WLAN EDGE COMPUTER

Group 07

Supervisor : Dr. Sampath Edirisinghe
18/ENG/112 : U.D.P.D Uduwela
18/ENG/037 : D.C. Jayamali
18/ENG/080 : V.R.K.M. Premabandhu
18/ENG/062 : M.A.D.T.L. Madurapperuma

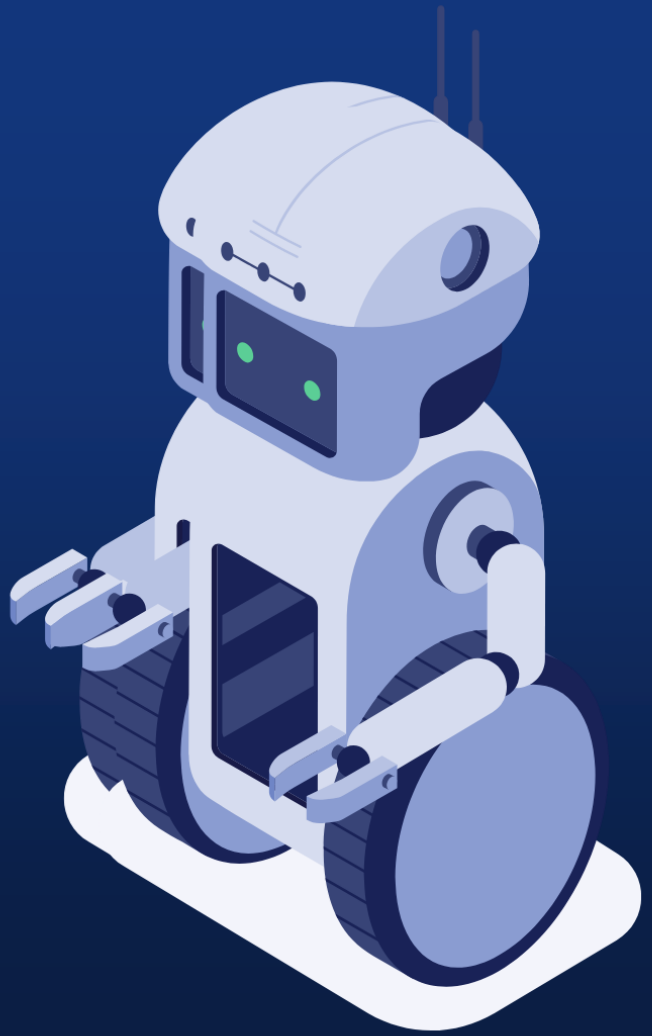
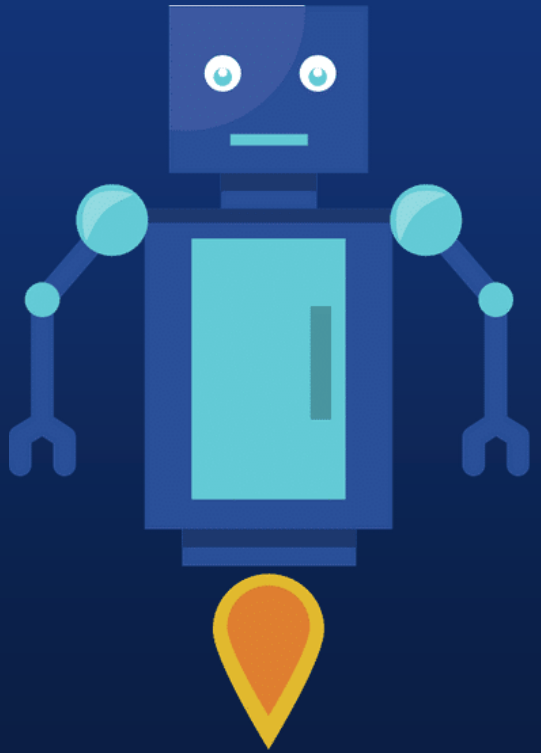


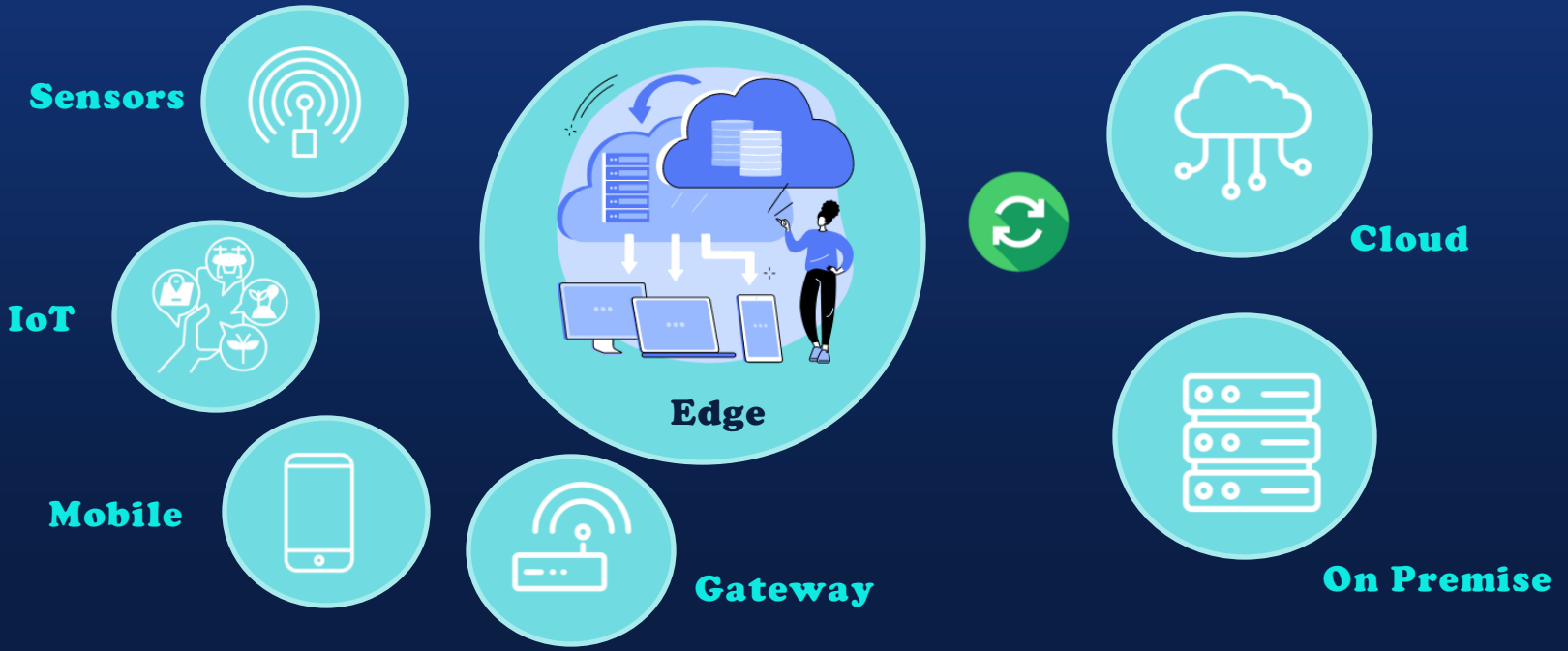
TABLE OF CONTENTS



BACKGROUND

What is Edge Computing?

- Edge computing stores and processes data locally , on the IOT or Mobile device it was created on ,at the "edge" of the network. Only necessary data is synced to cloud or an on premise server.
- Distributed computing paradigm that brings computation and data storage closer to the sources of data.

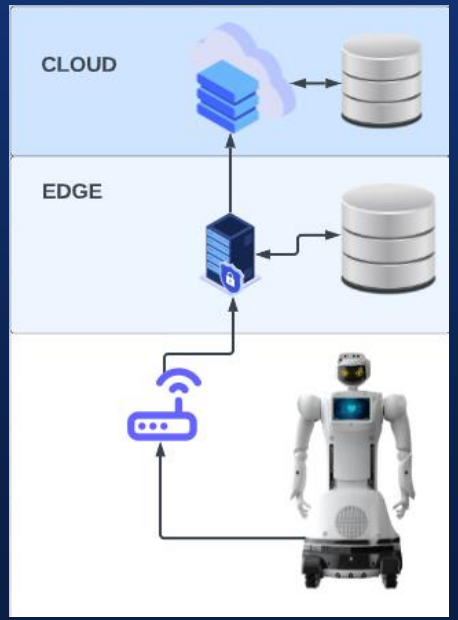


What is Edge Computing for robotics?

