Object Detection Pipeline:

Read video input and perform detection in each frame:

1. Select & get the region of Interest (ROI)

* Crops input frame based on the input parameter: ROI rectangle’s upper left corner position (x, y), width, and height
* Purpose of this step is to reduce false positives detected in the background
* Also reduces computing time by reducing to smaller number of pixels
* Since the pitcher position is fixed in games, the parameters will be same as long as the camera position is fixed

1. Convert the region in greyscale

* Since the object we’re trying to detect is white and round, we do not need RGB channels in the detection process

1. Apply Image Smoothing

* Uses **Gaussian smoothing**:
  + Method to use a Gaussian function to create a smoothing filter.
  + Removes noise, allows smoother transitions between different intensity levels, and improves the consistency of regions in the frame
* Smooths out the image using 7x7 kernel with sigma = 1.0

1. Apply thresholding to select pixels close to white

* Uses manual **thresholding** method:
  + Converts a grayscale image into a binary image.
  + Pixels with an intensity greater than or equal to the threshold are set to white, and those with lower intensity are set to black
* Pixel intensity less than 130 will be classified as black and between 130 and 255 will be white

1. Perform Edge Detection

* Uses **Canny Edge Detection**
  + Removes the noise with a 5x5 Gaussian filter
  + Computes gradients (derivatives) in the frame to identify areas of rapid intensity change by filtering with a Sobel kernel in both horizontal and vertical direction
  + Non-maximum Suppression - Thins the edges by suppressing non-maximum values in the gradient direction
  + Hysteresis Thresholding - Uses two thresholds to categorize edges
    - Pixels with gradient magnitudes above a high threshold are considered strong edges, while those below a low threshold are discarded
    - Pixels between the two thresholds are considered weak edges, which may be retained if they connect to strong edges

1. Identify circular shapes in current frame

* Uses **Hough Circle Transforms**
  + For each edge point in the image, the algorithm considers all possible circles passing through that point, varying in center coordinates and radius
  + Votes for possible circles by incrementing the value in the parameter space that corresponds to the circle parameters
  + Circle candidates with the most votes in the parameter space are returned as the detected circles
  + Adjust the parameters of the openCV method manually to detect most true positive circles and least false positive circles.
  + Parameters:
    - Circle Radius (Min and Max): Allows us to only look for circles close to baseball size in frame
    - Minimum Distance: Controls the minimum distance between circle centers and helps to prevent detecting redundant circles
    - Accumulation Threshold: Determines the number of votes required to identify a circle
    - Accumulator Ratio (DP): Inverse ratio of the accumulator resolution to the image resolution (coarser accumulator when it’s a higher value)

1. Pass the detected circle coordinates to object tracking