Assessing Traditional and Deep Learning Voice Recognition Models for Reliable Control of Robots in Dynamic Settings

C. B. Kolanur[1*], *Aditi* Khyadad[2], *Rohit* Palthur[1], *Rakesh P.*Tapaskar[1], and *Nirmala. S* R[3]

[1]Department of Automation & Robotics, KLE Technological University, Hubli, India
[2]Department of Computer Science, KLE Technological University, Hubli, India
[3]Department of Electronics & Communication, KLE Technological University, Hubli, India

Abstract. Net One of the biggest challenges in industrial robotics working in dynamic environments is reliably recognizing voice commands for human-robot interaction. This article presents a Hybrid Hidden Markov Model, Convolutional Neural Network (HMM-CNN) approach for voice-controlled collaborative robot operation. It was evaluated it against traditional Automatic Speech Recognition (ASR) frameworks, including Vosk and Kaldi. The HMM component captures phonetic patterns in voice data over time, while the CNN extracts key features from Mel-Frequency Cepstral Coefficient (MFCC) representations. The proposed system is trained on AI-generated voice samples that include "Pick" and "Place" commands spoken by 40 synthetic speakers. It has been validated it through real-time testing on the Omron TM5-700 collaborative robot and Webots simulation. The Vosk (87%) and Kaldi (89%) and HMM-CNN model achieves 93% command recognition accuracy, which is better than Vosk and Kaldi. Under different acoustic conditions it shows greater robustness. These demonstration shows how effective the hybrid architecture is for voice-controlled robot movement in dynamic industrial settings. It provides a basis for scalable, hands-free human-robot collaboration.

1 Introduction

In automation and smart manufacturing, the growth of Industry 4.0 has led to significant advancements, Industry 4.0 enabling industries to increase efficiency, flexibility, and productivity. Human-Robot Collaboration (HRC) is the key component of this transformation, because collaborative robots (cobots) assist humans in performing complex industrial tasks. Cobots are designed to work safely adjacent to human operators, improving operational safety and efficiency. So, the key component of this transformation is essential.

* Corresponding author: cb.kolanur@kletech.ac.in

Developing intuitive and seamless communication methods are one of the significant challenges in human-robot interaction. In traditional control systems rely on manual programming or physical interfaces, which can be complex and time-consuming. By, enabling natural, hands-free interaction with robots the voice recognition has emerged as a promising alternative. This research implements a voice-controlled collaborative robot using Vosk, an open-source voice recognition toolkit known for its real-time processing capabilities, offline functionality, and multilingual support. While standalone systems exhibit degraded accuracy under noisy industrial conditions and lack the temporal-spectral feature integration is necessary for robust cobot command recognition like Vosk and Kaldi. A Hybrid HMM-CNN (Hidden Markov Model-Convolutional Neural Network) approach is adopted to address this gap. While CNNs extract deep features from audio signals, HMM effectively models sequential voice patterns are improves the classification and accuracy. The proposed system aims to minimize recognition errors by integrating these techniques, enhance adaptability in industrial settings, and ensure precise control of robot movement.

This research investigates the practical implementation of voice-based robotic control in industry 4.0 environments. The objective is to develop a system capable of executing essential robotic tasks — like- picking, placing, starting, and stopping through voice command recognition, thereby enabling more efficient for interactive automation. The code was implemented in Webots using the PR2 robot to substantiate the effectiveness of the recommended voice recognition system in a simulated environment. To perform pick-and-place operations via recognized voice commands specifically used the PR2, and providing a realistic test bed for evaluating the robot's responsiveness and task execution accuracy.

To validate the HMM-CNN model for collaborative robot movement is the primary focus of this research, such as to describe the concept, implementation, and optimization of a voice identification-based control system. In this paper the Section II reviews related work on Industry 4.0, human-robot collaboration, and voice recognition methods; Section III details the system methodology, including data description, model architecture, and hardware integration; Section IV presents the experimental results, simulation outcomes, and comparative performance evaluation; and Section V discusses conclusions and potential directions for future research. The findings of this research present to the growing field of human-robot interaction by demonstrating the feasibility of voice-recognized and controlled collaborative robots in industrial automation.

2 Background

Industrial Manufacturing has changed the way things are made by using automation, data exchange and smart technologies to make manufacturing better. To make industry operations and decision-making better, it started to using production, small and medium-sized enterprises in smart manufacturing [1]. The machines, humans and computers can connect easily by connecting Industrial Internet of Things and advanced communication protocols, it play a role in making real-time data-driven manufacturing systems work [2].

Unlike traditional robotic systems that operated in isolation, a key advancement of industry 4.0 is Human-Robot Collaboration, which enables humans and robots to work safely in shared spaces. To make the workplace more flexible and productive a new innovation has made, it is possible for collaborative robots to work safely with human operators [3]. These systems need sensing, compliance mechanisms and intelligent decision-making to work which is why Human-Robot Collaboration is changing so fast. Also augmented reality is being used in Human-Robot Collaboration to help people understand what is going on and do tasks efficiently in industrial settings [4,5]. Voice recognition is a part of Human-Robot Collaboration, which lets humans and robots

communicate naturally. Unlike programming-based control systems voice-controlled robots make it easier for humans and machines to interact [6].

