Learning-based Model Predictive Controller for Drink Dispensing Robotic Arm Relying on Multimodal Inputs

Team Members

Joseph Thapa Magar (THA076BEI011)

Ritu Ram Ojha (THA076BEI024)

Rupak Mani Sharma (THA076BEI027)

Sujan Bhattarai (THA076BEI037)

Supervised By:

Er. Dinesh Baniya Kshatri

Lecturer

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

October 8, 2023

Presentation Outline

- Motivation
- Objectives
- Scope of Project
- Project Applications
- Methodology
- Result and Analysis
- Remaining Tasks
- References

Motivation

Monotonous and Tired Bartender

Command Operated Drink Serving Arm

Objectives

- To model and simulate a drink dispensing robotic arm
- To instantiate the robotic arm and perform performance comparison between simulation and reality

Scope of Project

- Project Capabilities:
  - Voice-based human-machine interaction
  - Proper detection of glass, dispenser and customer
  - Precise and responsive control of robotic arm

- Project limitations :
- Language understanding limited to specific languages or vocabulary
- Challenge in object recognition due to obstruction or poor visibility
- Robotic arm movement constrained to 4 degree of freedom

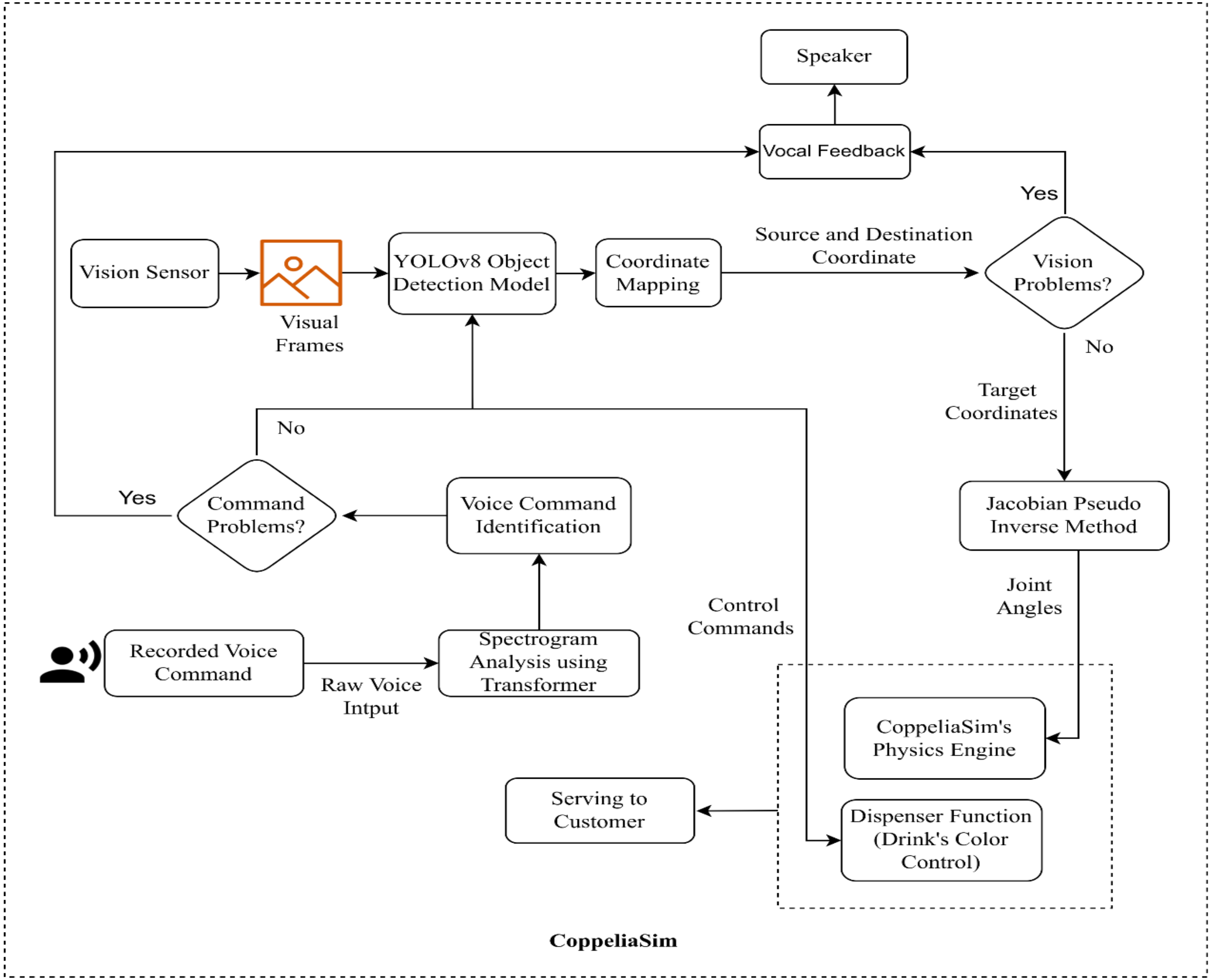
Project Applications

- Medical and Healthcare
  - Surgical procedures, rehabilitation, diagnostics, prosthetics
- Aerospace and Defense
  - Aircraft assembly, space exploration, defense applications
- Manufacturing and Industrial Automation
  - Assembly, welding, pick and place operations, packaging
- Hazardous Environments
  - Nuclear power plants, deep-sea exploration, mining, disaster response

Methodology-[1]
(Software and Hardware Requirements for Part-A)

- Fusion 360
  - CAD design of components.
- CoppeliaSim
  - Virtual Environment
  - Simulation Engine
- Deep Learning Framework
  - Pytorch was used.
  - Training vison and speech models.
- Audacity
  - Audio record and manipulation
- Python
  - Model integration with CoppeliaSim.
- Lua
  - Inverse Kinematics Implementation
- Google Collab GPU T4
  - Ram Usage: 3-12GB
  - Hardware resource for training

Methodology-[2]
(System Block Diagram)



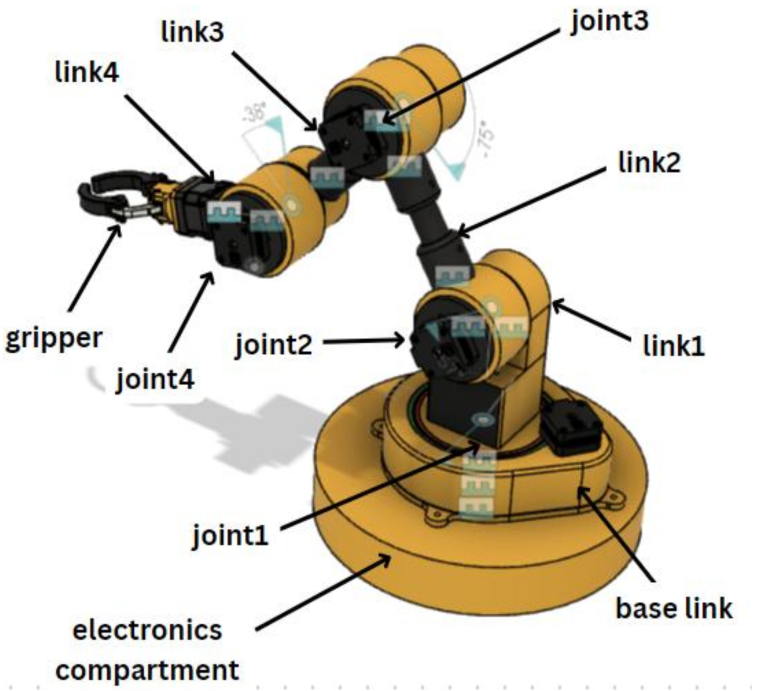
Methodology-[3]
(Working Principle)

- Microphone takes raw voice input from users
- Transformer takes raw voice and provides the name of drinks
- If error in recognizing command, provides vocal feedback
- If not, provides control signal to vision model
- Vision sensor captures the current frame of environment
- The frames are then pass into YOLOv8 model
- Obtained coordinates are mapped into world coordinates

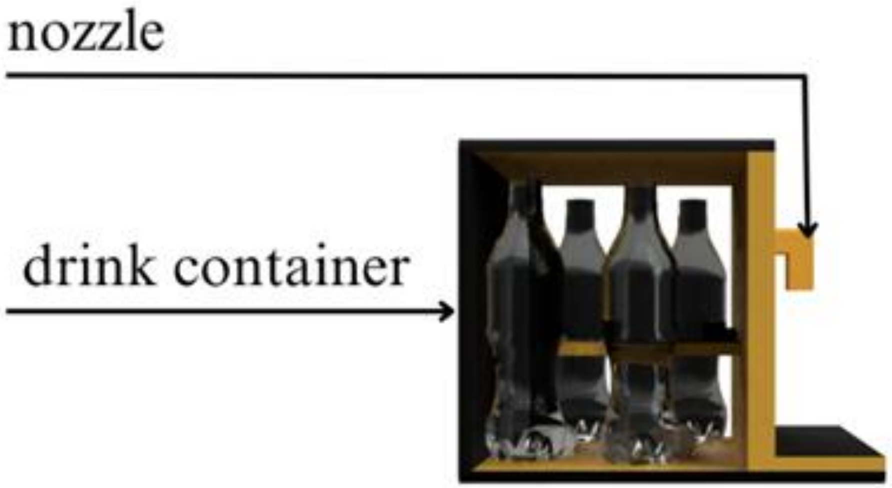
Methodology-[4]
(Working Principle)

- Any problems related to vision are given as vocal feedback
- The world coordinates are used to set target position
- CoppeliaSim IK model calculates required joint angles
- Built-in Position Controller controls the joint angles of robot
- The dispenser system dispenses the requested drink
- Integration of all models in virtual environment of CoppeliaSim

Methodology-[5]
(3D Design of Robot Arm and Dispenser)

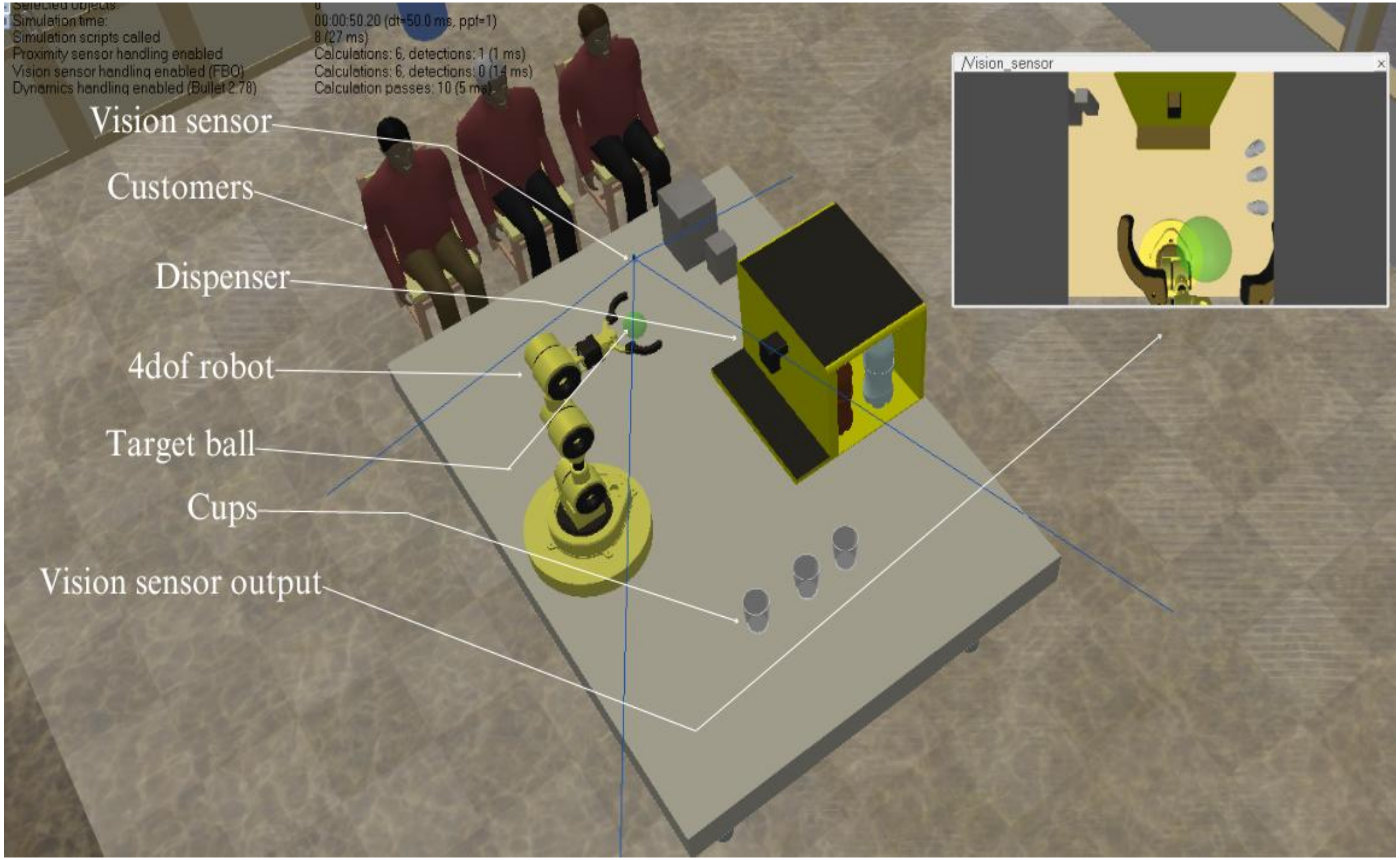


Robot Design

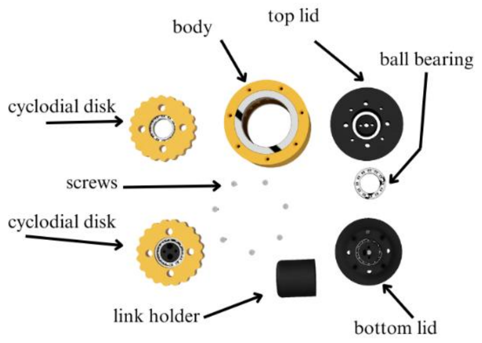


Dispenser Design

Methodology-[6]
(Virtual Environment)

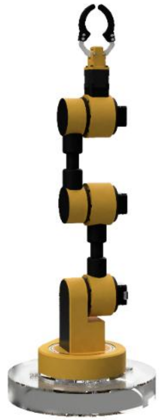


Methodology-[7]
(CAD to Simulation)

