import cv2

import numpy as np

def detect\_watch\_color(image\_path, box):

image = cv2.imread(image\_path)

x1, y1, x2, y2 = map(int, box)

cropped = image[y1:y2, x1:x2]

# Convert to HSV color space

hsv = cv2.cvtColor(cropped, cv2.COLOR\_BGR2HSV)

# Threshold for detecting black color

black\_mask = cv2.inRange(hsv, (0, 0, 0), (180, 255, 50)) # Low value, low saturation

# Threshold for detecting silver color (grayish with high brightness and low saturation)

silver\_mask = cv2.inRange(hsv, (0, 0, 150), (180, 50, 255)) # Higher brightness, low saturation

total\_pixels = cropped.shape[0] \* cropped.shape[1]

# Calculate the ratio of pixels that are black or silver

black\_ratio = cv2.countNonZero(black\_mask) / total\_pixels

silver\_ratio = cv2.countNonZero(silver\_mask) / total\_pixels

# Adjust the threshold for detection; if black or silver ratio is greater than 10% of the total pixels

if black\_ratio > silver\_ratio and black\_ratio > 0.1:

return 'black'

elif silver\_ratio > black\_ratio and silver\_ratio > 0.1:

return 'silver'

else:

return 'unknown'