

Group 21 -- PoolNet Prototype

Roderick L. Renwick

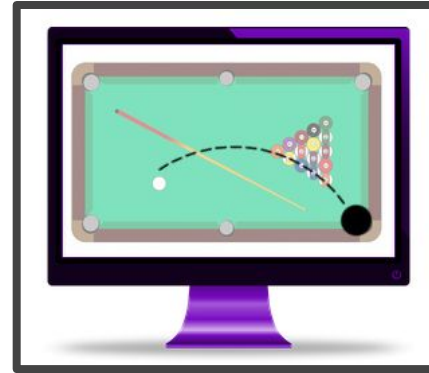
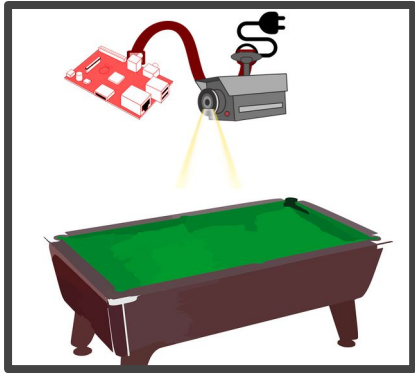
rrenwick@purdue.edu.com

<https://www.rodericklrenwick.com>

December 4th, 2021

Overview

The aim of this project is to present a solution capable of displaying the trajectory of a pool shot in real time as people play billiards. Users could map their shots and look at which one the computer suggests as being the most feasible.



Understanding the problems

- How do you display the trajectory of a pool shot?

er

Target audience

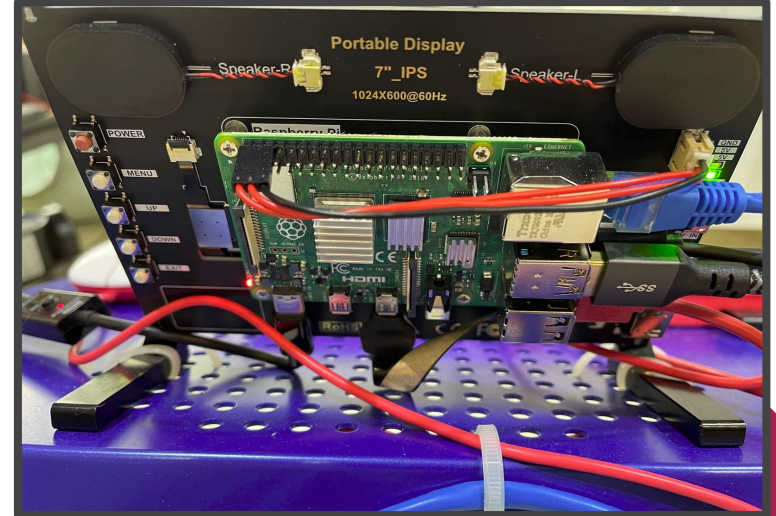
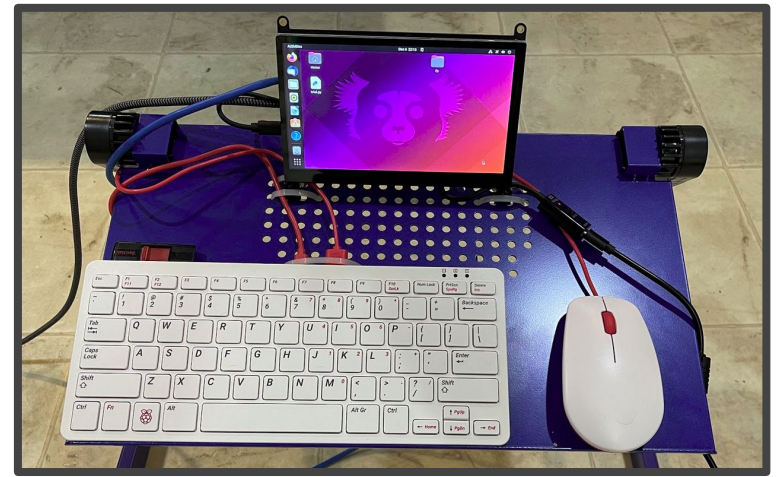
The targeted audience for this project is those who play billiards frequently and would be curious about suggestions to their own self-improvement.



