

Perception Module: Human Object Detection for Acme Robotics

OVERVIEW

- **Perception Module for Acme Robotics:** Building a Human Obstacle Detector.

Perception is a crucial component of a Mobile robot as it helps the robot sense it's environment and provide base for other modules to make a decision . Accuracy of perception module dictates the decision making of the robot. Avoiding Pedestrians obstacles is of highest priority and having a accurate perception module hence becomes crucial.

- Acme Robotics' Robotic System would benefit by having a accurate perception module that would give the 3d locations of the human obstacle in robot frame and hence this information can be acted upon directly by the navigation stack.

METHODOLOGY

- The detection tracking and depth estimation is done by applying deep learning models on video feed from a monocular camera. Deep Learning is common tool used nowadays for object detection tracking and depth estimation as well.

- The detection and tracking is done by existing Deep Sort method published in the recent years. The depth estimation is done through MonoDepth Estimation method, also published in the recent years.

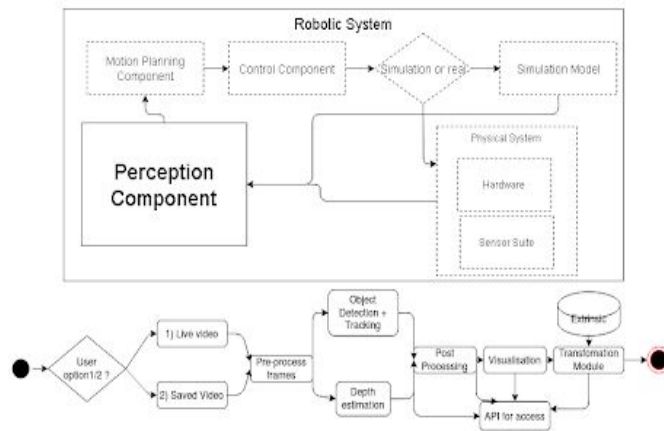
- Risks involved are associated with the accuracy of Deep learning based methods. Reaching 100% accuracy is not realistic and this is very true in case of Deep Learning based methods as well. The risks can be mitigated by extensive corner case testing to know and be prepared for our model's limitations

IMPLEMENTATION

Design Process: UML Class and activity diagrams. Agile Iterative Process and TDD

Tools: Make, Make, Cppcheck, Cpplint, CTest, Doxygen, Travis, Coveralls, OpenCV, Gtest, Gmock, Git, C++, Eclipse/VScode, Agile Iterative process, Pair Programming, Pytorch, TensorFlow

Testing: Unit testing of all modules mentioned in the activity diagram below the system diagram



RESULTS

Timeline

Oct 6th 2020 - Finish Proposal

Oct 9th 2020 - Finish UML Diagram,
Git repository with travis and coveralls,
Activity Diagram,
Class declarations and stub implementation

Oct 12th 2020 - Finish Unit tests, Update Readme, Doxygen
commenting Have a feature requirements, Prepare for transfer to
Phase 2 Driver.

Oct 14th 2020 - Implementation of classes.

Oct 17th 2020 - Pass all test cases and submit pull request.

Oct 19th 2020 - Create Video.

Oct 20th 2020 - Buffer day

Expected Results

- 1) 100% Travis Build passing
- 2) 90%+ Coverage.
- 3) Achive 90% author's claimed accuracies for models.
- 4) APIs mentioned in the API block in the activity diagram