Nirshal Chandra Sekar

☎ (763) 485-6687 ✉ chand861@umn.edu
nirshalchandrasekar.github.io
/NirshalChandraSekar in /in/nirshal-chandra-sekar

Education

University of Minnesota (UMN), Twin Cities, (MS in Robotics, 4.0 GPA) Sep 2023 - Present

- Computer Vision, Machine Learning, Deep Learning, Natural Language Processing

Vellore Institute of Technology, Vellore, (B.Tech Mechanical Engineering, 3.9 GPA) Jun 2019 - May 2023

Skills

Programming Languages: Python, C/C++, JavaScript, MATLAB

Libraries: OpenCV, PyTorch, Omniverse Replicator, Open3D, PyBullet, NumPy, scikit-learn

Tools: Git/Github, Linux, Docker, ROS/ROS2, Gazebo, NVIDIA Isaac Sim, Blender, SolidWorks

Work Experience

Robotics: Perception and Manipulation Lab, Graduate Research Assistant UMN — Jan 2024 - Present

- Engineered a **Real-Time Segmentation-Guided Grasping Pipeline** using **SAM**, **Contact GraspNet**, and **RealSense L515** lidar camera.
- Built a **Human Demos Guided Novel Object-Part Grasping Network** to grasp novel objects from a single video demonstration.
- Designed a robust sensor fusion framework to enable precise object localization and **real-time grasp planning** in dynamic environments.
- Leveraged foundational segmentation models to reduce the need for extensive training data, improving system efficiency and scalability.
- Conducted extensive testing and validation, ensuring reliable performance in real-world robotic manipulation tasks.

Nilfisk, Software Engineering Intern Brooklyn Park, MN — May 2024 - Dec 2024

- Developed a **custom annotation tool** using Segment Anything Model to enhance YOLOv8 object detection and segmentation pipelines.
- Streamlined annotation workflows to improve labeling efficiency and data preparation accuracy for ML models.
- Utilized **NVIDIA Isaac Sim's Replicator library** to generate **synthetic datasets** for detecting scraps on the factory floor
- Achieved a robust sim-to-real transfer and a **mAP of 92%** when validated on real-world data.

Technical Projects

Bi-Manual Manipulation using Diffusion Policy Github — ROS, PyTorch, OpenCV

- Designed and executed a **vision-based (CNN) diffusion policy** for bottle uncorking using PyTorch and dual UR5e arms.
- Trained on 188 teleoperated demos via RealSense L515 LiDAR and D405 stereo cameras, using ROS for data collection and action execution.
- Achieved a **74.7% task completion rate** across 30 rollouts, showcasing effective deployment of learned policies.

Image Generation using Diffusion Model Github — PyTorch, NumPy

- Coded an unconditioned **Denoising Diffusion Probabilistic Model** from scratch using a **UNet architecture** for noise prediction.
- Implemented the complete **forward and reverse diffusion processes** and trained on the **Stanford Cars dataset**.
- Achieved a **FID score of 34.7** between real and generated images.

3D Semantic Reconstruction Paper — OpenCV, COLMAP, YOLOv8

- Collaboratively performed **3D semantic reconstruction** using **Structure from Motion (SfM)** and **Multi-View Stereo (MVS)** with **COLMAP**.
- Conducted **2D semantic segmentation** with **YOLOv8** and linked 2D points to 3D points via a voting process.
- Generated a fully labeled **3D triangle mesh model** with **76% semantic labeling accuracy** across the reconstructed surface.

Object Detection using Convolutional Neural Networks — NumPy, OpenCV

- Implemented a **feed-forward CNN** for image classification and **Faster R-CNN** for object detection from scratch in **CSCI 5980**.
- Trained models on the **PROPS Detection Dataset**, achieving **30% mAP** and insights into neural network-based object detection.

Autonomous Navigation and Manipulation for Mobile Robots Github — Mobile Manipulation

- Led implementation of **Rapidly-exploring Random Trees (RRT)** for real-time motion planning, achieving **93% success rate** across 100 rollouts.
- Generated collision-free trajectories for picking, placement, and hand-off tasks while adhering to kinematic constraints.
- Utilized **Kineval Stencil** for simulation, testing, and validation of robot motion planning.