DeepSORT with OAK-D for Collaborative Robots

Ryan Barry
RIT Electrical Engineering Department
RIT
Rochester, USA
rpb3646@rit.edu

***Abstract*—Advances in computer vision technology have increased the capability of object detection and tracking for robotic perception. The integration computer vision technology with industrial or collaborative robots allows them to become reactive and accomplish a wider variety of tasks than simple puck and place routines. In this project, a Luxonis OAK-D spatial AI camera was used in conjunction with a Baxter collaborative robot to play a ball and cup game with a person. This was accomplished through a custom application of DeepSORT based on a YOLOv6 framework.**

I. GOAL

This project merged computer vision with Rethink Robotics' collaborative robot Baxter to create a ball and cup game between Baxter and a human player. The system used a custom implementation of Luxonis' real time Deep SORT adaptation, derived from the official Simple Online and Realtime Tracking with a Deep Association Metric repo [5]. Their adaptation is intended for real time use with the OpenCV AI Kit: OAK-D spatial AI camera, where Deep SORT is used as a tracking method on the YOLOv6 detection framework, and stereo cameras are used to detect an object's depth and return spatial cartesian coordinates in meters. In this project, a filtering algorithm was developed to track and store the locations of a predefined number of cups, or in the case of the game, 6 cups, for use in a ball and cup game between Baxter and a human. The camera was paired with ROS based Baxter to stack 6 cups in a pyramid shape with the hidden ball cup as the top cup in the tower.

II. RELATED WORKS

A. Object Tracking Overview

Object detection and tracking is a key aspect of the use of computer vision in robotics. Feedback from visual sensors allows robots to better interact with their environment. Before exploring object tracking as a means of robotic perception it is important to note the difference between object detection and object tracking. Object detection refers to the detection of an object in a still frame of a photo or video. This differs from object tracking, where the objective is to predict the location of an object in each consecutive frame of a video. This is typically done by means of machine learning and deep learning algorithms. There are several approaches to object tracking, with the most predominant methods being classification and feature extraction. The most difficult problems within the space of modern object tracking deals with occlusion, a state of which view of the object cannot be seen within the camera frame, and background noise within the frame.

B. Feature extraction as a means of object tracking

Feature extraction is a machine learning method that isolates objects by simplifying the image. They typically save time over large networks because they simplify the dataset and require much fewer variables to extract data.

The present state of the art feature extraction model is DAN-Superpoint. It was designed by the authors of [11], with their model consisting of an encoder, feature pyramid network, dual attention network, and two decoders. The feature pyramid architecture adopts a 6-layer convolutional neural network in a 32-32-64-64-128-128 channel shape and passes 3x3 kernels to generate a feature map [1]. The model uses bilinear interpolation to reduce the feature map to 64 channels, then convolves the image after performing multi-scale feature-fusion [1]. This is followed by a dual attenuation network that calculates an output feature map with spatial and channel weighting [1]. Their model was effective at extracting a feature map and proved to be a superior detection model in low texture environments [1].

While DAN excels with feature association, the authors of [2] sought to improve upon it with their own neural network. Their network contained a 4-component architecture: a Mask RCNN stem for bounding box estimations, feature extraction layers, appearance feature association, and bounding box association [2]. Their network used appearance features and bounding boxes, resulting in comparable or some cases better outputs than DAN [2].

In [3] a "fast feature tracker" was proposed that tracks objects based on relative movement in attempt to avoid the delay in detection rate brought about by more advanced object tracking algorithms. For a moving object, new bounding box corners are computed by tracking individual features of an object through consecutive frames. In parallel, their algorithm makes estimates on the presence of occlusions or false detections [3]. They found using feature tracking allowed them to move the camera system while keeping the subject in frame and creating a bounding box with less latency than typical high level tracking systems [3].

C. Multiple Object Tracking (MOT)

Multiple Object tracking is one of the leading challenges and use cases for computer vision in 2022, and in this project. Object detection as opposed to feature extraction poses a new dynamic as classification networks are typically used to extract objects from the frame, allowing tracking algorithms to follow the object throughout its time in view.

This is not always the case however, as the authors of [4] tackle this challenge through image segmentation and morphological operations based on color thresholds. Segmenting an image as opposed to classifying objects in it with a neural network is less computationally and temporally expensive [4]. Without considering occlusion, they were able to track the movement of bacteria with over 91% accuracy via Kalman filtering and the Hungarian algorithm [4].

The group responsible for [5] integrated visual restoration of images with single and multiple object tracking pipelines to enhance underwater robotic perception. Through their research, they were able to implement GAN-RS as a visual restoration network, which restores images from an underwater robot's camera to be passed into a single object

detection network [5]. The team was able to achieve fully autonomous grasping with a soft robotic arm upon integrating the framework to a remotely operated vehicle (ROV) [5].

Simple Online Realtime Tracking (SORT) is a semi-modern solution to multiple object tracking which focuses on Kalman filters combined with frame analysis using the Hungarian method to measure the overlap of bounding boxes [6]. The authors of [6] introduced a deep association metric to SORT, implementing a convolutional neural network model that was trained on over 1,100,000 images. The results of their test showed a dramatic decrease in identity switches, roughly 55% as often as SORT [6].