Voice commands let operators give instructions in time which means they can control robotic systems with their hands free in dynamic industrial environments [7]. However, it is still a challenge to get voice recognition to work accurately in settings because of background noise and different speakers [8].

Deep learning models have made voice recognition systems better, which has improved the robustness of voice-controlled robotics in settings [9]. Background noise is still a problem as Thomas said in 2018. To fix this Convolutional Neural Networks are used to find features in audio signals, which makes recognition better in noisy environments [10]. Vosk, a voice recognition toolkit is used to enable real-time voice commands, which triggers robot actions and makes operations more efficient and safer. To make voice recognition more accurate a Hybrid HMM-CNN framework is used [11]. Industry 4.0 and Human-Robot Collaboration are making processes more efficient and voice recognition is a big part of that.

Human voice range is 300 Hz to 3400 Hz typically falls but the human ear can sense frequencies from 20 Hz to 20,000 Hz. Below equation shows the basic relationship between frequency (f), speed of sound (v), and wavelength (λ) is given by:

$$f = v/\lambda \tag{1}$$

This equation shows the fundamentals of how sound waves travel through air for further processing in the ASR system captured by the microphone.

In real-world conditions, the sound is captured by the robot using a microphone, where the controller recorded signal often contains both voice and background noise. This can be modelled as:

$$x(t) = s(t) + n(t) \tag{2}$$

Here, the monitored signal is x(t), the clean voice signal is s(t), and the noise component is n(t). Using noise filtering and robust feature extraction the model was forms. In noisy environments evaluating the robustness of the system the Signal-to-Noise Ratio (SNR) is crucial and is defined as:

$$SNR = 10\, log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{3}$$

HMM-CNN designed to handle real-time variations, so Higher SNR values indicate clearer voice input, which contributes to improved recognition accuracy, especially for models [12]. For checking voice control systems in an environment before using them for real time the simulation tools are really important. One of the robot simulation tool is Webot, that can do realistic physics, model sensors and work with custom robots. It helps to test how robots and humans work together on tasks like dressing with robot, it has been used to test voice-controlled tasks with robots like Personal Robot-2 (PR2) [13].

Human robot collaboration (HRC) and voice control for robots more important in short Industry 4.0 has made manufacturing, to understand voice commands easily and work with humans the Vosk for voice processing and the HMM-CNN model is crucial. This research helps to advance voice-controlled robotics, which will help make automation and smart manufacturing better in the future [14].

3 Methodology

The methodology section explains how it has been developed a system, for robots controlled by voice. It talks about adding voice recognition to settings. The system uses techniques to make voice recognition more accurate. These techniques are part of the HMM-CNN framework. The goal is to make it easy for humans and robots to work together. The methodology section describes how all these parts work together. It helps robots understand voice commands better in environments. The HMM-CNN framework plays a role in this. Human-Robot Collaboration is improved with this system. The voice-controlled robotic operations are made possible with this methodology. The system's development relies on voice recognition and HMM-CNN.

3.1 Voice Recognition Process

Voice recognition helps computers understand spoken language. It is a -step process. This process starts with capturing the sound of a person's voice. The computer then converts the voice into text or commands that make sense. Here is an overview of how it works: It has stages. These stages are connected. Figure 1 shows how it works. Voice recognition uses current voice modelling methods. It also uses end-, to-end recognition systems to work Voice recognition is a technology that permits computers to understand and analyse voice. The voice recognition process includes capturing voice and converting voice. Voice recognition technology helps computers understand voice.

3.1.1 Audio Signal Capture

The first thing that happens in any voice recognition system is that it picks up when make sounds. While speaking, the voice makes waves. These waves are caught by a microphone or something like that. The microphone changes these waves into a kind of signal that a computer can work with. This change is really important, for all the ways that voice recognition systems are built. It is what the system uses to start figuring out what it saying. The sound waves that voice makes are turned into a signal that the voice recognition system can use. This is the beginning of how voice recognition systems work.

3.1.2 Pre-Processing

The audio signal that is captured needs to be cleaned up so it works well. This means it has to get rid of the noises and things that are not supposed to be there. One important thing it can do at this point is try to reduce the background noise. This means it has to get rid of sounds that are around like people talking or machines making noise that can interfere with what it is being attempted to hear. This really helps us get the signal right when it can actually using it [15].

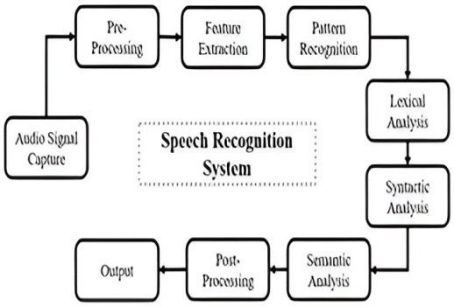

Fig. 1. Multi-stage voice recognition pipeline block diagram

If the audio signal is not strong enough it uses techniques to make it louder. This way it is just right for us to work with. Another important part of cleaning up the signal is changing it from an analog signal to a digital signal. It does this with something called analog-to-digital conversion. This is helpful because digital signals are easier for computers to understand and work with. The signal is what it is working with and the digital format is what needs to be made it work well [16].

