerometer output into digital values, controlling the robot's motors via an RF transmitter-receiver. This tutorial focuses on the technical aspects of controlling a robotic servo arm using these methods. [5] It integrates fingerprint sensor for contact-based scanning and is integrated with a gesture-controlled robot, connected via an accelerometer. Change of hand direction also changes the command for the robot The basic hardware modules utilized consists of Arduino Uno, Flex sensors, LM324 comparator IC, an L293D motor shield, and an RF Transmitter-Receiver circuit for wireless communication. [6] This paper presents an Arduino-based gesture-controlled robocar that interprets hand signals and spoken commands. It includes an accelerometer, ultrasonic sensor, RF transmitter, and Bluetooth module for communication and control. The system enables hands-free operation, enhancing user convenience. Applications span assisting individuals with disabilities, agriculture, and medical surgeries. [7]This paper explores the design of an affordable, adaptive technology for individuals with mobility impairments, using an Arduino microcontroller and accelerometer to control a wheelchair via hand movements. It examines mechanical adjustments and electronic components to balance cost-effectiveness with quality and performance. The research highlights the technology's potential to assist those with physical limitations. [8] Automatic Speech Recognition (ASR) that most frequently operates in real-life environments and business applications. There are different AppSource assistants exist to route incoming calls through voice commands and interact with users. ASR technology finds a wide range of commercial applications: from live captioning, virtual meetings to call tracking service. [9]The system described in this paper uses a more general procedure that operates on the entire speech recognition and text-conversion framework, acoustic modeling responsible for the identification of some basic linguistic units, language modeling predicting some words for the next context, and a decoding process converting feature information to text. It offers better quality and is applicable to a lot of different accents, dialects, and languages, and thus has increased effectiveness in its applications. [10] This system employs a DHH-HMM speech recognition model to convert audio into text, followed by transformer-based translation models that include a sentence extraction summarization feature. The speech recognition model is trained using the Kaldi toolkit along with the TEDLIUMv3 and LibriSpeech corpora.[11]Natural Language Processing (NLP) aims to understand and manipulate human language, enabling tasks like speech-to-text conversion. This paper focuses on processing Kannada using Hidden Markov Models (HMM) and statistical modeling. By implementing NLP, we can automate sentiment analysis and analyze previously inaccessible language data, improving communication in business meetings and assisting the deaf community. [12]This paper examines how neural networks can categorize speech signals, noting that simpler models are limited in recognizing just a few words. Their effectiveness largely depends on pre-processing techniques. They showcase speech recognition in MATLAB using Feedforward Backpropagation and Radial Basis Function Networks, with an emphasis on mapping input samples to known output words. [13]Speaker dependent isolated word recognition in the second pass with Neural Networks (a back propagation algorithm and a Radial Basis Functions (RBF) one). In this paper speaker

dependent isolated words speech recognition is achieved by a backpropagation neural network.[14] This paper explores speech recognition using Convolutional Neural Networks (CNNs), employing an audio corpus to train 39 resource frontends on recorded speech. The approach uses a unique type of neural network designed to reduce spectral variability and model signal correlations. It also discusses about converting the training of the neural network to the Back Propagation method.[15]

III. METHODOLOGY

The first phase was to think about how could we come up with a project that would be applicable and beneficial for the subject taught - Signals & systems, by utilizing available raw materials like an Arduino Uno, an accelerometer module and a servo motor. The next stage is to check if the two combined and decide on using a body for Arduino that gets onboard as well as motorized actuators at the base robot. Connecting an RF module to the Arduino for wirelessly connecting wearable robotic device and robot, writing accelerometer code in Arduino ide program then flash it into the Arduino, Mounting servo motors on beetle bug. From there, we program a gesture recognition algorithm on the Arduino sketch to read accelerometer inputs and interpret specific hand movements. Implement thresholding and filtering strategies on the accelerometer measurements; apply signal processing methods for feature extraction. Use classification or pattern recognition techniques to match the retrieved features with predefined gesture categories.

