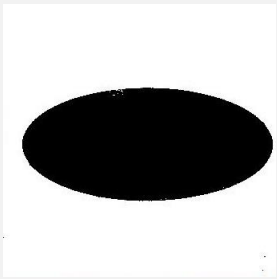
Object Detection and Tracking

Range detection: Object in an image can be detected based on their color ranges. Masks for any uniformly colored object in an image can be created by knowing their upper and lower threshold values in HSV space.

In range_detection.py, we can find color ranges for specific-colored objects by passing an image of the object. We can change the minimum and maximum values for different color channels and simultaneously see the updated object masks (black or white).

Object Image



Mask (Black/White)

Object tracking: Now that we can successfully create a mask for an image, we will use this concept to build an object tracking tool.

In object_detection.py, we can find implementation of preprocessing of video frames. Preprocessing involves resizing, filtering and changing the color space of the video frame. After preprocessing, masks can be created using the threshold information of object color obtained from range_detector.

GaussianBlur: Performs Gaussian filtering of 11 x 11 size kernel
cvtColor (cv2.COLOR_BGR2HSV): Converts BGR space to HSV color space
inRange: Creates the initial mask using the lower and upper threshold values for object color
erode: Performs eroding on image to remove noise in the mask
dilate: Performs dilating on mask to enhance the boundaries
findContours, grab_contours: Finds available contours in the mask
minEnclosingCircle: Finds largest contour in the mask and use it to compute the minimum enclosing circle center.

In the end, the obtained contours can be used to calculate center and radius enclosing an object. Further, the available information is processed to implement bounding curve and center tracking for a video.