1. Start Inference
2. Extract Frames from Video

3. Segment the input 3 channel image (the default input image size of YOLOv5s architecture is 3 × 640 × 640) into four slices with the size of 3 × 320 × 320 per slice, using a slicing operation.
4. Utilize Concat operation to connect the four sections in depth, with the size of output feature map being 12 × 320 × 320,
5. Through the convolutional layer composed of 32 convolution kernels, generate the output feature map with a size of 32 × 320 × 320.
6. Through the BN layer (Batch Normalization) and the Hardswish activation functions, output the results into the next layer.

7. Store each predicted class name and respective bounding box (bbx) coordinates in a list along with the frame number.
8. For each predicted class in the current frame, calculate euclidean distance ($d = \sqrt{[(x22 - x11)2 + (y22 - y11)2]}$, $where, (x11, y11)\ are\ the\ coordinates\ of\ one\ point.\ (x22, y22)\ are\ the\ coordinat$ $d\ is\ the\ distance\ between\ (x11, y11)\ and\ (x22, y22).$) between its bbx centre and all the other stored bbxs' centres received from the previous observations.
9. Calculate the std. deviation of all the distances calculated so far.
10. Discard the class name from consideration if current distance is greater than std dev calculated in previous step, otherwise add respective class name in a blank list, namely 'A'.
11. Take 5 most recent/ latest class names from blank list 'A'
12. Choose the most frequent class name from those 5 elements. If there are more than one element having the highest frequency, choose the class which has the most recent occurrence.
13. Repeat the steps unless all frames are encountered.