Rishikesh Avinash Jadhav

+1 (240) 854-0789 | College Park, MD 20740
rishikeshjadhav712@gmail.com | linkedin.com/in/rishikesh-avinash-jadhav | github.com/Rishikesh-Jadhav | Portfolio | YouTube

EDUCATION

University of Maryland, A. James Clark School of Engineering — **College Park, MD**
Master of Engineering in Robotics — *May 2024*

- Relevant Coursework: Computer Vision, 3D Vision, Advanced Techniques in Visual Learning and Recognition

MIT - World Peace University — **Maharashtra, India**
Bachelor of Technology in Robotics and Automation — *Jun 2022*

SKILLS AND INTERESTS

Programming Languages: Python (Scikit learn, NumPy, Pandas, Matplotlib, SciPy), C++.
Frameworks, Platforms, and Tools: OpenCV, PyTorch, PyTorch3D, PCL, Keras, TensorFlow, HuggingFace, Gradio, ROS, ROS2, Gazebo, Moveit, Moveit2, SLAM, Docker, GCP, AWS, Ubuntu, Linux, Raspberry Pi, SolidWorks, Git, Jupyter, Google Colab.
Fundamentals of Artificial Intelligence | IIT-Guwahati | 12 weeks |**:** Certification link

EXPERIENCE

The Robotics Algorithms & Autonomous Systems (RAAS) Lab — **College Park, MD**
Graduate Research Assistant — *Jan 2023 – Dec 2023*

- Improved localization accuracy of a sensorless mobile robot by **13%** using machine learning techniques, Aruco markers, and a UAV to localize it in the Gazebo simulator to avoid collisions.
- Implemented intelligent multi-robot coordination strategies utilizing the Simultaneous Localization and Planning paper, and integrated computer vision methods to elevate localization accuracy, focusing on autonomy.

MIT-WPU — **Maharashtra, India**
Research Assistant - Machine Learning — *Feb 2022 – Jun 2022*

- Applied machine learning algorithms to drive an **11% efficiency** boost in tomato sorting pipelines. Collected a dataset of **9000** tomato images, and trained a TensorFlow model with **93% accuracy**. Visualized and monitored training metrics using TensorBoard, paving the way for a significant surge in sorting precision.
- **3D** printed a working prototype featuring a hopper, conveyor, and Raspberry Pi camera module, paired with motor activation, resulting in the streamlined classification of tomatoes into **4** distinct bins based on their **4** ripening stages.

Phi Robotics — **Maharashtra, India**
Robotics Intern — *Nov 2021 – Feb 2022*

- Collaborated within a **4** member team, refining ML models and creating visualizations using dashboards. Delivered the optimal solution for the assigned problem statement through rigorous fine-tuning and experimentation.
- Developed a reusable library for the LPC1768 microcontroller, incorporating UART, GPIO, Timer, PWM, ADC, and DAC functionalities and integrated it into a project.

PROJECTS

Agile Robotics For Industrial Automation Competition - *Team Lead* | *Python, Pytorch, C++, ROS2, OpenCV* — ***Link***

- Developed a robust control system automating kitting and assembly tasks, coordinating **2** UR10e floor and gantry robots, and **4** AGVs for the healthcare industry. Overcame **5** agility challenges including high priority, and insufficient orders, and performed quality checks at regular intervals replacing faulty parts.
- Deployed a wide sensor suite and achieved an **8%** reduction in competition time and **40%** in sensor cost, using a fusion of RGBD and logical camera on the conveyor belt and **8** bins.
- Integrated ResNet for part detection, and classification followed by a dynamic pick and place algorithm for tracking and moving parts in the arena.

Birds Eye View Trajectory Prediction for Autonomous Driving | *Python, Pytorch, Deep Learning* — ***Link***

- Enhanced instance prediction for autonomous driving by adapting the state-of-the-art PowerBEV framework, resulting in a **5%** increase in IOU, **0.967** for Woven, and **0.956** for the Argoverse dataset.
- Reduced vehicle ID switching instances by **10%**, yielding a **15%** gain in overall Vehicle Prediction Quality (VPQ) values, improving model stability in scenarios with varying vehicle counts.
- Optimized the vision module after Grad-CAM visualizations leading to improved processing time by proposing a ViT-based feature extractor in place of the EfficientNet network used by the baseline.

