**KEY FUNCTIONS**

**1. calculate\_direction(bbox)**

* **Purpose**: Determines the movement direction of the detected object based on the centroid of its bounding box.
* **Parameters**:
  + bbox: A list of four integers representing the bounding box coordinates [x1, y1, x2, y2].
* **Logic**:
  + **Centroid Calculation**: Computes the centroid of the bounding box using the formula ((x1 + x2) // 2, (y1 + y2) // 2).
  + **Direction Determination**:
    - If prev\_centroid is None (i.e., no previous frame), it initializes prev\_centroid and prev\_bbox and returns "approaching".
    - Computes the change in x (dx) and y (dy) between the current and previous centroids.
    - If both dx and dy are positive, it returns "approaching", indicating the object is moving towards the camera.
    - Otherwise, it returns "moving away", indicating the object is moving away from the camera.
  + **Updates**: Updates prev\_centroid and prev\_bbox with the current values for future calculations.
* **Returns**: "approaching" or "moving away" based on the direction of movement.

**2. capture\_frames()**

* **Purpose**: Continuously captures frames from the camera and updates the global frame variable.
* **Logic**:
  + Uses a while True loop to keep capturing frames indefinitely.
  + **Thread Safety**: Utilizes frame\_lock to ensure that only one thread accesses the frame variable at a time, preventing data corruption.
  + Captures a frame from the camera using picam2.capture\_array() and updates frame.
  + **Frame Rate**: Introduces a delay of 0.03 seconds (approximately 30 FPS) between captures to manage the capture rate.
* **Returns**: None. It
* continuously updates the frame variable.

**3. control\_door(command)**

* **Purpose**: Controls the door state by activating/deactivating the LEDs and buzzer based on the command ("open" or "close").
* **Parameters**:
  + command: A string indicating the desired door action ("open" or "close").
* **Logic**:
  + **Open Door**: If the command is "open" and the door is not already open:
    - Turns on the green LED.
    - Turns off the red LED.
    - Activates the buzzer for 5 seconds to signal the door opening.
    - Updates the door\_state to "open".
  + **Close Door**: If the command is "close" and the door is not already closed:
    - Turns off the green LED.
    - Turns on the red LED.
    - Deactivates the buzzer.
    - Updates the door\_state to "close".
* **Returns**: None.

**4. control\_detection\_led(action)**

* **Purpose**: Manages the yellow LED to indicate whether detection is active.
* **Parameters**:
  + action: A string that specifies the LED action ("detect" or any other value to turn off the LED).
* **Logic**:
  + **Detect**: Turns on the yellow LED if the action is "detect".
  + **Off**: Turns off the yellow LED for any other action.
* **Returns**: None.

**5. control\_standby\_led(action)**

* **Purpose**: Manages the blue LED to indicate standby mode.
* **Parameters**:
  + action: A string that specifies the LED action ("standby" or any other value to turn off the LED).
* **Logic**:
  + **Standby**: Turns on the blue LED if the action is "standby".
  + **Off**: Turns off the blue LED for any other action.
* **Returns**: None.

**6. Main Loop**

* **Purpose**: Processes frames from the camera, performs object detection, and controls the door and LEDs based on detection results.
* **Logic**:
  + Checks if frame is None (indicating no frame available yet).
  + Copies the frame for processing to avoid race conditions with the capture\_frames thread.
  + **Frame Skipping**: Only processes every 5th frame to balance performance.
  + **Image Flipping**: Flips the image horizontally for a mirrored view.
  + **Object Detection**:
    - Uses the YOLO model to predict objects in the frame.
    - **Error Handling**: Catches and prints any errors from the model prediction.
  + **Detection Processing**:
    - Processes detected bounding boxes to determine object class and movement direction.
    - Calls control\_door and control\_detection\_led based on object detection results and movement direction.
    - Draws bounding boxes and labels on the image using OpenCV and cvzone.
  + **LED and Door Control**:
    - If no person is detected, turns off the detection LED, closes the door, and turns on the standby LED.
    - If a person is detected, turns off the standby LED.
  + **Display**: Shows the processed image in a window.
  + **Exit Condition**: Allows the program to exit when the 'q' key is pressed.
* **Returns**: None.

**7. Cleanup**

* **Purpose**: Ensures that resources are properly released when the program exits or is interrupted.
* **Logic**:
  + Closes all OpenCV windows.
  + Stops the camera.
  + Calls control\_door, control\_detection\_led, and control\_standby\_led to reset their states.
  + Turns off all LEDs and buzzer.
* **Returns**: None.