Zhang, Chen, and Wei applied Deep SORT, built on the You Only Look Once (YOLO) v4 framework [7], to track professional athletes during play [8]. Their experiment showed Deep SORT's successful retainment of tracking ids despite occlusion.

While DeepSORT is widely considered one of the best multiple object tracking applications in modern computer vision, it is not without its faults. Its biggest issue is the reassignment of tracking ids upon long occlusions.

In [9] an algorithm was developed with detection verification in mind, where an object must be identified as the same object as its initial framing before being assigned the same tracking id. The algorithm proved to be more effective at id reassignment of long-term tracks than DeepSORT, where an occluded object is more apt to be assigned a new id [9].

In most cases, occlusion is a major problem for multiple object tracking. For [10], a rendezvous cone-based scheme was developed for air based detection from an unmanned drone. They developed an algorithm to calculate a minimum area ellipse around bounding boxes representing tracked ground vehicles [10]. Simulations of their design proved to be successful at tracking multiple ground vehicles through occlusions at erratic velocities with the use of a single camera [10].

III. Implementation

The overall functionality of this project consists of two python programs. The object tracking program runs on python 3.7.9 with the cup stacking program running on an older version of python to be compatible with Baxter.

As mentioned in Part 1, this project used a custom implementation of Deep SORT. In the Luxonis DepthAI Deep SORT application, the overall network architecture consists of a YOLOv6 network and an embedding network. YOLO as a model is trained on 80 different classes. For this implementation, we only care about cups. The algorithm shown in Fig. 1 was used to filter out the other 79 classes. Deep SORT runs as normal, with this algorithm running sequentially to utilize the tracking ids of only cups. This algorithm was able to circumvent most problems with occlusions because it only allowed for a maximum of the defined number of cups at a given time. If a cup is occluded or missing from the frame the next "new" cup, which in reality is the cup that was lost, is assigned to the cup identifier of the occluded cup.

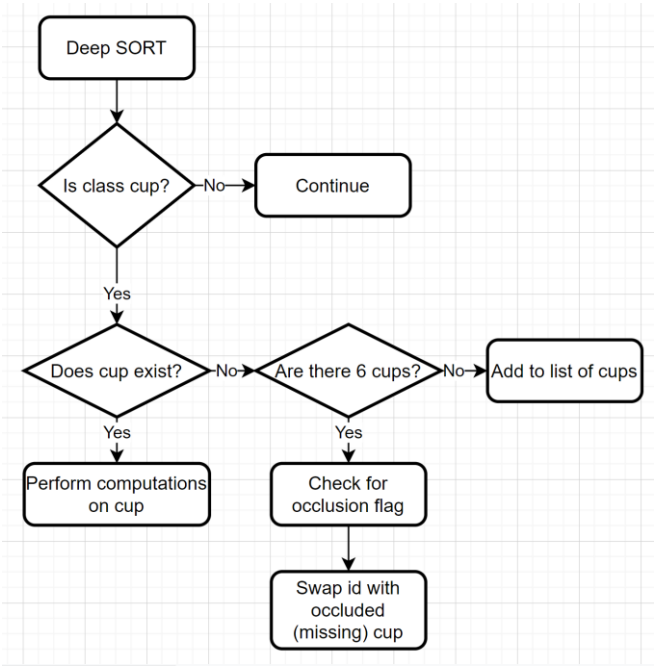

Fig. 1. Algorithm used for cup assignments and occlusion handling

Deep SORT introduced maximum cosine distance as an association method. For each tracking target, the last 100 frames are saved to create a feature vector [8]. The maximum cosine distance parameter determines how close a target can get to another target's location before they may swap ids. This is set to 0.2 by default but it was changed to 0.05 to prevent id swapping in this application.

After cups are mixed to the player's liking, their spatial coordinates are written to a text file. These coordinates, after a frame transformation depending on the camera placement, are read from the file by Baxter's program. Baxter then picks the cups from these locations and places them in a pyramid shape, with the hidden ball cup being the last picked and placed.

An different function was tested in which the cups are picked from predefined locations where the order in which they are picked depends on where in the frame the hidden ball cup is. This model is estimated to be more consistent because of slight variance in depth measurements that may cause the cup to be off center compared to the gripper, as this application does not use an overhead camera.

IV. Results

The algorithm above was successful in preventing occlusions, and the object tracking algorithm worked as intended when used with a dedicated graphics card. Frame rate was however a hardware bottleneck. The difference in frame rate between running the program with a dedicated graphics card and Intel integrated graphics was enough to cause id switching, ultimately rendering the program unusable without a graphics card. A comparison is shown in Fig. 2.

