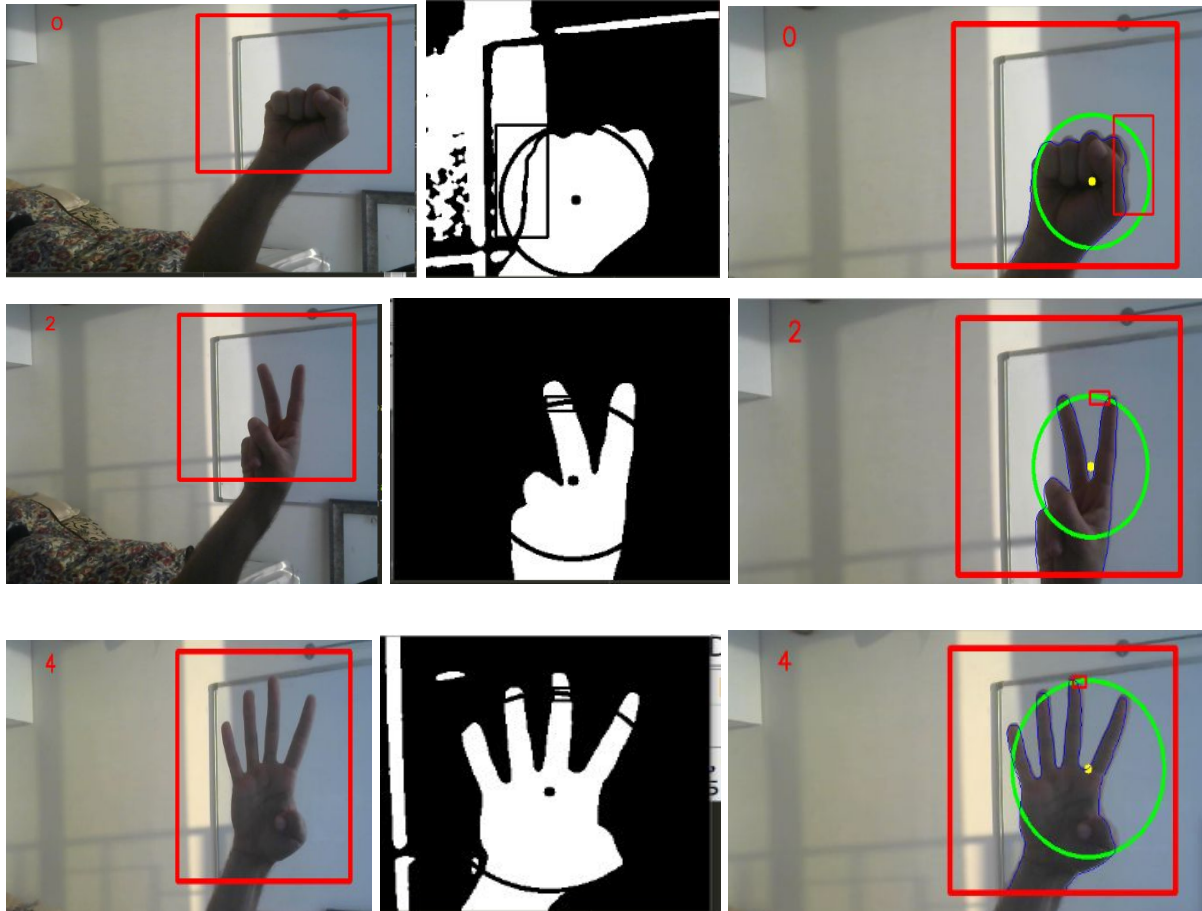


FINGER COUNT

Count how many fingers are being held up, implemented OpenCV.

DEMO



pre-processing

Background Extraction from a Video

Background extraction comes important in object tracking. If you already have an image of the bare background, then it is simple. But in many cases, you won't have such an image and so, you will have to create one. That is where Running Average comes in handy.

The function we use here to find Running Average is [cv2.accumulateWeighted\(\)](#)

we keep feeding each frame to this function, and the function keep finding the averages of all frames

I found that $\alpha = 0.02$ achieves good results for 60 frames.

Segment

Grabbed the hands segment from the region of interest.

In order to perform background subtraction, we first must "learn" a model of the background.

Once learned, this background model is compared against the current image and then the known background parts are subtracted away.

The object left after subtraction are presumably new foreground objects.

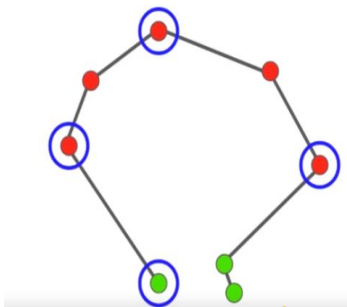
And then we can find the external contours from the image.

I had to blur the frames using Gaussian blur and morphology operations (opening) to achieve better results

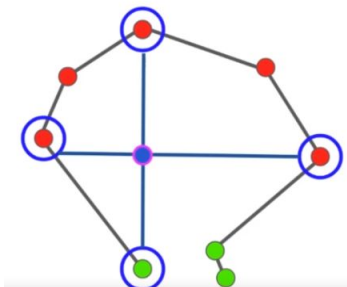
Count the fingers being held up

The idea is this:

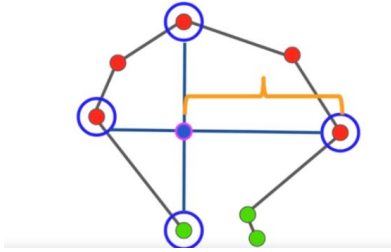
- Find the extreme points in the hand



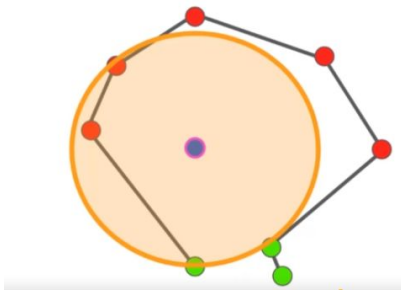
- Then use their intersection to estimate the center of the hand.



- Calculate the distance for the point further away from the center



- Using that distance to create circle(Any points that are outside of the circle,those should be extended fingers)



- Return the number of the fingers and procedures frames.