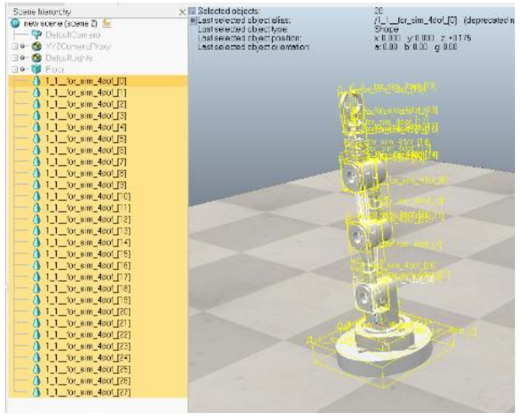


Simplification in Fusion 360

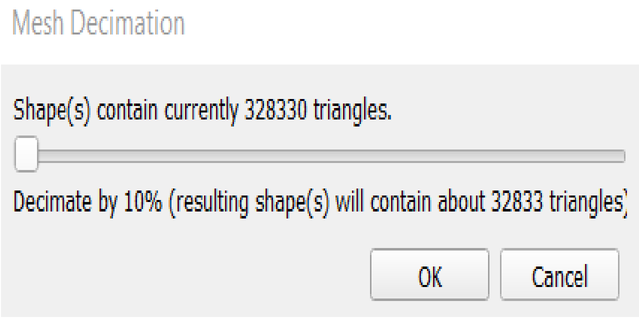
Fewer components Model



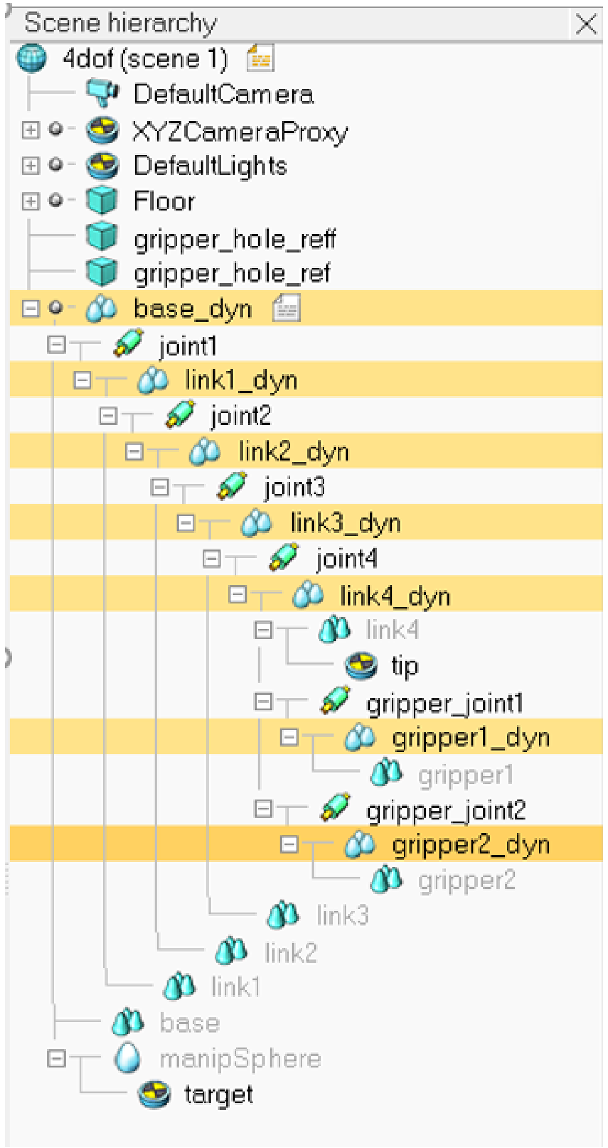
Mesh to sim



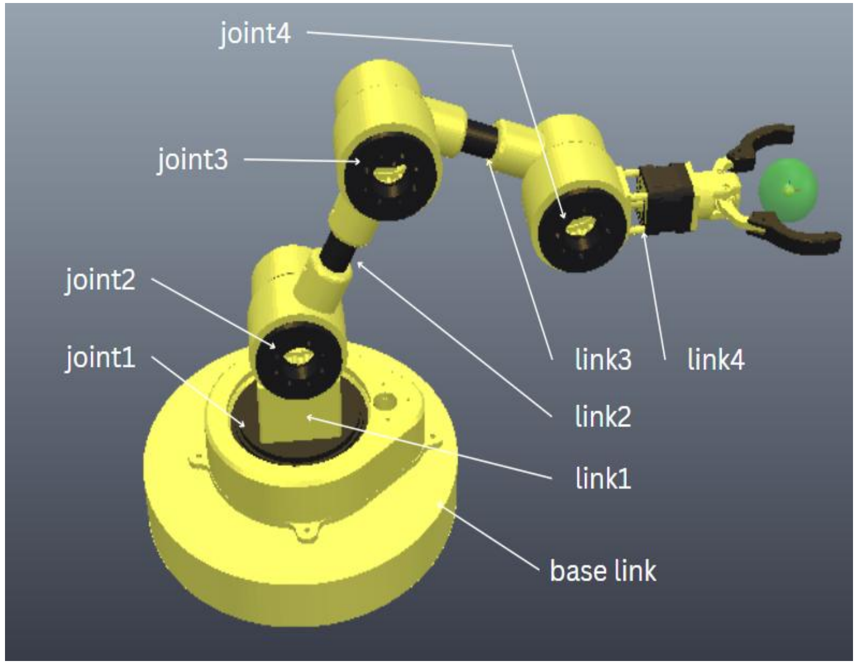
Triangle quantity reduction



Creating Scene Hierarchy



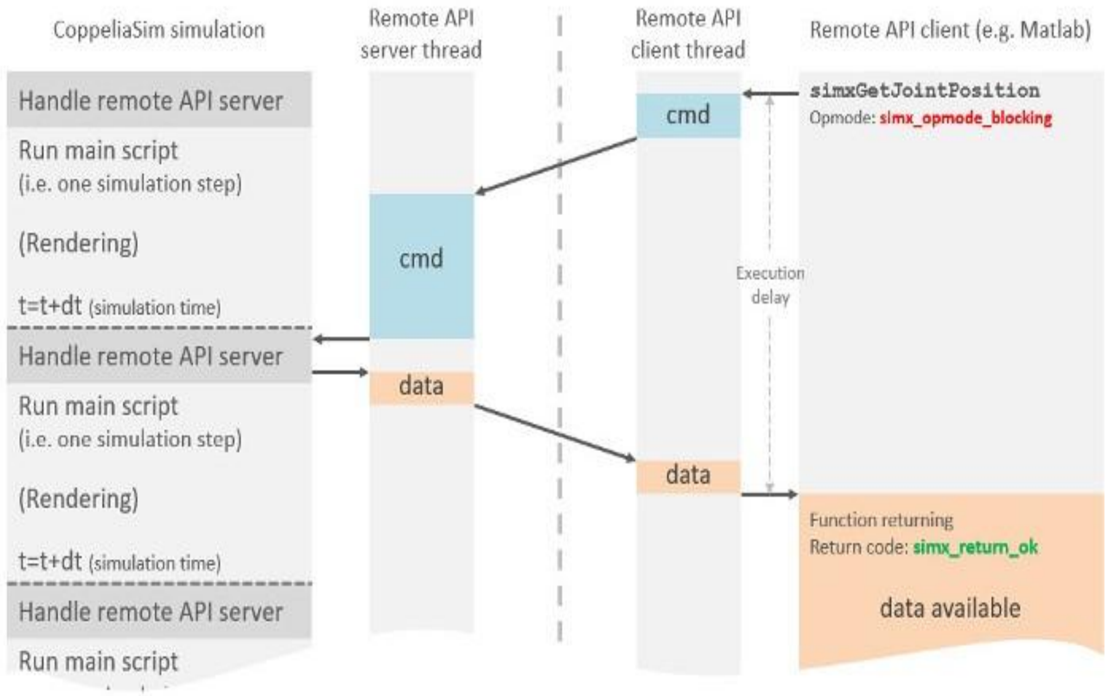
Methodology-[8]
(Inverse Kinematics Simulation)



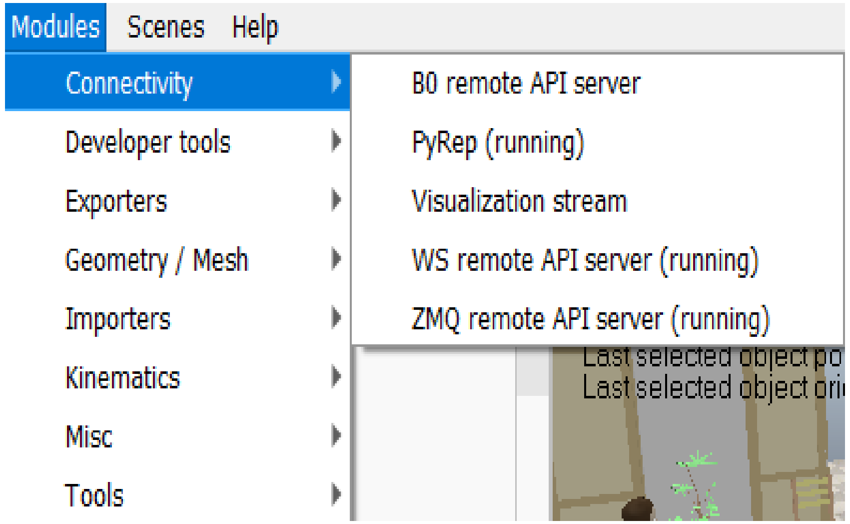
IK implemented in Lua script

```lua
function sysCall_init()
    -- Take a few handles from the scene:
    simBase=sim.getObject('.')
    simTip=sim.getObject('./tip')
    simTarget=sim.getObject('./target')

    ikEnv=simIK.createEnvironment()

    -- Prepare the 2 ik groups, using the convenience 

    ikGroup_undamped=simIK.createGroup(ikEnv)
    simIK.setGroupCalculation(ikEnv,ikGroup_undamped,s
    simIK.addElementFromScene(ikEnv,ikGroup_undamped,s
    ikGroup_damped=simIK.createGroup(ikEnv)
    simIK.setGroupCalculation(ikEnv,ikGroup_damped,sim
    simIK.addElementFromScene(ikEnv,ikGroup_damped,sim
end

function sysCall_actuation()
    if simIK.handleGroup(ikEnv,ikGroup_undamped,{syncW
        simIK.handleGroup(ikEnv,ikGroup_damped,{syncWo
    end
end

function sysCall_cleanup()
    simIK.eraseEnvironment(ikEnv)
end
```

Methodology-[9]
(ZeroMQ)

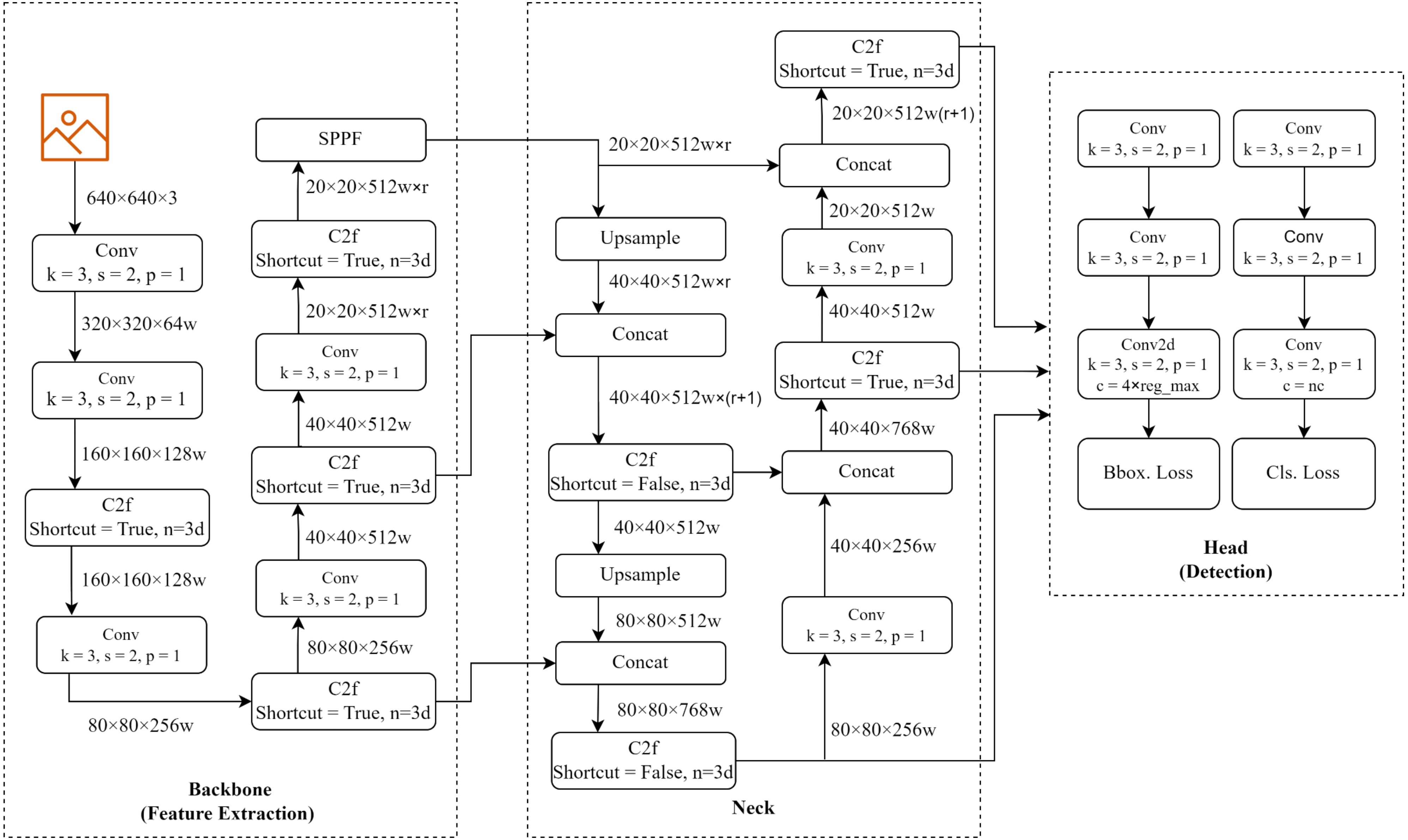


Client Server Communication API



Status of ZeroMQ Server

Methodology-[10]
(YOLOv8 Architecture)



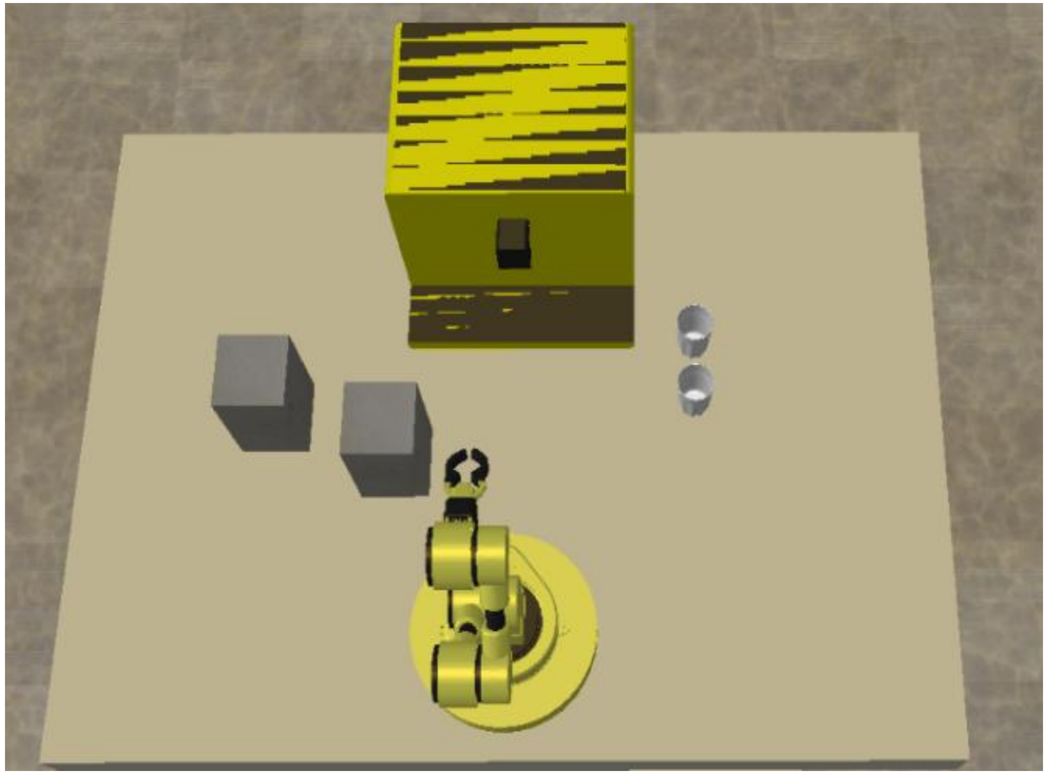
Methodology-[11]
(YOLOv8 Architecture (Backbone))

- C2f (Convolution to Fully Connect)
  - C2f enhances capacity without spatial modifications
  - It incorporates "shortcut" for skip connections
  - Preserves high-res details, captures abstract features
- SPPF (Spatial Pyramid Pooling with Fuse)
  - SPPF block captures multi-scale information in YOLO's backbone
  - It uses MaxPool2d layers with varying pooling sizes for dimension reduction
  - Different pooling sizes capture features at various scales
  - Concatenation combines multi-scale information into one feature map

