Human Detection and Tracking

Sahruday Patti
James A. Clark School of Engineering
University of Maryland, College Park
sahruday@umd.edu

Vishaal Kanna Sivakumar
James A. Clark School of Engineering
University of Maryland, College Park
vishaal@umd.edu

***Abstract*—Perception is an integral part of any autonomous system. Knowing the surrounding environment enables the system to take suitable actions and navigate to the desired goal location and also augments the control system of the robot to perform a variety of tasks. We want to design one important part of the control system, within the perception stack of acme robotics which is Human detection, and tracking and providing coordinates of humans present in the scene with respect to the Robot's Reference frame. In this Proposal document, we explain our approach to how we plan to implement this module of the control system and also outline the workflow, and software engineering practices we intend to follow during the development process.**

***Index Terms*—Human detection, Tracking, Yolov3-tiny, autonomous robot**

I. INTRODUCTION

Acme Robotics needs a Human Obstacle Detector for their new robotic product. Being able to detect humans, the robot can navigate to or away from humans autonomously which can then perform a variety of tasks in a wide variety of environments. In a warehouse, the robot may want to avoid humans and reach its destination to perform pick and place operations, In an autonomous car, it may help make decision-making quicker and improve the safety limits of the car. In search and rescue operations robots may navigate toward humans. Human detection in real-time provides several capabilities and certainly augments the robot's capabilities to perform tasks in any environment. Here, we document our approach to developing this module, our method to attain this capability, and the process we follow to develop this module.

II. IMPLEMENTATION

There are three modules in our system. We explain the importance and how we plan to implement each module in the following sections.

A. HUMAN DETECTION

We plan to use YOLO Network Pretrained on COCO Dataset to Detect Humans. The backbone consists of a Convolution Neural Network, which is followed by the prediction head with linear layers. The network takes images of specific dimensions as the input and predicts bounding boxes and the probability that the boxes contain a human. The bounding boxes produced by the network are redundant and have to be suppressed based on their intersection over union scores with other boxes. We use Non-maximal Suppression to achieve this and retain the desirable bounding boxes over humans present in the frame.

B. TRACKING

Tracking is a problem that can be dealt with differently depending on the task and amount of resources we have within the framework of our perception capabilities. We assume to have a monocular camera system. We Plan to implement a tracker, which tracks a person and assigns a specific ID to a specific person by the use a Least square fit over past observations and extend it to the Kalman filter to track humans. We plan to use Open CV Library's implementation of an object tracker and verify the quality of tracking and deliver a product that provides the best tracking in order to ensure the quality of our module.

C. TRANSFORMATION

Having obtained the bounding boxes from the detection module and the association Ids from the tracking module, we transform the 2D image coordinates to the 3D coordinates with respect to the camera frame. We select the average human height of a human and enforce an additional constraint over the transform equations obtained using the camera intrinsics to estimate the depth of the detected person. We can now transform the coordinates from the camera frame to the robot reference frame which then can be utilized by the robot control system to navigate/perform a variety of tasks depending on the application.

III. POTENTIAL RISKS AND MITIGATION

The Deep Learning models we plan to use are trained for object detection in general and we have to train again to detect humans with a data set. This might push our deadlines. To mitigate this, we plan to accommodate our initial time developing this module as this is the foremost important subsystem of this module. The Other potential problem is the tracking module. We plan to Implement a Kalman filter based tracker and this in itself can be a submodule as this has so many moving parts. One such moving part is being able to detect what new bounding boxes were added or present from the previous frame. For this, we plan to implement the Hungarian algorithm and this might push our time. We plan to mitigate this by using the OpenCV::EKF function if we push our deadlines. With the usage of a deep learning model, we potentially might push the robot's hardware limits.

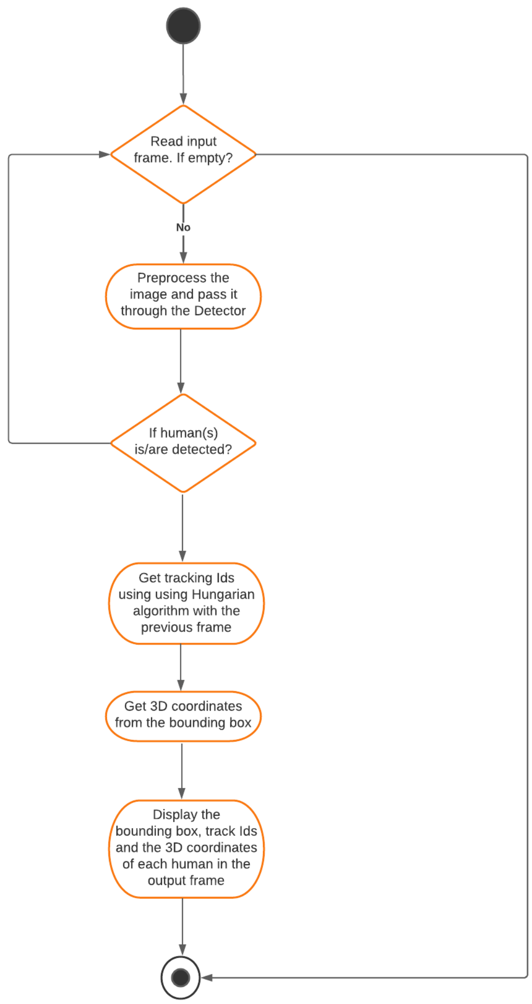

Fig. 1: UML Activity diagram of our proposed approach

IV. Software Process

The Software development Process involves several stages of development, modifications, and identification of changes and improvements. we believe that the processes we follow will determine the quality of the product we deliver. This section outlines our process.

A. Organization Structure

Software evolution is often characterized by iterative changes and for this reason, we plan to adopt the Agile Iterative Process as our team members are equally capable which enables us to break down the broad task of a module into smaller and smaller tasks and take feedback and evaluate the performance of the iteration and provide us insights to improve our developments process throughout the SDLC by the use of Product Backlog, Iteration Backlog, Daily meetings and work log.

B. Project Workflow

We are a team of two and we will follow a Test Driven Development Approach. This approach involves creating a design and implementing code stubs for unit tests in the initial plan and iteratively increasing our unit tests as our development of the module progresses. At each stage, we implement our code stubs to pass these tests. This test-driven development approach aids the agile iterative process we adopt and ensures the productivity, progress, and high quality of the software system we intend to provide.

V. Technical Dependencies

We develop our software in the Ubuntu 20.04 OS with the use of OpenCV and Eigen Libraries. We host our development process on Github, which helps in Version Control and aids the test-driven development approach we adopt. We use Google test for our Testing Framework.

VI. Technical Documentation

we use Google C++ Style-guide with cpplint validation and perform static code analysis with cppcheck. we adopt Doxygen style documentation of our code in order to provide Documentation for our source code.

VII. Project Deliverables

This module will take the input image mounted from a monocular camera of the robot and deliver the coordinates of the humans present in the frame with respect to the robot reference frame in real-time.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[2] Mills-Tettey, G. Ayorkor et al. “The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs.” (2007).