YOGESH GAJJAR

ygajjar@usc.edu | 213-399-0913 | linkedin.com/in/yogesh-gajjar/ | yogeshgajjar123.github.io/

EDUCATION

Masters of Science, Electrical Engineering — *Jan 2019 - Dec 2020*
University of Southern California — **GPA: 3.57**
Bachelors of Technology, Instrumentation and Control Engineering — *Jul 2011 - Jun 2015*
Institute of Technology, Nirma University — **GPA: 8.03**

SKILLS

Programming: C, C++, Python, Matlab, HTML, CSS
Framework: OpenCV, Scikit Learn, Keras, TensorFlow, PyTorch, Flask, Redis, PCL, NLTK
Architectures: YOLOv3/v4, RetinaNet, EfficientDet/Net, Mask-RCNN, DeepLab, ResNet-50, VGG-16, HoG
Coursework: Machine Learning, Deep Learning, Computer Vision, Robotics, Pattern Recognition, Image Processing
Others: Git, ROS, Bash, Latex, Arduino, Raspberry Pi, Nvidia Jetson SoC

EXPERIENCE

Computer Vision Intern, Frenzy Labs, Inc — *May 2020 - Aug 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.

Graduate Researcher, USC Cyber Physical Systems-VIDA Group — *Aug 2019 - Dec 2020*
- Lead the development, build, and bring-up execution of USC's first delivery AV prototype from start to finish. Software stack includes object detection, fusion, visual odometry, localization and mapping, 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.

PROJECTS

F-1/10th Self-Driving Car
- 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.

Road 2D Objects Detection
- 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
- 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 a room.

Intelligent Drivers Enhanced Assistance System
- 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 APIs; Won Ford sponsor award at CalHacks Hackathon.
- Queried to/by (JSON requests) Ford's infotainment SDK to initiate voice assistance to alert drivers.

Distracted Driver Detection
- 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
- Developed image processing algorithms using C++ that includes edge detection, half-toning, geometric image modification, and texture classification and segmentation.

Kalman Filter
- Coded a novel Kalman filter to estimate position of an object of unknown dynamics with and without noise constraints.

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

LEADERSHIP

- Secured Ford sponsor award out of top 300 teams at UC Berkeley CalHacks 6.0 Hackathon 2019 for building a platform called IDEAS (Intelligent Drivers Enhanced Assistance System) in 36 hours.