NCE Journal of Science and Engineering (NJSE), Vol. 2, June 2025

Drink Serving Robotic Arm Relying on Multimodal Inputs

Ritu Ram Ojha[†1 *], Rupak Mani Sharma[†], Sujan Prasad Bhattarai, Joseph Thapa Magar, and Dinesh Baniya Kshatri
[1]Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal
[†] Equal contribution
[*]Correspondence: rituramojha@gmail.com

Manuscript received January 19, 2025; accepted April 16, 2024

Abstract—**This paper presents a voice-controlled robotic arm with four degrees of freedom (DOF) designed to automate drink serving in a bartender domain. The system uses an overhead camera with You Only Look Once (YOLO) architecture to detect and localize cups, obstacles, and the drink dispenser within the robot's workspace. Voice orders are recognized using a SqueezeFormer model, enabling customers to place orders. Model Predictive Control (MPC) ensures precise motion of the robotic arm. The arm's joints employ a cycloidal drive for precise movement and reduced backlash, while inverse kinematics is calculated using the Gradient Descent method. The object detection model achieved a mean Average Precision (mAP50) score of 0.981, and the speech recognition model demonstrated a Word Error Rate (WER) of 0.081 on the test dataset. The system's reliability was validated through multiple experiments with customers, with the average time to complete a drink order measured at approximately 30 seconds.**

Keywords—**Cycloidal, gradient descent, model predictive control.**

I. INTRODUCTION

BARTENDERS rat bars handle repetitive tasks like serving drinks and managing inventory, which are essential for smooth service and consistent quality. However, challenges like high labor costs and staff fatigue persist. As a result, the use of robotics is advancing in this sector, offering solutions to improve efficiency and reduce the burden on human staff.

We have developed a voice-controlled robotic arm with four degrees of freedom to streamline the repetitive task of drink serving in the bartending industry. The system also includes an overhead camera that uses the YOLO object detection algorithm to identify and locate cups, drink dispenser, and any obstacles within the robot's operating area.

Our project simplifies robotic drink serving with an affordable, four-degree-of-freedom robotic arm, designed for precise control using cycloidal drives. This approach makes the system more accessible compared to high-cost alternatives while maintaining performance. The key contributions of our research are:

1. Reduction of the degrees of freedom of the robotic arm for drink serving robots.
2. Usage of cycloidal drives in the joints to provide precise control in an economically viable design.

II. RELATED WORKS

Many industrial applications of robotic system for drink serving include complex design such as an anthropomorphic structure that embodies a 3 DOF robotic head, a fixed torso, two 7DOF robotic arms as well as multi-lens camera system to detect and track users faces [1]. A commercial 6 DOF robotic arm was used along with multiple subsystems for ingredient dispensing and automatic washing [2].

A 3DOF robotic arm setup with monocular camera was designed that can recognize the color and depth of the object, enabling it to operate to the target position and grasp the object accurately. The color of the objects are identified based on HSV values and the inverse kinematics problem for the robotic arm was solved using geometric approach. Model Predictive Control (MPC) was used to get the optimal input torques on each joints of the robotic arm [3].

A 6-degree-of-freedom robotic arm was designed to recognize objects using a shape detection algorithm and to enhance pick-up accuracy by utilizing an ultrasonic sensor for measuring the distance between the objects and the arm. The robotic arm's movement was controlled via Amazon Alexa voice assistance, allowing for hands-free operation. To manipulate the arm's position for object grasping, an artificial neural network was employed, trained using data calculated from inverse kinematics equations [4].

The design and implementation of a robotic assistant system was based on the Niryo-One robotic arm, a 6-DOF arm, integrated with a camera mounted on the end effector. The system utilized shape and color detection techniques, along with the YOLOv3 algorithm, to identify and locate the requested object. For voice interaction, the Google Voice Recognition API was employed. Experimental results showed that the robotic arm was able to accurately detect and deliver the desired object with an accuracy rate of 96.66% [5].

Current research on robotic drink serving systems shows clear gaps, particularly in creating cost-effective robotic arms for bartending. While some designs focus on affordability, they often compromise on the precise control needed for tasks like

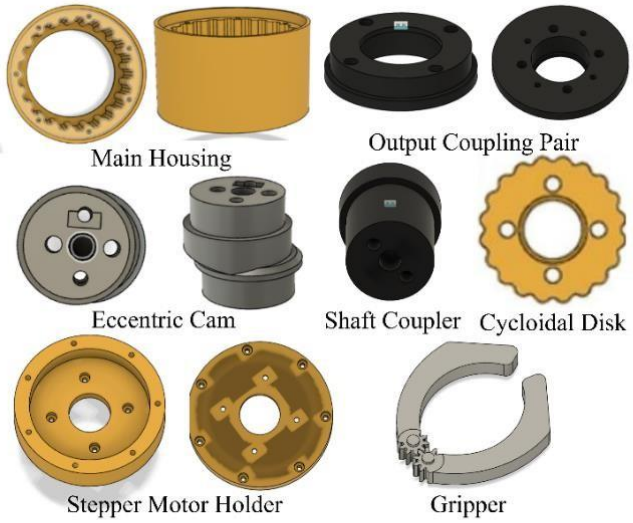

Fig. 1. 3D model of cycloidal drive parts.

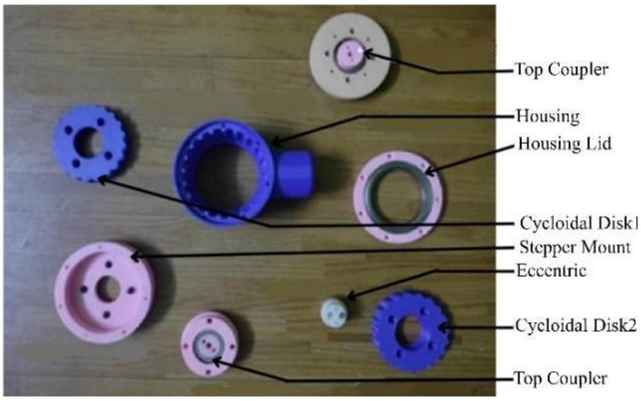

Fig. 2. 3D printed parts of the cycloidal drive.

drink serving. This highlights the need for solutions that balance both cost and accuracy in robotic systems.