- Ability to process information in industrial gateways and on-premises servers means robots can collect, process, and store information at the edge. This greatly reduces the latency and connectivity requirements for robotics operations, while alleviating security and privacy concerns.

Use Cases

- Industrial Robots
- Medical Robots
- Autonomous Vehicles
- Smart grid



RELATED WORK

Existing research papers contain work on

- Analysis of the current edge computing landscapes
- Current benefits and Challenges of edge computing
- ML systems have been deployed at the edge of computer networks, focusing on the operational aspects including compression techniques, tools, frameworks and hardware used in successful applications of intelligent edge systems
- General cloud-based architecture for real-time robotics computation, and then implementing a Particle Filtering-based SLAM algorithm in a multi-node cluster in the cloud.

Key Benefits of Edge

"Edge Computing brings data processing power at the edge of the network closer to the source."

Interoperability between Legal & Modern Devices

Cost-effective solution

Reliable operations with intermittent connectivity

Faster Response Time

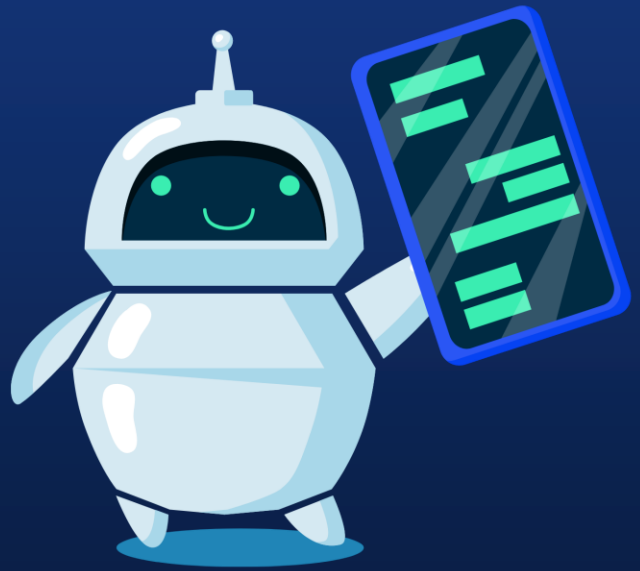
Security & compliances

Proposed Approach

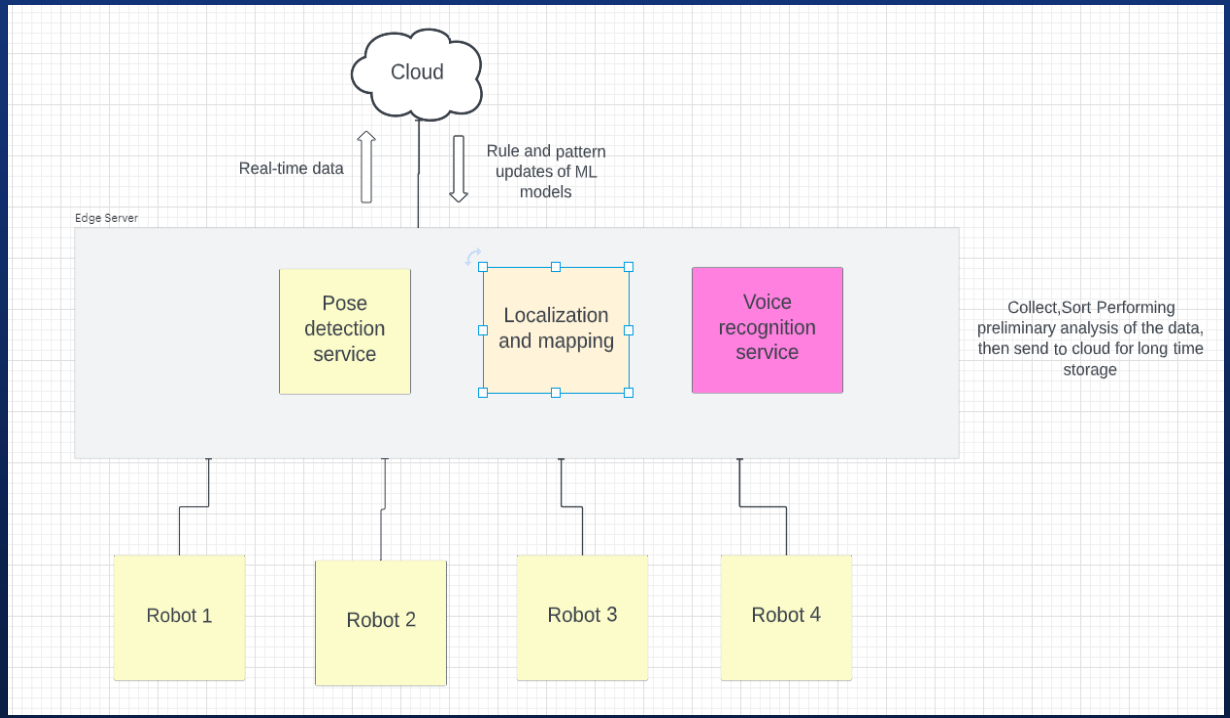
A multi-robot system which is based on edge computing that yields better real-time and network transmission performance

