

Rajan Vivek

Tensorflow_ros_train: Accelerating ROS-based Model Development and Lowering the Barrier to Entry for Newcomers

After a year of diving into autonomous navigation research, I've realized that this world can be pretty unfriendly to newcomers. The learning curve is steep. It's very easy to get bogged down in the details of complex ROS infrastructure and data synchronization and waste a lot of time just debugging data collection instead of learning from it. Also, while machine learning as a whole is quickly becoming democratized, the barrier to entry for newcomers in autonomous navigation seems disproportionately higher.

In my personal exploration of online autonomous navigation tutorials, I've realized that they often simply demonstrate the suite of implemented motion planners and pretrained models without providing much flexibility for the user to change them. Also, while much real-world data is available online for learning various machine learning techniques (such as labeled handwriting samples and housing market prices), the available data for sensor-based navigation is much more limited.

I aimed to create seamless integration of Tensorflow and ROS to help newcomers and experienced researchers alike take full advantage of what both have to offer. My project promotes:

Human-Centered Design: The wrapper classes and methods are designed to be as intuitive and convenient as possible. The open-source nature and lower learning curve will promote inclusivity in the field, as users can easily gather their own data, make modifications to the codebase, and benefit from community-generated data repositories online.

Robust Model Development: Easier data collection means more data and more rapid prototyping and testing, leading to greater robustness. The lower barrier to entry means more diverse minds being put to work.

Multiple Metrics: ROS provides incredible customizability in simulated sensor design and robot geometries. This varied data means versatile models can be designed and pitted against one another.

My future plans for this project include optimizing it for the soon-to-be-released ROS Noetic, which will solve the Python compatibility issues, releasing it to the open-source community for feedback and further development, and creating a tutorial based on the package to allow newcomers to quickly learn both ROS and Tensorflow and start making their own impact.