1. **Chương trình con xử lý ảnh:**

**1. create\_grid\_mask(vertical, horizontal)**

* Tác dụng: Hàm này tạo một mặt nạ để loại bỏ các đường lưới không cần thiết khỏi ảnh.
* Cách thực hiện:

- Kết hợp các đường đứt ngang và dọc để tạo ra lưới.

- Áp dụng ngưỡng động và làm đậm để che phủ nhiều diện tích hơn.

- Sử dụng biến đổi Hough để tìm các đường thẳng trên lưới.

- Vẽ đường lưới và trả về mặt nạ chỉ chứa các số còn lại trên ảnh.

**2. get\_grid\_lines(img, length=10)**

* Tác dụng: Hàm này tìm các đường đứt ngang và đứt dọc trên ảnh.
* Cách thực hiện:

- Sử dụng hàm hỗ trợ ***“grid\_line\_helper”*** để tìm các đường đứt ngang và đứt dọc với chiều dài đã cho.

**3. find\_contours(img, original)**

* Tác dụng: Hàm này tìm đường viền của vùng quan tâm trên ảnh nhị phân.
* Cách thực hiện:

- Sử dụng ***“cv2.findContours”*** để tìm các đường viền trong ảnh nhị phân.

- Sắp xếp các đường viền theo diện tích giảm dần.

- Tìm đa giác đơn có 4 góc và diện tích lớn nhất, xác định vùng quan tâm.

**4. warp\_image(corners, original)**

* Tác dụng: Hàm này chuyển đổi ảnh thành một ảnh vuông dựa trên các điểm góc.
* Cách thực hiện:

- Xác định chiều rộng của ảnh vuông dựa trên khoảng cách giữa các điểm góc.

- Tính ma trận chuyển đổi perspective và thực hiện chuyển đổi perspective.

**5. split\_into\_squares(warped\_img)**

* Tác dụng: Hàm này chia ảnh vuông thành các ô vuông nhỏ.
* Cách thực hiện: Duyệt qua từng ô vuông và thêm nó vào danh sách ***“squares”***.

**6. clean\_squares(squares)**

* Tác dụng: Hàm này làm sạch từng ô vuông trong danh sách.
* Cách thực hiện: Sử dụng hàm ***“clean\_helper”*** từ module ***“process\_helpers”*** để làm sạch từng ô vuông và xác định xem nó có phải là một số hay không.

**7. recognize\_digits(squares\_processed, model)**

* Tác dụng: Hàm này nhận diện các chữ số trong các ô vuông đã được xử lý.
* Cách thực hiện:

- Chuẩn hóa kích thước của từng ô vuông.

- Sử dụng mô hình để dự đoán chữ số trong mỗi ô vuông.

**8. draw\_digits\_on\_warped(warped\_img, solved\_puzzle, squares\_processed)**

* Tác dụng: Hàm này vẽ chữ số đã giải lên ảnh đã chuyển đổi.
* Cách thực hiện:

- Duyệt qua từng ô vuông và vẽ chữ số đã giải lên ảnh chuyển đổi.

**9. unwarp\_image(img\_src, img\_dest, pts, time)**

* Tác dụng: Hàm này hủy chuyển đổi perspective của ảnh và kết hợp với ảnh gốc.
* Cách thực hiện:

- Tính ma trận chuyển đổi perspective và thực hiện hủy chuyển đổi perspective.

- Kết hợp ảnh đã hủy chuyển đổi perspective với ảnh gốc.

- Thêm thông tin thời gian vào ảnh.

1. **Chương trình con giải bài toán Sudoku**

**1. solve\_sudoku(size, grid)**

* Tác dụng: Hàm này giải một bảng Sudoku có kích thước và trạng thái khởi đầu cho trước.
* Cách thực hiện:

- Sử dụng thuật toán chia và trị để giải Sudoku.

- Tạo các điều kiện cần thiết cho việc giải, sau đó gọi hàm solve để tìm nghiệm.

**2. exact\_cover(X, Y)**

Tác dụng: Hàm này chuyển đổi mô hình chính xác từ dạng ma trận sang danh sách và từ điều kiện sang tập dữ liệu.

Cách thực hiện: Chuyển đổi dữ liệu từ dạng ma trận sang dạng danh sách, tạo X và Y.

**3. solve(X, Y, solution)**

* Tác dụng: Hàm này thực hiện thuật toán Dancing Links để giải bài toán chính xác che phủ.
* Cách thực hiện:

- Chọn một cột có số lượng lựa chọn nhỏ nhất.