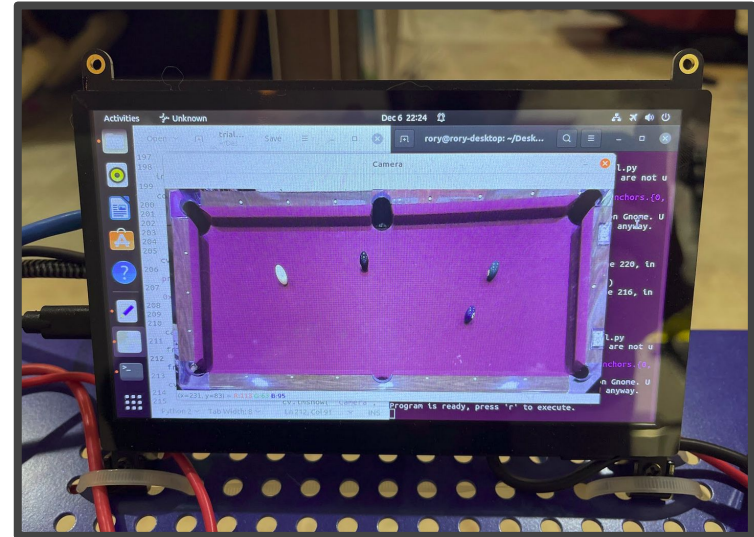
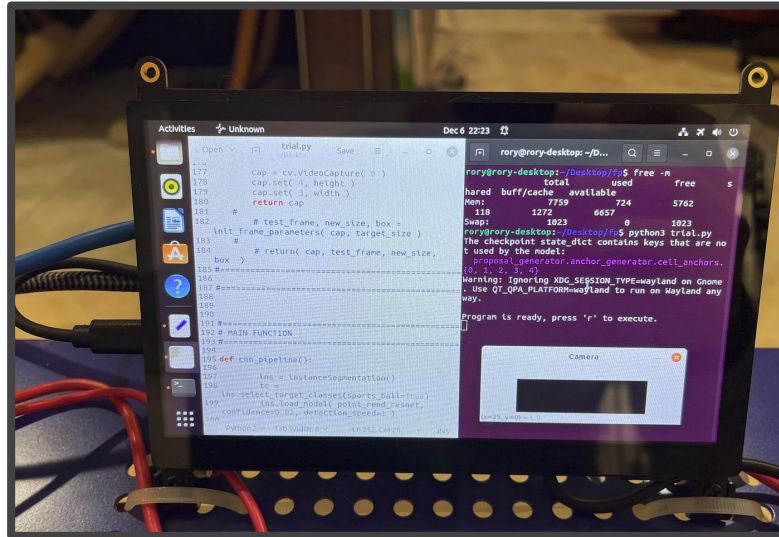
Core Questions to Consider

- **What is the problem:** Human limitations when playing pool/billiards.
- **Who cares for a solution:** People curious about a solution path when unable to find one.
- **Where can it be used:** In any controlled environment (i.e., indoors with stable lighting and clear visibility) where a camera can be stationed. For prototype simplicity, this implementation will be done via a ceiling mount to position the camera lens parallel with the pool table.
- **When can it be used:** 24/7/365.
- **What deficiencies are present:** Time constraints and training datasets for billiards.
- **Why is embedded vision needed:** To enable autonomous inferencing IRT as it determines the position of the balls, the angle and path needed to sink the next target.
- **Why is this solution unique:** Hopefully there is a basic API to project the board state to (with the position of the balls on the table). From there, take the projected line of path and draw it onto a webcam feed running in real time. Essentially pointing a camera at a pool table to then project the best calculated shoot onto a display, drawing out the projected paths. This solution case makes it uniquely adaptable to any table that has adequate lighting.
- **Why is this solution better:** At its current state, it is not a superior solution since it is not consistently capturing all balls on the table.

