STB600 Lab 6: Image pattern recognition

Atieh Sahraeidolatkhaneh: atieh.sahraeidolatkhaneh@hv.se

Yongcui Mi: yongcui.mi@hv.se

1. Task 1: Image recognition using template matching

Template Matching is a method for searching and finding the location of a template image in a larger image. It can be used for various tasks, such as object recognition and detection, video tracking, and image recognition. OpenCV comes with a function cv.matchTemplate() for this purpose. It simply slides the template image over the input image (as in 2D convolution) and compares the template and patch of input image under the template image. It returns a grayscale image, where each pixel denotes how much does the neighbourhood of that pixel match with template.

Several comparison methods are implemented in OpenCV. `cv2.TM_CCOEFF_NORMED` and `cv2.TM_CCORR_NORMED` methods are the most widely used. `cv2.TM_CCOEFF_NORMED` computes the correlation coefficient between the template and the target image. The correlation coefficient is a measure of the similarity between the two images. The resulting values are normalized between 0 and 1, with 0 indicating no match and 1 indicating a perfect match. `cv2.TM_CCORR_NORMED` computes the cross-correlation between the template and the target image. The cross-correlation is a measure of how similar the two images are and the resulting values are normalized between 0 and 1.

In this task, a template image is given as shown in Fig. 3 and several test images are also given as shown in Fig 4. Search in the given test images and recognize footballs in these images. If a football is identified, frame it with a rectangle with the same size as the template image. Otherwise, indicate in the image that no football is found.

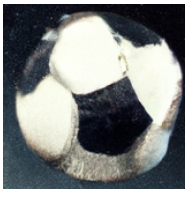

Fig. 1. Template image

Fig. 2. Test image

```
#
#Load the template and test images
# Perform template matching using the cv2.matchTemplate() method
# Use the cv2.minMaxLoc() method to find the location with the highest
    similarity score
# Use the cv2.rectangle() method to draw a rectangle if a match was
    found.
# Use cv2.putText() to indicate that no match was found.
```

2. Task 2: Detect multiple objects inside Image with different rotations

In the first task, you did a basic template matching. To detect multiple objects or templates you need to filter the result matrix generated by cv2.matchTemplate. If you are going to detect multiple objects, you need to filter the result matrix and find all (x, y)-coordinates that have a score greater than a threshold. Try to find all the matching templates with different orientation inside the test image and draw rectangle on the detected objects.

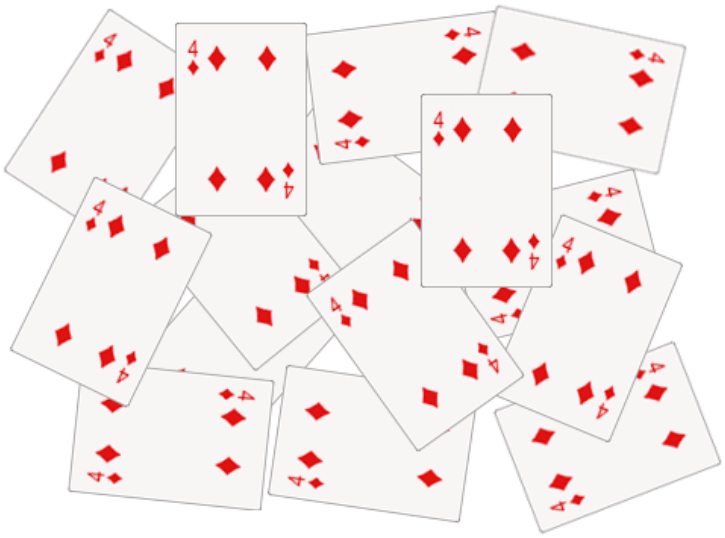

Fig. 3. Test image

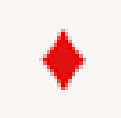

Fig. 4. Template image

```

# Use 'np.where' to find all (x,y) coordinates when correlation score is
    greater than threshold value

# loop over the final bounding boxes and draw the bounding box on the
   image
```

3. Additional resource

References