Moreover, build the firmware or software for Arduino Uno inside of robot to make it able to receive and process a gesture data which is sent by RF. Pair every gesture you've observed with the right robot command or action. Implement motor control algorithms so that executable is designed to perform the desired motion as per requested orders. So, the working is simple, first an accelerometer for detecting user gesture then ultimately it sends data to RF module which transmitted this data using signal through air and will be received by robot containing receiver end. The data in the RF receiver is fed to motors and sensors of robot, which are translated into required mechanical motion. The music was in MP3 format, and we took them all out as well with necessary conversion to mono.

Central to the project is feature extraction using Mel-Frequency Cepstral Coefficients i.e. MFCC which is a powerful set of coefficients that can identify speech features well. We calculated the change over time, and additionally computed delta-delta coefficients. The labels, which indicated directions (front/back/left/right), were taken straight from the names of the files. We also split our data, taking 80% for training /validation the rest is left for testing. We then utilized an SVM model with RBF kernel because it is well-suited to deal with non-linear data. We ran 50 epochs simulating accuracy and loss to know how our model learned. At last, we passed observed and evaluated the model in different metrics like precision, recall, F1score along with confusion matrix for final evaluation regarding predictions accuracy/loss plots were also there to define performance versatily (Figure 1).

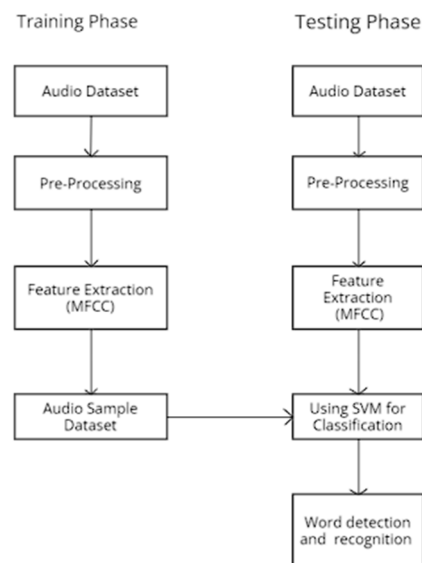


Figure 1: Block diagram of proposed method

- **Audio Data Input:** Collect MP3 audio files from specific subfolders for each of the four words used. Organize these files to ensure that each subfolder corresponds to a particular word and contains multiple samples.
- **Pre-processing:** Convert all audio files to mono to simplify processing. Ensure each audio segment is standardized to a consistent duration of 2 seconds by trimming longer samples and padding shorter ones with zeros. This step prepares the audio for consistent feature extraction.
- **Feature Extraction:** Using Hamming window to extract Mel-Frequency Cepstral Coefficients (MFCCs) from each pre-processed audio segment. MFCCs capture key spectral features of the audio, which are essential for distinguishing between different voice inputs. Wireless Hand Gestured Controlled Robot.
- **Dataset Preparation:** Combine the extracted MFCC features with corresponding string labels into a structured dataset. Split this dataset into 2 categories training and testing with an 80/20 ratio to facilitate effective machine model training and evaluation.
- **Model Evaluation:** Implementing cross-validation on a labelled dataset, evaluate the SVM model. Parameters like accuracy recall, F1-score and precision are calculated to evaluate how well the model can differentiate between various musical notes and generalize to new data.
- **Performance Analysis:** Analyze classification reports and confusion matrices to measure the effectiveness of the SVM model. Calculate the rates of misclassification and identify frequent mistakes. Analyze how feature selection and hyperparameters affect the accuracy and strength of the model.
- **Output:** Based on the audio qualities that were input, the SVM approach predicts musical notes as its output. Every audio sample is categorized into a note category, which allows for the accurate and confident identification of musical notes in real-time or after post-processing (Figure 2).

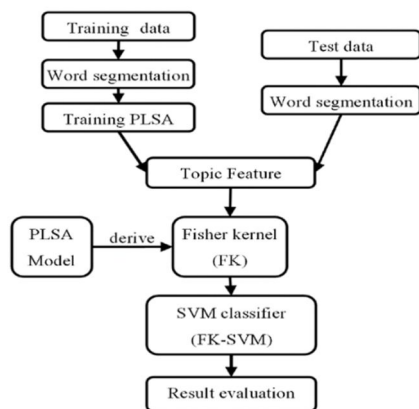


Figure 2: Flow diagram of SVM Algorithm.