Hardware Infrastructure



Software Execution



Original Img



Table BBoxes



Largest BBox



Cropped Original



Cropped BBox



Cropped Segmentation



Extracted



Segmented



Masked



Table Detection

Original Img



Table BBoxes



Largest BBox



Table Segmentation

Cropped Original



Cropped BBox



Cropped Segmentation



Table Masking

Extracted



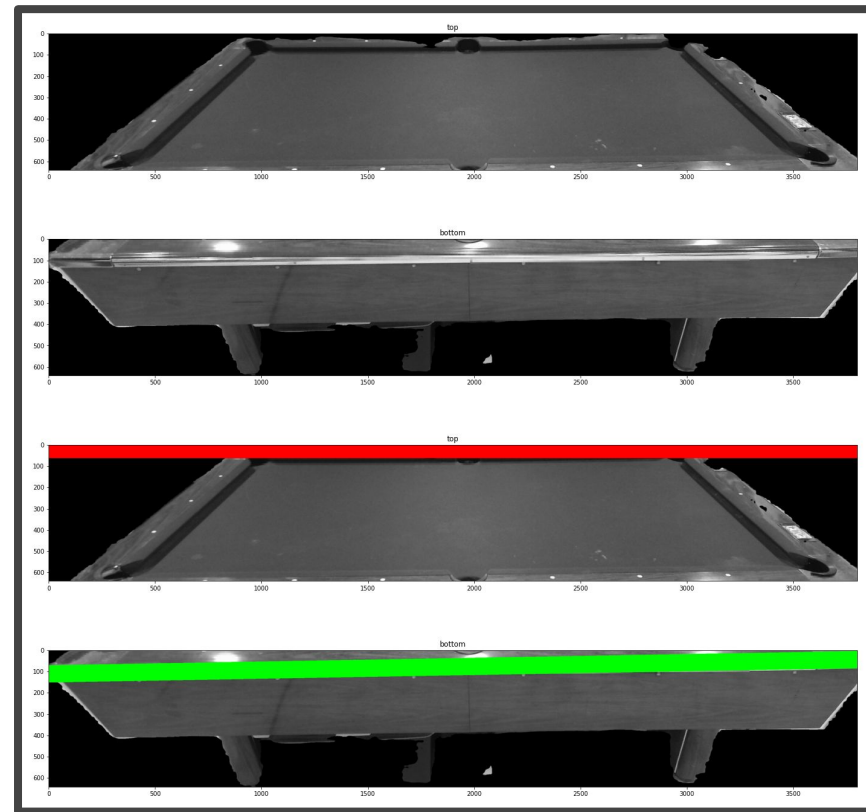
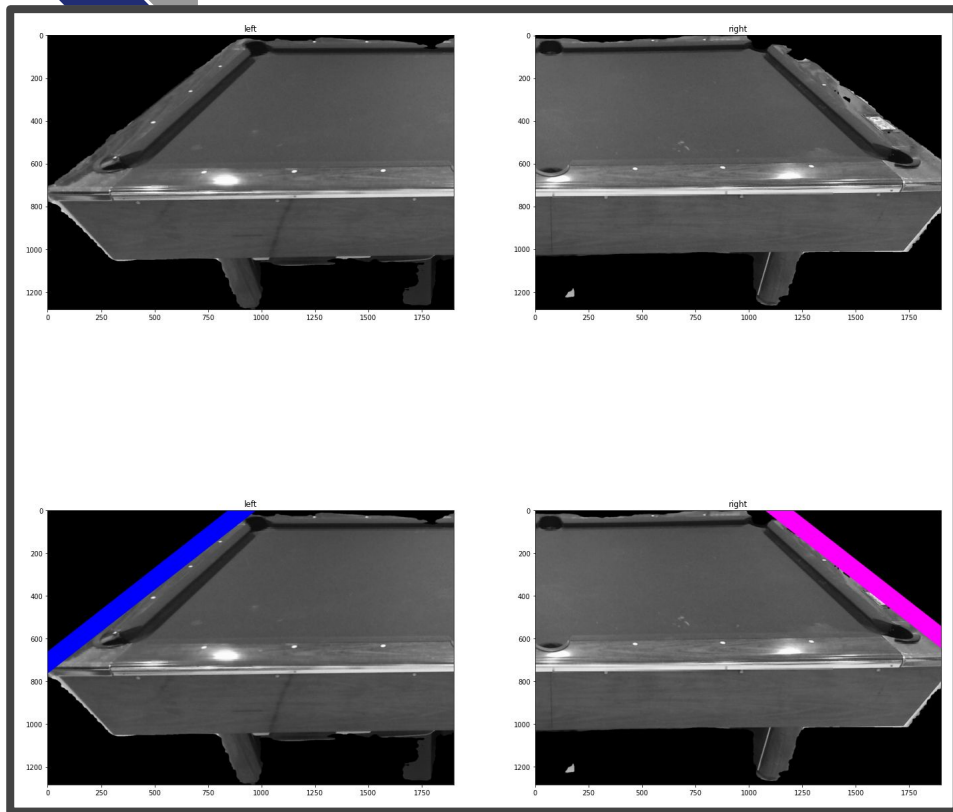
Segmented