III. System Design and Methodology

A. Mechanical Design

The robotic arm consists of four joints and a gripping mechanism. Each joint includes components such as a housing case, cycloidal disk, eccentric shaft, shaft coupler, motor mount, and output coupling pair, all designed in Fusion360 as shown in Fig.1 and 3D printed in Polylactic Acid (PLA), a biodegradable 3D printing material, as shown in Fig. 2. Industrial curtain rods serve as the links between joints, and the gripper is also 3D printed. For drink dispensing, the system uses four DC pumps, each designated for a specific drink, along with pump-switching circuitry, as shown in Fig. 3. The pump-switching circuitry consists of four NPN transistors, which activate based on the drink selected by the customer, enabling the corresponding pump.

B. Experimental Setup

The tabletop setup features a 4DOF robotic arm with stepper motors (Nema 17) driving the joints utilizing stepper driver (TB6600), an MG90 servo for the gripper, controlled by an Arduino Uno. A Raspberry Pi handles multimodal computation, a microphone for the voice input, an overhead camera to scan the tabletop, a Bluetooth speaker for feedback, and a drink dispenser, all powered by a 12V supply as shown in Fig. 4.

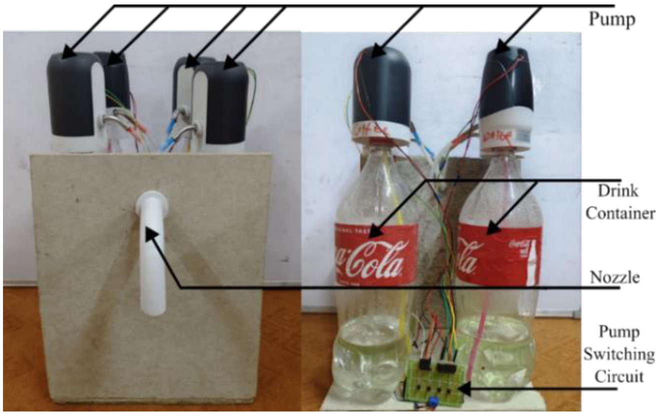

Fig. 3. Drink dispenser.

C. Working Principle

The microphone captures the customer’s drink order, which is processed by an Automatic Speech Recognition (ASR) model to convert it into text. Simultaneously, the overhead camera monitors the tabletop, sending real-time images to a YOLO object detection model to identify objects such as cups, dispensers, obstacles, and nozzles, as shown in Fig. 4. The detected object coordinates are then mapped to the physical environment’s coordinate system. Any anomalies in the voice or vision inputs—such as invalid drink orders, the absence of cups, or obstacles on the workspace—trigger voice feedback through the speaker. The inverse kinematics algorithm calculates the necessary joint angles for the robotic arm, which are further processed by a model predictive controller to generate the joint angle trajectories. These trajectories are translated into control signals by the Arduino, which directs the robotic arm's joints to execute the required movements. The overall workflow is shown in Fig. 5.

D. Automatic Speech Recognition

Dataset Preparation: The first step was to finalize the number of drinks that the system is going to serve and form orders according to that. Four drinks were finalized with five orders variations for each one of them aiming for generalization. However, the system is not limited to the specific order variations listed in the (I). It can recognize and process any order that uses vocabulary from these phrases, allowing for flexible and natural speech patterns. A wake word, 'Hey Arm', has been integrated into the ASR system to ensure that the model only responds to orders after detecting this specific trigger phrase. The audio recordings for each of the order were collected by collaborating with multiple schools and colleges around Kathmandu. The participant pool consisted of students ranging from high school level to graduate studies, as well as teachers and campus staff ensuring representation across

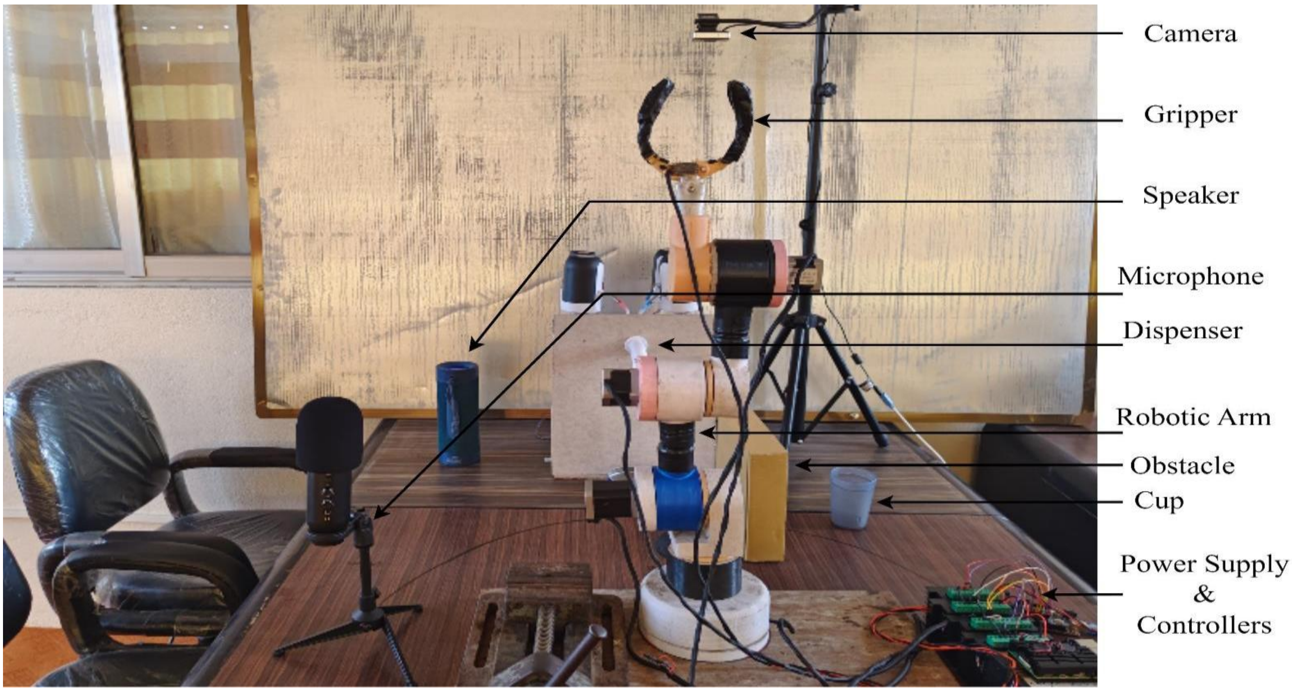

Fig. 4. Complete project setup.

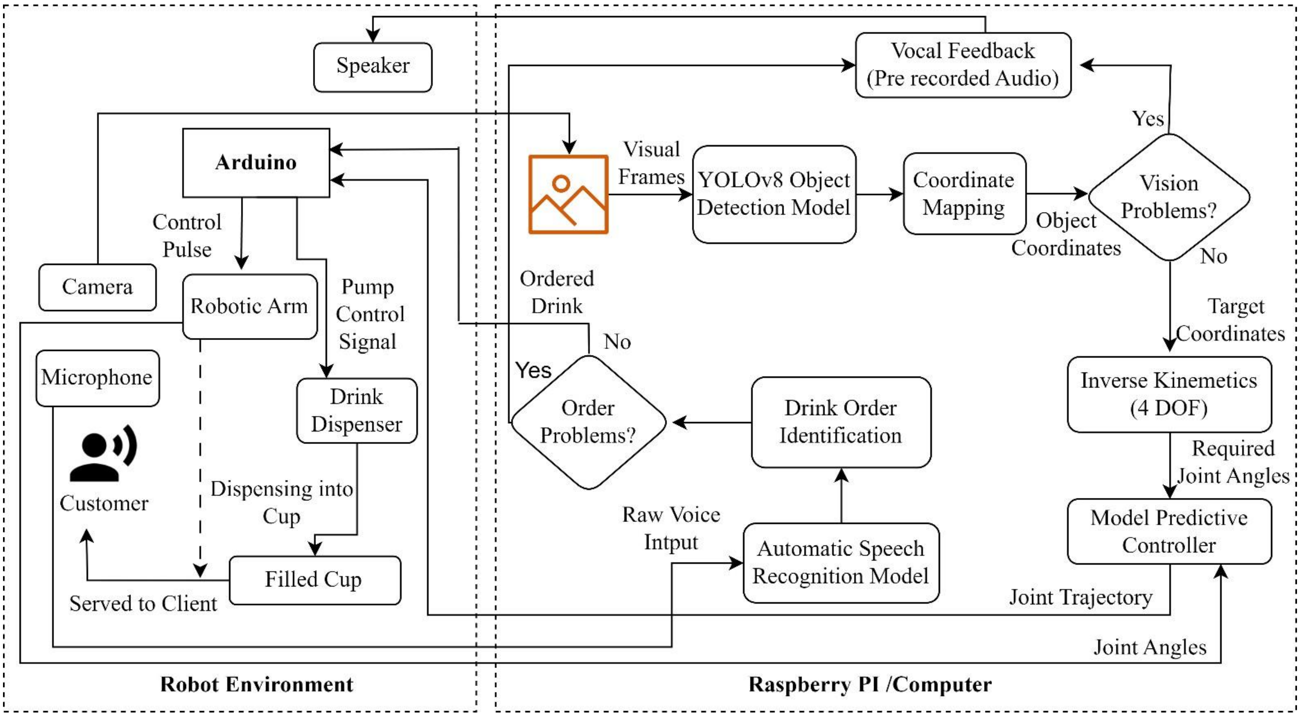

Fig. 5. System Block Diagram

various age groups. A total of 207 audio sample for different variations of each drink orders were recorded, as shown in the Fig. 6. The audio recordings, captured in WAV format with mono channel at a sample rate of 16 kHz, utilizes Fantech MCX01 Leviosa, a professional condenser microphone.

Frequency Domain Conversion: To convert recorded voice sample from the amplitude-time domain to the frequency-time domain, spectrogram is employed. This is crucial for analyzing phonemes and various linguistic features, which correspond to specific frequency components. The conversion to the frequency domain generally involves performing a Short-Time Fourier Transform (STFT) on the audio signal to generate a spectrogram. Following this, a Mel filter bank is applied to the spectrogram to produce a Mel spectrogram, which offers a representation of the audio signal that is more aligned with

TABLE I: DIFFERENT VARIATIONS OF SAMPLE ORDERS

Variation	Order
1	Provide me with a cup of <drink name>
2	Fill up a cup with <drink name> please
3	I would like a cup of <drink name>
4	Can you get me a cup of <drink name>
5	Make me a cup of <drink name>

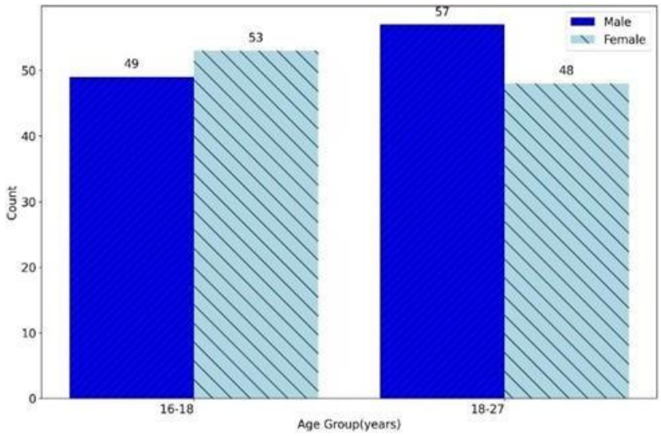

Fig. 6. Audio sample counts for drink orders.