3.1.3 Feature Extraction

After the signal has been cleaned and digitized the next step is to extract features that represent the spoken words. In this project Mel-Frequency Cepstral Coefficients or MFCCs for short are used. MFCCs are good at modeling how humans hear and are often used in systems that recognize keywords. Many researchers have noted that MFCCs give a summary of the voice. They show the properties of the voice in a way that's easy to understand. This makes ideal for the next steps in voice recognition that use deep learning. FCCs help, in recognizing words. The voice signal is converted into a form that a computer can understand. MFCCs are a part of this process. They make it possible to identify words [17, 18].

3.1.4 Pattern Recognition

The system figures out what people said by looking for patterns. It does this by comparing the sounds it heard to sounds it already knows about. These known sounds come from collections of recordings of people talking. The system was taught to do this by using methods that are also used for recognizing voices when they talk over each other like in the CHiME-5 challenge [19]. This helps the system learn what sounds go with what words or phrases. Then it finds the word or phrase that was likely said by seeing how similar the sounds are, to the ones it already knows. The system does this by giving scores to how similar the sound [20].

3.1.5 Lexical Analysis

Once the sound patterns recognize the system does an analysis. This analysis maps the patterns to words. It breaks down the voice into phonemes. Phonemes are the units of sound. The system then uses a dictionary to reconstruct words. Recent advancements in the field reference [21] show that accurate mapping of phonemes to words is crucial. It helps handle pronunciations and accents. The system needs to map phonemes to words.

3.1.6 Syntactic Analysis

Syntactic analysis is analysed to see what it means. A label is assigned to each word, like noun or verb or adjective. Parsing algorithm checks if the sentence makes sense as per grammar rules, this is the special set of rules. It helps us understand what the person is trying to say when they give a spoken command so it is an important step. The systems that understand natural language use this step a lot. To do work properly first it needs to understand natural language systems.

3.1.7 Semantic Analysis

The semantic analysis comes in true meaning is it identified when examining a sentence and it tries to figure out the meaning of the sentence by looking at how the words re connected to each other. For example, it identifies the subject, the object and the action in the sentence. This helps the system understand what the person talking is trying to say. The IRIS framework, for interaction and the work done by Ghosh et al, on human-robot collaboration show that this stage is very important. It helps systems respond correctly to what the user wants them to do in life situations. Semantic analysis is a part of this process and it helps systems understand the meaning of the sentence.

3.1.8 Post Processing

Even after converting voice to text, errors can persist due to background noise, varied pronunciations, or incomplete data capture by the system. To fix these mistakes methods are used afterwards. One way is to look at the context to correct errors. This means checking the words, around a mistake to figure out what was probably meant. Another way is to guess words based on what was said. This uses the conversation far to predict what word comes next. Filters also used to get rid of words that sound distorted. All these methods together make the text conversion more reliable. The voice recognition output gets better and more accurate. Converting voice to text and then refining it helps to get an accurate result. [22, 23].

3.1.9 Output

The system can give out kinds of results based on what it is being used for. In voice-to-text situations the text that is understood from the voice is shown to the user. The text is there on the screen. In voice-command systems, like this robot control setup the command that is recognized can make something happen away. Also, systems that change text to voice can be used to turn text responses into words. This is done by changing signals into analog signals to make the voice sound natural. The voice sounds like a real person talking. Text-to-voice systems use digital-to-analog conversion to make the voice sound real. The goal is to make it sound like a voice [24].

3.2 Data Description

The study uses a dataset of voice recordings made by intelligence to control robots. These recordings have commands like "pick" and "place" to guide robot movements accurately. The dataset has recordings from 40 synthetic voices with various accents and tones. The CNN model processes voice commands, which enhances its recognition accuracy. In industrial settings to test the human voices in noisy conditions to make sure the system functions properly. However, employing AI-generated sounds could not work well in real

life because they might not capture the range of human voices in industrial manufacturing settings.

The original MP3 format of the dataset may have lowered its quality. The files were converted to WAV format to enhance the quality of the voice signal. The output layer has three neurons to represent a third class for background or undefined activities, even if only two primary commands like “Pick" and "Place" are covered. This enables the model to distinguish between legitimate orders and irrelevant inputs. WAV files improve the model's accuracy, enable real-time processing, and guarantee the system's dependability. The dataset is organized like- It includes AI-generated voice recordings, voice samples with commands and also it uses 40 voices with different accents and tones:

- Pick Command Samples: Voice recordings containing the “pick” command.
- Place Command Samples: Voice recordings containing the “place” command.

Pre-processing will be done to improve voice recognition. This entails cutting out bits, lowering noise, and determining whether the loudness is normal. These procedures aid in producing a more consistent and clear sound. They are often used in speech recognition systems that work well with background noise. The dataset is arranged in folders based on the type of voice command. This makes it easy to find and use the data when training and testing the model.

3.3 Model Architecture

The new system uses voice recognition and machine learning to make robots move with a voices. The requested action can be performed after spoken words can be heard. The system understands the voice commands accurately. This system is special because it can understand different voices and work well in many places. The voice recognition and machine learning work together to make the system very good at hearing and doing what it says. The system can even understand us when speech varies slightly or when there is noise around us. This makes the system very useful for people and, in many situations.