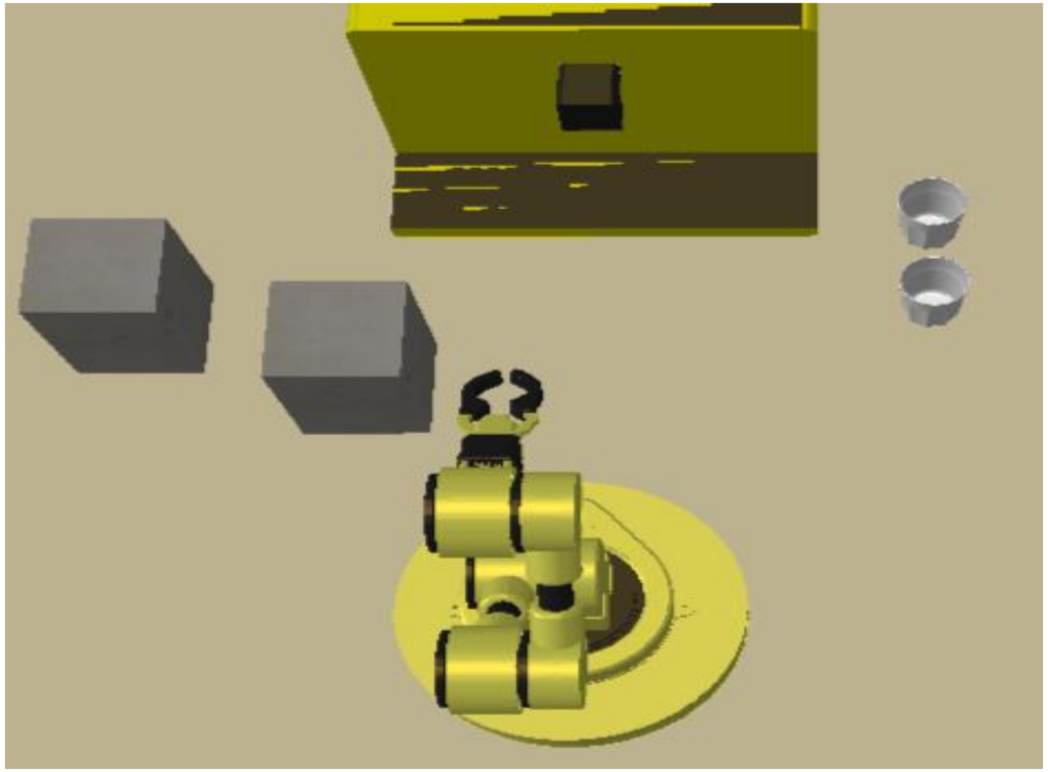
Methodology-[12]
(YOLOv8 Architecture)

- Neck
  - The neck refines backbone features in YOLO architecture
  - Up sampling and concatenation increases spatial resolution and feature scale
- Head
  - The detection layer in YOLO has two sets of convolutions
  - The first set captures spatial info
  - The final Conv2d layer predicts class probabilities for each box

Methodology-[13]
(Data Augmentation for YOLOv8 Model)

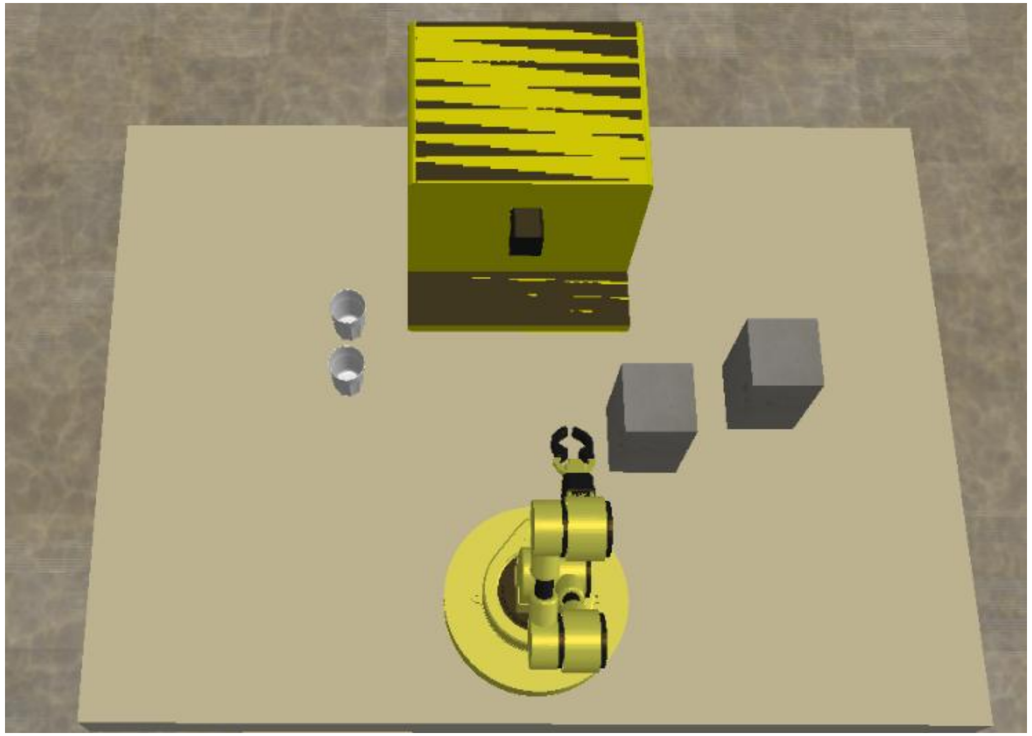


Original Image

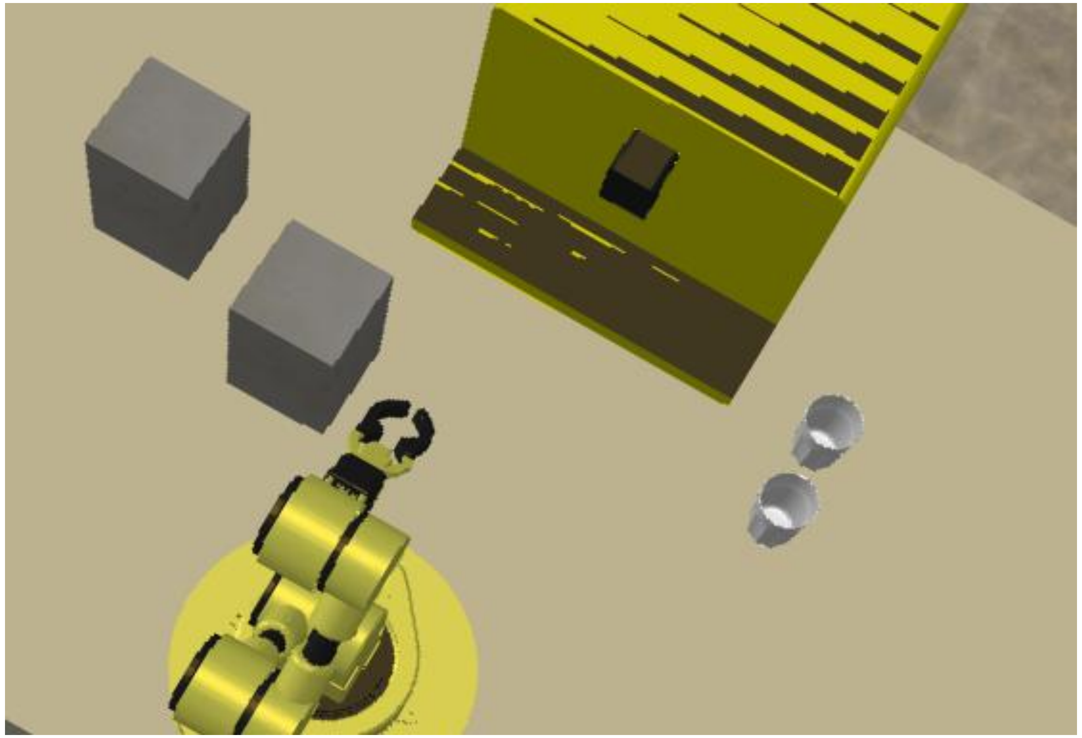


Cropped Image

Methodology-[14]
(Data Augmentation for YOLOv8 Model)

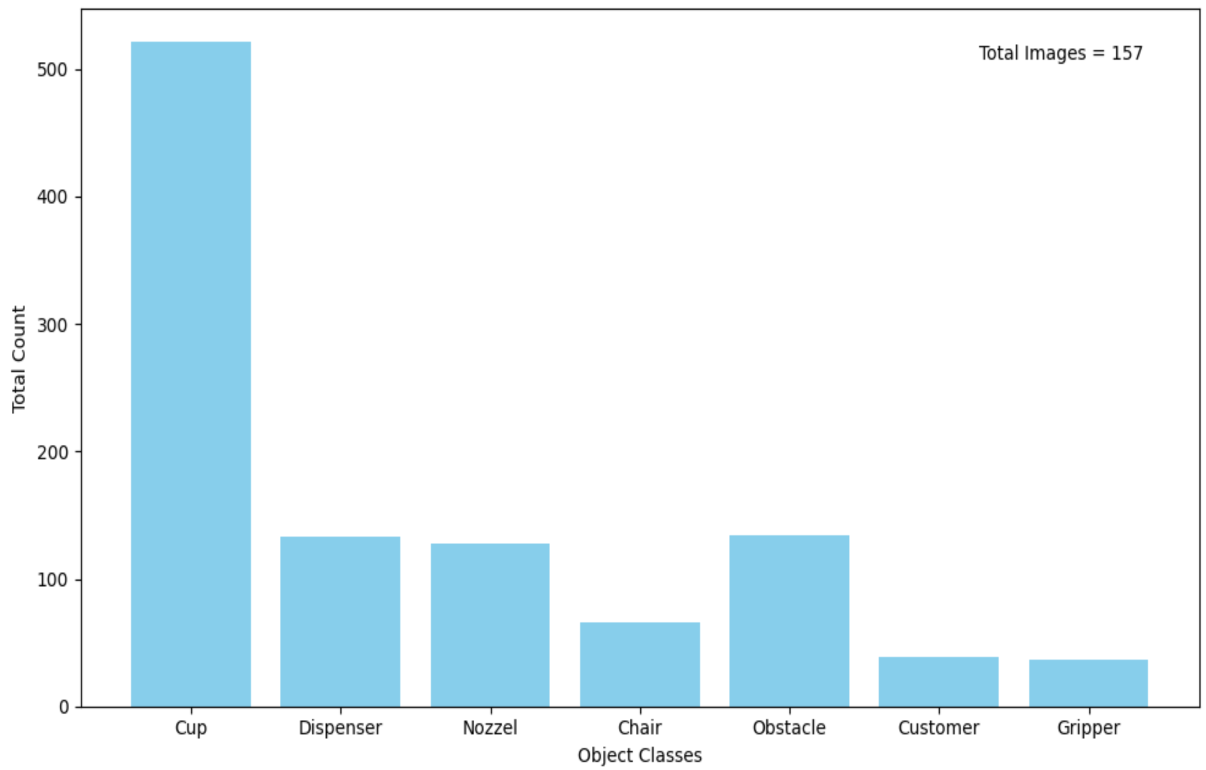


Flipped Image

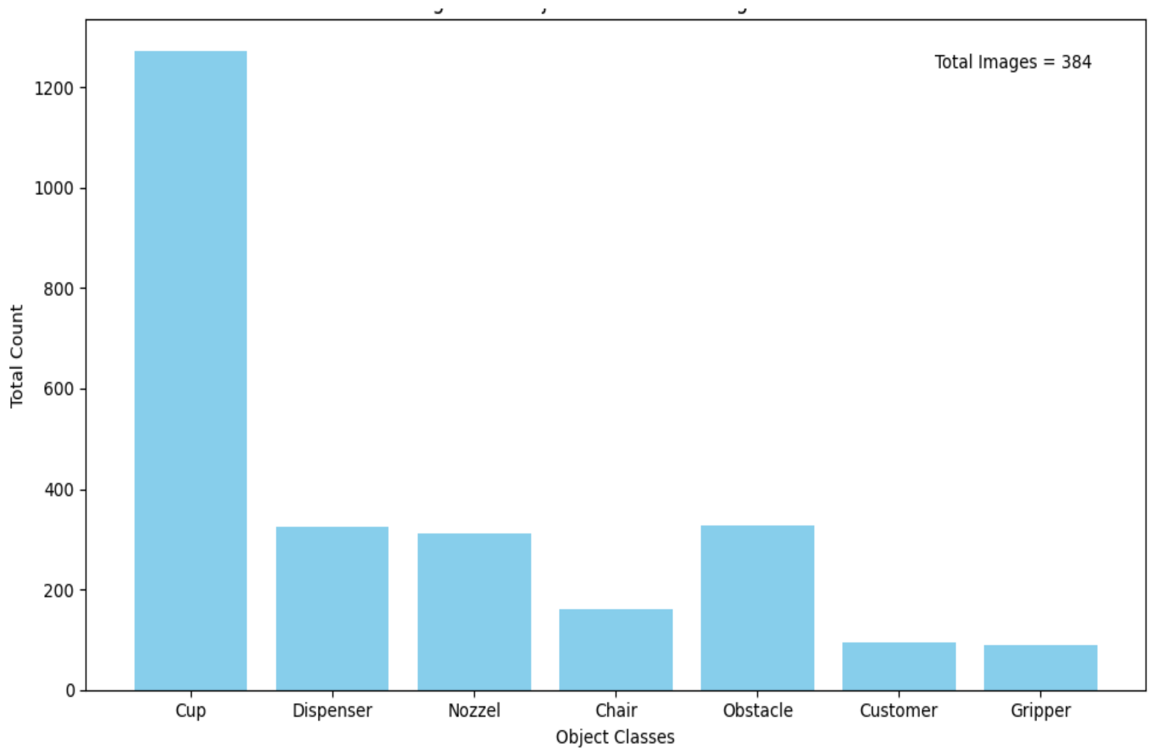


30-degree Rotated Image

Methodology-[15]
(Object Counts Before and After Augmentation)