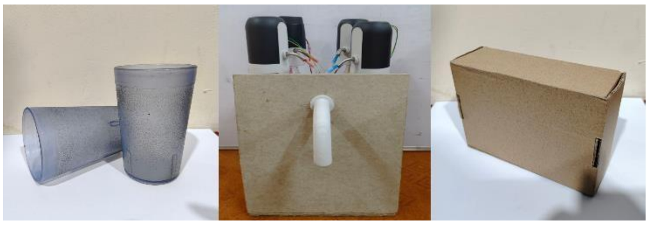

Fig. 7. Raw glimpse of objects utilized for the dataset.

human auditory perception.

Training Squeezeformer Model: The pretrained Squeezeformer-CTC model is used which has been trained on Librispeech 960 hours corpus. It utilizes a Google Sentence Piece tokenizer with vocabulary size 128, and transcribes text in lower case English alphabet along with spaces, apostrophes and a few other characters. For the proposed system, the encoder of the Squeezeformer model is frozen, and only the decoder is fine-tuned on the custom dataset. The model processes raw audio waveforms through its preprocessor to generate mel spectrograms, which pass through an encoder comprising depth wise separable convolutions, multi-head self-attention, feed-forward networks, and convolutional modules with relative positional encoding and time reduction. The encoder's final embeddings are mapped to output labels by the decoder, a 1D convolution layer. The model is trained using Connectionist Temporal Classification (CTC) loss to align predictions with target sequences.

E. Object Detection Model

Dataset Preparation and Training: The pictures of cups, obstacles, nozzle, and dispenser are taken from an overhead camera with a resolution of 1280x720 at different positions, orientations, and under varied lighting conditions. The objects utilized in the making of dataset are shown in the Fig. 7.

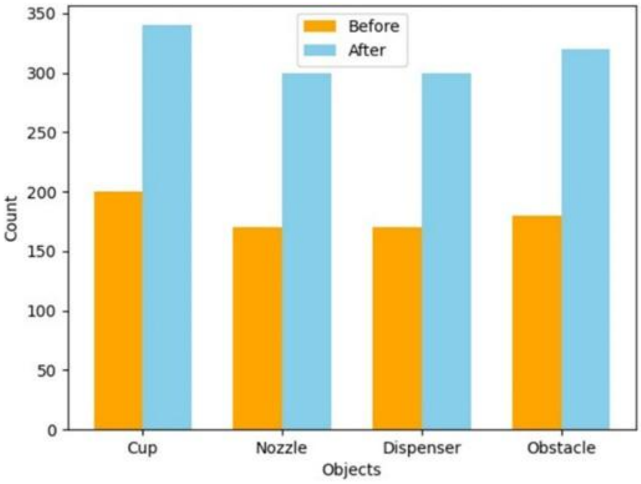

Fig. 8. Object count before and after image augmentation.

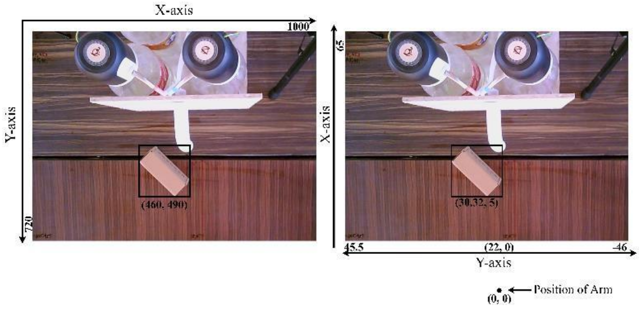

Fig. 9. Overhead camera view of physical environment.

The images are annotated using the online tool CVAT. After annotation, multiple image augmentation techniques like flip, rotation, and changes in brightness were applied to increase the size and variability of the dataset. Following augmentation, there were around 1250 object instances across all classes in the dataset as shown in Fig. 8. The dataset was then exported in Darknet format, which is used for the YOLOv8 model. It was further divided into training and testing sets. Finally, the YOLOv8 detection model was finetuned on our dataset.

F. Coordinate Mapping

Robotic vision systems require mapping the coordinates of the detected objects from the camera frame to the physical environment. A total of 9 camera coordinates of different locations inside the field of view of the camera were meticulously listed along with their corresponding physical environment coordinates, as shown in Fig. 9. They were utilized to derive the transformation matrix that maps the camera coordinates to the physical environment coordinate.

G. 4-DOF Robotic Arm

Kinematic Model of the Robotic Arm: The kinematic model of the robotic arm is formulated using the DH convention that ultimately provides the homogeneous transformation matrix that tells both the rotation and position of one frame n relative

TABLE II: NOMENCLATURE FOR KINEMATIC MODEL

Symbol	Remarks
θ_i	Angle of i^{th} joint
a_i	Length of i^{th} link
α	The amount of rotation of $frame_{i-1}$around axis x_i to get axis z_{i-1} to match z_i
r	The amount of displacement from $frame_{i-1}$to the $frame_i$ measured only in x_i direction
d	The amount of displacement from $frame_{i-1}$to the $frame_i$ measured only in z_{i-1} direction
c_i	$cos\theta_i$
s_i	$sin\theta_i$
s_{ij}	$\sin(\theta_i + \theta_j)$
c_{ij}	$cos(\theta_i + \theta_j)$
c_{ijk}	$cos(\theta_i + \theta_j + \theta_k)$
s_{ijk}	$sin(\theta_i + \theta_j + \theta_k)$
$\dot{x}, \dot{y}, \dot{z}$	Linear velocities of end effector in x, y, z direction
w_x, w_y, w_z	Angular velocities of end effector in x, y, z direction
θ_ι	i^{th} joint velocities

