# 模板匹配

## 目标：

* 使用模板匹配查找图像中的对象
* 函数：cv.matchTemplate()，cv.minMaxLoc()

## 理论

模板匹配是一种在较大图像中搜索和查找模板图像位置的方法。为此，OpenCV附带了一个函数cv.matchTemplate()。它只是在输入图像上滑动模板图像（如在2D卷积中），并比较模板图像下的输入图像的模板和补丁。在OpenCV中实现了几种比较方法。 （你可以查看文档以获取更多详细信息）。它返回一个灰度图像，其中每个像素表示该像素的邻域与模板匹配的程度。

如果输入图像的大小（WxH）且模板图像的大小（wxh），则输出图像的大小为（W-w + 1，H-h + 1）。获得结果后，可以使用cv.minMaxLoc()函数查找最大/最小值的位置。将其作为矩形的左上角，并将（w，h）作为矩形的宽度和高度。那个矩形是你的模板区域。

注意：如果你使用cv.TM\_SQDIFF作为比较方法，则最小值会给出最佳匹配。

## OpenCV中的模板匹配

## 实例1，匹配单个对象

|  |
| --- |
| **import** cv2 **import** numpy **as** np **from** matplotlib **import** pyplot **as** plt *# 单模板匹配* img = cv2.imread(**'../mydata/lena.jpg'**,0) *# 填写图片名称* template = cv2.imread(**'../mydata/template.jpeg'**,0) target2 = img.copy() *# 获取template宽高* w, h = template.shape[::-1]  *# All the 6 methods for comparison in a list ,注意:cv2.TM\_CCORR效果不好* methods = [**'cv2.TM\_CCOEFF'**, **'cv2.TM\_CCOEFF\_NORMED'**, **'cv2.TM\_CCORR'**,  **'cv2.TM\_CCORR\_NORMED'**, **'cv2.TM\_SQDIFF'**, **'cv2.TM\_SQDIFF\_NORMED'**] **for** meth **in** methods:  target = target2.copy()  method = eval(meth)   *# Apply template Matching* res = cv2.matchTemplate(target,template,method)  *# cv2.normalize(res, res, 0, 1, cv2.NORM\_MINMAX, -1)  # 查找最大值和最小值* min\_val,max\_val,min\_loc,max\_loc = cv2.minMaxLoc(res)  *# 不同的匹配方式有不同的处理方法* **if** method **in** [cv2.TM\_SQDIFF, cv2.TM\_SQDIFF\_NORMED]:  top\_left = min\_loc  **else**:  top\_left = max\_loc  bottom\_right = (top\_left[0]+w,top\_left[1]+h)  cv2.rectangle(target,top\_left,bottom\_right,255,3)  plt.subplot(121), plt.imshow(res, cmap=**'gray'**)  plt.title(**'Matching Result'**), plt.xticks([]), plt.yticks([])  plt.subplot(122), plt.imshow(target, cmap=**'gray'**)  plt.title(**'Detected Point'**), plt.xticks([]), plt.yticks([])  plt.suptitle(meth)  plt.show()  cv2.waitKey(0) cv2.destroyAllWindows() |

### 效果：（会弹出6次窗口）

|  |
| --- |
| * cv.TM\_CCOEFF   [image62](https://camo.githubusercontent.com/a24c2422671fb6ce883517c2dc0286a4f08c8cfc285e9fecee8a45f91c2e4a02/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f63636f6566665f312e6a7067)   * cv.TM\_CCOEFF\_NORMED   [image63](https://camo.githubusercontent.com/59e4503edb145d3c01f4165e1170c5d4aa92881a01fc34235433bbfb0c55be39/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f63636f6566666e5f322e6a7067)   * cv.TM\_CCORR   [image64](https://camo.githubusercontent.com/e74393c2c5c49b24dfa16987270514db58cbb9423e7506328da2327541ac56e5/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f63636f72725f332e6a7067)   * cv.TM\_CCOEFF\_NORMED   [image65](https://camo.githubusercontent.com/3d65b41d13a72129828c9600bec0f27e59e406f067b643c516d16af40f15334e/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f63636f72726e5f342e6a7067)   * cv.TM\_SQDIFF   [image66](https://camo.githubusercontent.com/9151a72f988392c7f65651ac93b2e1893e5b190189489ba278d150c7ec15f816/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f7371646966665f352e6a7067)   * cv.TM\_SQDIFF\_NORMED   [image67](https://camo.githubusercontent.com/559ee9307a010ffefe442718ec4ba91993886416a202df5492dea5c11b94ea82/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f74656d706c6174655f7371646966666e5f362e6a7067) |

**你可以看到使用cv.TM\_CCORR的结果不如我们预期的那样好。**

## 与多个对象匹配的模板

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |