import cv2

import numpy as np

# MJPEG stream URL

stream\_url = "http://192.168.184.6:5000/video\_feed"

# Open the video stream

cap = cv2.VideoCapture(stream\_url)

# Create CLAHE object for contrast enhancement

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))

while True:

ret, frame = cap.read()

if not ret:

print("Failed to grab frame from stream.")

break

frame = cv2.resize(frame, (640, 480))

# Convert to grayscale

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# Apply CLAHE to enhance contrast

enhanced = clahe.apply(gray)

# Gaussian blur to reduce noise

blurred = cv2.GaussianBlur(enhanced, (5, 5), 0)

# Edge detection

edges = cv2.Canny(blurred, 50, 150)

# Line detection using Hough Transform

lines = cv2.HoughLinesP(edges, 1, np.pi/180, threshold=100, minLineLength=50, maxLineGap=20)

if lines is not None:

for line in lines:

x1, y1, x2, y2 = line[0]

cv2.line(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)

# Find contours for walls or large objects

contours, \_ = cv2.findContours(edges.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for contour in contours:

if cv2.contourArea(contour) > 500:

cv2.drawContours(frame, [contour], -1, (0, 255, 0), 2)

# Show processed output

cv2.imshow("Processed Frame", frame)

cv2.imshow("Edges", edges)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()