TABLE III: DH PARAMETER FOR KINEMATIC MODEL

n(link)	Parameters			
	θ	α	r	d
1	θ_1	90°	0	a_1
2	θ_2+90°	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4-90°	0	a_4	0

TABLE IV: NOMENCLATURE FOR INVERSE KINEMATICS PROBLEM

Symbol	Remarks
H_k	Current homogeneous transformation matrix of end effector w.r.t base
α	Step size
H_d	Desired homogeneous transformation matrix of end effector w.r.t base
Δx	Vector: difference in desired and actual pose of the end effector
e_p, e_o	Vectors: position and orientation error between current and desired pose
θ_k	Vector: joint angles at time instant k
$FK(\theta_i)$	Forward Kinematics method; provides end effector x, y, z position according to given joint angles
$\Delta\theta$	Vector: Change in joint angles
J^+	Pseudo-Inverse of Jacobian matrix

to another frame m. It is denoted as Hm. The DH parameter is demonstrated in Table III.

$$H_n^{n-1} = \begin{bmatrix} c(\theta_n) & -s(\theta_n)c(\alpha_n) & s(\theta_n)c(\alpha_n) & r_n c(\theta_n) \\ s(\theta_n) & c(\theta_n)s(\alpha_n) & -c(\theta_n)s(\alpha_n) & r_n s(\theta_n) \\ 0 & s(\alpha_n) & c(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The overall transformation of end effector w.r.t base frame is given by:

$$H_4^0 = \begin{bmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & i_1 \\ c_{234} s_1 & -s_1 s_{234} & -c_1 & i_2 \\ s_{234} & c_{234} & 0 & i_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where

$$i_1 = -c_1(-\alpha_4 c_{234} + \alpha_2 s_2 + \alpha_3 s_{23})$$
$$i_2 = -s_1(-\alpha_4 c_{234} + \alpha_2 s_2 + \alpha_3 s_{23})$$
$$i_3 = \alpha_4 s_{234} + \alpha_2 c_2 + \alpha_3 s_{23} + \alpha_1$$

The first three row's element of fourth column of the given matrix provides the x, y, z position of the end-effector given the joint angles which are the forward kinematics equations. The inverse kinematics problem is solved using gradient descent. This method relies on the Jacobian matrix, which relates the rate of change of the end-effector's position and orientation to changes in the joint angles of the robot. A complete inverse kinematics solution not only calculates the necessary joint angles to position the end-effector at the target but also ensures that the end-effector is correctly oriented when it reaches the destination.

Algorithm 1 Inverse Kinematics via Gradient Descent

Require: H_d and H_k at step k, J^+, α, $F(\theta_i)$

Loop

Stop after $||\theta_k|| \approx 0$

Evaluate

$$\Delta x = \begin{bmatrix} e^p \\ e^0 \end{bmatrix}$$

Compute $\Delta\theta = J^{+\Delta x}$

Increment θ_k to converge end effector pose on the desired pose:
$\theta_{k+1} = \theta_k + \alpha J^{+\Delta\theta k}$ (3)

Dynamic Model of the Robotic Arm: The dynamic model of the robotic arm gives information of torque and other forces resulting in the motion of the robot. The dynamic model was derived using Euler-Lagrange methods and is given by:

$$(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \quad (4)$$

where θ is the joint angle, $\dot{\theta}$ is the joint velocity, $\ddot{\theta}$ is the joint acceleration, $D(\theta)$ is the manipulator inertia matrix, C is the vector of Coriolis and centrifugal forces, g is the vector of gravity forces and τ is joint torques. The above equation can be rewritten with a separate highest derivative as follows:

$$\ddot{\theta} = -D^{-1}C\dot{\theta} - D^{-1}g + D^{-1}\tau \quad (5)$$

Now, this equation is suitable for further modification that enables us to obtain the standard linear-like state-space model

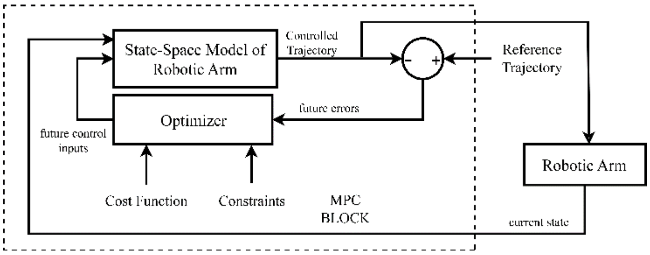

Fig. 10. System block diagram of model predictive controller.

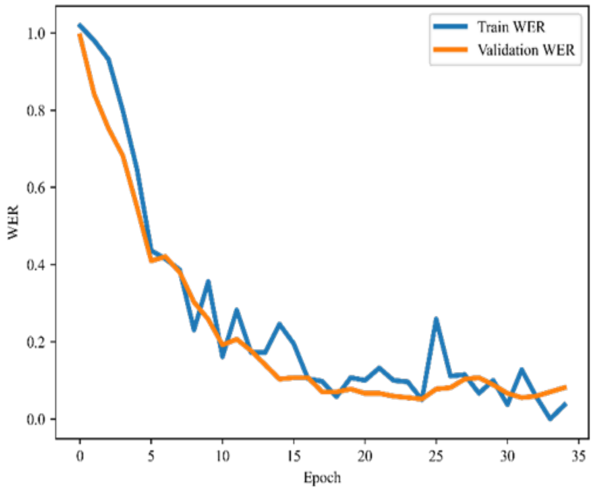

Fig. 11. Word error rate curve.

for control design. It can be written as follows:

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0_{4\times4} & I_{4\times4}\\ 0_{4\times4} & -D^{-1}C_{4\times4}\end{bmatrix}\begin{bmatrix}\theta_{4\times1}\\ \dot{\theta}_{4\times1}\end{bmatrix} + \begin{pmatrix}0_{4\times4}\\ I_{4\times4}\end{pmatrix} + \begin{bmatrix}0_{4\times4}\\ I_{4\times4}\end{bmatrix} u_{4\times1} (6)$$

$$[\theta_{4\times1}] = [I_{4\times4} \quad 0_{4\times4}]\begin{bmatrix}\theta_{4\times1}\\ \dot{\theta}_{4\times1}\end{bmatrix} \qquad (7)$$

which follows the form:

$$x_{k+1} = Ax_k + Bu_k \qquad (8)$$

$$z_k = Cx_k \qquad (9)$$

Also, the calculation of joint torques from an auxiliary vector of control actions u is given by:

$$\tau = Mu + g \qquad (10)$$

MPC uses the model of the plant to make predictions about the future plant output behavior. It also uses the optimizer which ensures that the predicted output tracks the reference trajectory. The MPC controller needs to find the best predicted output so that it is closest to the reference. So, it simulates multiple future scenarios, also accounting for the given input/output constraints and the predicted output with the smallest cost function provides the optimal solution of the future inputs that is fed onto the model, as shown in Fig. 10.

Now, from the derived state-space model (8) and (9), x_k is the state, u_k is the control input, z_k is the output that we want to control, and A, B and C are the system model matrices. The goal of a model predictive controller is to determine a sequence of control inputs, $u_{k+i|k}, i = 0,1,2, \dots, \mathrm{v} - 1$ at a specific time step k and control horizon (v). These inputs are intended to steer the system's output, $z_{k+i}, i = 1,2,3, \dots, \mathrm{f}$ towards a reference trajectory (the specified control path) within the prediction horizon (f). To accomplish this, the controller utilizes the current system state x_k and the system model matrices A, B, and C to forecast the future output

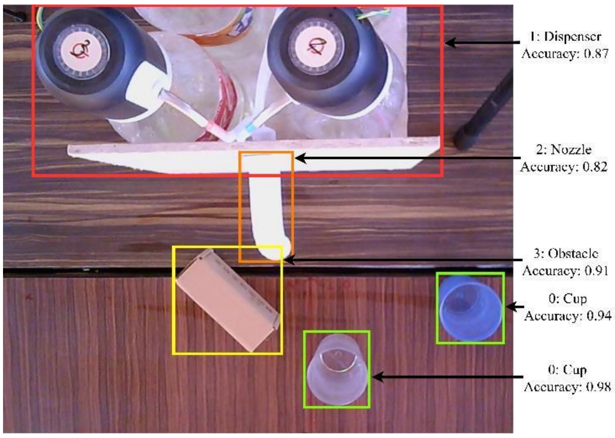

Fig. 12. Various objects detected along with the confidence score.

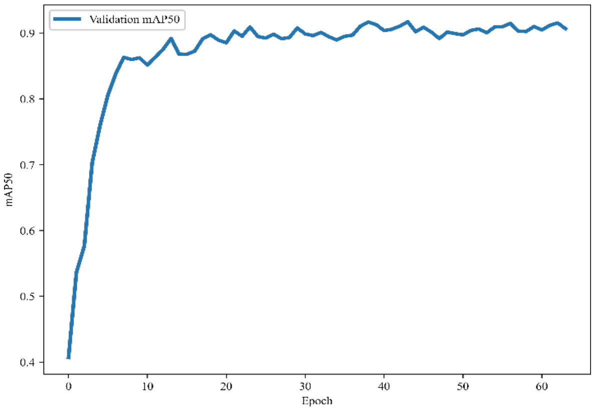

Fig. 13. Mean average precision score.

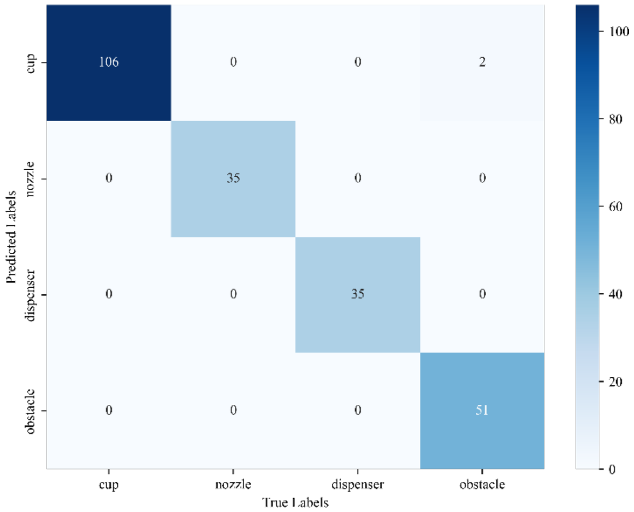

Fig. 14. Confusion matrix.

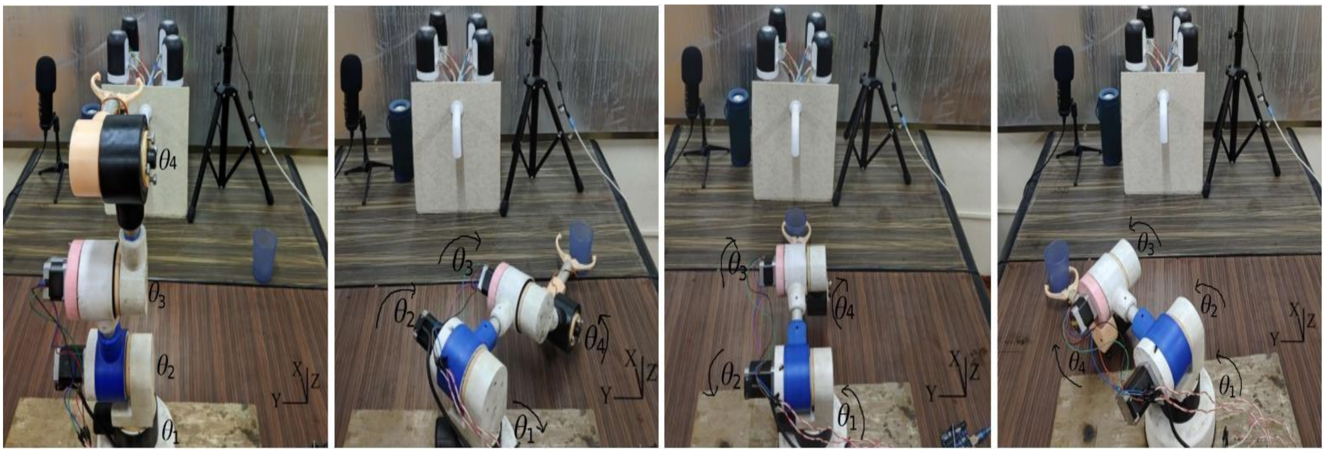

Fig. 15. Different actions performed by robotic arm from being at home position to serving drink to customer (left to right).