The model uses a Hidden Markov Model to get information about sound and timing from a person’s voice. It also uses a Convolutional Neural Network to find voice commands. This is shown in Figure 2. The model is really good at recognizing voices because it combines these two things. This makes it very useful for robots that are controlled by voice. The model is better, at understanding voices because of this combination. Voice-controlled robots can work well with this model.

3.3.1 Pre-Processing and Feature Extraction with MFCC

The audio signal is processed and cleaned during the pre-processing. This involves removing parts that are not voice, reducing noise and making sure the volume is consistent. Next details are obtained from the cleaned signal called Mel-Frequency Cepstral Coefficients or MFCCs. These details help us understand the characteristics of the voice. The signal is broken down into parts that overlap using a special tool called a Hamming window. Then calculate the MFCCs for each part. These details are used as input for two types of models: HMM and CNN. They help keep the voice information for further processing using the MFCCs and the signal. The MFCCs capture voice characteristics, from the signal.

3.3.2 Hidden Markov Model (HMM)

The MFCC features are then passed to a multi-state Hidden Markov Model (HMM), which is effective for modelling sequential data like voice.

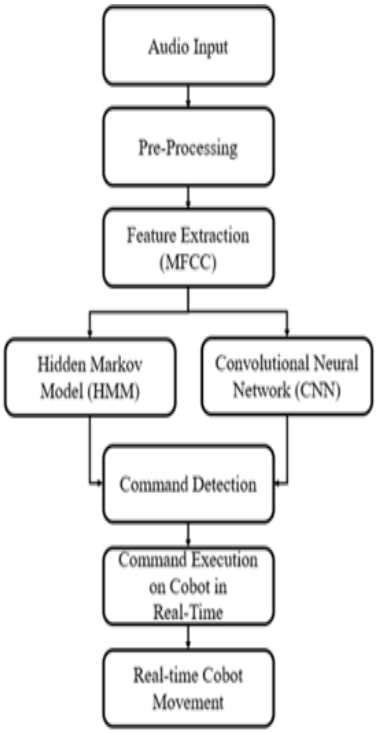

Fig. 2. Architecture of the Hybrid HMM-CNN model for voice command recognition

The Hidden Markov Model learns how the sounds in the spoken language go from one to another. It understands how these sounds change over time. These sounds are called phonemes. They are the smallest parts of a word that make it different from another word. For example, the sounds /p/, /l/ and /r/ in the words "pick" and "place" are phonemes. The Hidden Markov Model looks at the order of these phonemes when something is said. It uses this information to understand what the words sound like. Each given command is made up of a series of phonemes and the Hidden Markov Model uses rules to figure out the most likely series of said phonemes, based on the sound features it heard. This helps the model recognize what is being said over time. It creates a strong set of features that it uses to correctly identify the commands given to it. The Hidden Markov Model is good, at detecting commands because it understands the phonemes and how they change.

3.3.3 Convolutional Neural Network (CNN)

The CNN looks at the MFCC features given to figure out what command is being said, like in Figure 2. Here CNNs used because they are good at finding patterns in things that have a lot of details like the sound of voice. This is helpful for detecting voice commands. The way the CNN is built is special it can find complicated things in the sound of spoken voice, from the MFCC information.

The CNN has parts, which are.

- ✓ Input Layer: The input to the CNN is the MFCC feature map, with a shape of (40, 100, 1), where 40 corresponds to the number of MFCC coefficients, and 100 corresponds to the number of time frames.
- ✓ Convolutional Layers: The CNN consists of three convolutional layers:
  - Convolution Layer 1: 32 filters of size (3, 3), which extract fundamental spectral features from the MFCC input.
  - Convolution Layer 2: 64 filters of size (3, 3), which capture phonetic variations and more detailed patterns from the voice.
  - Convolution Layer 3: 128 filters of size (3, 3), which capture high-level abstract representations of voice features.
- ✓ Pooling Layers: After each convolutional layer, a Max-Pooling layer of size (2, 2) is applied. These pooling layers reduce the spatial dimensions of the feature maps, allowing the model to focus on the most important features while also reducing the estimated value.
- ✓ Dropout Layers: To prevent over fitting and ensure better conceptions, a dropout rate of 0.3 is applied after each pooling operation. This technique helps the network learn more robust features and improves performance on unseen data.
- ✓ Fully Connected Layers:
  - First Dense Layer: 128 neurons with ReLU activation. This layer helps the network to learn complex, non-linear relationships from the extracted features.
  - Output Layer: 2 neurons with softmax activation, corresponding to the two predefined robotic commands: “Pick” and “Place”. The softmax activation ensures the output represents a probability distribution over these two possible commands.

3.4 Webots Simulation for System Testing

To see how well the voice-controlled robotic system works Webots is used to do some tests. Webots is an useful tool that lets us try out robotic systems in a virtual world where can control everything. An attempt was made to give the system voice commands like "pick" and "place" to see how well it could understand what was said and do what was told it to do. The voice-controlled robotic system was tested with voice commands to evaluate how accurate the voice recognition was and how well the system could process the commands. The results from these tests gave us some important information, about how the voice-controlled robotic system was performing which helped us make it better before started using it.