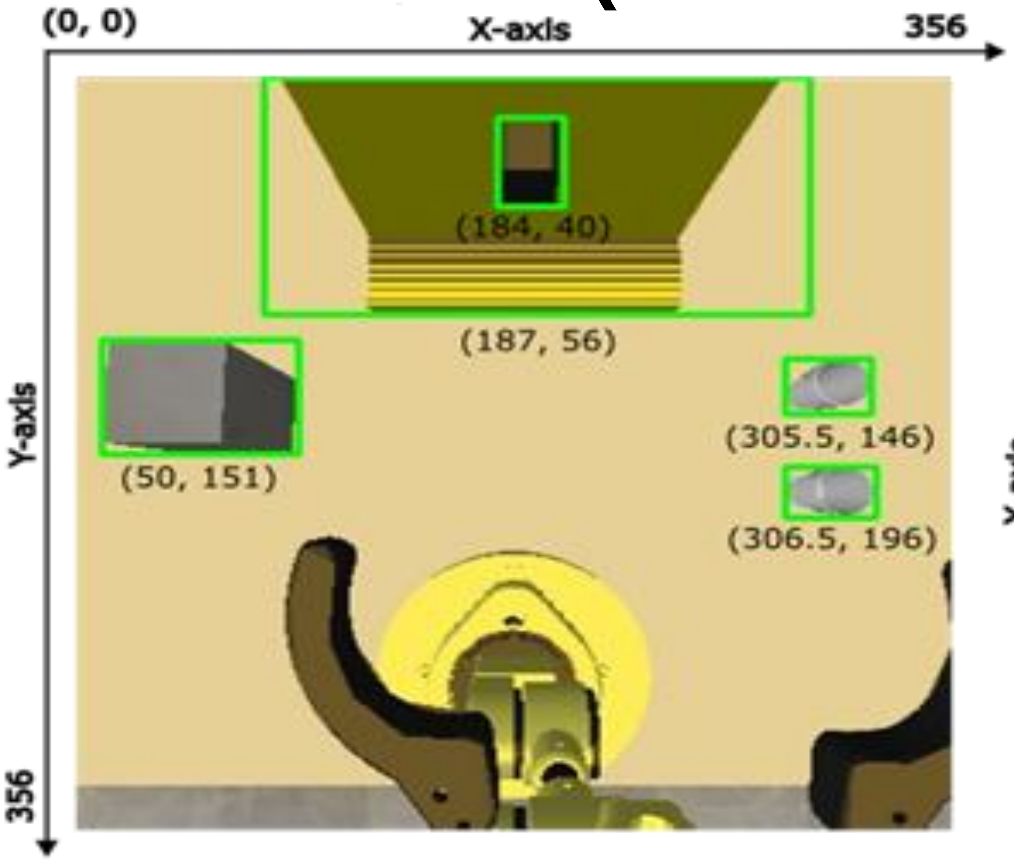


Object Counts Before Augmentation

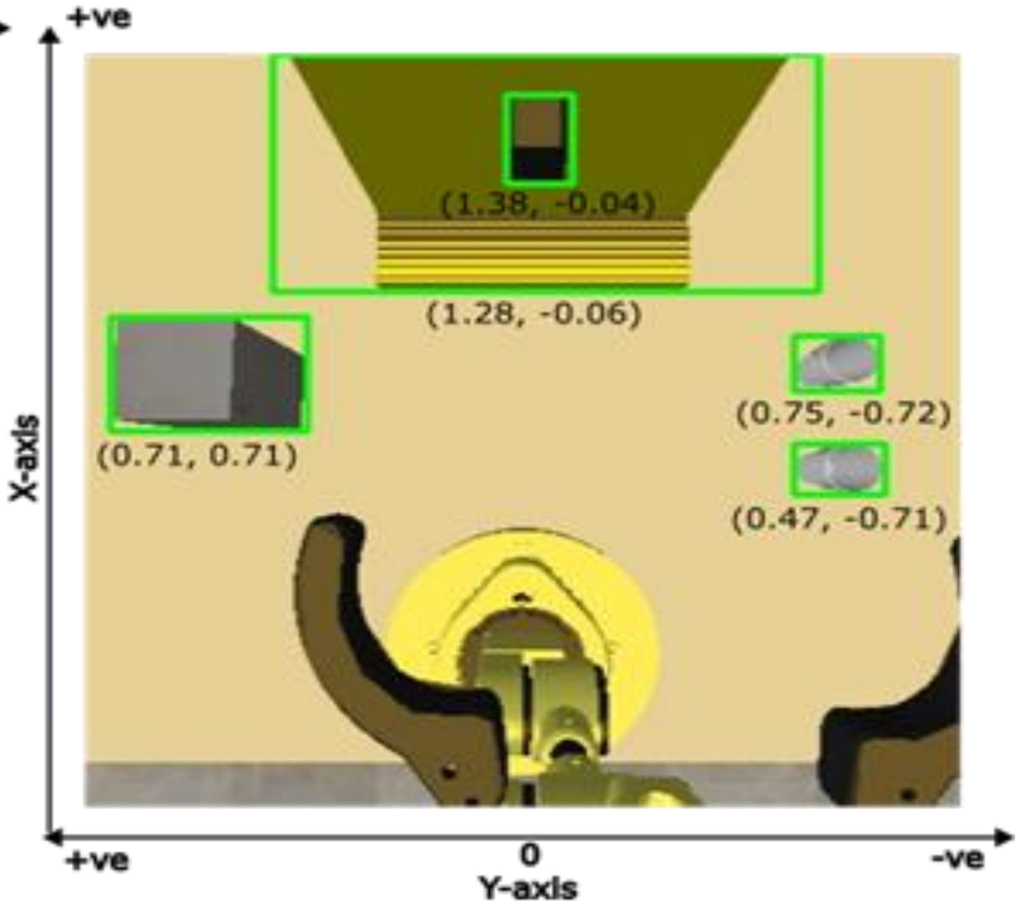


Object Counts After Augmentation

Methodology-[16]
(Coordinate Mapping)



Camera Coordinate System



Simulation Coordinate System

Methodology-[17] (Coordinate Mapping)

$$H = \begin{bmatrix} 9.69e-06 & -5.75e-03 & 1.63 \\ -5.97e-03 & -3.37e-08 & 1.06 \\ -6.63e-07 & 4.47e-04 & 1.00 \end{bmatrix}$$

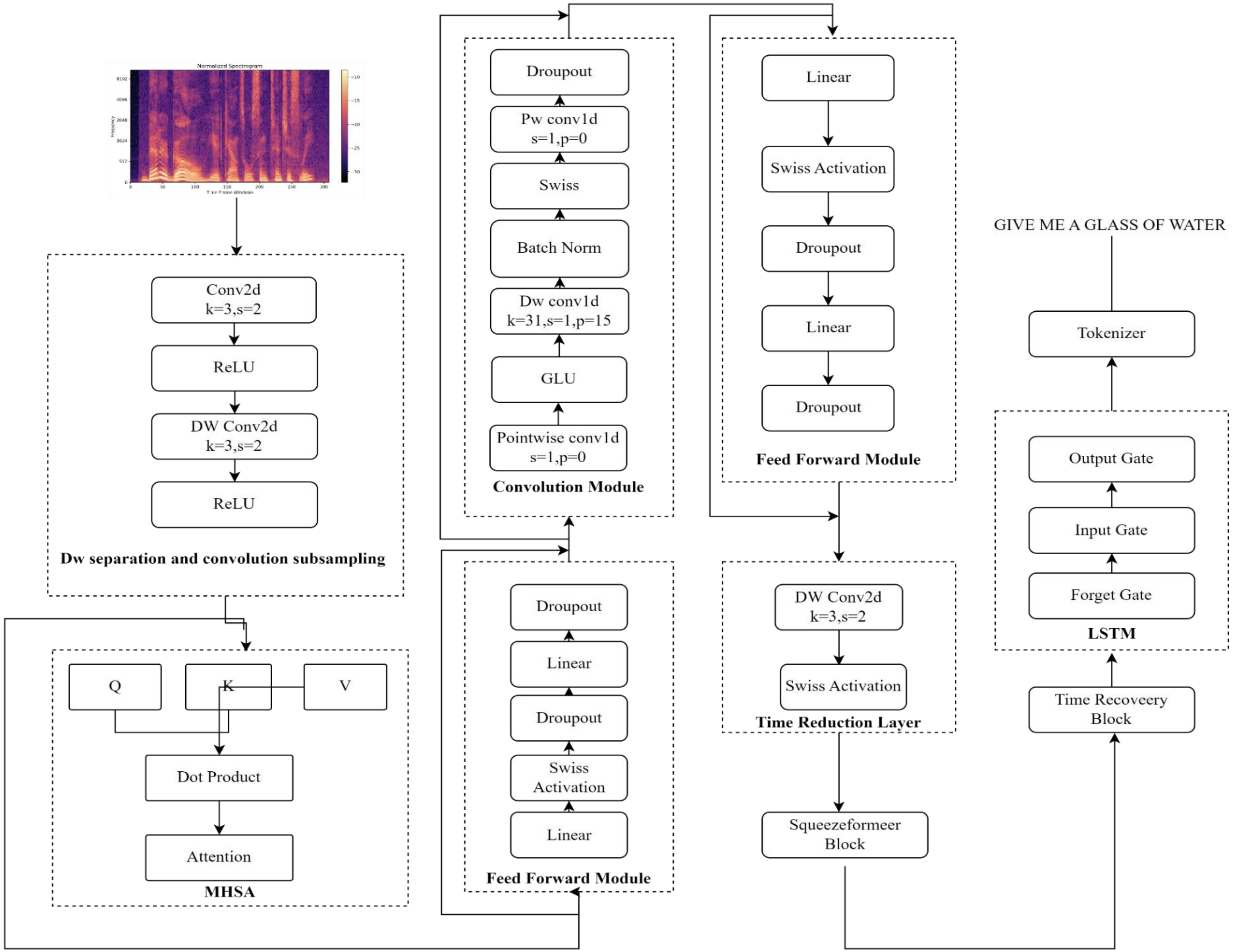
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9.69e-06 & -5.75e-03 & 1.63 \\ -5.97e-03 & -3.37e-08 & 1.06 \\ -6.63e-07 & 4.47e-04 & 1.00 \end{bmatrix} \cdot \begin{bmatrix} 184 \\ 40 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.38 \\ -0.04 \\ 1 \end{bmatrix}$$

- (H) matrix is used to map camera coordinates into simulation coordinates
- Computing (H) matrix requires camera coordinates and its corresponding simulation coordinates
- Sample calculation shown for dispenser's nozzle

Methodology-[18] (Possible Vision Problems)

S.N.	Problem	Description	Remedy
1.	Path Obstruction	Objects obstruct the robotic arm's movement trajectory	Incorporate an alert feedback mechanism for detection
2.	Glass Unavailable	Empty glasses are not available	Prompt the user to place a glass in the designated area
3.	Drinking Glass Unreachable	The drinking glass is out of reach of the robotic arm	Alert the user that the glass is unreachable
4.	Vision Sensor Blockage	The vision sensor is obstructed, affecting accuracy.	Alert users to remove the blockage

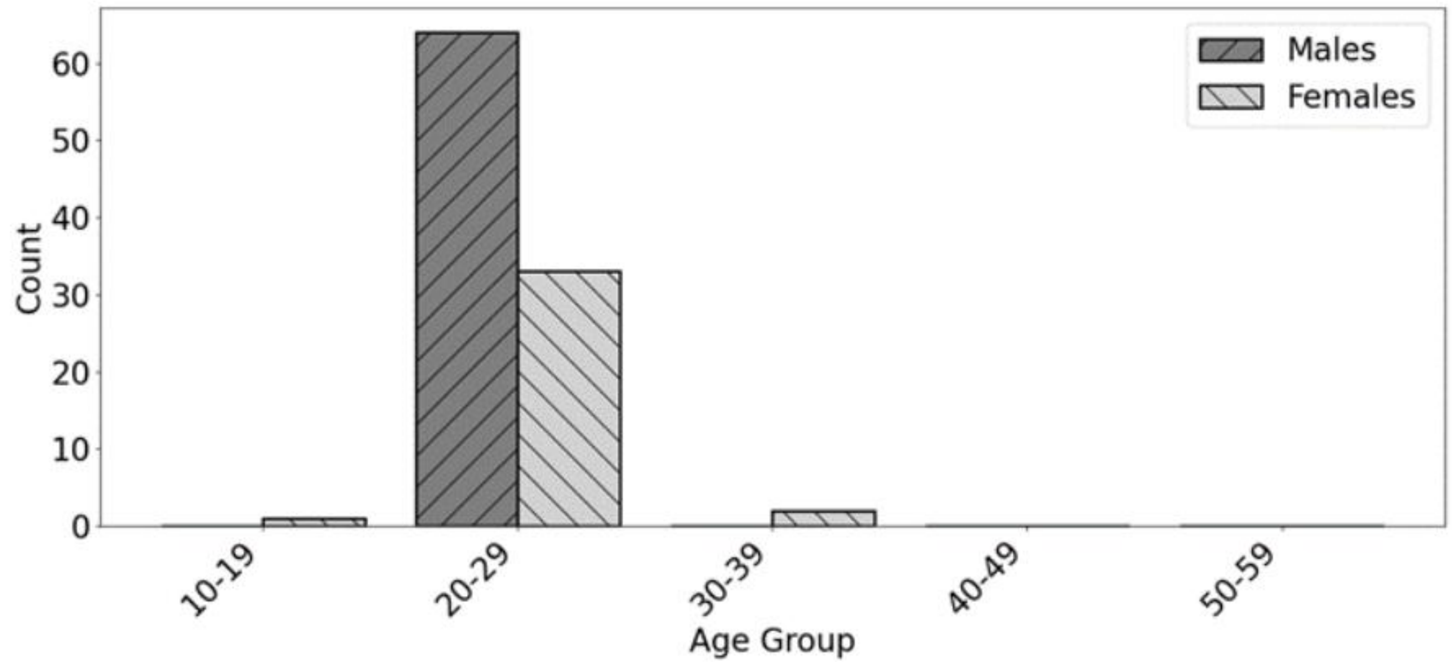
Methodology-[19]
(ASR Architecture)



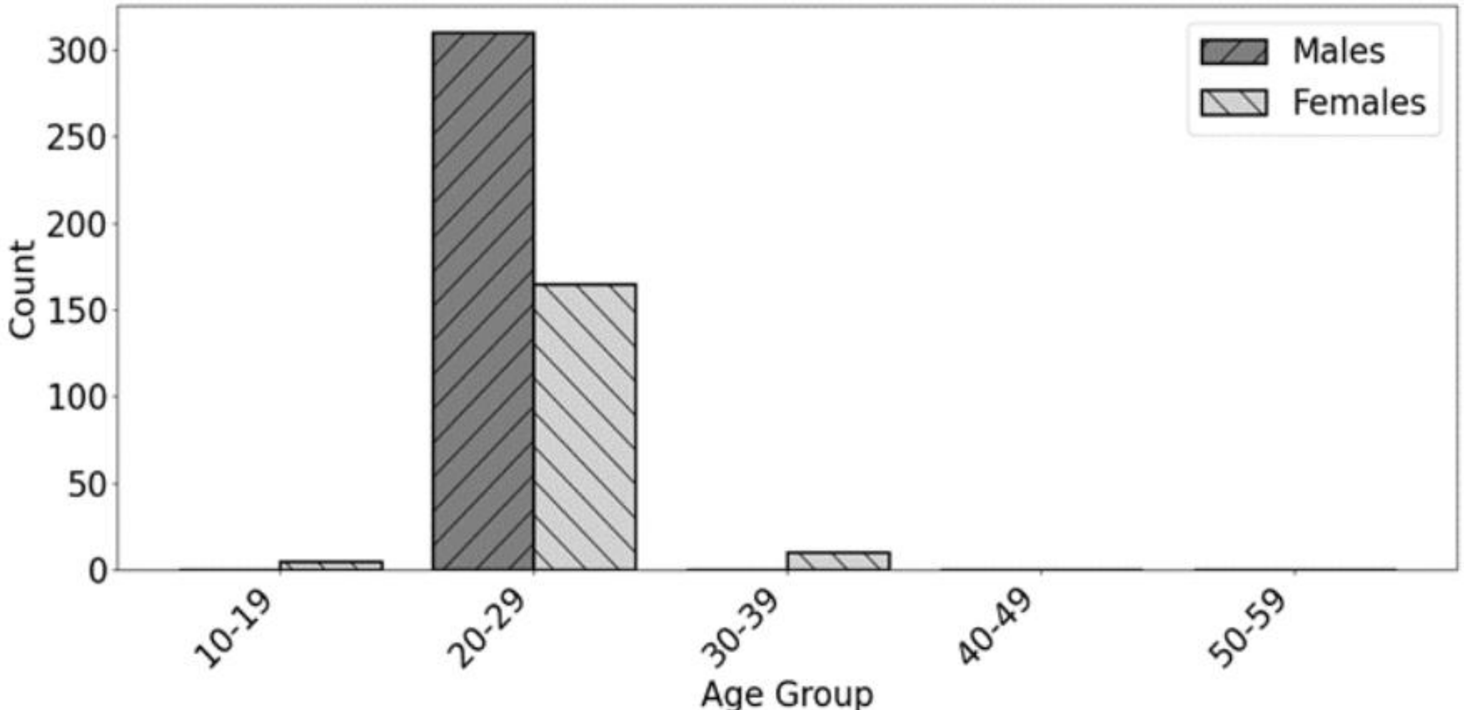
Methodology-[20]
(ASR Architecture)

- Spectrogram and positional embedding pass through Conv layers
- Next, it undergoes MHSA (Multi-Head Self-Attention) layer
- The output further passes through a Feed Forward layer
- Subsequently, it is processed by a Convolution layer
- Followed by another Feed Forward operation and Time Reduction
- Then, time recovery is applied
- Full transcription is achieved through an LSTM decoder

Methodology-[21] (Voice Recordings)