Adaptive Neural Network-based Navigation for Autonomous Driving | *Python, TensorFlow, Deep Learning* — ***Link***

- Improved autonomous vehicle control with imitation learning using Microsoft's Airsim. Conducted **3** reliability experiments using distinct Region of Interest variations on a custom MPC-collected dataset of over **3000** images.
- Implemented a resilient hybrid neural network architecture and data augmentation techniques resulting in a **20%** improvement in predicting steering, throttle, and brake pressure inputs by integrating both image and numerical data.

3D Indoor Mapping and Object Segmentation | *Python, Pytorch, 3D Reconstruction, Object Detection* — ***Link***

- Achieved a **17%** boost in indoor scene mapping accuracy by designing a novel pipeline using, SimpleRecon, and Point-Voxel CNN for object segmentation using point clouds with RGB images as inputs.
- Executed an effective ETL pipeline, reducing 3D scene reconstruction time by **7%,** and showcased model robustness by attaining **93%** accuracy on the Stanford 3D Indoor Scene Dataset.
- Generated intermediate 3D point clouds for fine-tuning the model, attaining a **12%** enhancement in point cloud quality.

Multi-Modal Image Generation using Prompts | *Python, Generative AI, Model Deployment* — ***Link***

- Orchestrated an image processing pipeline using Grounding-DINO for zero-shot object detection, followed by the Segment Anything Model for segmentation, and Stable Diffusion v2 after A/B testing with Contextual Attention GAN for inpainting.
- Enhanced PSNR and SSIM scores by **7%** via denoising and sharpening filters, resulting in refined inpainting by smoothing imperfections, and sharpening edges.
- Designed a Gradio UI to streamline user interaction with the multi-modal image processing model. Prepared the model and deployed it on HuggingFace Spaces.

Virtual X-Ray Vision App | *Python, Pytorch, Machine Learning, Model Deployment, and Monitoring* ***Link***

- Engineered a medical imaging solution using the UW-Madison GI Tract dataset. Conducted A/B testing of the Segformer model and a U-Net for ROI segmentation in CT and MRI scans, yielding a **10%** improvement in inference latency.
- Used WandbLogger for experiment tracking and monitoring training metrics. Improved user accessibility by integrating a Gradio UI, followed best practices for file packaging, and deployed the model on HuggingFace Spaces.

Heart Disease Classifier | *Python, Scikit learn, Data Cleaning, Data visualization, Feature Engineering* ***Link***

- Built an end-to-end ML pipeline for classification using the Cleveland database. Conducted EDA to reveal data distributions, identify missing values, and find anomalies.
- Achieved accuracies of **88.52%** with Logistic Regression, **90.04%** with XGBoost, and **83.40%** with Random Forest models, after hyperparameter tuning using randomized and grid search.
- Evaluated model performances by visualizing ROC curves, AUC scores, confusion matrices, precision, recall, and F1 scores. Visualized feature importance using PDP plots to identify key features influencing classification decisions.

Modeling and simulation of a mobile manipulator | *Python, Solidworks, ROS, Gazebo, Rviz, Moveit* ***Link***

- Designed a **9**-degree-of-freedom mobile manipulator robot using SolidWorks and imported the URDF into Gazebo for exploration and visualization using Rviz of object manipulation, and pick-and-place tasks.
- Validated forward and inverse kinematics, incorporating a differential drive mobile base mounted with a robotic arm to clean large-sized objects and implemented a closed-loop control mechanism, for navigation.

Vehicle Velocity Prediction using Optical Flow - *Team Lead* | *Python, PyTorch, OpenCV, YOLO, Raspberry Pi, IMU* ***Link***

- Integrated optical flow techniques and object detection, analyzing **2** classic computer vision methods including Lucas-Kanade and Farneback, alongside deep learning algorithms such as RAFT and FlowNet for velocity estimation.
- Combined RAFT with YOLOv3 for simultaneous vehicle detection and velocity estimation. Successfully deployed this model on a Raspberry Pi 4-based robot rover for autonomous driving application.

Enhanced Path Planning with RRT, RRT*, RRT*smart, RRT-connect | *Python, ROS, Gazebo* ***Link***

- Implemented **3** variations of the RRT path planning algorithm, including RRT*, RRT* smart, and RRT-connect, to plan routes from a user-defined start to goal avoiding obstacles.
- Used the navigation stack for visualizing in Rviz and evaluated the algorithms, based on criteria such as cost path, computation time, and tree density. Finally, executed the algorithms on Turtlebot3 within a 3D simulated maze in Gazebo.