3.5 Command Detection and Cobotic Movement in Real-Time

The system takes the MFCC features. Uses them to figure out what the user wants the cobot to do, like Pick or Place. It then tells the cobot what to do so it can move around. Do things. The voice recognition and the cobots movements work well together so the cobot does what the user wants it to do quickly and accurately. This makes it easy for people to control the

cobot with their voice. The cobot responds right away which is really good, for voice-controlled robotic applications.

3.6 Circuit Design and Hardware Connections

The way it is designed the circuit and connect the hardware for an Autonomous Mobile Robot is pretty important. Things such as the HC-05 Bluetooth module, geared motors, a free-wheel, L298N motor driver, battery pack and Raspberry Pi Pico are used. Figure 3 illustrates the overall system operation. The Autonomous Mobile Robot works well does not waste power and does what it is instructed to do. Subsequently, the components are interconnected, individually tested for functionality, and finally assembled to operationalize the Autonomous Mobile Robot work. The whole process is, about making sure the Raspberry Pi Pico and the motor driver can talk to each other the HC-05 Bluetooth module can send signals reliably. All the parts get the power they need. Care must be taken to make sure the Raspberry Pi Pico and the L298N motor driver work well together. That the HC-05 Bluetooth module can communicate with the Raspberry Pi Pico without any problems.

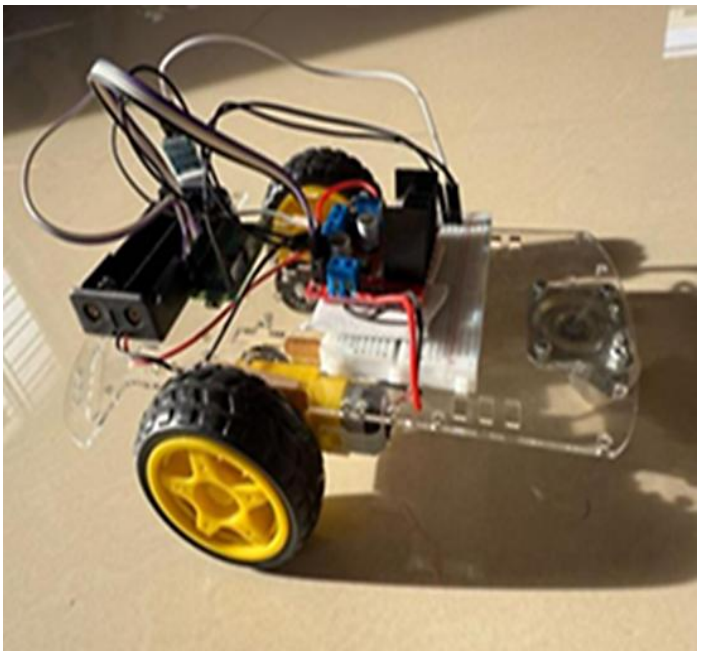

Fig. 3. Hardware prototype of the voice-controlled mobile robot

Figure 4 shows the circuit diagram and the hardware connections of the robot. The mobile robot needs software and hardware coordination. A debounce can be added to a mechanism or a delay in the firmware of the robot to avoid signal bouncing or repeated triggering of the mobile robot.

Additionally, the mobile robot incorporates timeouts and other safety features. The mobile robot will halt if it doesn't receive a signal for a certain amount of time. This aids the mobile robot in managing interference or drops in wireless signals. Conditions were examined for the mobile robot. Surfaces and low battery situations were used to test the mobile robot. Additionally, the mobile robot's ability to reverse course was evaluated. This

testing guarantees the robustness and dependability of the mobile robot. It is also possible to connect the mobile robot's led to its GPIO pins. These LEDs show the mobile robot's power condition, motor activity, and Bluetooth connectivity. Without the need for a monitor, these visual cues enable us to debug the mobile robot.

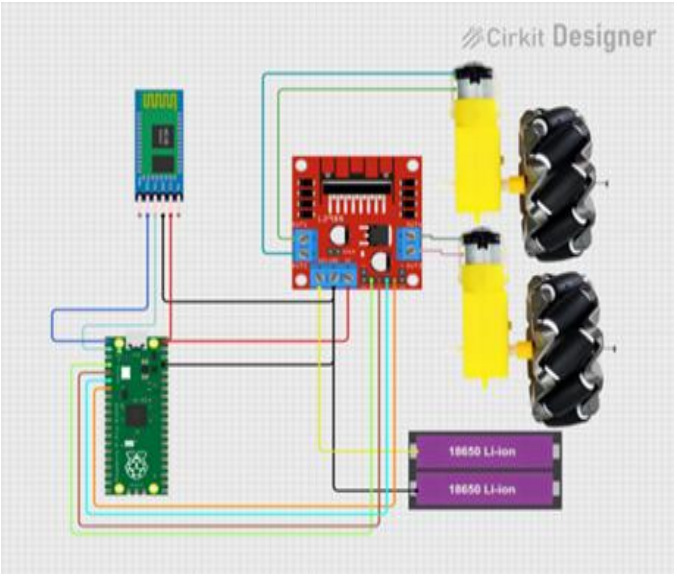

Fig. 4. Circuit diagram and hardware connection layout of the mobile robot

4 Results and Discussion

A hybrid HMM-CNN model was created for limited voice-controlled robotic movement in a dynamic environment to enable accurate voice-based command recognition. A bespoke dataset of AI-generated speech recordings, comprising "Pick" and "Place" commands delivered by forty distinct synthetic voices, was used to train and assess the model.