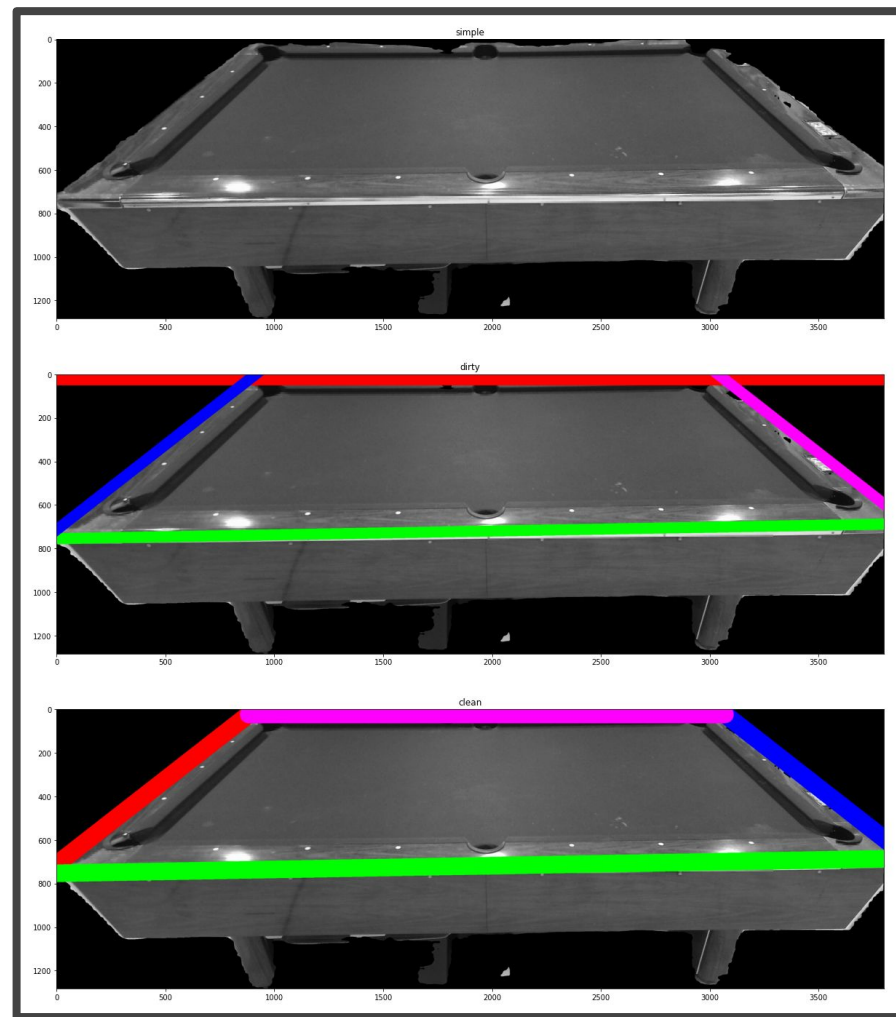
Masked



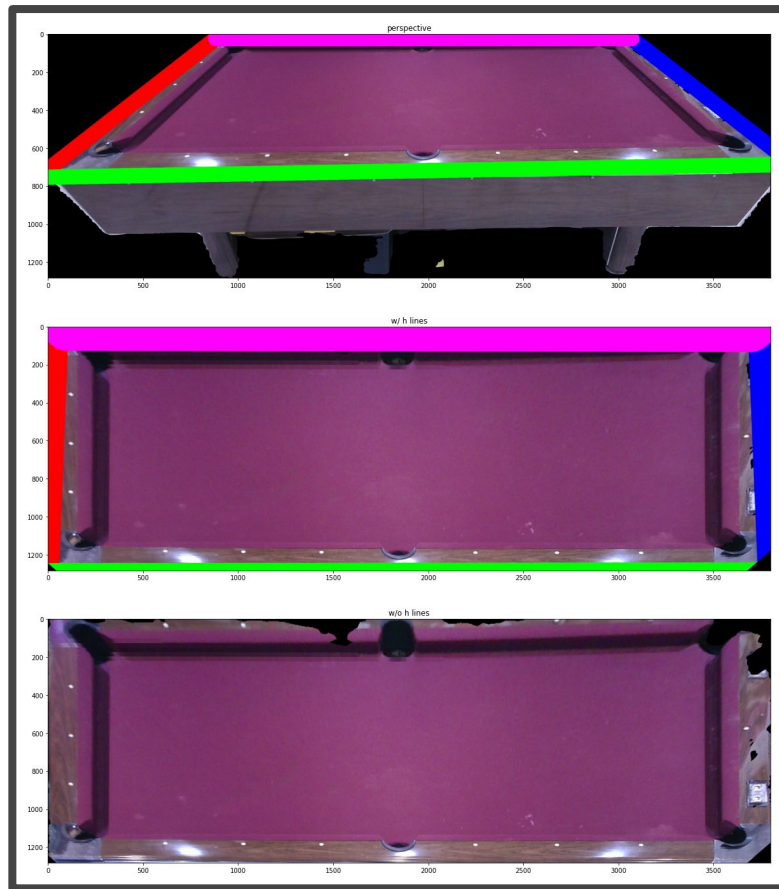