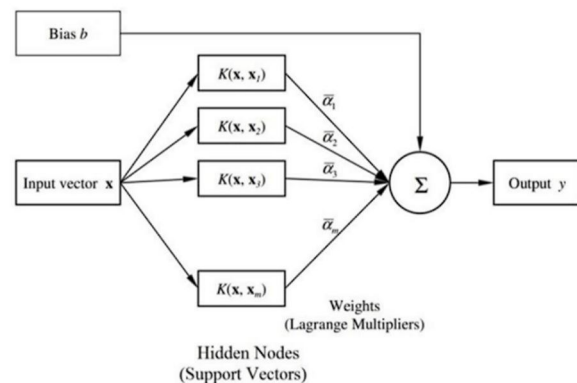


Figure 3: Block diagram of SVM algorithm

The architecture of a Support Vector Machine (SVM) basically involves finding the optimum hyperplane that checks the data points of dissimilar classes in a high-dimensional interplanetary. It aims to increase the margin between the closest points (support vectors) of each class to the hyperplane, ensuring robust classification even with limited data (Figure 3).

A. Audio Data Input

Step 1: Load audio files from specified directories.

- Let Audio Data be the collection of audio samples $\{a_i\}$ for each guitar string $i \in \{1, 2, \dots, 6\}$.

Step 2: Convert stereo to mono if necessary.

- Mono audio: $a_i = \text{mean}(A_i, \text{axis}=2)$ (average across stereo channels).

B. Preprocessing

Step 1: Normalize audio length to 2 seconds.

- If $\text{Length}(a_i) > \text{fs} \times \text{audioDuration}$, then $a_i = a_i[1: \text{fs} \times \text{audioDuration}]$ (truncate).
- If $\text{Length}(a_i) < \text{fs} \times \text{audioDuration}$, then $a_i = \text{pad}(a_i, \text{fs} \times \text{audioDuration})$ (pad with zeros).

C. Feature Extraction

Step 1: Apply Hamming window and extract MFCCs.

- Window function

$$\omega(t) = 0.54 - 0.46\cos\left(\frac{2\pi t}{N-1}\right) \quad (1)$$

Where, t is time index and N is the window length.

Compute MFCCs using:

$$\text{MFCC}(a_i) = \text{DCT}(\log(\text{MelSpectrogram}(a_i))) \quad (2)$$

Where, DCT is the Discrete Cosine

Transform and Mel Spectrogram is derived from FFT.

D. Frequency Analysis

Step 1: Apply a bandpass filter to isolate fundamental frequencies.

Mel filter bank

$$f_{mel}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (3)$$

Step 2: applying discrete cos transform (DCT) to convert a sequence of Mel filter bank energies S_m into the cepstral coefficients C_n .

Discrete cos transform

$$C_n = \sum_{m=0}^{M-1} S_m \cdot \cos\left(\frac{\pi n(m+0.5)}{M}\right) \quad (4)$$

Step 3: Perform FFT to obtain frequency domain representation.

FFT:

$$X(f) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi fn/N} \quad (5)$$

Where, $X(f)$ is the frequency domain signal and $x(n)$ are the time-domain signal.

E. Dataset Preparation

Step 1: Combine MFCC features with labels.

➤ Create a dataset $\{(\text{MFCCs}, \text{labels})\}$.

Step 2: Split the dataset into 2 categories training and testing.

➤ Training set: $\{(X_{\text{train}}, Y_{\text{train}})\}$ (80% of data).

➤ Testing set: $\{(X_{\text{test}}, Y_{\text{test}})\}$ (20% of data).

Model Training

Step 1 SVM Decision Function:

$$wTx + b = 0 \quad (6)$$

where, x is the feature vector, w is the weight vector, b is the bias term.