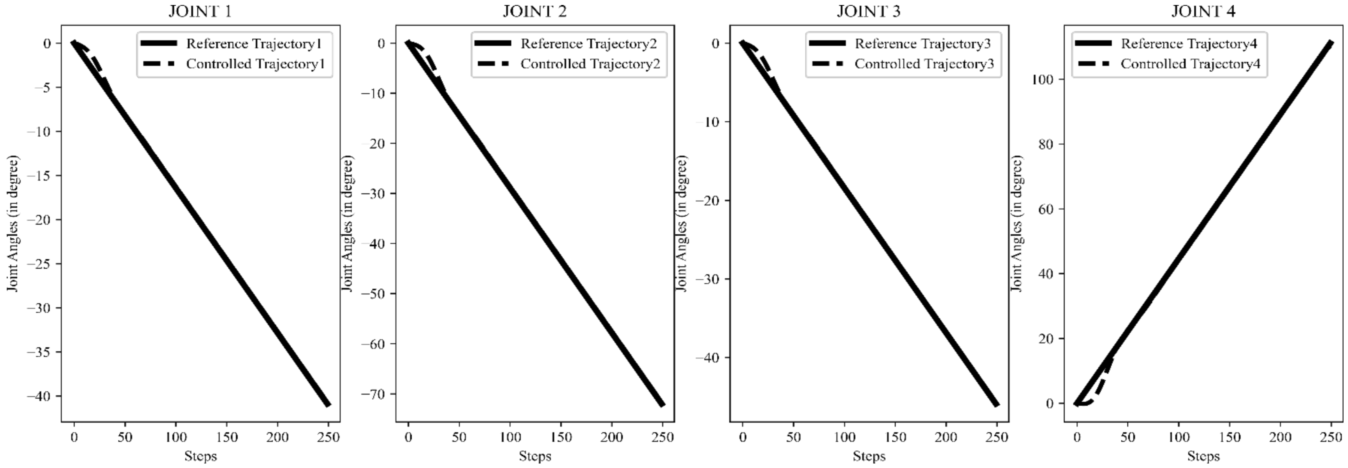

Fig. 16. Reference trajectory tracking for different joints while grabbing empty cup by the robotic arm.

trajectory $z_{k+i|k}, i = 0,1,2,3, \dots, \mathrm{f}$. The inputs are then optimized by minimizing the deviation between the forecasted and reference trajectories. The reference trajectories for each joint are 250 evenly spaced values from the current joint angles to the target joint angles determined by inverse kinematics. The goal is to track a reference trajectory which is given by:

$$Z^d = \begin{bmatrix} Z^d_{k+1} \\ Z^d_{k+2} \\ \dots \\ Z^d_{k+f} \end{bmatrix} \qquad (11)$$

Now, the optimization problem is to determine the vector u minimizing the overall cost function given by:

$$^{min}{}_{u}(J_z + J_u)$$

where

$$J_z = (Z^d - Z)^T W_4 (Z^d - Z) \qquad (12)$$

$$J_u = (W_1 u)^T W_2 (W_1 u) \qquad (13)$$

where W_1, W_2, and W_4 are user-defined weight matrices to penalize input changes error (J_u) and trajectory tracking error (J_z), respectively. Ultimately, to obtain the solution of the model predictive control problem, the cost function described above should be minimized.

IV. Results and Discussion

A. Automatic Speech Recognition Model

The speech recognition model's performance is assessed using WER. It gauges the percentage of inaccurate words in the recognized transcriptions. It was employed to monitor the model's performance during both training and testing phases. The WER gradually decreases with an increasing number of epochs for both the training and validation datasets, reaching 0.081 for the validation dataset, suggesting that the model has achieved a high level of accuracy recognizing and transcribing speech, as shown in Fig. 11.

Sample Calculation of WER:
Reference: "fill a cup of coffee please"
Prediction": "fill a cup of coffeee please"
Number of substitutions (S) = 1
Number of deletions (D) = 0
Number of insertions (I) = 0
Number of words in the reference (N) = 6

$$WER = \frac{S + D + I}{N} = \frac{1}{6} = 0.17$$

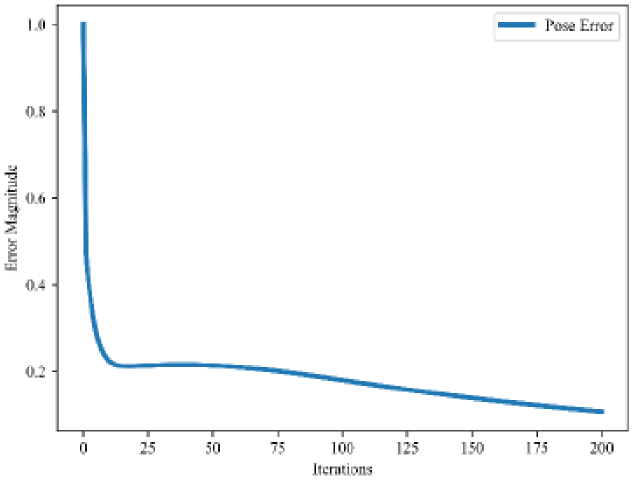

Fig. 14. Pose error while deriving inverse Kinematics solution.

