Yogesh Gajjar

ygajjar@usc.edu (213) 399-0913 yogeshgajjar123.github.io yogesh-gajjar yogeshgajjar

Ford sponsor winner at UC Berkeley Cal-Hacks Hackathon and Graduate Researcher developing USC's first autonomous F1/10th robot prototype.

Education

Master of Science in Electrical and Computer Engineering **December 2020**
University of Southern California GPA: 3.57/4.0
Coursework: *Machine Learning, Intro to Self-Driving, Pattern Recognition, Computer Vision, Image Processing, Robotics*

Bachelor of Technology in Instrumentation and Control Engineering **May 2015**
Institute of Technology, Nirma University GPA: 8.02/10.0

Technical Skills

Programming	C, C++, Python, Matlab, HTML, CSS
Framework	OpenCV, Scikit Learn, Keras, TensorFlow, PyTorch, Flask, Redis, PCL, NLTK, Gazebo, Rviz
Architectures	YOLOv3/v4, RetinaNet, EfficientDet/Net, Mask-RCNN, DeepLab, ResNet-50, Deep Sort, VGG-16, HoG
Other	Git, ROS, Latex, Linus/UNIX, GCP, Raspberry Pi, Nvidia Jetson SoC's

Experience

Graduate Researcher (USC CPS-VIDA Group) **August 2019-Present**
- Lead the development, build, and bring-up execution of USC's first AV delivery vehicle prototype from start to finish. Software stack includes object detection, visual odometry, localization, planning, and controls.
- Spearhead research on developing Signal Temporal (STL) monitors, and vision-based Timed Quality Temporal (TQTL) monitors for ROS to track, and quantify perception robustness.
- Integrate ROS object tracking, lane-line detection, and semantic segmentation architectures with AV software stack.

Computer Vision Intern (Frenzy Labs, Inc) **May 2020-August 2020**
- Proposed and developed a network architecture by integrating state-of-the-art R-CNN object detection and H-CNN classification network that improved apparel classification/detection performance by 5%.
- Devised an end-to-end testing pipeline with RESTful request dispatching using Flask framework to accelerate model evaluation and deployment with reproducibility and traceability.
- Optimized state-of-the-art backbone CNN and R-CNN network with an increased 2% accuracy.

Projects

F-1/10th Self-Driving Car **C++, ROS, Jetson Xavier**
- Built a 1/10th scale autonomous delivery vehicle prototype for outdoor environment with Perception (Yolov3), Localization (Cartographer), and Path Planning (ROS Navigation) software stack.
- Tailored to perform online STL monitoring and TQTL based perception prediction to ensure safe autonomy.

Yolov3 Road 2D Objects Detection **C++, Python, ROS, Darknet**
- Trained independent Yolov3 object detection algorithm to detect road objects, traffic lights, and traffic signs on Berkeley DeepDrive 100k images and Bosch Traffic Light dataset to achieve a loss of 3.4%.
- Deployed trained weights on F1/10th vehicle for a real-time road object detection on public roads.

Localizing and Navigating Robot **C++, Python, ROS**
- Established ROS master-slave network and localized Turtlebot3 with GMapping (Rao-Blackwellized particle filter SLAM) to generate a 2D occupancy grid map of room with LiDAR.
- Utilized 2-D point cloud map to navigate TurtleBot3 to a real-time goal location in the room.

Intelligent Drivers Enhanced Assistance System **Python, OpenCV, Google Cloud API**
- Designed and tested a platform to detect drowsiness and emotions of driver with added rest-spot and mood-based song detection feature using OpenCV and Google Cloud API's; Won Ford sponsor award at Cal-Hacks Hackathon.
- Queried to/by (JSON requests) Ford's infotainment SDK to initiate voice assistance to alert drivers.

Distracted Driver Detection **Python, OpenCV**
- Predicted state of driver from 45,000 images falling under 10 classes with a vanilla CNN architecture and pre-trained ResNet-50 architecture. Achieved a robust 98% accuracy.

Image Processing Algorithms **C++, Object-Oriented**
- Developed image processing algorithms using C++ that includes edge detection, half-toning, geometric image modification, texture classification and segmentation.

Kalman Filters **Python, State Estimation**
- Coded a novel Kalman filter to estimate the position of an object of unknown dynamics with and without noise constraints.

Autonomous Lane-Keeping Robot **Python, OpenCV, Controls**
- Demonstrated working of a lane-keeping robot using image processing techniques and PID controller to drive motors.