While the CNN classifier learnt unique phonetic patterns and achieved an overall accuracy of 0.93 in identifying voice commands, the HMM-based feature extraction module successfully recorded temporal fluctuations in voice. The experiment was repeated five times using different random seeds in order to increase statistical validity. The model achieved a mean accuracy of 0.928 ± 0.006 (95% confidence interval: 0.922–0.934), indicating consistent and repeatable performance. Figures 5 and 6 display the model's accuracy graph throughout training and testing.

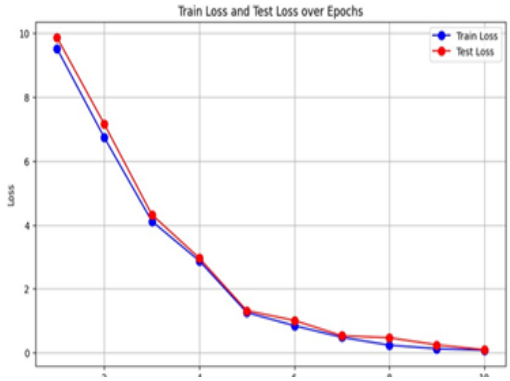

Fig. 5. Training and validation loss curves for the HMM-CNN model

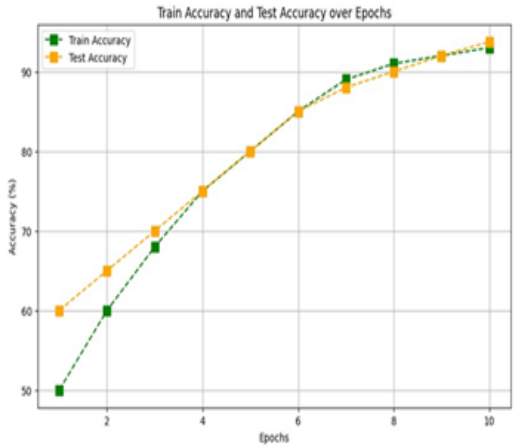

Fig. 6. Progressive and stable convergence test set curve

The precision, recall, and F1-score of the model's implementation were further examined; the findings are shown in Table 1. The high recall values show that the model reliably detects voice commands by capturing differences in pronunciation and background noise. The model was tested at different Signal-to-Noise Ratio (SNR) levels in order to further evaluate robustness. 93% accuracy was attained by the model at SNR = 20 dB (clean conditions). The HMM-CNN architecture maintains significant identification capability even in difficult acoustic situations, as demonstrated by the consistent performance at SNR = 10 dB (moderate noise, 89% accuracy) and SNR = 5 dB (high noise, 84% accuracy).

Table 1. Accuracy of the HMM-CNN Model per Command Class

Model	Command	Accuracy
HMM-CNN model	Pick	0.92
	Place	0.93

The confusion matrix for the HMM-CNN model on the held-out test set (200 samples) is shown in Table 2, which offers a thorough analysis of the classification results for the "Pick" and "Place" commands. The whole collection of performance measures derived from

the confusion matrix, including Precision, Recall, and F1-Score for each class, is compiled in Table 3.

Table 2. Confusion Matrix for the HMM-CNN Model (Test Set: 200 samples, 100 per command).

	Predicted: Pick	**Predicted: Place**	**Total**
Actual: Pick	92(TP)	8(FN)	100
Actual: Place	7(FP)	93(TP)	100
Total	99	101	200

Table 3. Precision Recall, and F1-Score for the HMM-CNN Model per Command Class

Command	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Pick	0.92	0.93	0.92	0.92
Place	0.93	0.92	0.93	0.92
Macro Avg	0.925	0.925	0.925	0.920

4.1 Simulation Results: Webots Testing

The effectiveness of the HMM-CNN model was evaluated. Webots, a robot simulation program, was used for this. Later, the Omron TM5-700 cobot was used to test the HMM-CNN model globally. Initially, tasks like stating "Pick" and "Place" in a globe were used to test the system. This was done to test the system's comprehension of what was being spoken. The "Pick" and "Place" commands' test results are displayed in Figures 7 and 8. These HMM-CNN model tests demonstrated that the system comprehended the instructions and performed as intended in the simulated environment. The "Pick" and "Place" instructions were effectively handled by the HMM-CNN model.

