Bryce George and Colin Henson

CS 4800-001

Presented on November 5, 2020

Work Undertaken for Demo 2

Our work to prepare for the second demo consisted of selecting a method for bowl detection and integrating it with the CSRT and Multi-Tracker mechanisms identified for Demo 1. After investigating a number of different Deep Learning based methods for detecting objects, we discovered the ImageAI Python package from Moses Olafenwa and decided to use it to train a custom object detector based on the YOLOv3 algorithm.

In order to train the object detector, we needed to build a custom data set with a minimum of approximately a thousand examples of the targeted object, the bowls. We initially selected approximately 200 frames from segments of the video which were representative of the many different scales and perspectives the exhibit is examined from. We trained an initial set of models and evaluated them using the ImageAI package. Based on the model evaluations, we selected the model with the highest mean Average Precision score for further testing. The detector functioned well overall; however, it failed to recognize bowls in a few instances and also interpreted a roughly circular dark stain on one of the pool’s edges as a bowl. At that point, we elected to increase the size of our data set with the hope that it would improve the detector’s accuracy. We added another 100 or-so frames to the data set and re-trained the object detector. Upon repeating the model evaluation, selection, and testing we were significantly more satisfied with the result.

With an object detection mechanism that met our requirements, the next step was to integrate it with the object tracking mechanisms from the first phase of the project. We created a prototype which uses the YOLOv3 object detector to identify bowls in the first frame of a video. The bounding boxes of the detected objects are then used to initialize a CSRT tracker for each bowl, which are then aggregated into a single Multi-Tracker object. The Multi-Tracker is then updated with each subsequent frame of the video, yielding the tracked positions of each of the objects in the video. This prototype proves that integration between the object detection and object tracking mechanisms is possible.

The remaining work for Demo 3 includes implementation of the redetection-tracking sequences and the graphical visualizations of bowl paths and interactions. Our plan for doing the former is to implement a threshold-bound, closest centroid identification algorithm: the final frame of a tracking sequence will be run through the object detection mechanism, at which point, the object detections between the two will be reconciled. The sets of centroids from the tracker and object detector will be computed and then compared. Closest centroids between the two sets—within a threshold—will be identified as the same object. Meanwhile, object detection will be preferred to provide the opportunity to add new objects which have come into the scene and to remove objects which have left the scene during the preceding tracking sequence. Meanwhile, visualization will be driven by collecting an object’s position for each frame in a tracking sequence. The “trail” indicating a bowl’s path history will be rendered by placing a colored dot at each position in the current and preceding *n*-frames. The dots will range in size according to how recently the bowl was at that position, growing smaller for subsequent frames as the bowl moves further away.