# Rohit Bhikule

+1(445)-208-7978 | bhikulerohit@gmail.com | LinkedIn | GitHub | Portfolio

## Education

- **University of Pennsylvania** — Philadelphia, PA
  *Master of Science in Mechanical Engineering and Applied Mechanics (Robotics); GPA: 3.76* — *Aug. 2021 – May. 2023*

## Experience

- **Terran Robotics (Bloomington, IN)** | *Robotics Software Engineer* — *June 2023 - Present*
  - Designed end-to-end neural network to hammer earthern clay into walls, increasing speed by 30% and 40% redundancy reduction. Designed task-specific prediction heads to improve end-to-end performance by using multimodal data.
  - Developed a GTSAM-based ROS localization pkg for cable-driven parallel robots using AprilTags and encoders, implemented a parallel processing pipeline using Ray library to achieve 10 Hz pose output with 5 mm accuracy.
  - Developed A* based obstacle avoidance planner to navigate across walls on a construction build site. Scripted gradient descent based optimization algorithm for extrinsics calibration.
  - Architected pipeline to convert stl CAD files of house walls to 2-dimensional goal depth maps and plan operations. Scripted python deployment scripts for managing services across kubernetes cluster for robot operations.
- **Skymul (Atlanta, GA)** | *Robotics Intern (Vision, Perception)* | *Video* — *May 2022 - Aug 2022*
  - Built a novel algorithm to detect rebar intersections and pose from noisy pointclouds real-time at 6 Hz in a densely multilayered rebar network on quadraped robot. Developed algorithm to increase confidence score of rebars in pointclouds obtained by photogrammetry.
- **F1/10th Autonomous Racing (mLAB UPenn, PA)** — *Jan 2023 - May 2023*
  - Acheived **1st** place in the 12th F1Tenth Autonomous Grand Prix at the CPS-IoT 2023 conference by implementing a finite state machine to switch between overtaking, adaptive cruise control and pure pursuit modes on an F1Tenth car.
  - Enhanced the accuracy of Local-INN model to predict pose at 40 Hz from 2D LiDAR scans with IMU fusion using UKF. Deployed the neural network using TensorRT on the F1Tenth car for real-time validation.

## Projects

- **Computer vision deep learning frameworks** | *Skills: Pytorch, OpenCV, CUDA, Feature Pyramid Network (FPN)*
  - **MaskRCNN** : Implemented a FPN based two-staged model to predict instance segmentation masks. *Github*
  - **SOLO** : Implemented FPN based model *(Segmenting objects by location)* to predict segmentation masks. *Github*
  - **YOLO** : Scripted YOLOv1 pipeline from scratch in Pytorch to predict bounding boxes. MAP achieved: 0.43. *Github*
- **Semantic mapping and Pointpainting** | *Skills: Pointcloud, Sensor fusion, Pytorch, SegFormer* *Github*
  Fused LiDAR and Camera data to paint pointcloud as per semantic information from images to classify each point using SegFormer NN (Transformers-based) on KITTI dataset. Obtained the BEV for the autonomous vehicle. *Video*
- **Localization and Estimation** | *Skills: Particle Filter, UKF, EKF, Visual-inertial odometry*
  - **SLAM** : Integrated the orientation and odometry information from IMU and 2D LIDAR scan to build occupancy grid map of environment by updating the log odds while simultaneously performing particle filter based localization. *Github*
  - **3D Orientation tracking** : Implemented a Quaternion based Unscented Kalman filter to track 3D orientation from IMU data and compared it against ground truth data obtained from motion capture system. *Github*
  - **Visual Inertial Odometry** : Implemented controller on quadrotor to follow path obtained by A*. Fused the IMU and stereo pair information to estimate 3D pose of a flying robot using two different algorithms - Error State Kalman Filter(ESKF) and Multistate Constraint Kalman Filter(MSCKF). *Github*
- **LLMs** | *Skills: Quantization, Pruning, Knowledge distillation*
  - **Llama2** : Scripted Llama2.7B in pytorch from scratch implementing KV cache, rotary positional encoding. Used pretrained weights for inference validation. *Github*
  - **Vanilla-transformer** : Implemented Attention is all you need from scratch and utilized Distributed Parallel Data training to train the model on multiple GPUs on AWS instance for a language translation task. *Github*
- **Controls and Planning** | *Skills: MPC, LQR, iLQR, Value iteration, Drake solver, Gazebo*
  - **MPC manipulator arm** : Developed and implemented a Model Predictive Control (MPC) system for a 7-DoF robotic manipulator, enabling real-time collision-free trajectory planning in obstacle cluttered environments. Designed forward and inverse kinematics solutions for dynamic block grasping using AprilTag detection. *Github*

## Skills

- **Languages**: C/C++, Python, Matlab, Arduino
- **Frameworks**: ROS, ROS2, Kubernetes, Docker, Git, Linux, Pytorch, Numpy, OpenCV, Open3D, Redis, MongoDB