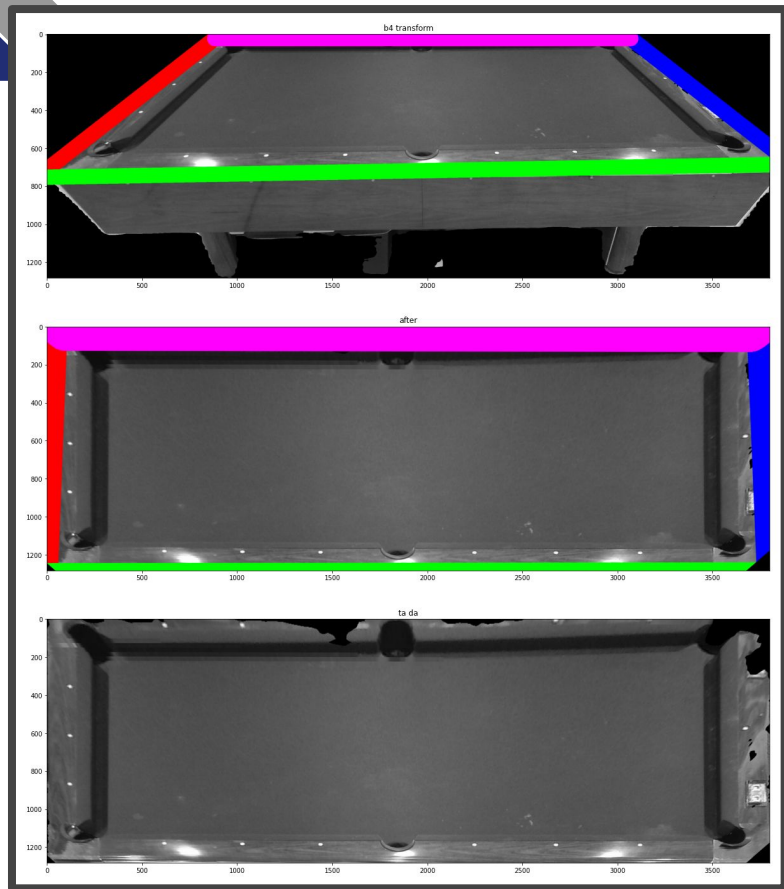
Line Detection



