Neato Tag!

Anil Patel, Katya Soltan

- In 4-5 sentences, what is the big idea of your project?
    - The project is to create a person locating algorithm that enables a Neato to play tag with another person. There would be two main robot modes. In the first mode, the robot is identifying a person and attempting to “tag” them with the bump sensor. The mode is switched by the bump sensor, and in the second mode, the robot is identifying the location of the person that is “it” and avoiding them.
- Have you found any papers or blog posts that have done something similar to what you are proposing?
    - To play tag, the robot needs to detect a target to follow or run away from, which uses object detection. OpenCV has support for this, and there are lots of posts about out-of-the-box person detection packages. However, is this something we should use, or should we go down another path (see questions)?
    - If we were to implement a very robust tag game, with multiple people coming in and out of the frame, we would also need to track our current object of interest. We could use GOTURN for example.
- In terms of the system that you will have running on the robot, what is your MVP? What is your stretch goal?
    - The fastest thing we can get working is tracking a person in space with no robot controls using OpenCV tools.
    - The MVP for this project is playing 1 v 1 tag between one Neato and one human (no extra humans or obstacles popping in and out of frames). The robot should be able to identify a person

    - A stretch goal would be handling multiple people, tracking which people are “it” vs. pedestrians, and robust object avoidance from the lidar in conjunction with vision-based person tracking for final robot controls.
    - Additionally, we can train the robot to use the object tracking algorithm to determine the person’s position in space, and control how fast/if the robot moves based on that information.
- Describe your learning orientation (top-down versus bottom-up) and why you have chosen it. In particular if you choose bottom-up, make sure you specify what this will mean (e.g., which algorithms will you implement, will you eventually switch to a standard toolkit, etc.).
    - We are both more interested in getting a cool project to work using existing tools. Additionally, because we are both ML noobs, it makes sense to use a top-down approach to get introduced to all of the concepts and learn how to frame ML problems before diving into the weeds of the actual algorithm.
- What is your data collection plan? How do you plan to get the data needed for your project? How much data do you think you'll need? Are there existing datasets you can leverage?
    - See questions below. In terms of actual data, there seem to be a lot of pre-trained packages that can identify objects, including humans. However, how much learning will we get if we just use an out-of-the-box solution versus collecting data ourselves and trying to process it?
- What sorts of learning algorithms will you apply? You could choose these based on what you think will work the best or what you want to learn about the most.
    - The GOTURN algorithm makes the most sense at first glance, as it is well-documented, and known for being effective in object tracking. For the initial generation of bounding boxes, we will need to either use an

available person-detection tool, or potentially explore some deep learning methods for identifying bounding boxes for objects.

- What non-learning baseline algorithm will you compare to?
    - We would compare the learning-based algorithm to a solution from the warmup project, where we created a person-following algorithm using the lidar and its scan data.

Questions:

1) How much training data is “enough” and how can we easily classify/label training data if that number is significantly large? Do we need to draw bounding boxes manually for 6,000 images?
2) How representative of the background/environment that the robot will be operating in to “play tag” does the training data need to be? For example, can learning from training pictures taken in the classroom be applied to other settings?
3) Does it make sense to collect a bunch of images while the robot is driving around and then feed it through an available person detection algorithm? Or, should we be trying to create bounding boxes of what we want the robot to see in each image? Or use the lidar data and approximate where a person/object probably is and use it to figure out where the person in the image is?
4) How worried should we be about tracking a person from far to very close, where most of their features will not appear in the frame anymore?
5) An option we thought about but were unsure of was using the lidar person detection from the warmup project to automatically label images with an X,Y position of the person. This would facilitate data collection but make the project about correlating the location with the image not finding the bounding box of the person in the frame.