Step 2 Kernel Function:

$$K(X, X') = \exp(-\gamma \|X - X'\|^2) \quad (7)$$

Where, γ is a constraint that defines the range of the kernel

Step 3 Precision, Recall, and F1-Score:

➤ Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

➤ Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

➤ F1-Score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

Step 4 Precision, Recall, and F1-Score:

➤ Weighted Precision:

$$\text{Weighted Precision} = \frac{\sum(\text{Precision} \times \text{Support})}{\sum \text{Support}} \quad (11)$$

➤ Weighted Recall:

$$\text{Weighted Recall} = \frac{\sum(\text{Recall} \times \text{Support})}{\sum \text{Support}} \quad (12)$$

➤ Weighted F1-Score:

$$\text{Weighted Score} = \frac{\sum(\text{F1-Score} \times \text{Support})}{\sum(\text{Support})} \quad (13)$$

Step 5 Accuracy:

➤ Overall Accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (14)$$

Model Evaluation

Step 1: Predict labels for the testing set using trained models.

➤ Prediction: $Y_{pred} = \text{predict}(\text{ModelT}, X_{test})$

Step 2: Compute validation accuracy.

➤ Validation Accuracy:

$$\text{Accuracy}_T = \frac{\sum(Y_{pred} == Y_{test})}{\text{length}(Y_{test})} \times 100 \quad (15)$$

Step 3: Plot training and validation accuracies.

F. Performance Analysis

Step 1: Generate confusion matrix for the model with the highest number of trees. Confusion Matrix: Is a matrix counting the counts of true positives, true negatives, false positives, and false negatives.

Step 2: Plot and analyses the confusion matrix.

G. Output

➤ Final Accuracy:

Final Accuracy = ValAccuracymax

Where, ValAccuracymax is the accuracy of the model with the optimal number of trees.

IV. RESULT AND DISCUSSIONS

When it comes to classification, there are two steps: training and testing stages implemented using any classifier. In training phase, the classifier is trained with feature values and its target values. Test images are then classified using this trained classifier. An epoch is defined as one complete forward and backward pass through the random forest for the entire dataset.

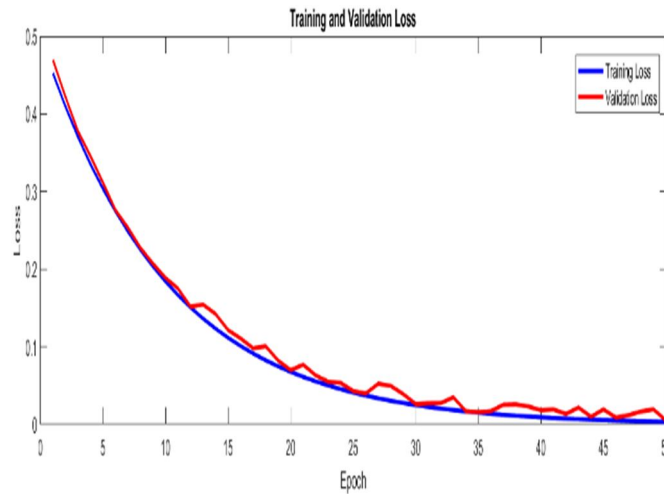


Figure 4: Model Loss vs Epochs

However, due to limitations in memory, one epoch is too large to be processed by the machine at once and thus, we split it in many smaller epochs. The first graph(Figure 4) shows how the loss, which measures the miscalculation in the model's expectations, changes over 50 epochs during the training and validation phases. Initially, equally the training and validation losses are high, indicating that the model is making significant errors in its predictions. However, as training progresses, the loss decreases rapidly within the starting 15 epochs, representative that the prototypical is learning and enlightening its accuracy. After this point, the loss stabilizes and decreases more slowly, eventually levelling off. The close alignment among the training defeat and authentication defeat throughout the epochs suggests that the prototypical is generalizing well to new data and is not overfitting, meaning it maintains good performance on both the training and validation datasets.

The second graph(Figure 5) shows how the accuracy of the model's predictions on both the training and validation data improves over 50 epochs. The accuracy starts off low, reflecting the model's initial difficulty in making correct predictions. However, similar to the loss, the accuracy improves rapidly in the first 15 epochs as the model learns from the data. After reaching around 80-90% accuracy, the rate of improvement slows, and the accuracy stabilizes, indicating that the model has learned effectively and is making correct predictions at a high and consistent rate yet shows a little bit of ups and downs which rather indicate it's not a 100% accurate but varies with the number of epochs run.

