Edge detection: active Gaussian thresholding

cv2.adaptiveThreshold(target\_gray, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY, 11, 2)

Template matching method: correlation coefficient

result = cv2.matchTemplate(edged, target\_edge, cv2.TM\_CCOEFF)

Scaling numbers: 0.5, 2.0, 10

for scale in np.linspace(0.5, 2.0, 10)[::-1]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0.5, 2.0, 10 | **Waldo** | **Wenda** | **Wizard** | **Odlaw** | **Woof** |
| **City** | Found | Found | Found | Found | X |
| **Beach** | Odlaw | Found | Found | Found | X |
| **Zoo** | Found | Found | Found | Found | X |
| **Store** | Odlaw | Waldo | ? | Found | X |
| **Ski Resort** | Odlaw | Found | Found | Found | X |
| **Train Station** | Odlaw | Found | Found | Found | X |
| **Museum** | Found | Found | Found | Found | X |

These specs with 10 scales between 0.5 and 2.0 worked pretty well for Waldo’s friends, except for Woof. Searching for Waldo still finds Odlaw instead sometimes. The ‘?’ for the Wizard at the store threw an error (error: (-215:Assertion failed) !\_src.empty() in function 'cvtColor') not sure why.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0.6, 1.2, 7 | **Waldo** | **Wenda** | **Wizard** | **Odlaw** | **Woof** |
| **City** | Found | Found | Found | Found | X |
| **Beach** | X | Found | X | X | X |
| **Zoo** | X | Found | Found | Found | X |
| **Store** |  |  |  |  | X |
| **Ski Resort** |  |  |  |  | X |
| **Train Station** |  |  |  |  | X |
| **Museum** |  |  |  |  | X |
|  |  |  |  |  |  |
| 0.7, 1.3, 3 | **Waldo** | **Wenda** | **Wizard** | **Odlaw** | **Woof** |
| **City** | Found | X | Found | Found | X |
| **Beach** | X | Found | X | Wenda | X |
| **Zoo** | X | Found | Found | X | X |
| **Store** | X | Odlaw | X | X | X |
| **Ski Resort** | Odlaw | Found | Found | Found | X |
| **Train Station** | Odlaw | Found | X | Found | X |
| **Museum** | X | Found | Found | Found | X |

The search with these numbers was much faster but less accurate than the top numbers. No errors but many times the “best result” was not even close to the target image.