Solutions to Task 1

- The approach to solving Task 1 is based on identifying the cells visible to each cop. The idea to tackle down the problem at hand is to find the all cells each cop can see and append the cells to the list. To do this:
    - We first make the viewing range for each cop in the given grid by dividing the FoV by 2, and subtracting and adding it to the orientation of each cop. For example, in the sample subset the range of cop1 was 150 degrees to 210 degrees because their orientation and FoV was 180 and 60 degrees each.
    - Once that is done, we check the angle each corner of a cell makes to determine if it is seen. It is done by the following formula:

      $$\theta = atan2(-\frac{xcorner - xcenter}{ycorner - ycenter})$$

    - If either of the corners makes an angle in the range (and not equal to either of the bounding ranges) the whole cell is considered seen and is added to the list of visible cells for the associated cop.

- From there we can cross check if the thief is in the line of sight by seeing if their location is mentioned in the aforementioned lists which essentially gives us the first part of the result.

- Once that is done, we need to identify the closest safest cell to the thief which can be done by using BFS to see which unmonitored cell is the closest to the thief be referencing it to the lists of visible cells.

- To make the question viable for multiple data sets, multiple safety checks were applied. They were:
    - Ensuring Orientation of each in the range of 0 – 360 degrees and it is normalized.
    - Putting conditions to avoid situations where the upper bound is less than lower bound. This would happen when the upper bound crosses 360 degrees and is normalized back in the range of 0-360 degrees.
    - Ensuring FoV cannot be more that 360 degrees. Once that happens, the problem excludes the cop and treats the cop as if they have no sight and their FoV is 0 degrees.
    - It raises an error when the thief is not present or cannot be detected.

AIM
AIM · INTELLIGENT · MACHINES

Solutions to Task 2

- To approach this question, my approach is to train a model which tracks the soccer ball at hand.
- The question at hand demands for the ball being juggled to be track.
- To train the model to detect the ball at hand, few frames were extracted as training and validation frames.
- These frames were used to annotate images (on CVAT) and produce Yolov8 compatible dataset for ball detection.
- Train a Yolov8 model with the received dataset images.
- Load the trained model and apply it to the video to detect the soccer ball in the video, creating bounding boxes around it as it moves. The video was then saved in the output directory.
- Produce a .csv file with the {frame_number}, {x_center}, {y_center}, {x_size} and {y_size} and save it in the output directory, where:
    - frame_number corresponds to ordinal number of the frame for which the ball is detected.
    - x_center and y_center are pixel corresponding to the center of the ball in x and y axis
    - x_size and y_size denote the size of the bounding box in x and y axis.