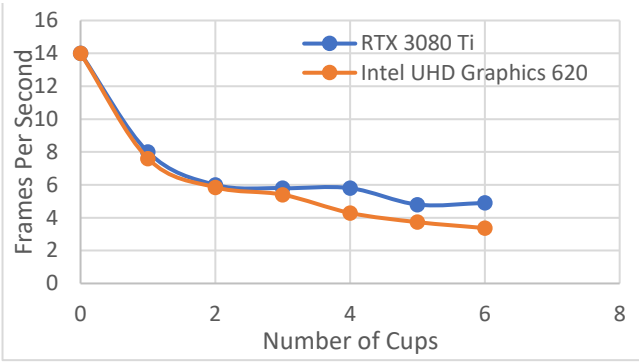

Fig. 2. Frame rate comparison of object tracking application between RTX 3080 Ti dedicated GPU and Intel UHD 620 integrated graphics

The decision was made to perform cup tracking with the cups right side up as opposed to upside down as a traditional ball and cup game is played. The YOLO network was trained to recognize upright cups resulting in a consistently low classification confidence in upside down cups. Fig. 3 below shows this phenomenon.

Fig. 3. Comparison in classification confidence between upside down cups and right side up cups

Setting the maximum cosine distance to 0.05 resulted in a tracking performance with no mistakes when tested with a dedicated GPU. It was found that the system does not have a problem with tracking if moving cups are moved in front of stationary cups. This prevents long term occlusions that could result in random reassignment of ids.

Baxter was consistently able to stack cups as long as the cup is in the center of the gripper. A completed stack is shown in Fig. 4.

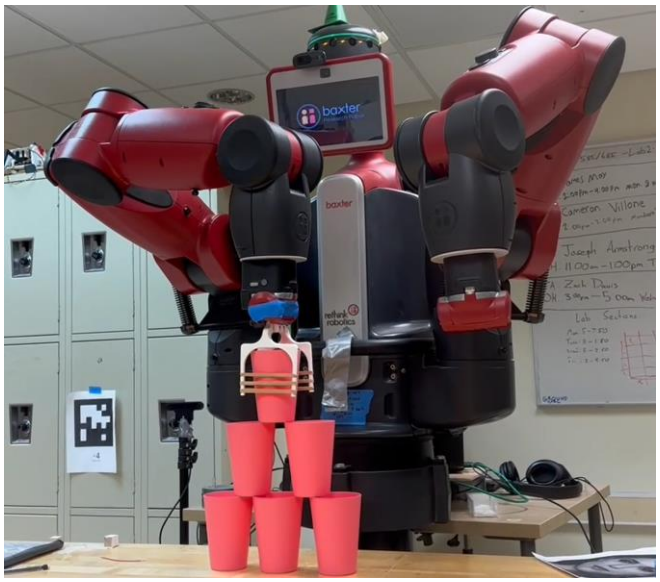

Fig. 4. Baxter stacking cups after a person is finished mixing. The hidden ball cup is placed on top of the tower

V. Future Works

Additional work for this project may include making it more lightweight and portable to not limit its implementation to high power laptops. A hand camera for gripper readjustments could rid the system of some inconsistencies where the cup is not centered when picking.

Acknowledgment

Thank you Dr. Sahin, Cameron Villone, and Zach Davis for your help and support on this project.

References

[1] Z. Li, J. Cao, Q. Hao, X. Zhao, Y. Ning, and D. Li, “Dan-SuperPoint: Self-supervised feature point detection algorithm with dual attention network,” *Sensors*, vol. 22, no. 5, p. 1940, 2022.

[2] R. hu, S. Bouindour, H. Snoussi, A. Cherouat and C. Chahla, "Multiple Cues Association for Multiple Object Tracking Based on Convolutional Neural Network," 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 117-122, doi: 10.1109/AIKE.2019.00030.

[3] W. Pairo, P. Loncomilla, and J. R. del Solar, “A delay-free and robust object tracking approach for Robotics applications,” Journal of Intelligent & Robotic Systems, vol. 95, no. 1, pp. 99–117, 2018.

[4] Y. Qian, H. Shi, M. Tian, R. Yang and Y. Duan, "Multiple Object Tracking for Similar, Monotonic Targets," 2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2020, pp. 360-363, doi: 10.1109/CYBER50695.2020.9279162.

[5] Y. Lu, X. Chen, Z. Wu, J. Yu, and L. Wen, “A novel robotic visual perception framework for underwater operation,” Frontiers of Information Technology & Electronic Engineering, vol. 23, no. 11, pp. 1602–1619, 2022.

[6] N. Wojke, A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645-3649, doi: 10.1109/ICIP.2017.8296962.

[7] A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[8] Y. Zhang, Z. Chen and B. Wei, "A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1312-1316, doi: 10.1109/ICCC51575.2020.9345010.

[9] J. Redmon, S. Divvala, R. Girshick and

[10] H. Park, S.-H. Ham, T. Kim, and D. An, “Object recognition and tracking in moving videos for Maritime Autonomous Surface Ships,” Journal of Marine Science and Engineering, vol. 10, no. 7, p. 841, 2022.

[11] K. Dhal, P. Karmokar, A. Chakravarthy, and W. J. Beksi, “Vision-based guidance for tracking multiple dynamic objects,” Journal of Intelligent & Robotic Systems, vol. 105, no. 3, Jul. 2022.