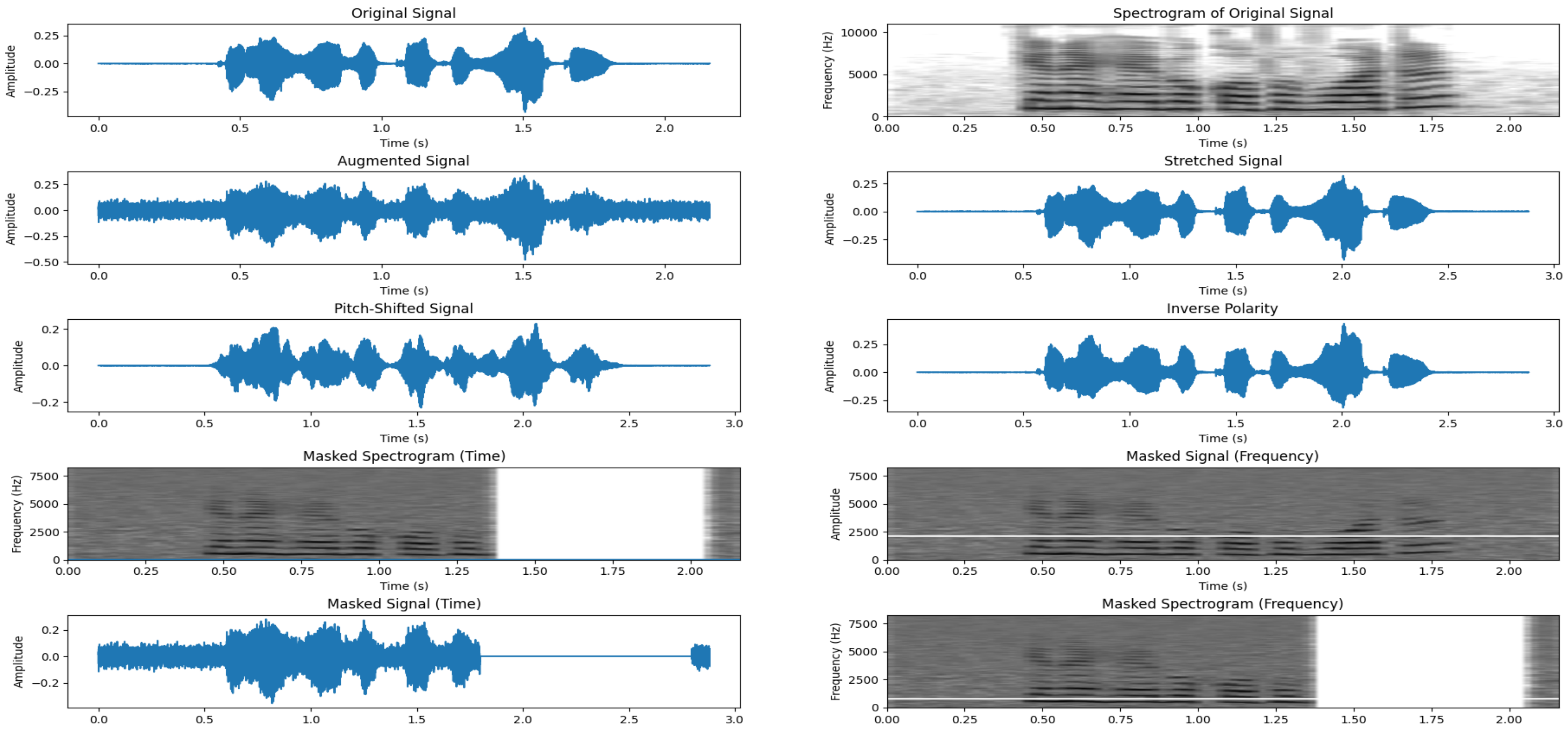


Before Augmentation

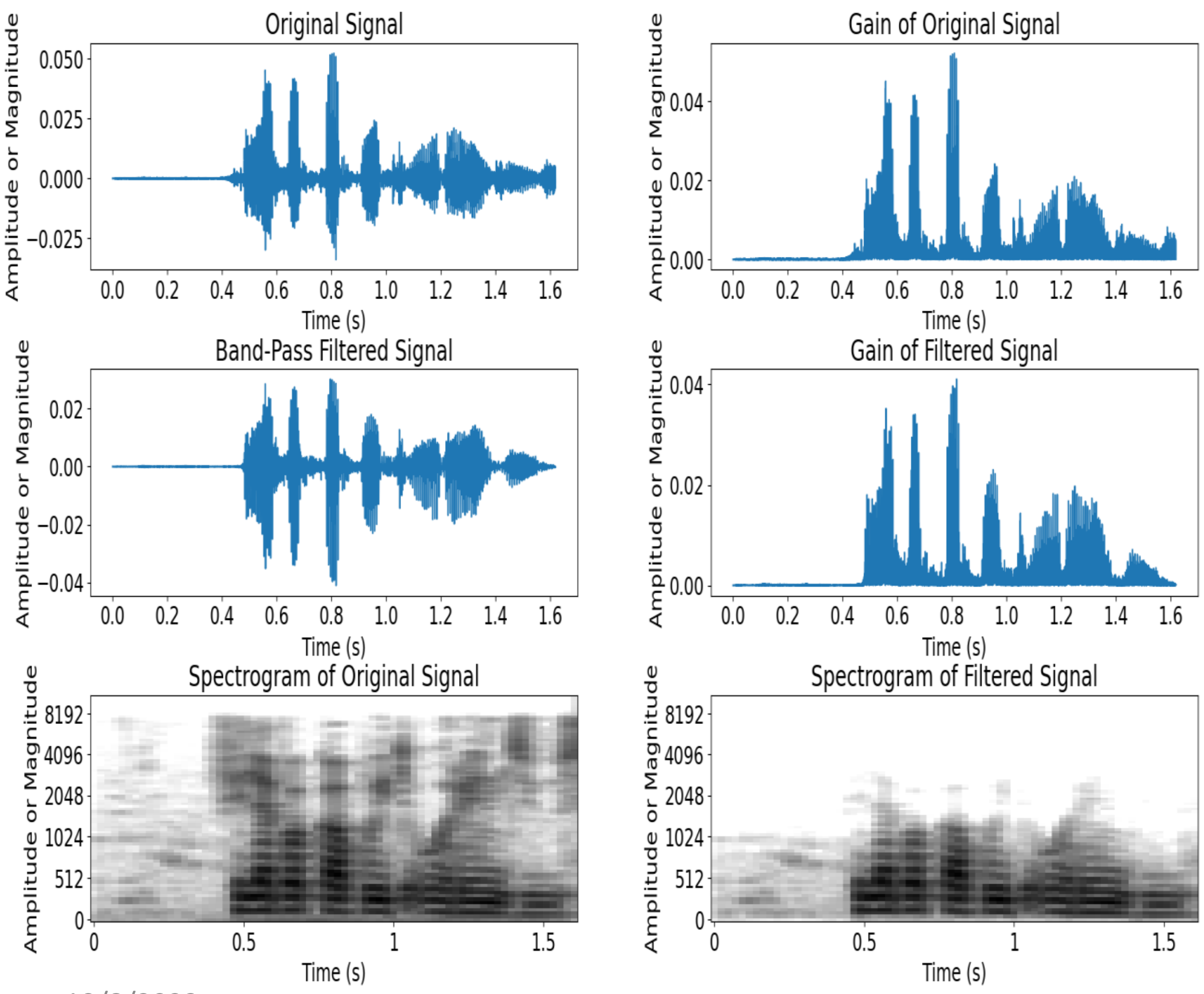


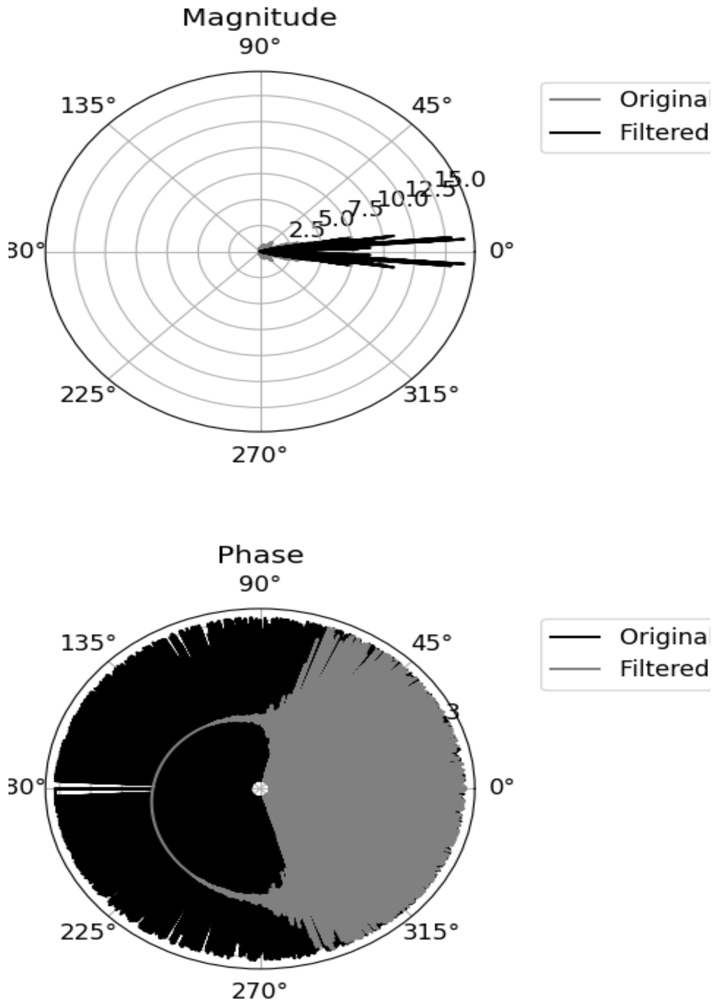
After Augmentation

Methodology-[22]
(Audio Augmentations for Female Voice)



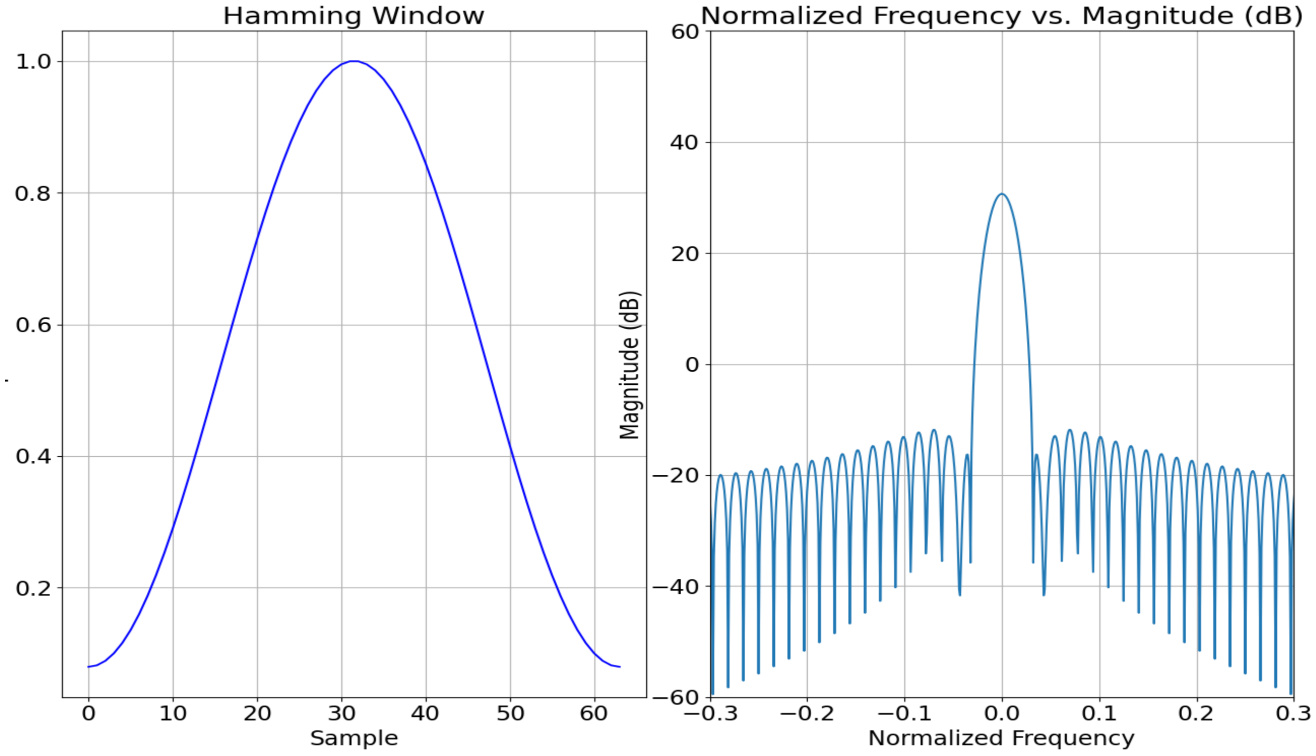
Methodology-[23]
(Background Noise Removal Using Band Pass Filter)





Methodology-[24]
(Hamming window)

Parameters	Values	No of Samples
Sample rate	16000Hz	1600
Frame length	20ms	320
Frame shift	10ms	160



Methodology-[25]
(Voice Commands and Feedbacks)

Drink	Variation 1	Variation 2	Variation 3
Water	Bring me a glass of water!	Can I have a glass of water?	A glass of water, please.
Sprite	Can I have a glass of sprite?	Fetch me a glass of sprite.	Sprite, please.
Orange Juice	A glass of orange juice, please.	I will have one glass of Orange Juice.	Orange Juice for me.
Coffee	I will have a glass of coffee.	Brew me some coffee	Can I have some coffee

Commands

Case	Feedback
No drink available	I’m sorry but we are currently out of that drink, would you like something else
Obstacle in the way	There’s an obstacle in the way. Please clear the path
Incorrect Command	I’m sorry, I didn’t understand your command ,Please give me a drink related command
Target position out of reach from arm	The target is out of my reach

Feedbacks

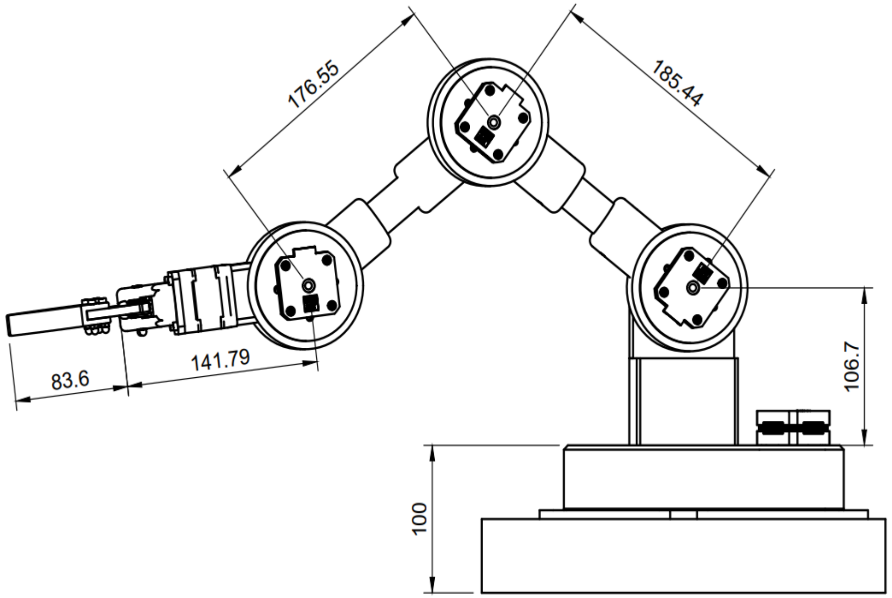
Methodology-[26] (Confusion Matrix for Commands)

Predicted Labels

True Labels	YOU	ARM	ORANGE	I	PROVIDE	CAN	SPRITE	JUICE	OF	PLEASE	COFFEE	COLD	HAVE	UP	POUR	SERVE	WOULD	LIKE	GET	MAKE	FRESH	FILL	WATER	WITH	DELICIOUS	CHILLED	HOT	HEY	ME	A
YOU	21	0	0	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ARM	3	67	0	1	0	0	0	0	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ORANGE	0	0	41	0	0	0	0	3	4	4	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	7
I	0	0	0	49	0	1	0	0	0	1	0	3	2	0	0	4	1	3	0	0	0	0	0	0	0	0	0	1	6	7
PROVIDE	0	0	0	0	12	2	0	0	0	1	0	0	5	1	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1
CAN	0	0	0	6	0	42	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0
SPRITE	0	0	0	0	0	0	43	0	2	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15
JUICE	0	0	4	0	0	0	0	41	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
OF	0	0	3	0	0	0	5	3	144	6	3	0	0	0	0	0	0	0	0	0	0	6	0	0	1	0	0	0	6	31
PLEASE	0	0	7	5	0	1	0	3	0	106	0	0	0	1	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	16
COFFEE	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
COLD	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	5	0
HAVE	1	1	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UP	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
POUR	2	0	0	0	1	5	0	0	0	0	0	0	0	0	15	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0
SERVE	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	32	2	0	0	0	0	0	0	0	0	0	0	3	2	11
WOULD	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	32	0	0	0	0	0	5	0	0	0	0	1	5	7
LIKE	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	1	0
GET	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	2
MAKE	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
FRESH	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	1	0	0	0
FILL	0	0	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	47	0	0	1	0	0	0	0	9
WATER	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0	61	0	0	0	0	1	6	6
WITH	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	2
DELICIOUS	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	7	0	0	0	0	1
CHILLED	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	6
HOT	0	0	0	2	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	67	0	0	0
HEY	9	0	0	0	1	0	0	0	0	0	0	0	1	0	0	7	5	2	0	0	0	0	0	0	0	0	0	92	10	3
ME	0	0	2	0	0	1	0	0	2	0	0	0	0	0	0	0	7	7	0	0	0	4	10	0	0	0	0	5	195	27
A	0	0	3	1	0	0	1	2	18	8	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	7	15	201