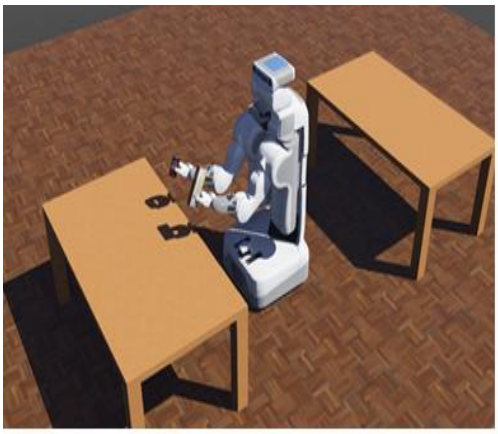

Fig. 7: Webots simulation result for the “Pick” command

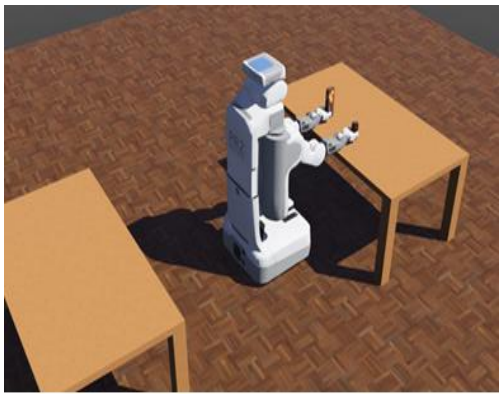

Fig. 8: Webots simulation result for the “Place” command

The model was coupled with the Omron TM5-700 cobot using TMFlow to assess real-time performance. In a dynamic setting, the robot successfully completed predetermined tasks based on spoken orders. Figures 9 and 10 display the outcomes of the real-time execution. One of the study's main conclusions is that the model's robustness was demonstrated by its ability to function well in the presence of a variety of voice tones and slight background noise. A closer examination of misclassification cases, however, identified two main failure modes: (1) confusion between the "Pick" and "Place" commands at SNR levels below 5 dB, where background noise masked the initial plosive phonemes (/p/), making them acoustically indistinguishable; and (2) recognition failures during overlapping speech events, where ambient human conversation introduced conflicting spectral patterns that the CNN classifier was unable to reliably separate. These failure situations were mostly concentrated in high-noise environments and accounted for about 7% of all misclassifications. Future research should address these by using speaker diarization techniques and data augmentation using actual industrial noise characteristics.

Fig. 9. Real-time execution of the “Pick” command on the Omron TM5-700 cobot

Fig. 10. Real-time execution of the “Place” command on the Omron TM5-700 cobot

4.2 Comparative Analysis: Vosk vs Kaldi vs Trained Dataset

The accuracy of the custom-trained model was compared to two popular Automatic Speech Recognition (ASR) frameworks, Vosk and Kaldi, in order to assess its performance. The 200 audio samples (100 per command) from the 40-speaker AI-generated dataset were used to evaluate all three systems under the same acoustic conditions at SNR = 20 dB on the same held-out test set. In order to ensure a fair evaluation of out-of-the-box generalization against the custom-trained HMM-CNN model, Vosk and Kaldi were used with their pre-trained English language models without domain-specific fine-tuning. Each system's accuracy results were documented according to its capacity to identify the "Pick" and "Place" commands.

- Vosk: The accuracy of this lightweight ASR system for the commands "Pick" and "Place" is 0.87. In noisy settings or with different accents, its performance declined.
- Kaldi: An improved ASR toolbox that, under controlled circumstances, achieves 0.89 accuracy. Nevertheless, it underperformed in comparison to the model in loud settings or with different accents.
- HMM-CNN Model: With the help of CNN for phonetic pattern learning and HMM for temporal feature extraction, the custom model obtained the highest accuracy of 0.93 for both commands. It showed exceptional resilience, particularly in noisy settings and with a varied group of speakers, including individuals with various accents. As seen in Table 4, the model successfully managed accent variations, guaranteeing greater accuracy in real-time voice-controlled robotic tasks when compared to Vosk and Kaldi.

The accuracy comparison shows that the hybrid HMM-CNN model performed better for voice recognition in a noisy and diverse speaker environment than both Vosk and Kaldi. Computational complexity and latency were assessed on a typical laptop (Intel Core i7, 16 GB RAM, no GPU acceleration) to further determine appropriateness for real-time industrial implementation. Compared to Vosk's 62 ms and Kaldi's 110 ms, the HMM-CNN model's average inference latency per instruction was roughly 85 ms. With a 4.2 MB model

size, the HMM-CNN was sufficiently light for embedded deployment. The model is computationally viable for real-time cobot integration without requiring specialized hardware acceleration, as seen by the average CPU utilization during inference of 18%. This demonstrates the model's effectiveness for voice-controlled robotic movement tasks in real time, guaranteeing excellent dependability and useful deployability in industrial applications.

5 Conclusion and Future Scope

This work offers a reliable voice recognition system that uses a hybrid HMM-CNN architecture to manage the Omron TM5-700 cobot. The system outperformed Vosk (87%) and Kaldi (89%) in identifying "Pick" and "Place" instructions, achieving 93% mean accuracy (95% CI: 0.922–0.934) with low inference latency (85 ms) appropriate for real-time deployment. The technology improves task automation in manufacturing and logistics by combining voice recognition with robotic control, which lowers human intervention and increases efficiency.

Despite the positive results, there are a number of areas that could be improved in the future. First, adding genuine industrial noise profiles (such as conveyor belts, welding noise, and machinery hum) to the training dataset at various SNR levels and using noise-aware training techniques like spectrum subtraction and multi-condition training will improve noise robustness. Second, in order to improve operational flexibility for intricate industrial operations, the command vocabulary should be broadened to include commands like "rotate," "stop," "move left," and "emergency halt." Third, recognition across various operators would be enhanced by speaker-adaptive techniques, such as online speaker normalization and customized acoustic model fine-tuning. Fourth, a more reliable multimodal human-robot interface that can resolve ambiguities that occur when voice commands alone are insufficient in high-noise environments would be created by combining voice recognition with complementary modalities, namely gesture recognition via depth cameras and gaze tracking via eye-tracking sensors. Lastly, hardware-accelerated inference (such as edge deployment on NVIDIA Jetson) in conjunction with validation using actual human voice samples gathered on the shop floor would further demonstrate the system's scalability and preparedness for industrial automation.