- Dùng đệ quy để thử tất cả các lựa chọn có thể và tiếp tục với các lựa chọn hợp lý.

**4. select(X, Y, r)**

* Tác dụng: Hàm này chọn một dòng và loại bỏ các dòng và cột có liên quan.
* Cách thực hiện:

- Chọn một dòng và loại bỏ các dòng và cột liên quan khỏi mô hình.

**5. deselect(X, Y, r, cols)**

* Tác dụng: Hàm này hủy chọn một dòng và khôi phục các dòng và cột đã loại bỏ.
* Cách thực hiện: Hủy chọn một dòng và khôi phục các dòng và cột đã loại bỏ.

**6. solve\_wrapper(squares\_num\_array)**

* Tác dụng: Hàm này làm giao diện giữa giải thuật giải Sudoku và ứng dụng cụ thể.
* Cách thực hiện:

- Chuyển đổi dữ liệu đầu vào từ chuỗi sang mảng 2 chiều.

- Gọi hàm giải Sudoku và trả về kết quả.

1. **Chương trình chính**

**1. Khởi tạo và cấu hình camera:**

* ***“cap = cv2.VideoCapture(0)”:*** Tạo đối tượng VideoCapture để kết nối với camera.
* ***“cap.set(3, frameWidth)***” và ***“cap.set(4, frameHeight)”:*** Thiết lập kích thước khung hình (chiều rộng và chiều cao) cho video stream.
* ***“cap.set(10, 150)”:*** Thiết lập độ sáng của camera.

**2. Tải mô hình nhận diện số từ hình ảnh:**

***“my\_model = model\_wrapper.model\_wrapper(None, False, None, "model.hdf5")”:*** Tạo mô hình từ module `model\_wrapper` để nhận diện số trong các ô của Sudoku. Truyền vào đường dẫn của mô hình đã được huấn luyện.

**3. Vòng lặp chính xử lý video stream:**

* ***“time\_elapsed = t.time() – prev”:*** Đo thời gian giữa các frame để đảm bảo frame rate.
* ***“success, img = cap.read()”:*** Đọc frame từ video stream.

**4. Xử lý và nhận diện Sudoku từ frame:**

***“processed\_img = preprocess.preprocess(img)”:*** Tiền xử lý ảnh để chuẩn bị cho quá trình nhận diện Sudoku.

***“corners = process.find\_contours(processed\_img, img\_corners)”:*** Tìm các đỉnh của Sudoku trong ảnh sử dụng hàm `find\_contours`.

***“warped, matrix = process.warp\_image(corners, img)”:*** Warp (biến đổi) ảnh để có góc nhìn phẳng và trả về ma trận biến đổi.

***“vertical\_lines, horizontal\_lines = process.get\_grid\_lines(warped\_processed)”:*** Tìm các đường lưới trong ảnh đã biến đổi.

***“mask = process.create\_grid\_mask(vertical\_lines, horizontal\_lines)”:*** Tạo mặt nạ để tập trung vào vùng chứa số.

***“numbers = cv2.bitwise\_and(warped\_processed, mask)”:*** Áp dụng mặt nạ để lấy các vùng chứa số.

***“squares = process.split\_into\_squares(numbers)”:*** Tách ảnh thành các ô con chứa số.

***“squares\_processed = process.clean\_squares(squares)”:*** Làm sạch và chuẩn bị các ô con để nhận diện số.

***“squares\_guesses = process.recognize\_digits(squares\_processed, my\_model)”:*** Nhận diện số trong các ô con và trả về dạng chuỗi biểu diễn giả định của Sudoku.

**5. Hiển thị kết quả và giải Sudoku:**

* Kiểm tra xem Sudoku đã được giải trước đó hay chưa và cập nhật kết quả dựa trên điều kiện đó.
* Nếu Sudoku đã được giải, hiển thị kết quả đã giải và thời gian giải.
* Nếu Sudoku chưa được giải, thực hiện giải Sudoku và lưu kết quả.
* ***“cv2.imshow('window', img\_result)”:*** Hiển thị video stream với kết quả Sudoku đã được giải hoặc giả định.
* ***“cv2.waitKey(1) & 0xFF == ord('q')”:*** Kiểm tra xem có phím tắt 'q' được nhấn hay không để kết thúc chương trình.

**6. Dọn dẹp và giải phóng tài nguyên:**

***“cv2.destroyAllWindows()”*** và ***“cap.release()”:*** Đóng tất cả các cửa sổ hiển thị và giải phóng tài nguyên camera.