```
from ultralytics import YOLO
import sys

# Diccionari clau: nom-video, valor: objecte a detectar
objectNames = {
    ’Alladin’:’person’,
    #’Aquarium1’:’person’,
    #’Aquarium2’:’person’,
    ’Badminton1’:’person’,
    ’Badminton2’:’person’,
    ’Basketball’:’person’,
    ’Bharatanatyam’:’person’,
    ’Bike’:’person’,
    ’Billiards1’:’sports ball’,
    ’Billiards2’:’sports ball’,
    ’Boat’:’boat’,
    ’Boxing1’:’person’,
    ’Boxing2’:’person’,
    ’Boxing3’:’person’,
    ’BreakfastClub’:’person’,
    ’CarChase1’:’car’,
    ’CarChase2’:’car’,
    ’CarChase3’:’car’,
    ’Dashcam’:’car’,
    ’DriftCar1’:’car’,
    ’DriftCar2’:’car’,
    ’Drone1’:’person’,
    ’Drone2’:’person’,
    ’Drone3’:’person’,
    ’Elephants’:’elephant’,
    ’Helicopter’:’airplane’,
    ’Hideaway’:’person’,
    ’IceSkating’:’person’,
    ’ISS’:’person’,
    ’Jet1’:’airplane’,
    ’Jet2’:’airplane’,
    ’Jet3’:’airplane’,
    ’Jet4’:’airplane’,
    ’Jet5’:’airplane’,
    ’KinBall1’:’sports ball’,
    ’KinBall2’:’sports ball’,
    ’KinBall3’:’sports ball’,
    ’Lion’:’cow’,
    ’Mohiniyattam’:’person’,
    ’MotorcycleChase’:’motorcycle’,
    ’Parakeet’:’bird’,
    ’PolarBear1’:’bear’,
    ’PolarBear2’:’bear’,
    ’PolarBear3’:’bear’,
    ’Puppies1’:’dog’,
    ’Puppies2’:’dog’,
    ’Rope’:’person’,
    ’Sam’:’person’,
    ’Violonist’:’person’,
    #’ZebraFish’:’person’
}

# Retorna cantonada de dalt a l’esquerra de la caixa
def yoloBoxToTopLeft(coordinates, width, height):
    (cx, cy) = ((coordinates[0]+coordinates[2])/2,(coordinates[1]+coordinates[3])/2)
    minx = cx - width/2
    miny = cy - height/2
    return (minx,miny)

# Retorna Overlapping Ratio
def overlapRatio(xmin1, ymin1, xmin2, ymin2, width, height):
    xmax1 = xmin1 + width
    xmax2 = xmin2 + width
    ymax1 = ymin1 + height
    ymax2 = ymin2 + height

    xmin = min(xmax1,xmax2) - max(xmin1,xmin2)
    if xmin < 0:
        xmin = 0
    ymin = min(ymax1,ymax2) - max(ymin1,ymin2)
    if ymin < 0:
        ymin = 0
    area_interseccio = xmin*ymin
    area_unio = 2*width*height - area_interseccio
    return area_interseccio/area_unio

# CÃ lcul del overlapping del frame segons oclusiÃ³ i multiple selecciÃ³
def selectOverlapRatio(points, xmin, ymin, width, height, isLost):
    if isLost:
        if len(points):
            return 0
        else:
            return 1

    if not len(points):
        return 0

    # Escollir caixa amb millor confidence level
    max_c = 0
    max_p = (0,0)
    for (conf, point) in points:
        if conf > max_c:
            max_c = conf
            max_p = point

    (x,y) = max_p
    return overlapRatio(xmin, ymin, x, y, width, height)

def runVideo(videoname, model):
    # Llegir fitxer dades
    with open(’TinyTLP/’ + videoname + ’/groundtruth_rect.txt’, ’r’) as f:
        BBS = [[int(num) for num in line.split(’,’)] for line in f]

    results = model.predict(source="TinyTLP/" + videoname + "/img")
    overlappingRatios = []

    # CÃ lcul millor caixa
    i = 0
    for result in results:
        [_,xmin,ymin,width,height,isLost] = BBS[i]
        points = []
        for r in result.boxes:
            # Descartar objectes
            if result.names[r.cls.item()] != objectNames[videoname]:
                continue
            [coords] = r.xyxy.tolist()
            # Afegir punt com a possible candidat
            points.append((r.conf.item(), yoloBoxToTopLeft(coords, width, height)))

        # Afegir overlapping ratio de la millor caixa
        overlappingRatios.append(selectOverlapRatio(points, xmin, ymin, width, height, isLo
st))
        i += 1

    # Escriptura de valors al fitxer
    with open(videoname + ’Internet.txt’, ’w’) as k:
        for o in overlappingRatios:
            k.write(str(o) + ’\n’)


model = YOLO("yolov8m.pt")
```

```
#per a un sol video rebut com a parÃ metre
#videoname = sys.argv[1]
#runVideo(videoname, model)

#per a fer-los tots
for v in objectNames.keys():
    runVideo(v, model, method)
```