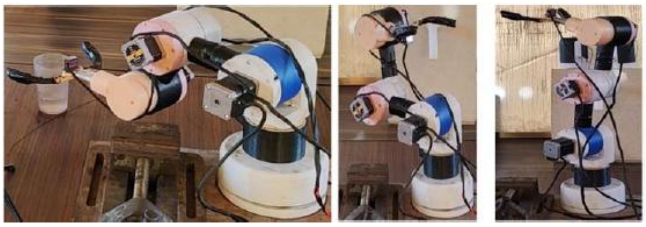

Fig. 15. Robotic arm movement while returning to home position.

We carried out multiple experiments with male/female customers and validated the time required for the robotic arm to complete one drink order, from recognizing the customer's voice order to serving the drink to the customer and returning to the home position, as approximately 30s.

B. Object Detection Model

The object detection model's performance is assessed using two main metrics: Mean Average Precision and Confusion Matrix. The mean average precision summarizes the precision-recall curve across different classes and the confusion matrix helps in identifying which classes are being confused with each other providing the detailed view of how well the model is classifying object. The overhead camera image showcases multiple detected objects, each outlined by bounding boxes generated by an object detection algorithm, highlighting their positions and spatial distribution within the scene, as shown in Fig. 12. The model's precision in detecting and classifying objects, measured at an Intersection over Union (IoU) of 0.5 for the validation dataset, gradually increases with the number of epochs, approaching a value of 0.981.

This trend suggests that the model has achieved near-perfect detection performance, as shown in Fig. 13. For the validation datasets, the confusion matrix shows high accuracy, with all predictions aligning almost perfectly with the true labels, as shown in Fig. 14. The off-diagonal values are all zero except for two instances of obstacle misclassified as cup.

C. Action Performed by the Robotic Arm

TABLE V: System Operation Time per Step

Step	Time Required (in seconds)
Automatic Speech Recognition	≈3
Scan Table Top	≈3
Robotic Arm Grabbing Empty Cup	≈5
Filling the Cup with Ordered Drink	≈6
Serving Drink to the Costumer	≈5
Returning to Home Position	≈8
Total	**≈30**

The actions performed by the robotic arm from being in the home position to serving the drinks to costumer are demonstrated. At the beginning, Robotic Arm is rested at its home position with all joints angles at zero degrees. Upon receiving a valid drink command from the customer, the cup's location on the table is determined using input from an overhead camera and the coordinate mapping algorithm. Inverse kinematics is then applied to compute the required joint angles for the robotic arm to grasp the cup. If the overhead camera detects any obstacles on the table, vocal feedback is provided, instructing the user to remove the obstruction before the robotic arm can proceed. During the inverse kinematics calculations, the pose error gradually decreases, as illustrated in Fig. 17. The four robotic joints then follow the reference trajectory of angles to reach the target position, as shown in the Fig. 16. After grabbing the empty cup, the arm reached towards the dispenser to fill the cup with drink ordered by the costumer. The robotic arm then serves the drink to the customer and return to the home position as shown in Fig.15.

The second joint requires a maximum torque of 5 Nm to return to the home position smoothly after serving a drink to a costumer, without any jerky movement. However, the stepper motor for the second joint can only provide up to 2.61 Nm of torque. To solve this, instead of trying to lift the arm directly, we first rotate the fourth joint counter clockwise, followed by rotating the third joint in the same direction, to help bring the arm back to its home position as shown in Fig.18. We carried out multiple experiments with male/female customers and validate the time required for the robotic arm to complete one drink order, from recognizing the customer's voice order to serving drink to the customer and returning to the home position, as approximately 30s.

V. Conclusion

A voice-controlled 4-DOF robotic arm to automate the repetitive task of drink serving in the bartending industry was developed. The system incorporates an overhead camera with YOLO object detection to locate cups, dispensers, and obstacles, ensuring smooth operation. By using cycloidal drives, the design achieves precise control while remaining cost-effective. The system was validated through multiple experiments, with an average time of approximately 30 seconds to complete a drink order. This work not only reduces labor costs in bartending but also showcases the potential for robotics in assistive applications, such as helping individuals with mobility challenges.

VI. Acknowledgement

This research received workspace and tools from the Department of Electronics and Computer Engineering as well as the Robotics and Automation Center of Thapathali Campus. Some major electronic components were sponsored by Real Time Solutions Pvt. Ltd., and the audio dataset was constructed using recordings from students of Thapathali Campus, Nepal Mega School, and Texas International College.

References

[1] N. E. John, A. Rossi, and S. Rossi, “Personalized Human- Robot Interaction with a Robot Bartender,” Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, Jul. 2022, doi: https://doi.org/10.1145/3511047.3537686.

[2] J. Ramirez, C. Laurel, M. Tafur, and M. Sigüenza, “Development and Implementation of a Robotic Bartender for Automatic Pisco Sour Preparation,” 2024 10th International Conference on Mechatronics and Robotics Engineering (ICMRE),pp.33–38,Feb.2024, doi:https://doi.org/10.1109/icmre60776.2024.10532178.

[3] [3] Zhou, Z., Zhang, Y. and Li, Y. (2023) ‘Model predictive control design of a 3-dof robot arm based on recognition of spatial coordinates’, 2023 9th International Conference on Mechatronics and Robotics Engineering (ICMRE), doi:10.1109/icmre56789.2023.10106581.

[4] Z. Wang, D. Chen, and P. Xiao, “Design of a Voice Control 6DoF Grasping Robotic arm Based on Ultrasonic Sensor, Computer Vision and Alexa Voice Assistance,”IEEEXplore,Aug.01,2019. https://ieeexplore.ieee.org/document/8964744 (accessed Nov. 22, 2021).

[5] G. Nantzios, N. Baras, and M. Dasygenis, “Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision,” Automation, vol. 2, no. 4, pp.238–251, Oct.2021, doi:https://doi.org/10.3390/automation2040015