Computer Vision Project Documentation

In this document I will present my work done on Task 1 and 2 from this project.

Regarding Task 1, in order to run it you have to run task1.py from near the train, test and evaluations folders and place the 4 txt files that I uploaded in the full-configuration-templates folder. In order to run from the test you need to update the paths accordingly.

My approach for this task was the following:

- I classify the photo as being part of one of the 4 lanes presented in the photos using template matching.
- Knowing in which lane we are, we load the bounding boxes of the 10 pins standing from the according txt file from the *full-configuration-templates* folder.
- These 4 txt files contain the bounding boxes of the 10 pins from the lane the picture is taken. I generated these files using a combination of YOLO algorithm and manual annotation.
- Now we try to detect the current pins in the image using YOLO V8x. I did some parameter tuning making it look only after two classes (39-bottle, 75-vase) which resemble a bowling pin and confidence of 0.02 in order to detect more objects.
- Now we iterate through the requested bounding boxes of some of the stable 10 pins. For each requested pin, we iterate through our detected objects. If we find at least one detected bounding box that shares at least 25-30% of the area with the original bounding box of the pin from the txt file we say that that pin is standing in the image(save it as 1, else 0).

Regarding Task 2 in order to run it you have to run task2.py from near the train, test and evaluations folders.

My approach for this task was the following:

- We break each video into frames and try to detect the bounding box of the ball.
- First we skip the first frame because we already have the bounding box for it.
- For each other frame we use a parameter tuned YOLO V8x with a confidence of 0.005 and iou of 0.5 to detect any object in the frame (we don't specify classes anymore).
- We calculate the iou between the bounding box detected in the previous frame and the one of the objects in the frame.
- We filter out those objects that have iou<0.2 with the bounding box in the previous frame.
- Now we will have 3 cases:
    a. If there still objects left, the object with the highest iou is considered the good bounding box for the bowling ball in that frame

b. If there aren't any objects left and we have tracked the object in at least 2 frames until now (i.e. we are at least at the 3rd frame from the video) we look at the bounding boxes from the previous 2 frames and we: get the direction from frame i-2 to frame i-1, get the median area and the distance between the those 2 previous frames. With this information we create the bounding box having the median area, in the calculated direction and at the same distance from frame i-1 as frame i-1 is from frame i-2.
c. If there aren't any objects left and we haven't tracked the object in at least 2 frames until now (i.e. we are at the 2nd frame from the video) we consider the previous bounding box as the current one.