Project Proposal: Human Detection and Tracker

Aneesh Chodisetty
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: aneeshch@umd.edu

Bhargav Kumar Soothram
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: bsoothra@umd.edu

Joseph Katakam
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: jkatak73@umd.edu

OVERVIEW:

Object detection and tracking is a computer vision task that has become essential for many robot applications. Object detection and tracking are the base for autonomous robots in specific. Suppose there is an autonomous delivery robot on a mission, it has to identify objects, track them, convert their coordinates to the robot's frame of reference and then try to avoid them. This project is focused on solving this problem using existing state-of-art object detection algorithms.

As per Acme Robotics' requirement, we will use the input from a monocular camera to detect and track an object. This object is then converted to the robot's frame of reference. Acme will interface this package in their robotics-based product that will be launched next year.

METHODOLOGY:

Our system uses the YOLOv5 model that is trained using the COCO dataset and is built using C++. COCO dataset is large-scale object detection, segmentation, and captioning dataset. Our system takes an image from a monocular camera, pre-processes the image, passes this image to the trained model, filters out the human object with the highest confidence, and then outputs the location of the object in the robot's frame of reference.

The key advantage of using YOLO is that it processes frames at the rate of 45 fps (larger network) to 150 fps (smaller network) which is better than real-time. Also, the network is able to generalize the image better. Hence better detection of trained images.

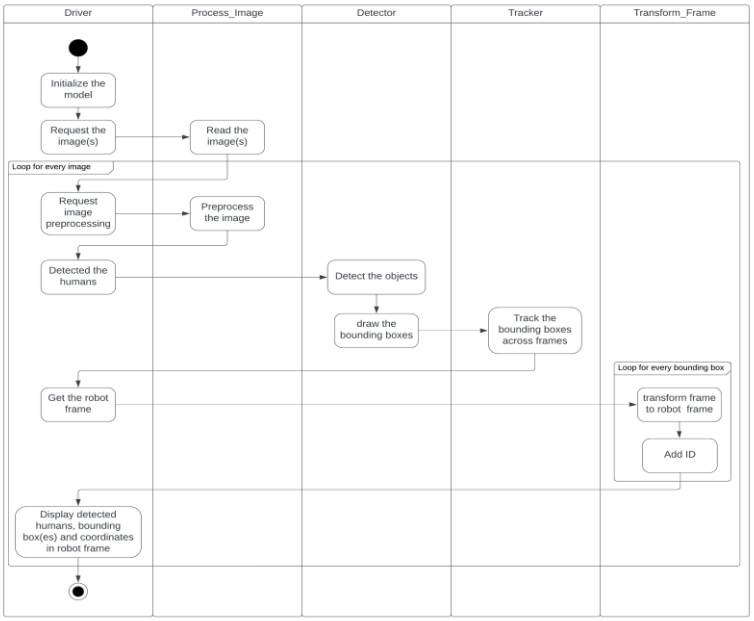

Fig. 1. Flow of our proposed system

POTENTIAL RISKS AND MITIGATION:

- YOLO is a memory expensive model. Hence, before this module is interfaced, the running system on robot should meet the necessary requirements. Seeing that the prices of RAM have been coming down over the years, the solution can be a simple inexpensive purchase. Another alternative could be to take the weights of the model and quantize them so that less memory is consumed, and it remains fast.
- COCO Dataset has been used to train the model. This dataset consists of daytime images. Hence, the model doesn't work for nighttime or infrared images. If the client has such requirements, then this model can be extended to train with the required dataset.
- YOLO's accuracy is low when it comes to detecting smaller objects. Therefore, the application of this module has to factor this in, before deploying it in real-time.

DESIGN AND DEVELOPMENT:

Agile software development model will be used for the development process where tasks will be tracked using a backlog table. The software is designed in a Test-Driven Development fashion and implemented using Pair programming technique. The tasks will be outlined for every sprint and after each sprint, the roles of the pair-programming group will be interchanged.

DELIVERABLES:

1. Proposal Documentation
2. UML Diagrams
3. Project Package with demonstrated OOPs concepts
4. Cl using Travis
5. Code Coverage using Coveralls
6. Unit Tests using Google Test Framework
7. Developer Level Documentation
8. Static code analysis with cppcheck
9. Google C++ Style guide with cpplint validation

TECHNICAL PROCESS:

'C++ 14' Language will be used to develop the project using 'Visual Studio Code' environment. To perform image related operation, the open-source library 'OpenCV (4.6.0) is used, and the entire project is built using CMAKE (3.16.3).

Before the project is released, the module will go through an extensive check for bugs, coding style, documentation and working version. Unit tests will be created to check properly functionality of code and complete code coverage will be achieved. Documentation will be done using Doxygen and maintained for each component, with proper code commenting