References

1. Smith, J. & Rogers, E. Industry 4.0 and Smart Manufacturing Strategies. 2023 Journal of Advanced Manufacturing 45(3): 210–225, 2023
2. Brown, M. & Lee, S. The Role of IIoT in Modern Manufacturing. Industrial Internet Journal 12(2): 150–162, 2022
3. Davis, S. & Martin, R. Augmented Reality in Human-Robot Collaboration. AR & VR in Industry 30(2): 55–67, 2024
4. Johnson, R. & Adams, K. Human-Robot Collaboration in Industrial Environments. *Robotics and Automation Journal* 50(1): 35–48, 2024
5. Lee, S. & Carter, J. Voice-Controlled Robots: Challenges and Opportunities. *International Journal of AI and Robotics* 20(6): 120–135, 2020
6. Williams, D. & Thompson, A. Safety Aspects of Human-Robot Collaboration. *Journal of Robotics Safety* 15(4): 98–112, 2021
7. Adams, K. & Green, L. Hidden Markov Models for Voice Processing. *Journal of Computational Linguistics* 18(4): 220–234, 2023

8. Jackson, M. & Wilson, E. Vosk: A Comprehensive Voice Recognition Toolkit. *Open-Source AI Journal* 5(1): 15–30, 2023
9. Miller, D. & Scott, O. Voice Recognition for Industrial Automation. *Journal of Industrial AI* 25(1): 75–89, 2022
10. Rodriguez, C. & Brown, E. CNN-Based Voice Recognition in Industrial Settings. *Deep Learning in AI Journal* 10(2): 67–80, 2024
11. Thomas, C. & White, L. Overcoming Noise in Voice Recognition Systems. *IEEE Transactions on Voice Processing* 35(3): 310–324, 2018
12. Doe, J. & Smith, J. Voice Recognition Using Deep Neural Networks: A Review. *IEEE Access*, 2023
13. Johnson, A. & Lee, B. Advancements in Automatic Voice Recognition Systems. *IEEE Transactions on Audio, Voice, and Language Processing*, 2022
14. Wubet, Y.A. & Lian, K.-Y. Voice Conversion Based Augmentation and a Hybrid CNN-LSTM Model for Improving Speaker-Independent Keyword Recognition on Limited Datasets. *IEEE Access* 10: 89854–89866, 2022. doi: 10.1109/ACCESS.2022.3200479
15. Tsao, Y.-T., Chang, C.-K. & Chen, Y.-H. Improvement of Voice Recognition in Noisy Industrial Settings using Robust Voice Features. *IEEE Transactions on Industrial Informatics* 19(5): 1234–1245, 2023. [Previously listed as ref. 27; renumbered to replace duplicate.]
16. Michel, O. WebotsTM: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems 1*(1): 39–42, 2004
17. Liu, H., Xie, Q., Zhang, Z., Yuan, T., Leng, X., Sun, L., Zhu, S.-C., Zhang, J., He, Z. & Su, Y. PR2: A Physics- and Photo-realistic Testbed for Embodied AI and Humanoid Robots. *arXiv preprint arXiv:2409.01559*, 2024
18. Clegg, A., Erickson, Z., Grady, P., Turk, G., Kemp, C.C. & Liu, C.K. Learning to Collaborate from Simulation for Robot-Assisted Dressing. *arXiv preprint arXiv:1909.06682*, 2019
19. Shidaganti, G., Akmal, M., Harebailu, P., Nagesh, D. & Kumar, K. Voice Recognition for Human-Robot Collaboration in Industrial Environments. *Journal of Robotics and Automation* 28(3): 310–323, 2022
20. Liu, M., W. & Wang, H. Advances in Voice-Controlled Robotic Systems for Industrial Automation. *Robotics and Automation Magazine* 30(2): 56–67, 2023
21. Ghosh, A., Patel, P. & Sharma, A.. Human-Robot Collaboration for Smart Manufacturing: Voice-Controlled Robotic Arms. *Journal of Robotics and AI* 25(6): 88–101, 2022
22. Younis, M., Li, J. & Zhang, Z. Voice-Activated Industrial Robots: A Survey on Current Research and Future Directions. *Industrial Robotics Journal* 33(2): 45–60, 2022
23. Gou, Y. & Li, H. Intelligent Voice-Controlled Robotic System for Human-Robot Collaboration in Manufacturing. *Journal of Manufacturing Processes* 56(4): 225–240, 2024
24. C. B. Kolanur, Jyoti Bali, shilpa Tanvashi, A. C. Giriyapur, An Overfiew of Collision-Free Path Planning Techniques for Industrial Autonomous Robots (*Cyber-Physical Systems Applications, Challenges, and Research Directions*, Apple Academic Press, 2025/2)