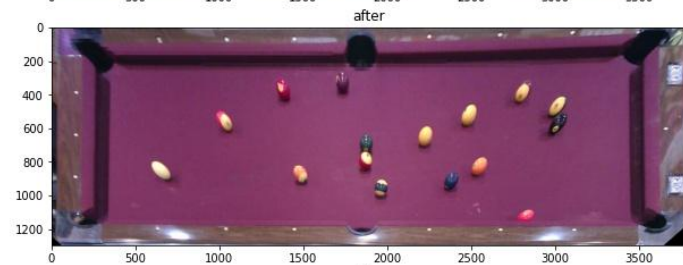
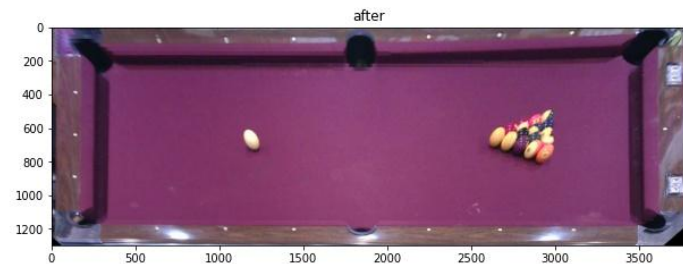
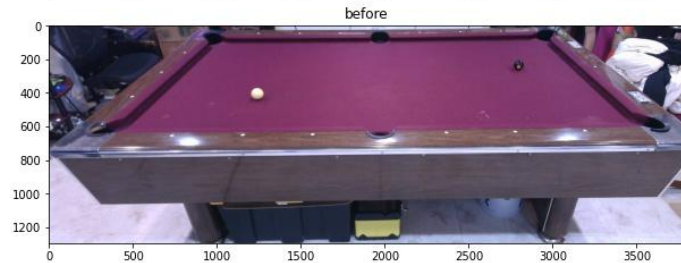
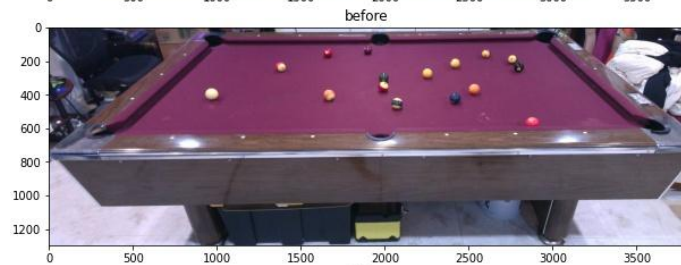
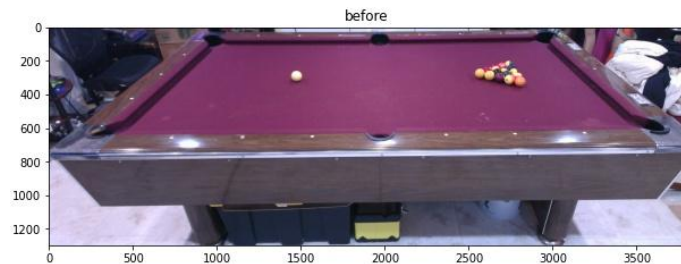
Line Detection Result