A robot can perform

- Localization and mapping
- Voice recognition
- Pose estimation



Proposed Approach



Proposed Approach cntd..

Voice Recognition

A voice command is given to a selected robot

Recognized voice command is then directed to the server

According to the given command the server commands the selected robot in the edge computing network to perform the relevant action

Proposed Approach cntd..

Pose Detection

A video sequence of the human figure is directed to the server

According to the given video, the pose of the human figure is then detected and necessary details are sent to the robot to perform further actions

Proposed Approach cntd..

Localization and Mapping

A robot that can localize itself and find a navigation path in the environment using a pre-built map, which is stored in the network edge.

Localization

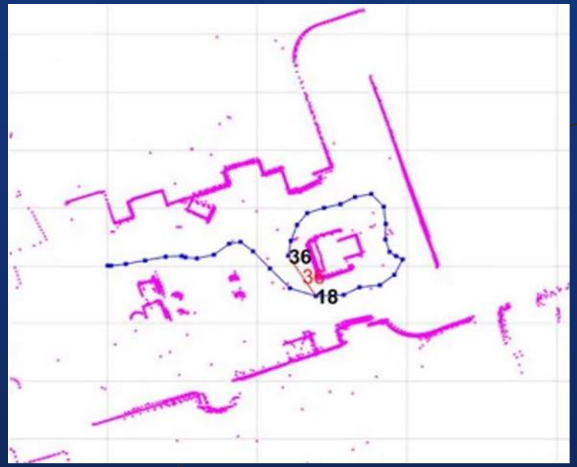
- Localize itself in the environment
- Pre-built map in the server

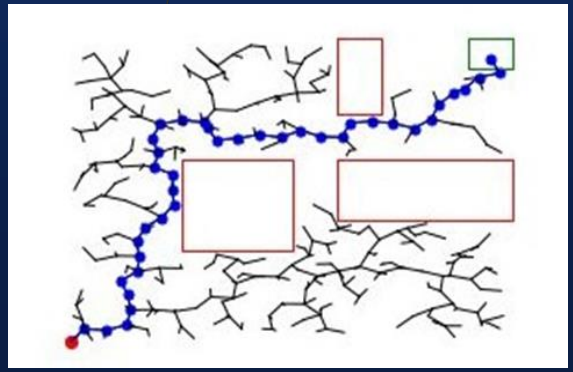
Mapping

- Two approaches
- A* algorithm –Jump point search
- Rapidly exploring Random Tree (RRT/RRT*)

Obstacle avoidance

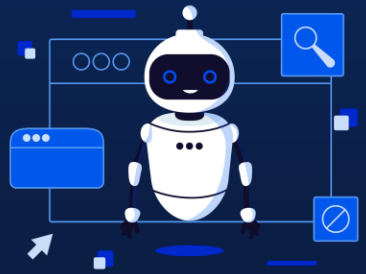
- ORB SLAM 2
- Update the map with the obstacles
- Find an alternative path