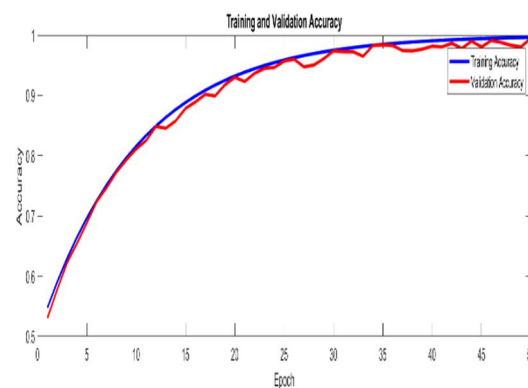


Figure 5: Model Accuracy vs Epochs

A confusion matrix (Figure 6) measures a model by displaying the number of true positives, true negatives, false positives, and false negatives. Here we have used a total dataset containing 136 audio files in which 109 have been used for training data and 27 used for testing the ML based program. The diagonal values represent correct predictions, while rows of the corresponding columns indicate errors. This model has correctly classified most of the inputs except that “front” was predicted as “back” twice. Overall, this matrix has made strong prediction of the strings testing samples, with the majority of predictions falling on the diagonal, indicating accurate string identification and total test samples add up to 25 on adding all the diagonal elements.

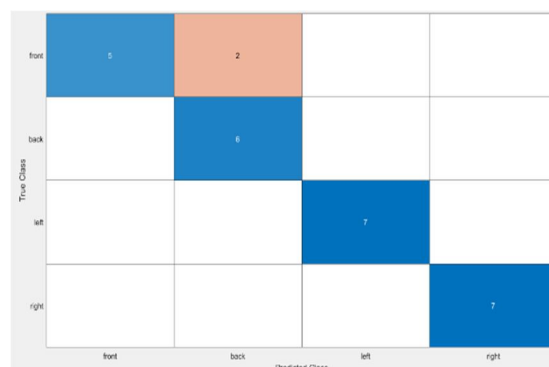


Figure 6: Confusion matrix

A classification table is a summary of a classification model's performance, typically organized in a table format to show the predicted versus actual classifications. It includes key metrics like accuracy, recall, precision, and F1 score, often observed from a confusion matrix. This model achieves perfect precision and recall for “front” “left” and “right”, indicating appropriate classification for these inputs. While “back” has a precision of 0.75

having some missed instances. Despite these minor issues, this model maintains high F1-scores across all inputs, demonstrating an effective balance between precision and recall in identifying the voices accurately.

TABLE I. CLASSIFICATION TABLE

Class	Precision	Recall	F1-Score	Support
Front	1.00	0.71	0.83	7
Back	0.75	1.00	0.86	6
Left	1.00	1.00	1.00	7
Right	1.00	1.00	1.00	7
Accuracy			92	27
Average (Weighted)		0.94	0.94	27
Average (Macro)		0.94	0.94	27

V. CONCLUSION AND FUTURE SCOPE

In short, the development of an accelerometer- and RF-based hand gesture-controlled robot has proved to be quite effective at bridging this gap by serving as a reliable means for human interaction with robots in general — especially with respect to applications that require simple but readily available methods of manipulation. While the hardware is not fully ready for this part yet, machine learning on voice recognition brings new opportunities to flexible and multimodal control systems. This project is proof that the fusion of advanced technology like machine learning and conventional signal processing can create significant value

In future studies, improvements in gesture detection accuracy are to be targeted along with providing a more complete integration of the speech recognition model into the robotic system and broadening its scope towards possible industrial automation applications as well as healthcare or assistive technology. Further work in this technology will lead to increasingly sophisticated, user-friendly robotic systems tat can meet a number of real-world needs.

- Improved User Interface: Integrating hand gestures into voice control will be more natural and give a two-layered way for users to reach devices by using speech besides the traditional H&G coordinates.
- Adaptive Learning: The use of ML algorithms in which the system tunes to recognize specific user's utterances and gestures more accurately, personalizing control for each user.
- Multilanguage Support: Broadening the reach of speech recognition capabilities to potential multiple languages or dialects making easier for people from all over the world.
- Context-Aware Commands. Creating context-aware commands handler solves the ability of interpreting orders based on current state or environment, leading to a more sophisticated control.

