# SMART ROBOTIC ARM MANIPULATION (S.R.A.M)


## B.M.S Institute of Technology and Management

## Department of Computer Science and Engineering

**Team Members:**

1. S N Shashank

   Email: 1by19cs128@bmsit.in

   phone: 8431897882

2. Saksham Bejwani

   Email: 1by19cs132@bmsit.in

   Phone: 7739097413

3. Naitik Goyal

   Email: 1by19cs089@bmsit.in

   Phone: 8490730114

**Project Guide:**

Prof. Ambika G N

Email: ambikagn@bmsit.in

Phone: 9902323632



## Keywords

ROS – Robot Operating System

Robotic arm

YOLO – You Only Look Once

DeepSpeech – speech recognition library

## Introduction / background

- Elders and handicapped individuals who do not have caretakers face several difficulties in their home environment. One of the major challenges they may face is difficulty with daily tasks such as picking the desired object from the nearby table etc.
- But by doing so in the absence of the care takers will lead to more accidents and injuries.
- Robotic arms allow bedridden individuals to fetch things without the need for extensive training or physical dexterity. Additionally, the use of robotic hands can reduce the physical strain on caregivers, allowing them to focus on other tasks and providing relief to the patient.
- The main objective is to design a robotic arm that can assist in basic activities of daily living, such as getting thinks from the table near to them. This technology can help patients maintain their independence and reduce their reliance on caregivers, which is especially important for those who live alone or do not have access to regular caregiving services.
- The arm is controlled by a combination of speech recognition, object detection, and ROS, which allows for precise and accurate movements. The use of AI and machine learning algorithms enables the robotic arm to adapt and learn new tasks over time, providing more personalized care to patients. This technology has the potential to revolutionize the way we care for elderly and bedridden patients, providing them with greater autonomy and improving their quality of life.

# Objectives

- To build and program a robotic arm that can move in 3D space of real world.
- Train the state-of-the art object detection model to detect and localize objects in the scene.
- Implement speech recognition algorithms/models to take user voice commands as input to the system.
- Develop controller scripts to move the arm and grip, pick and place objects in the world.
- Develop and program scripts to transform coordinates between different frames.
- Tune the controllers to make the arm precise enough to pick correct object and get it without dropping and collision.
- Finally building a pipeline to integrate all the individual components into one system.

# Methodology

- The robotic arm is built using 3D printed parts, with servo motors acting as actuators to move the arm.
- These are connected to Nvidia Jetson nano through GPIO pins.
- Kinect 360 camera is used to get the RGB-D information.
- A microphone is used to get the speech input to the ROS.
- The ROS and required libraries are installed in Nvidia Jetson nano.
- The user speech input is transcribed to English using DeepSpeech speech recognition library.
- The image is fed to the python pipeline script which is running on the jetson nano.

- The YOLO will detect the objects from the given input image and returns the object coordinates.
- These coordinates are then transformed with respect to the world coordinate frame and are given to the MoveIt (PyKDL) path planner.
- The path planned buy the planner is executed by the actuators by using ROS controllers.

