Crime forecasting: a machine learning and computer vision approach crime prediction

Code for fingerprint detection:

import os

import cv2

sample = cv2.imread("SOCOFing/SOCOFing/150\_\_M\_Right\_index\_finger\_CR.BMP")

best\_score = 0

filename = None

image = None

kp1, kp2, mp = None, None, None

counter = 0

for file in [file for file in os.listdir("SOCOFing/Real")][:1000]:

if counter % 10 == 0:

print(counter)

print(file)

counter += 1

fingerprint\_image = cv2.imread("SOCOFing/Real/" + file)

sift = cv2.SIFT\_create()

keypoints\_1, descriptors\_1 = sift.detectAndCompute(sample, None)

keypoints\_2, descriptors\_2 = sift.detectAndCompute(fingerprint\_image, None)

matches = cv2.FlannBasedMatcher({'algorithm': 1, 'trees': 10},

{}).knnMatch(descriptors\_1, descriptors\_2, k=2)

match\_points = []

for p, q in matches:

if p.distance < 0.1 \* q.distance:

match\_points.append(p)

keypoints = 0

if len(keypoints\_1) < len(keypoints\_2):

keypoints = len(keypoints\_1)

else:

keypoints = len(keypoints\_2)

if len(match\_points) / keypoints \* 100 > best\_score:

best\_score = len(match\_points) / keypoints \* 100

filename = file

image = fingerprint\_image

kp1, kp2, mp = keypoints\_1, keypoints\_2, match\_points

print("BEST MATCH: " + filename)

print("SCORE: " + str(best\_score))

result = cv2.drawMatches(sample, kp1, image, kp2, mp, None)

result = cv2.resize(result, None, fx=4, fy=4)

cv2.imshow("Result", result)

cv2.waitKey(0)

cv2.destroyAllWindows()