Methodology-[27]
(Kinematics of Robotic Arm)

- Deals with the motion of robotic arm
- Doesn’t consider forces and torques associated with it
- Provides the homogeneous transformation matrix
- Denavit-Hartenberg (D-H) parameters are used to find the transformation matrix



Methodology-[28]
(D-H Parameters and Homogeneous Transformation Matrix)

$$H_4^0 = \begin{bmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & c_1(a_4 c_{234} + a_2 c_2 + a_3 c_{23}) \\ c_{234} s_1 & -s_1 s_{234} & -c_1 & s_1(a_4 c_{234} + a_2 c_2 + a_3 c_{23}) \\ s_{234} & c_{234} & 0 & a_4 s_{234} + a_2 c_2 + a_3 s_{23} + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Transformation Matrix

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

- D-H parameters can be used to derive the homogeneous transformation matrix

Methodology-[29]
(Homogeneous Transformation Matrix)

$$H_4^0 = \begin{bmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & c_1(a_4 c_{234} + a_2 c_2 + a_3 c_{23}) \\ c_{234} s_1 & -s_1 s_{234} & -c_1 & s_1(a_4 c_{234} + a_2 c_2 + a_3 c_{23}) \\ s_{234} & c_{234} & 0 & a_4 s_{234} + a_2 c_2 + a_3 s_{23} + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Provides rotation matrix as well as displacement vector of end effector w.r.t base frame
- The first three elements of the third column provide equations for forward kinematics

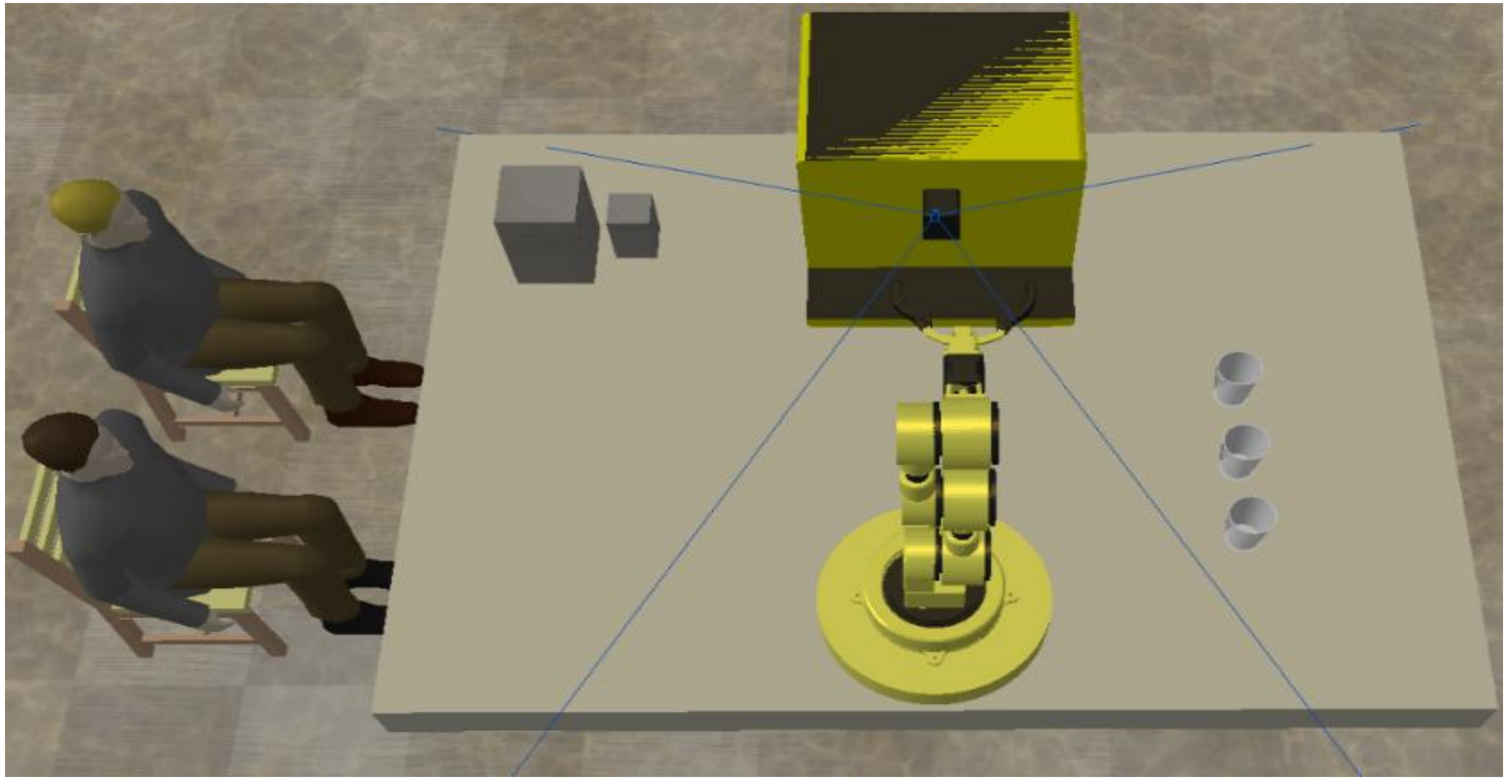
Methodology-[30]
(Jacobian Pseudo-Inverse Method)

$$J = \begin{bmatrix} -s_1(a_4c_{234} + a_2c_2 + a_3c_{23}) & -c_1(a_4s_{234} + a_2s_2 + a_3s_{23}) & -c_1(a_4s_{234} + a_3s_{23}) & -a_4c_1s_{234} \\ c_1(a_4c_{234} + a_2c_2 + a_3c_{23}) & -s_1(a_4s_{234} + a_2s_2 + a_3s_{23}) & -s_1(a_4s_{234} + a_3s_{23}) & -a_4s_1s_{234} \\ 0 & a_2c_2 + a_4c_{234} + a_3c_{23} & a_4c_{234} + a_3c_{23} & a_4c_{234} \end{bmatrix}$$

Jacobian Matrix

- Relates the changes in joint angles with changes in end effector positions
- Is a rectangular matrix and so no inverse is possible
- Pseudo inverse of (J) is derived
- Until the tip reaches the target position, new set of joint angles are calculated

Results and Analysis-[1]
(Robotic Actions)



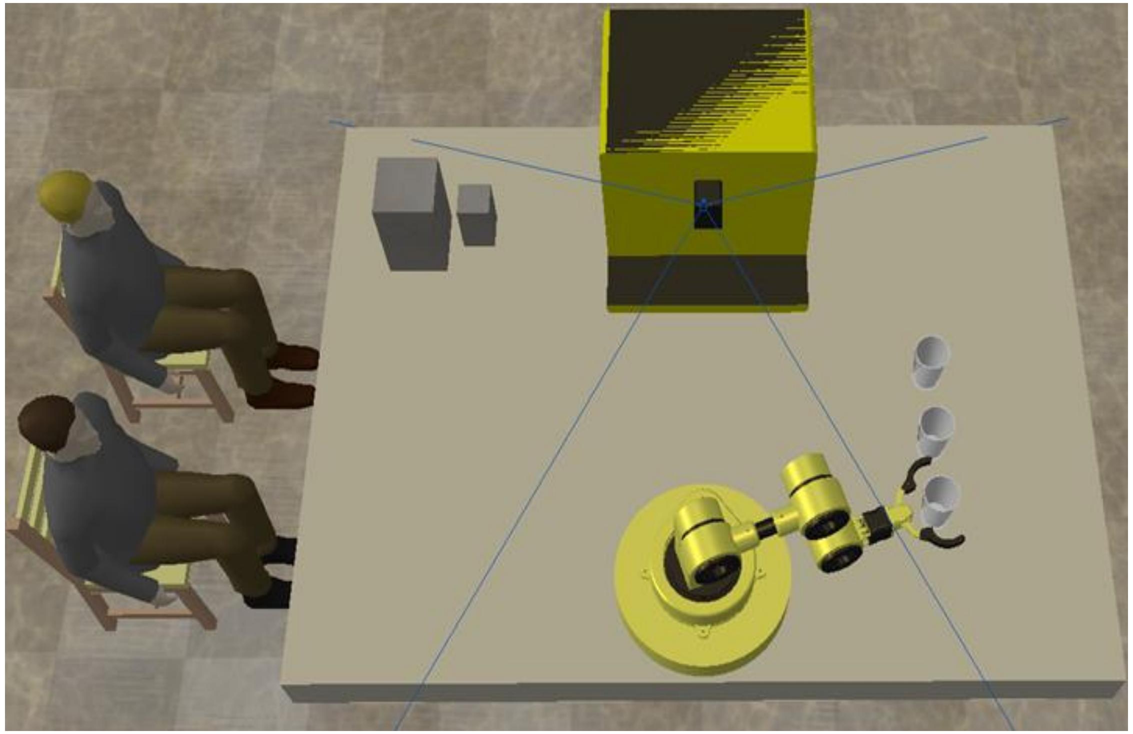
Robot Arm at Default Position

Results and Analysis-[2]
(Robotic Actions)

$$H = \begin{bmatrix} 0.32 & -0.047 & -0.94 & 0.30 \\ -0.93 & 0.13 & -0.33 & -0.84 \\ 0.14 & 0.98 & 0 & 0.02 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H^* = \begin{bmatrix} 0.33 & -0.06 & -0.96 & 0.29 \\ -0.86 & 0.13 & -0.33 & -0.80 \\ 0.18 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- H denotes the matrix derived manually
- H* is extracted from simulator

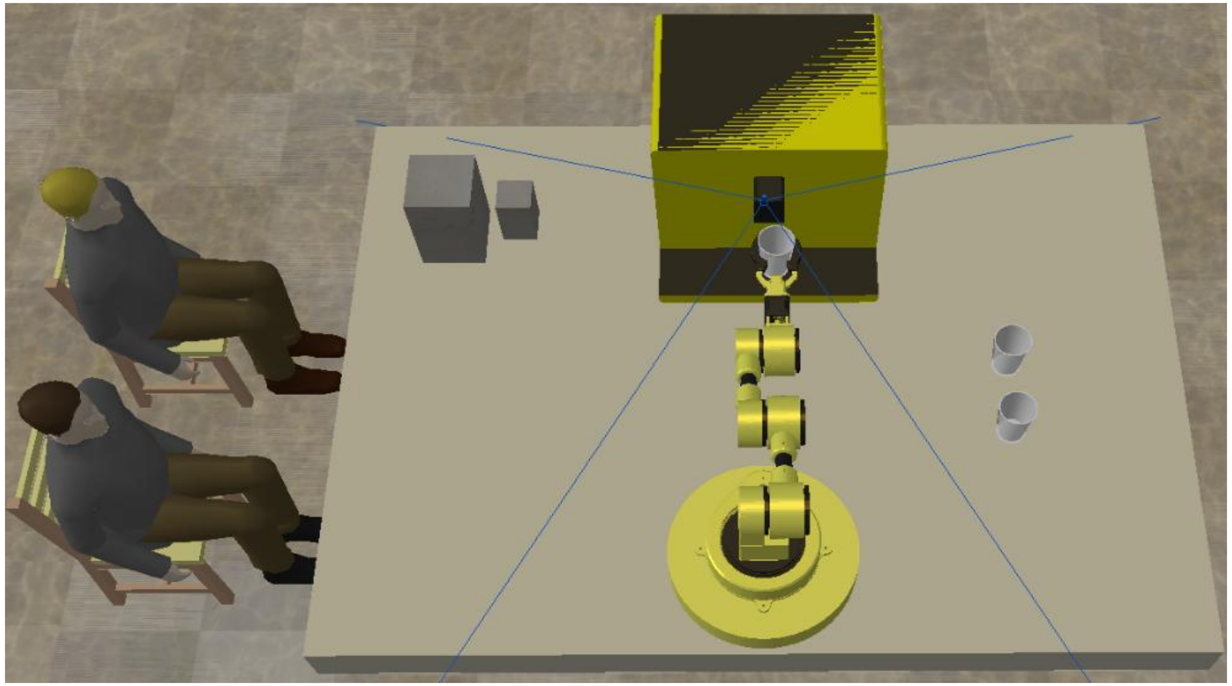


Robot arm grabbing empty glass

Results and Analysis-[3]
(Robotic Actions)

$$H = \begin{bmatrix} 0.87 & -0.48 & 0.015 & 0.95 \\ 0.01 & 0 & -0.09 & 0.01 \\ 0.47 & 0.87 & 0 & 0.52 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H^* = \begin{bmatrix} 0.9 & -0.55 & 0.05 & 0.92 \\ -0.06 & -0.08 & -0.1 & 0.04 \\ 0.52 & 0.93 & 0 & 0.52 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

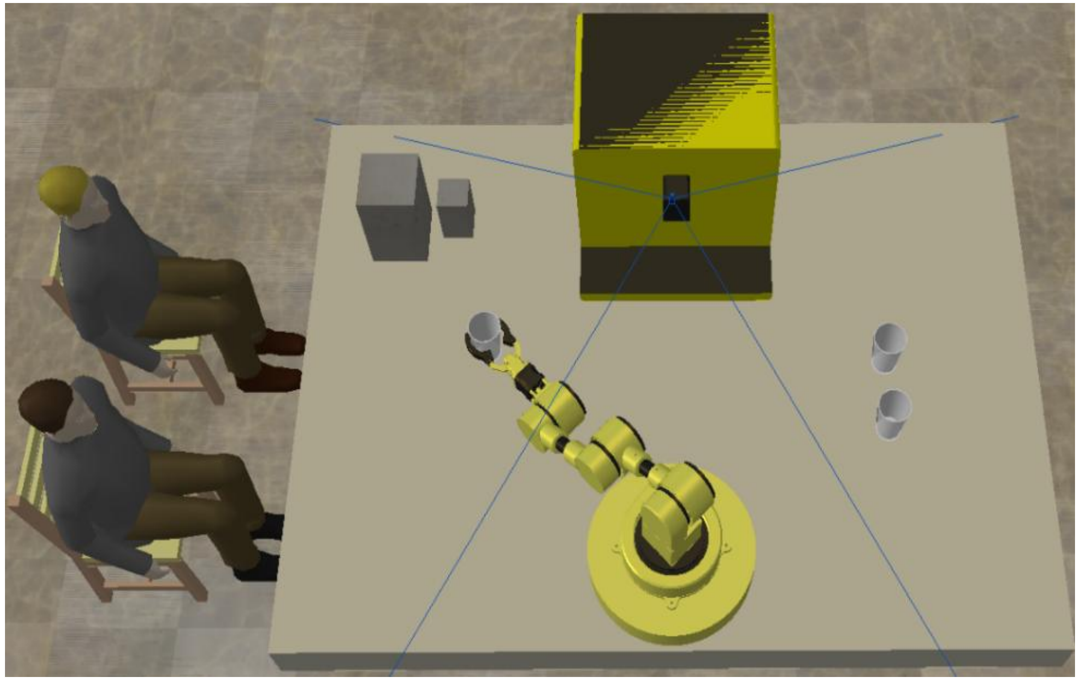


Dispensing the Ordered Drink

Results and Analysis-[4]
(Robotic Actions)

$$H = \begin{bmatrix} 0.52 & -0.47 & 0.71 & 0.62 \\ 0.53 & -0.46 & -0.72 & 0.62 \\ 0.67 & 0.75 & 0 & 0.66 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