Objectives

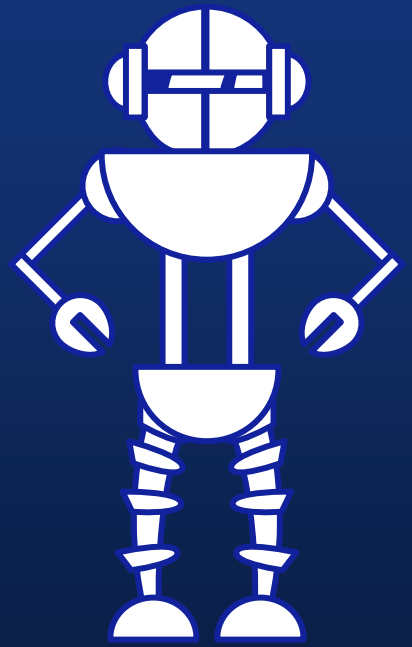
- Building an edge computing network of robots that's capable of performing relevant actions based on the commands given to it
- Ensuring the system recognizes voice commands precisely and the selected robot performs the necessary action accordingly
- Ensuring robots can localize and map his surroundings and perform autonomous navigation



REQUIRED TECHNOLOGY

Voice Recognition

(Python, DL Algorithms)



Microservice Architecture

(Kubernetes, docker& Spring boot)