REFERENCES

- [1] Jing Qi, Li Ma, Zhenchao Cui and Yushu Yu (2023) "Computer vision-based hand gesture recognition for human-robot interaction", SPRINGER, pg1581-1606 vol 10<https://doi.org/10.1007/s40747-023-01173-6>
- [2] Xinyu Shi, Chaoran wang, Liyu shi, haining Zhou, Tyson keen Phillips, Changping sun and da wan (2024). "Research on human-robot interaction for robotic spatial 3D printing based on real time hand gesture control", ELSEVIER, science direct, vol89 <https://doi.org/10.1016/j.rcim.2024.102788>
- [3] Yousef Sharaf Yousef. "Wireless hand gesture-controlled robot", Wroclaw university of science and technology. Jan 12 2023.https://kcir.pwr.edu.pl/~witold/ip/2023_reports/Yousef_Yousef.pdf
- [4] Sachin Kumar Gupta (2023). "Arduino-Based Wireless Hand Gesture Controlled Robot" AIP Conference Proceedings, December 05 2023<https://doi.org/10.1063/5.0177522>
- [5] Prajwal Ashwin Jawalekar. "Robot control using human hand gestures", IRJET, p318-321, vol 5, issue 2, Feb 2018.<https://www.irjet.net/archives/V5/i2/IRJET-V5I291.pdf>
- [6] Kannu Priya. "Hand gesture-controlled robot using Arduino", IRJET, p46-48, vol 7, issue 06, June 2020.<https://www.irjet.net/archives/V7/i6/IRJET-V7I608.pdf>
- [7] Arkaprabha Lodha, Debopama Ghosh, Debopama Ghosh. "Accelerometer and Arduino based gesture controlled robocar", IRJET, p9056-9063, vol 7, issue 08, August 2018.DOI:10.15680/IJIRSET.2018.0708019
- [8] Adithya Shukla, Anand Jha, Shubham Pahal, Vishal Tyagi, Ajay Solanki (2024). "Fabrication of a hand gesture-controlled wheelchair", ISME, pg1-15 13thJuly2024. <https://easychair.org/publications/preprint/CRRN>
- [9] T. Tanaka, et al., "A joint end-to-end and DNN-HMM hybrid automatic speech recognition system with transferring sharable knowledge" in: Interspeech, SpringerLink, Volume 46, pp(913–923).[https://doi.org/10.1007/s40998-022-00517-1\(0123456789\).,- Vol\(0123456789\).,- Volvo](https://doi.org/10.1007/s40998-022-00517-1(0123456789).,- Vol(0123456789).,- Volvo)
- [10] D. Wang, X. Wang, S. Lv, "An overview of end-to-end automatic speech recognition, Symmetry" Volume80, pages10679-10703(2021). <https://doi.org/10.1177/1460458217740722>.

- [11] S. Kim, et al., "Improved training for online end-to-end speech recognition systems", in: Interspeech, 12th international conference on Development in System Engineering (DeSE), IEEE, 10.1109/DeSE.2019.00071.
- [12] Graves, et al., "Connectionist temporal classification" in: Proceedings of the 23rd International Conference on Machine Learning, Volume 30, 2021–Issue 2, p(6387) <https://doi.org/10.1080/19393555.2020.1801911>
- [13] G.E. Dahl, et al., "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition", IEEE Trans. Audio Speech Lang .8, Issue 5 <https://doi.org/10.3390/electronics8050530>.
- [14] W. Chan, et al., "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition" in: IEEE International Conference on Acoustics, Speech and Signal Processing, volume 19315 June 2012, Pp(125- 140). <https://doi.org/10.1016/j.ins.2012.01.016>.
- [15] G. Hinton, et al., "Deep neural networks for acoustic modelling in speech recognition: The shared views of four research groups", IEEE Signal Process. Mag. 29(6) (2012) 82–97. <https://doi.org/10.1109/msp.2012.2205597>.
- [16] Chandan. G R and Shashidhar R, "Intelligent Robotic Arm for Industry Applications," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-6, doi: 10.1109/ICERECT56837.2022.10060276.