import cv2

import numpy as np

def find\_squares(frame, min\_area, max\_area):

# Convertir el cuadro a escala de grises

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

# Aplicar deteccion de bordes y encontrar contornos

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

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

squares = []

for contour in contours:

# Aproximar el contorno

approx = cv2.approxPolyDP(contour, 0.02 \* cv2.arcLength(contour, True), True)

# Verificar si el contorno tiene 4 lados (es cuadrado o rectangular)

if len(approx) == 4 and cv2.isContourConvex(approx):

# Calcular el area del contorno

area = cv2.contourArea(contour)

# Verificar si esta dentro de los limites

if min\_area <= area <= max\_area:

# Verificar si tiene proporciones de un cuadrado

(x, y, w, h) = cv2.boundingRect(approx)

aspect\_ratio = float(w) / h

if 0.95 <= aspect\_ratio <= 1.05: # Ajusta segun tu tolerancia

squares.append(approx)

# Dibujar los cuadrados detectados

cv2.drawContours(frame, squares, -1, (0, 255, 0), 3)

return frame