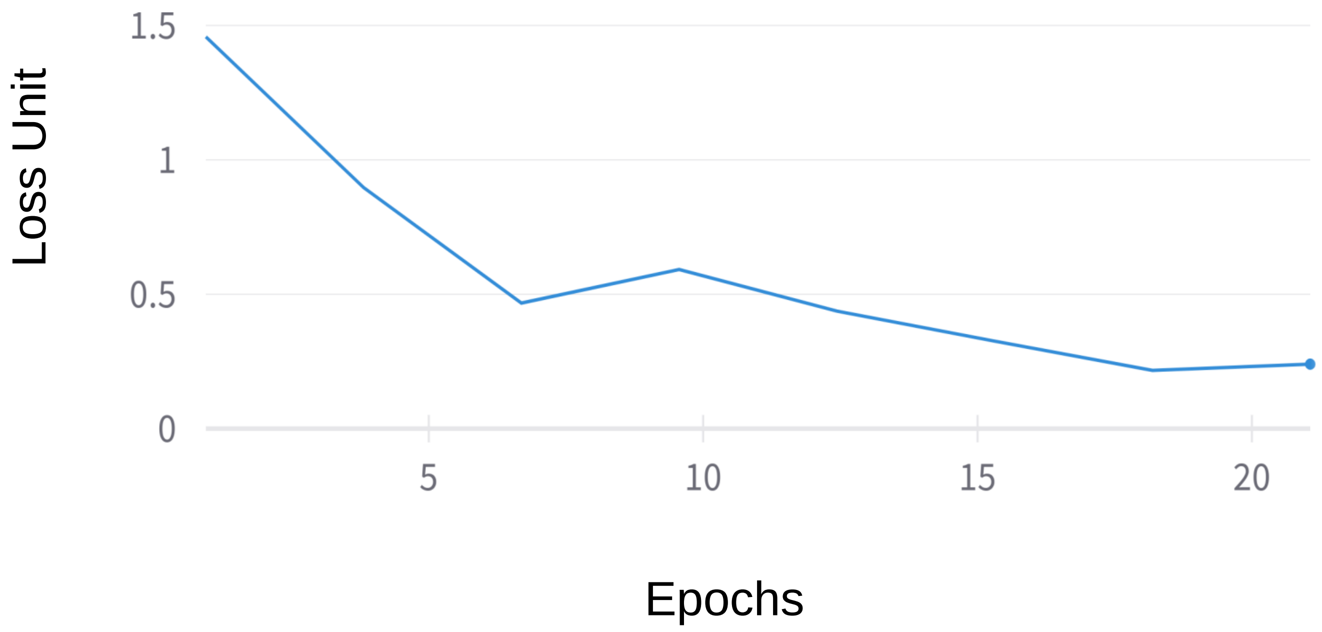
$$H^* = \begin{bmatrix} 0.61 & -0.52 & 0.70 & 0.70 \\ 0.45 & -0.44 & -0.65 & 0.54 \\ 0.66 & 0.68 & 0 & 0.60 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Robotic arm serving drink to client

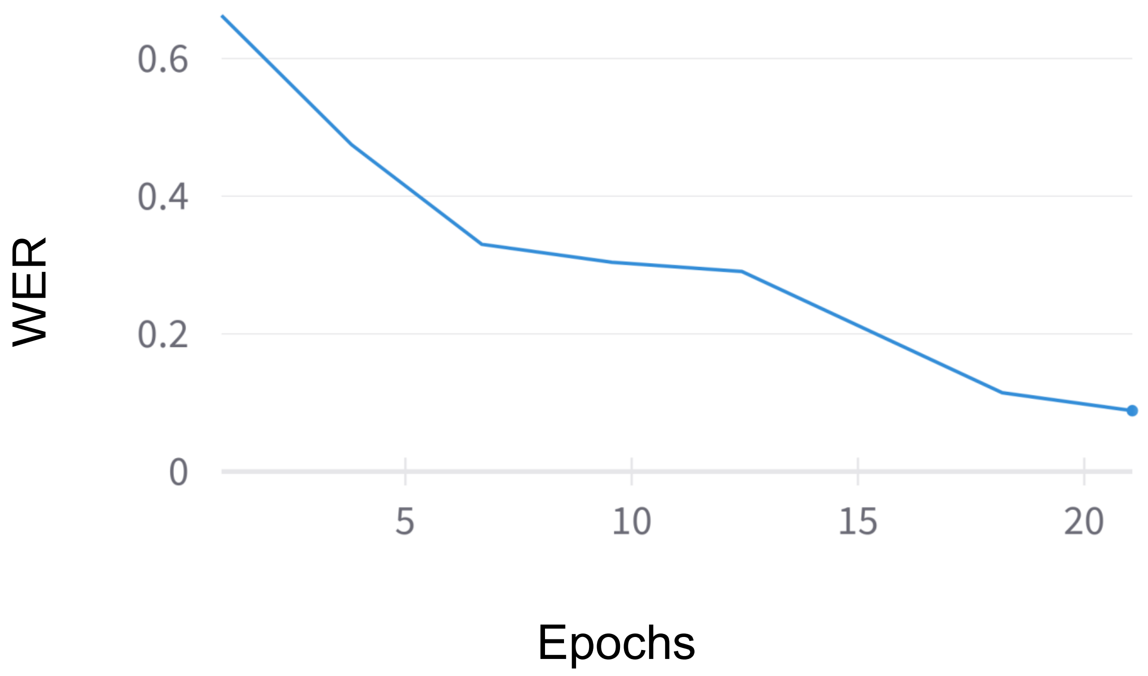
- Maximum discrepancy between H and H* is 0.1
- Reason of discrepancy is due to lack of trajectory planning

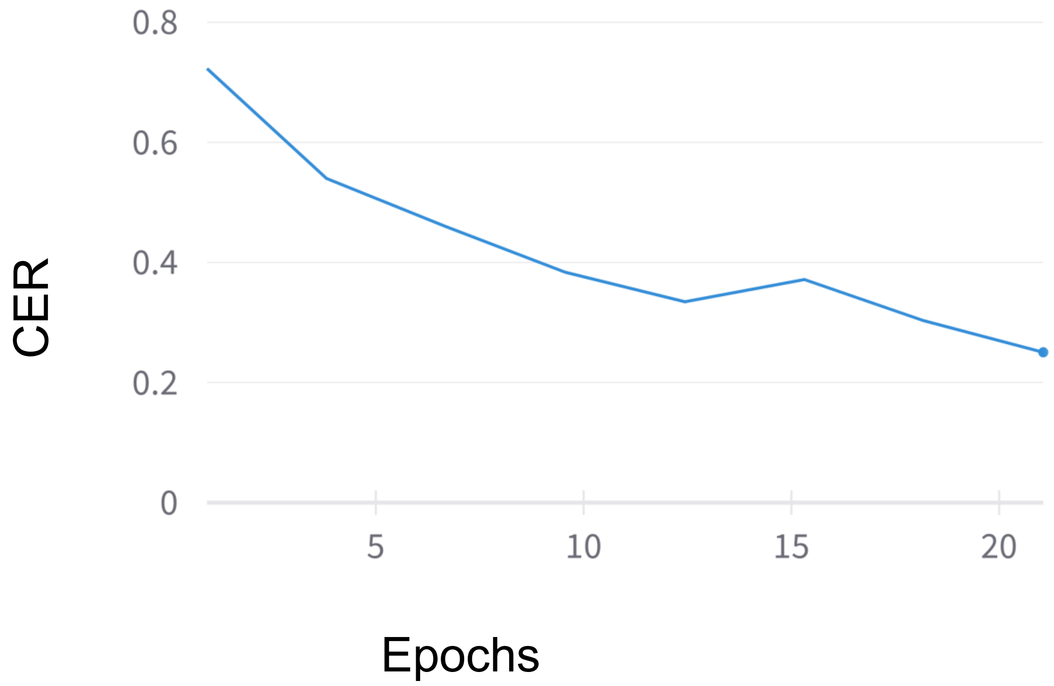
Results and Analysis-[5] (ASR Training (loss))



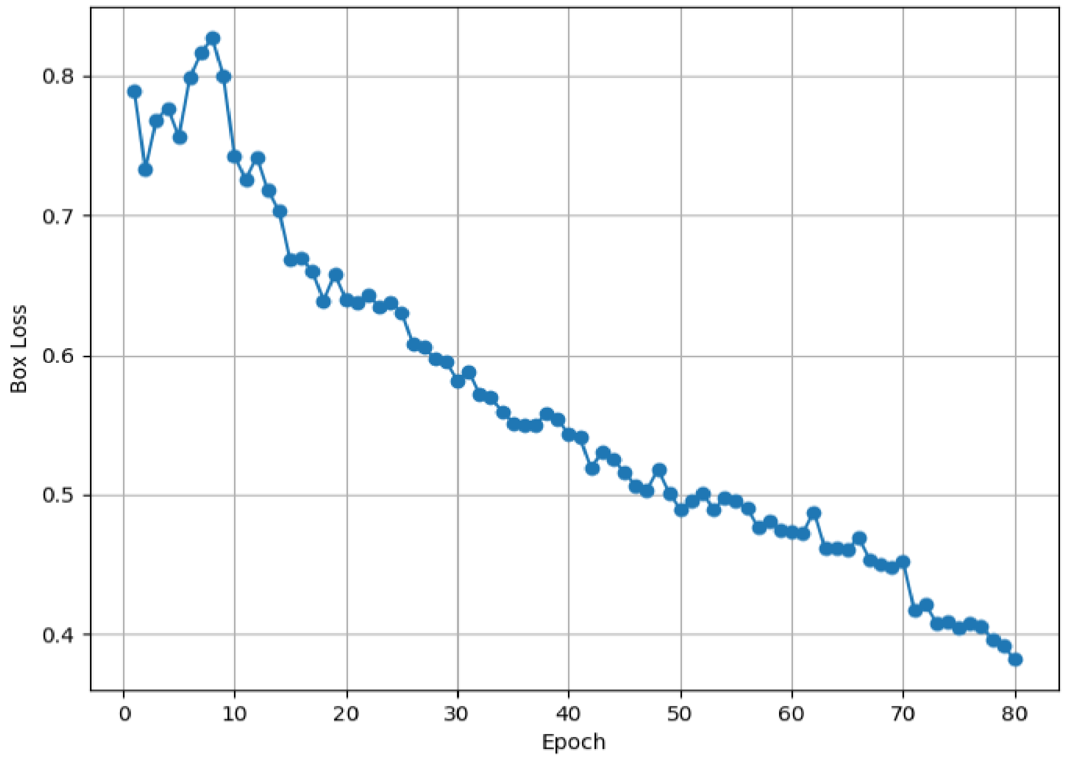
Hyperparameter	Values
Batch Size	50
Epochs	30
Learning Rate	0.001
Dropout	0.1

Results and Analysis-[6]
(ASR Training (CER,WER))

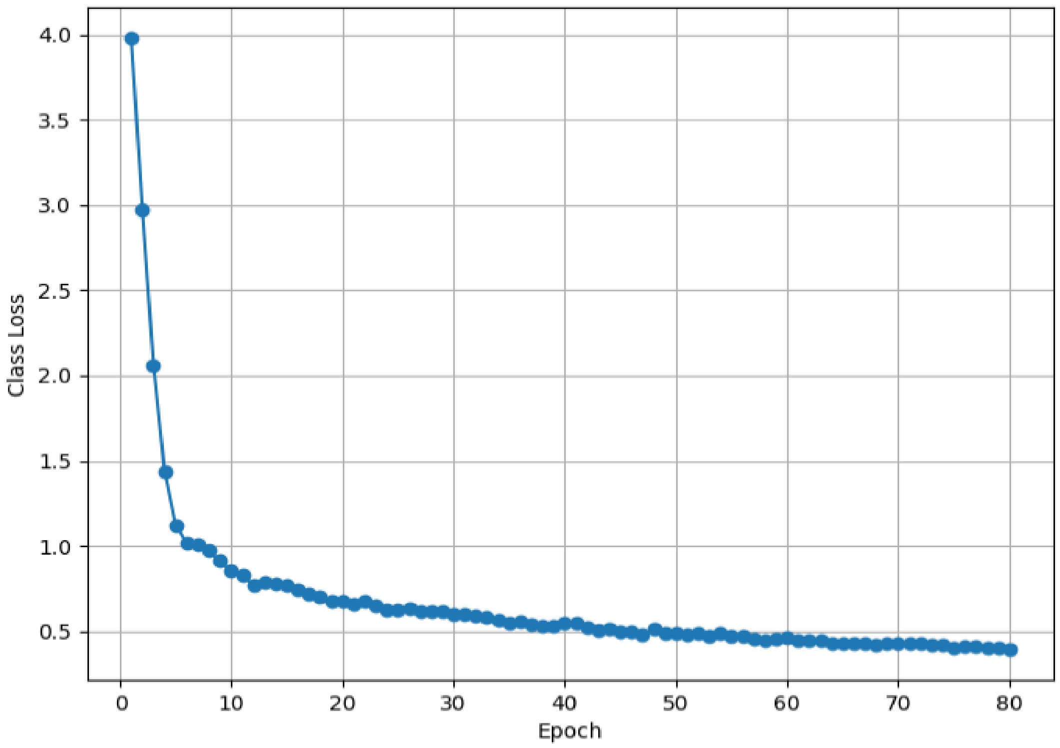




Results and Analysis-[7]
(YOLOv8 Training (Losses))

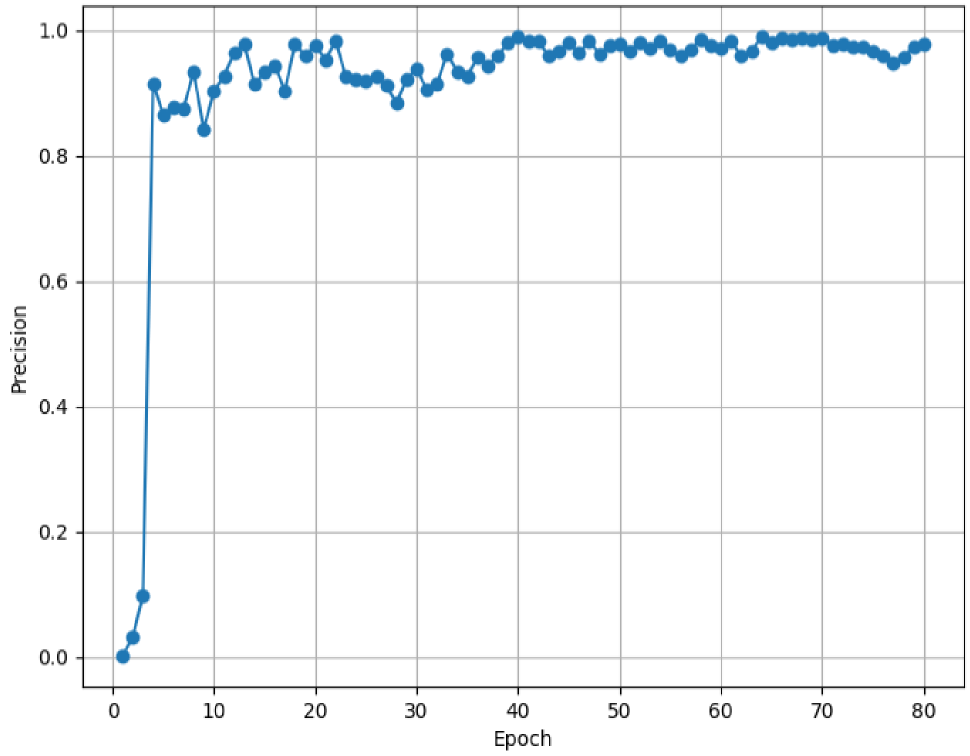


Box Loss vs Epoch

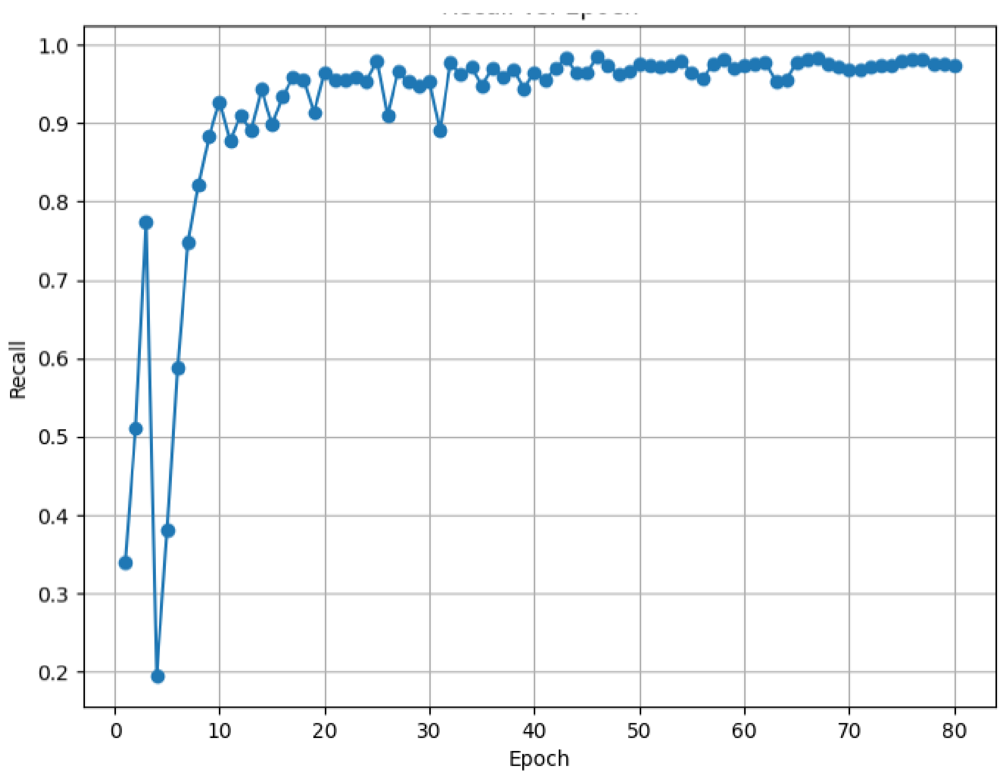


Class Loss vs Epoch

Results and Analysis-[8]
(YOLOv8 Training (Accuracy Metrics))

