```
import sys
import roboflow

# Diccionari clau: nom-video, valor: objecte a detectar
objectNames = {
    'Alladin':'Person',
    'Aquarium1':'Fish',
    'Aquarium2':'Fish',
    'Badminton1':'Person',
    'Badminton2':'Person',
    'Basketball':'Person',
    'Bharatanatyam':'Person',
    'Bike':'Bike',
    'Billiards1':'ball',
    'Billiards2':'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':'Helicopter',
    'Hideaway':'Person',
    'IceSkating':'Person',
    'ISS':'Person',
    'Jet1':'Jet',
    'Jet2':'Jet',
    'Jet3':'Jet',
    'Jet4':'Jet',
    'Jet5':'Jet',
    'KinBall1':'ball',
    'KinBall2':'ball',
    'KinBall3':'ball',
    'Lion':'Lion',
    'Mohiniyattam':'Person',
    'MotorcycleChase':'MotorBike',
    'Parakeet':'Bird',
    'PolarBear1':'PolarBear',
    'PolarBear2':'PolarBear',
    'PolarBear3':'PolarBear',
    'Puppies1':'Dog',
    'Puppies2':'Dog',
    'Rope':'Person',
    'Sam':'Person',
    'Violinist':'Person',
    'ZebraFish':'Fish'
}

# Retorna cantonada de dalt a l'esquerra de la caixa
def centerToTopLeft(coordinates, width, height):
    (cx, cy) = coordinates
    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 seleccciÃ³
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]

    overlappingRatios = []

    # CÃ lcul millor caixa
    for i in range(1,101):
        # Nom de les imatges segons valor de i
        name = "./TinyTLP/" + videoname + "/img/000" + str(i) + ".jpg"
        if i < 10:
            name = "./TinyTLP/" + videoname + "/img/0000" + str(i) + ".jpg"
        if i >= 100:
            name = "./TinyTLP/" + videoname + "/img/00" + str(i) + ".jpg"

        prediction = model.predict(name)

        [_,xmin,ymin,width,height,isLost] = BBS[i]
        points = []
        for p in prediction:
            # Descartar objectes
            if p["class"] != objectNames[videoname]:
                continue
            coords = (p["x"], p["y"])
            # Afegir punt com a possible candidat
            points.append((r["confidence"], centerToTopLeft(coords, width, height)))

        # Afegir overlapping ratio de la millor caixa
        overlappingRatios.append(selectOverlapRatio(points, xmin, ymin, width, height, isLost))

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


rf = roboflow.Roboflow(api_key="7ryBC8sKb0QeK9S2EXmK")
project = rf.workspace().project("upc-3sj4d/ara_si")
model = project.version("1").model

#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)
```