Perspective Transform



Perspective Transform Samples



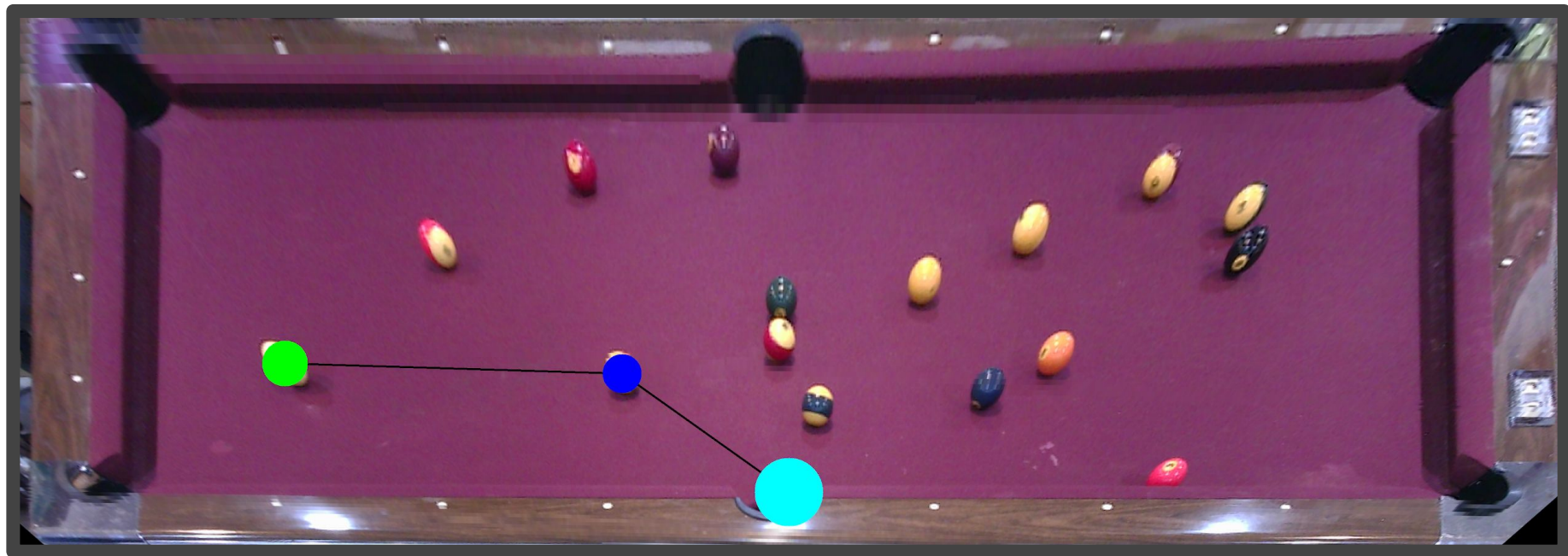
Object Detection Samples



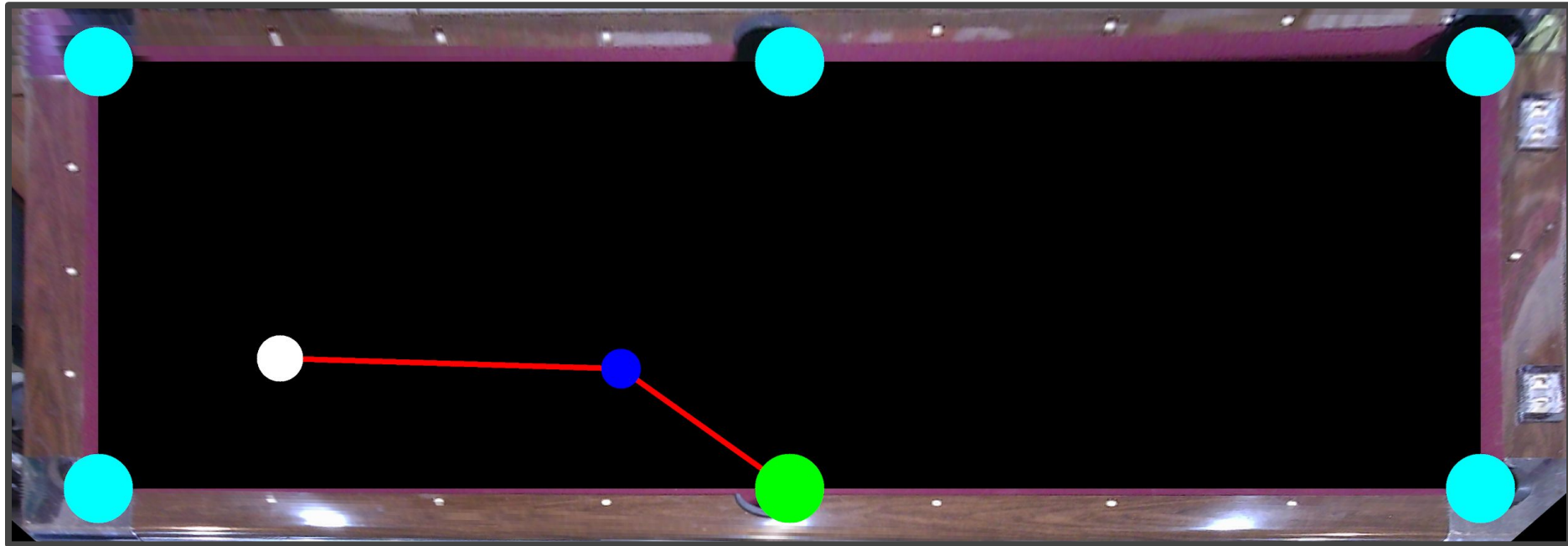
Instance Segmentation Sample Result



Mapped Shot Trajectory



Displayed Shot Trajectory





Feedback -- Main Critique

- Real time image segmentation often fails to capture all sports balls (i.e., pool balls) within image frame, thus leading to cascading challenges
- With enough time, would have liked to implement API call to physics game engine used to compute best feasible shot
- Dynamic capturing of pool table boundaries are not perfect, and the results show when transforming perspective via homography matrix

Thank you!