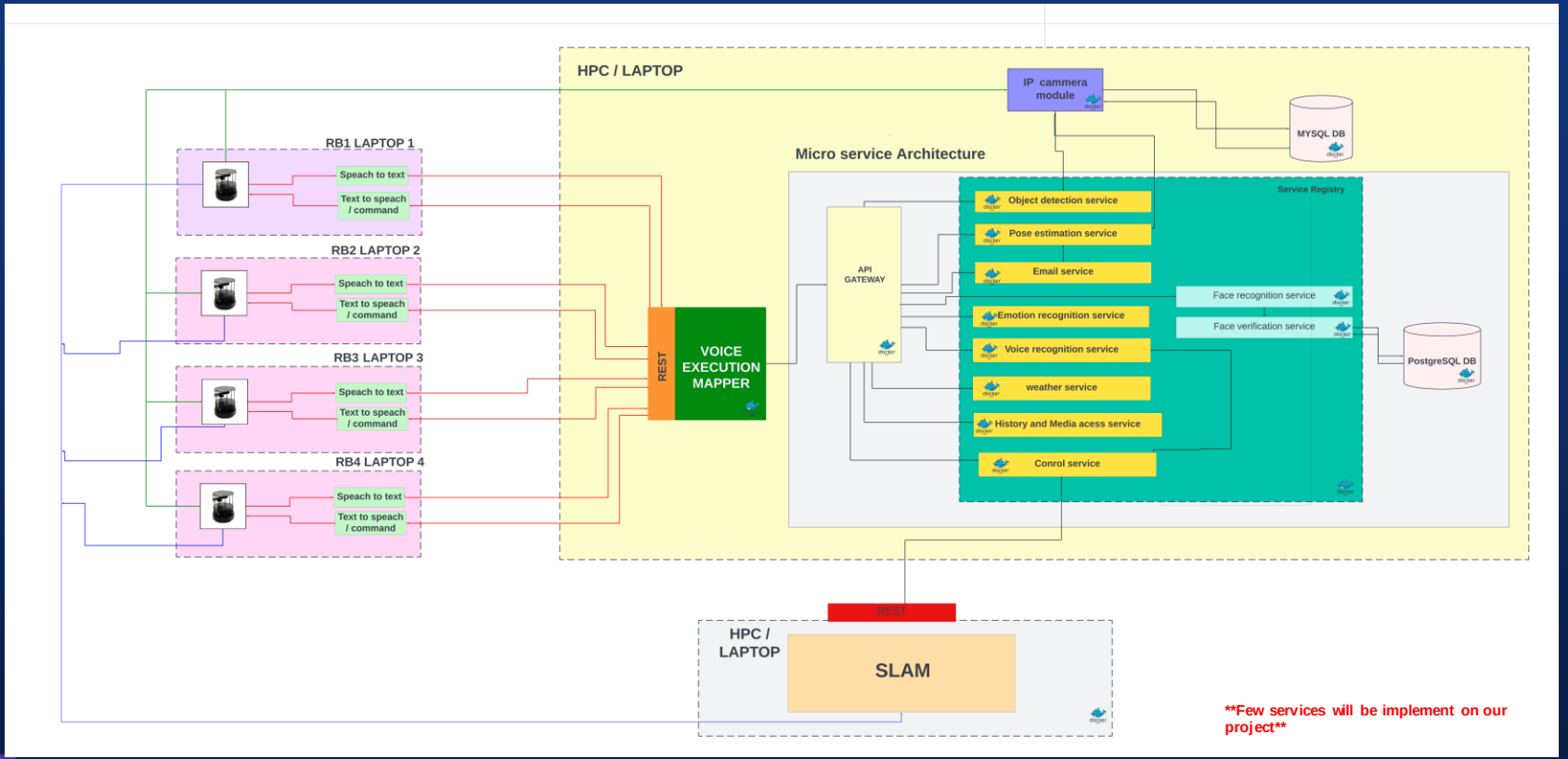
Pose Detection

(Python, MediaPipe Library)

SLAM & ROS

(ORB SLAM & ROS)

IMPLEMENTATION PLAN



Few services will be implement on our project

CHALLENGES

- API latency issues will occur while communicating between the client(Kobuki robots) and the server.
- SLAM Implementation for multipole robots on Edge computer would be a major challenge due to resource issues.
- The noises coming for the surrounding would make issue to grab the correct voice commands and redirect the robots request to correct service.
- Incapability of find a large dataset for train voice recognition model

Milestones

1. Background research and literature survey
2. Selecting suitable Edge SLAM, pose estimation, face recognition and voice recognition algorithms.
3. preparing the robots by installing ubuntu version 20.04 and ROS Noetic
4. Building the client application for the robot
5. Building the server side application using Microservices and REST APIs
6. Create a dummy ML algorithm and add as a service.

Milestones cntd.

7. Implementing voice recognition module

8. Implementing pose estimation modules

9. containerized all the services and add them to the back-end server

10. Implementing robots' navigation using SLAM

11. Add K8s to manage our docker containers

12. System Integration system testing and debugging

TIMELINE

		STAGE(WEEKS)																						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Designing	Designing the work flow digrams for robots actions																								
	Perpare the 2 Kobuki robots by installing ubuntu 20.04 with ROS (Noetic Ninjemys)																								
Implementation	Implement server side using Spring microservice architecture and test with dummy service																								
	Pose esitimation service Implementation																								
	Voice recognition service Implementation																								
	Voice execution mapper Implementation																								
	SLAM Implementation On edge																								
	Ros algorithm Implementation for robot movements																								
	Dockerize all the services																								
	Add Kubenertes for microservice Architecture																								
Testing	Testing and Debugging																								
Maintenance	Finalizing and enhance the perforamnce																								

REFFERENCES

[1]. “Wiki,” *ros.org*. [Online]. Available: http://wiki.ros.org/Documentation. [Accessed: 02-Nov-2022].

[2]. Supun Kamburugamuve School of Informatics and Computing, S. Kamburugamuve, S. of I. and Computing, Hengjing He State Key Lab of Power System, H. He, State Key Lab of Power System, Geoffrey Fox School of Informatics and Computing, G. Fox, David Crandall School of Informatics and Computing, D. Crandall, U. of B. 2, H. L. University, U. of W. Britanny, U. of Q. Roo, and O. M. V. A. Metrics, “Cloud-based parallel implementation of Slam for Mobile Robots: Proceedings of the International Conference on Internet of Things and cloud computing,” *ACM Other conferences*, 01-Mar-2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2896387.2896433. [Accessed: 02-Nov-2022].

[3]. A. Fortes, “Deep Learning based human pose estimation using opencv and MediaPipe,” *Medium*, 04-Jun-2021. [Online]. Available: https://medium.com/nerd-for-tech/deep-learning-based-human-pose-estimation-using-opencv-and-mediapipe-d0be7a834076. [Accessed: 01-Nov-2022].

[4]. “Production-grade container orchestration,” *Kubernetes*. [Online]. Available: https://kubernetes.io/. [Accessed: 01-Nov-2022].

[5]. “Spring makes java simple.,” *Spring*. [Online]. Available: https://spring.io/. [Accessed: 02-Nov-2022].

THANK YOU

ANY QUESTIONS?