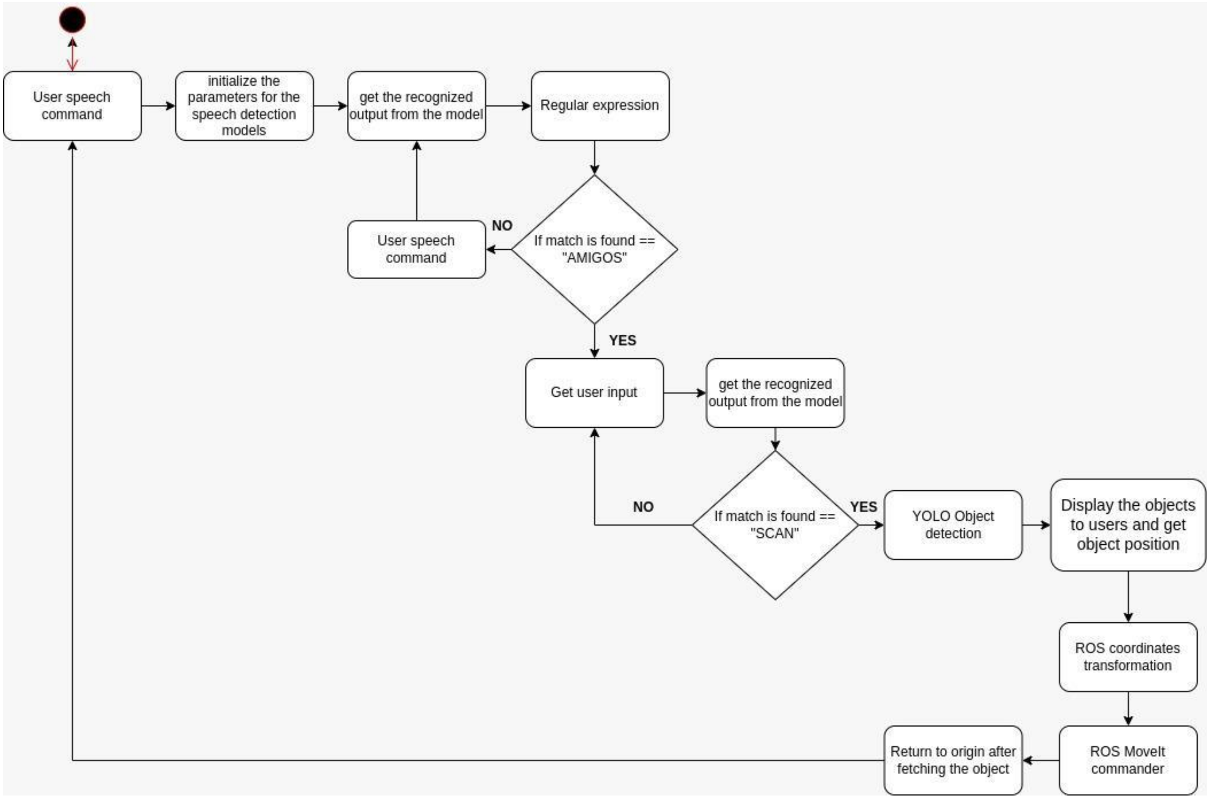


**Fig 1. Control flow diagram of the pipeline**

# Results and Conclusions

- The goal of this project is to build a smart robotic arm manipulator that can move around in the world and fetch the target object. We propose a voice controlled, perception based end-to-end robotic arm manipulator.
- our project successfully developed and tested a simulated robotic arm manipulator system integrating YOLO object detection, DeepSpeech speech recognition, MoveIt motion planning framework, and the PyKDL path planner. The combined capabilities of these components enabled accurate object detection, reliable speech recognition, efficient motion planning.
- This simulated system now is being deployed to a hardware.

# What is the innovation in the project?

The current robotic arm systems lack intelligence and perception. These industry level arms are not suitable for household and remote usage because of their function, usage, cost, and size.

Our proposed project makes sure that the arm is more user friendly by integration of speech recognition and Deep Learning, this integration makes it hassle free and more intuitive.

This can be mounted anywhere such as railing of the bed, chair etc, making this more portable.

# Scope for future work

- Enhanced Object Detection: Improve the object detection capabilities by exploring advanced variants of YOLO or other state-of-the-art object detection algorithms. This could involve refining the accuracy, speed, and robustness of object detection, as well as handling more complex scenarios such as crowded environments or occlusions.
- Intelligent Grasping and Manipulation: Combine the object detection capabilities with intelligent grasping algorithms to enable the robotic arm to autonomously plan and execute grasping actions based on the detected objects. This could involve developing algorithms for optimal grasp planning, adaptive grasping, or learning-based approaches to improve the efficiency and success rate of grasping tasks.
- Natural Language Interaction: Further enhance the speech recognition component by incorporating natural language understanding and dialog management techniques. This would allow the robotic arm to interpret and respond to more complex and context-rich spoken commands, enabling more interactive and intuitive human-robot interactions.
- Adaptive and Learning Systems: Incorporate machine learning and adaptive techniques to enable the robotic arm manipulator to learn from experience and improve its performance over time. This could involve developing algorithms for self-calibration, self-correction to changes in the environment or task requirements.