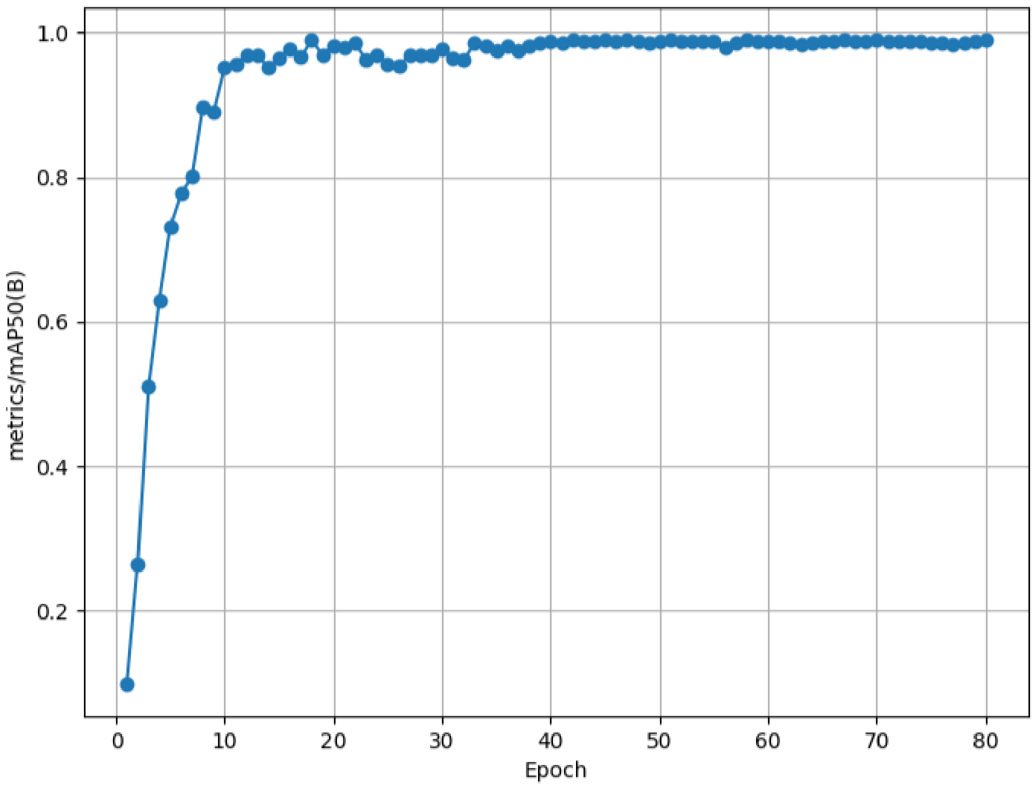


Precision vs Epoch

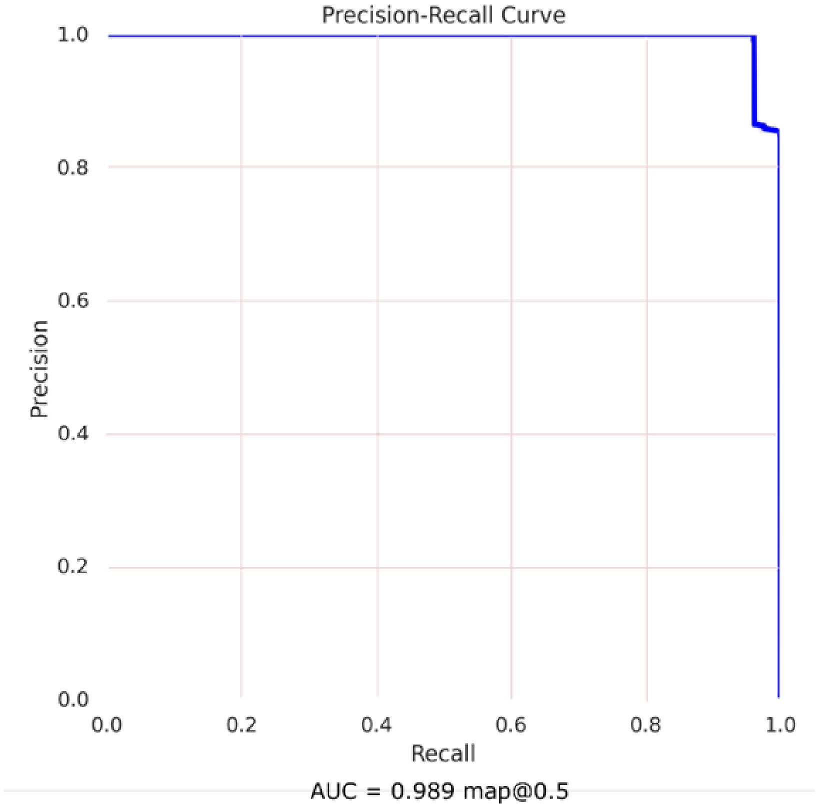


Recall vs Epoch

Results and Analysis-[9]
(YOLOv8 Training (Accuracy Metrics))

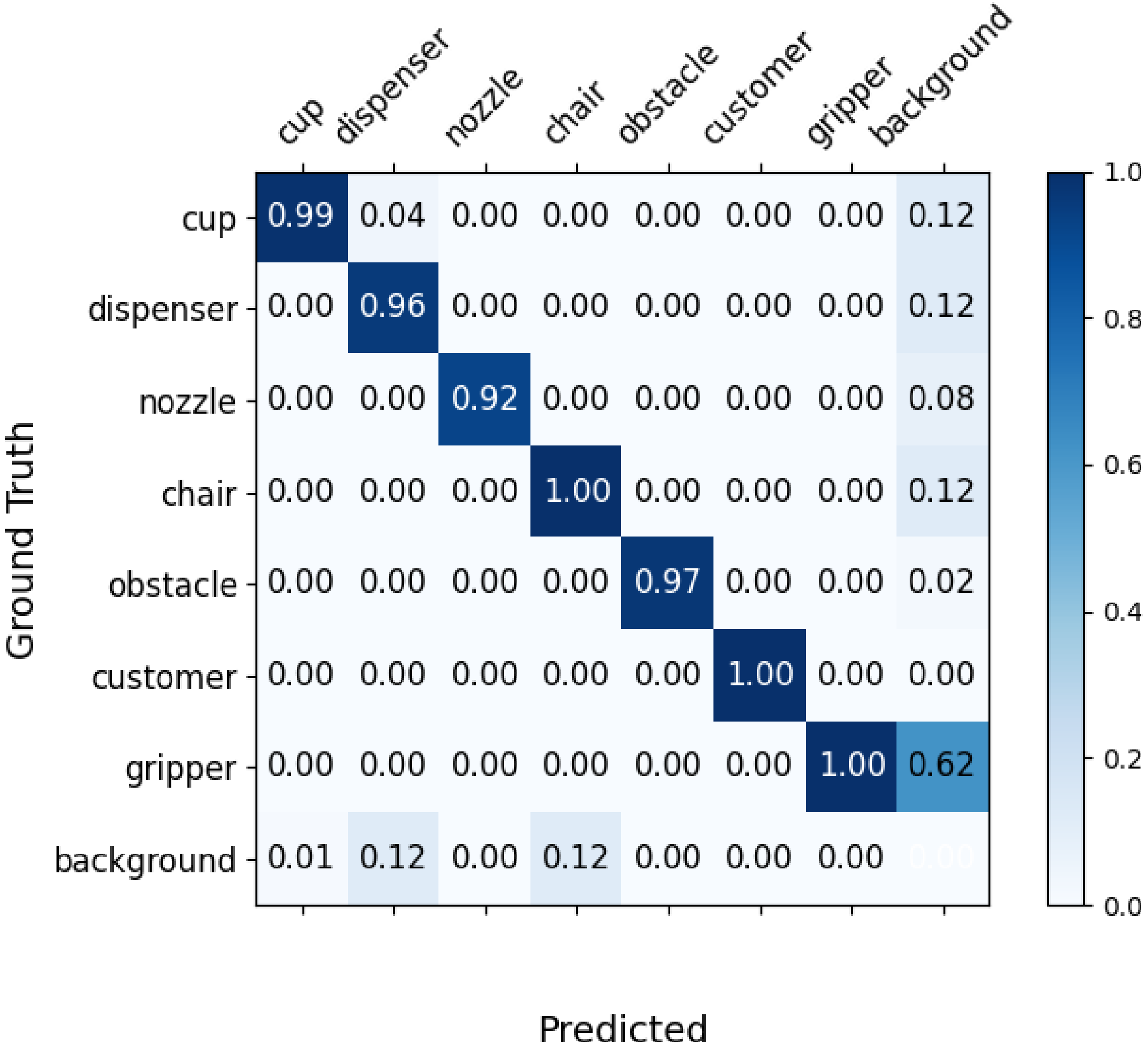


mAP50 vs Epoch

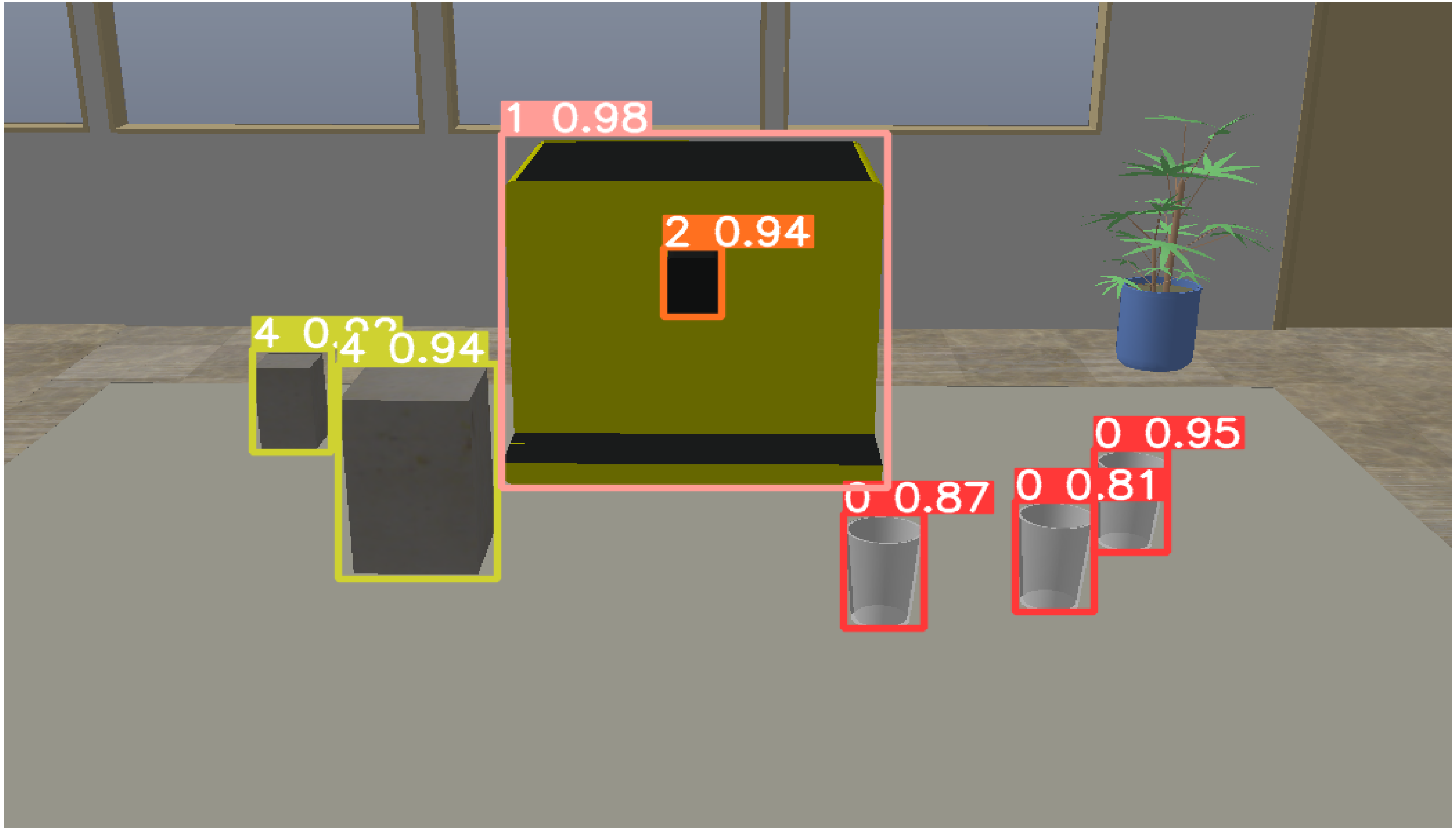


PR Curve

Results and Analysis-[10]
(YOLOv8 Training (Confusion Matrix))



Results and Analysis-[11]
(YOLOv8 Training (Detections))



Remaining Tasks

- Implementation of MPC controller in both software and in embedded platform
- 3D printing of all components of 4-DOF robotic arm and building dispenser system
- Integration of whole system in embedded platform and testing its functionality
- Perform performance comparison between simulation and reality

References-[1]

[1] Voice Transformer Network: Sequence-to-Sequence Voice . https://www.isca-speech.org/archive_v0/Interspeech_2020/pdfs/1066.pdf

[2] L. Dong, S. Xu and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5884-5888, doi: 10.1109/ICASSP.2018.8462506.

[3] S. Revay and M. Teschke, “Multiclass Language Identification using Deep Learning on Spectral Images of Audio Signals,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.04348

[4] L. Rafael Stefanel Gris and A. Candido Junior, “Automatic Spoken Language Identification using Convolutional Neural Networks.” [Online]. Available: http://www.freesound.org

References-[2]

[5] P. Kaur, Q. Wang, and W. Shi, “Fall Detection from Audios with Audio Transformers,” Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.10659

[6] A. A. Q. Mohammed, J. Lv, and M. D. S. Islam, “A deep learning-based end-to-end composite system for hand detection and gesture recognition,” *Sensors (Switzerland)*, vol. 19, no. 23, Dec. 2019, doi: 10.3390/s19235282.

[7] M. Musaev, I. Khujayorov, and M. Ochilov, “Image Approach to Speech Recognition on CNN,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2019. doi: 10.1145/3386164.3389100.

[8] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.01778

References-[3]

[9] R. P. A. Petrick and M. E. Foster, “Planning for Social Interaction in a Robot Bartender Domain.” [Online]. Available: www.aaai.org

[10] A. V. Oppenheim and A. S. Willsky, Signals and Systems, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996. [Online]. Available:https://www.academia.edu/37486178/Signals_and_Systems_2nd_Edition_by_Oppenheim_

[11] Ultralytics, "Ultralytics/ultralytics: Open-source deep learning inference & training on YOLOv3/YOLOv4/PyTorch," GitHub. [Online]. Available: https://github.com/ultralytics/ultralytics