## AJITH KUMAR JAYAMOORTHY

🌐 Personal Website | 🛅 LinkedIn | 💭 GitHub | 🤳 +1 774-253-0708 | 🔽 ajithkumarjayamoorthy@gmail.com

#### **EDUCATION**

#### Worcester Polytechnic Institute

Master of Science in Robotics Engineering (GPA - 3.88/4.00)

Indian Institute of Information Technology D&M, Kancheepuram

Bachelor of Technology in Mechanical Engineering (GPA - 8.65/10.00) **COURSEWORK & SKILLS** 

Coursework: Computer Vision, Deep Learning, Machine Learning, Motion Planning, Robot Dynamics Languages: Python, MATLAB, C++

Softwares: Linux, Gazebo, Rviz, Blender, AutoDesk Inventor, Unity, Fusion 360

Tools: Pytorch, ROS, Keras, Numpy, Scikit, OpenCV, PCL, Open3D, Scipy, Matplotlib, Pandas

#### **EXPERIENCE**

#### Deep Learning Intern-ADAS, Veoneer USA

Real-time Radar-based Obstacle Detection using Deep Learning

- Improved near-distance obstacle detection accuracy by 2%, using modified loss function.
- Assisted in evaluating radar tracker data using RMSE and track association model
- Researched alternate approaches for anchor points for detection boundaries based on limitations of radar data

#### Machine Learning Engineer, Veoneer India Private Ltd.

Interior Cabin Sensing for occupant detection & classification (NASSCOM ER&D Showcase 2021)

- Constructed FCNN training framework using Python, achieved 99.5% accuracy in MATLAB evaluations.
- Supported C++ implementation for real-time testing, accomplishing 97.5% accuracy in multi-occupant scenarios
- Performed feature Analysis and derived feature development
- Developed training material and served as a part-trainer for an in-office machine learning training program.

#### Machine Learning Intern, Veoneer India Private Ltd.

Interior Cabin Sensing for occupant detection & classification

- Extracted significant features from radar data and optimized Logistic Regression model training
- Verified model generalization, achieving 97.8% accuracy with minimal test data

#### PROJECTS

Obstacle Depth Estimation for micro-mobility platform using DL | Python, ROS, PyTorch Feb 2023 - May 2023

- Established an efficient data collection pipeline for the Gazebo-ROS Kinect camera model.
- Integrated and tuned YOLOv5 model for object detection & classification and achieved an accuracy of 97.2%.
- Estimated depth of all obstacles by mapping bounding box on the depth image.

#### Semantic Mapping of LiDAR Point Clouds | Python, PyTorch

- Semantically segmented the corresponding camera images of the point cloud using PSP net.
- Reprojected point cloud into segmented images for semantic mapping with camera extrinsic on the KITTI dataset.
- Utilized Open3D for constructing a LIDAR point cloud map using point-to-point ICP algorithm.

#### Stereo Camera & IMU Sensor fusion using EuRoC dataset | Python

- Deployed vision-aided inertial odometry for robust and efficient state estimation of a quadrotor flying a trajectory.
- Actualized a filter-based stereo camera method that uses the Multi-State Constraint Kalman Filter (MSCKF).

#### Image-based 3D Object Reconstruction using SfM and NeRf | Python, PyTorch

- Implemented end-end classical structure from motion algorithm using images of WPI's unity building.
- Programmed NeRF using Pytorch for Lego data through custom data loader, parser, network, and loss functions.

#### Auto-Calibration for Camera | Python, PyTorch

- Corner Detection using OpenCV, followed by estimation of Camera Intrinsic Matrix (K) and Extrinsic (R & t)
- Performed Non-linear Geometric Error Optimization using Scipy's optimize.

### Estimation of Homography using Deep Learning | Python, PyTorch

• Accomplished Homography estimation between two images through Supervised HomographyNet.

### **Panorama Stitching** | Python

- Executed corner detection with Adaptive Non-Maximal Suppression (ANMS) for balanced distribution.
- Augmented and matched features using descriptors and applied RANSAC for outlier rejection

## Collision avoidance for mobile robot in dynamic hospital environment | Python, ROS, Unity Feb 2022 - May 2022

- Implemented hierarchical motion planning: Global Planner RRT<sup>\*</sup>, Local Planner VO, RVO, and HRVO.
- Programmed Velocity Obstacle (VO) and Reciprocal Velocity Obstacle (RVO) methods for 4 robots in 2D simulation.

## Sep 2022 - Dec 2022

Aug 2022 - Dec 2022

# Lowell, MA, USA

May 2022 – Dec 2022

Aug. 2021 - May 2023

July 2015 - May 2019

Worcester, MA, USA

Chennai, TN, India

Oct 2020 - July 2021

Jun 2020 - Sep 2020

Bengaluru, India

Bengaluru, India