BÁO CÁO BÀI TẬP FIND & COUNT SỬ DỤNG TEMPLATE MATCHING

Lớp: Xử lý và phân tích hình ảnh

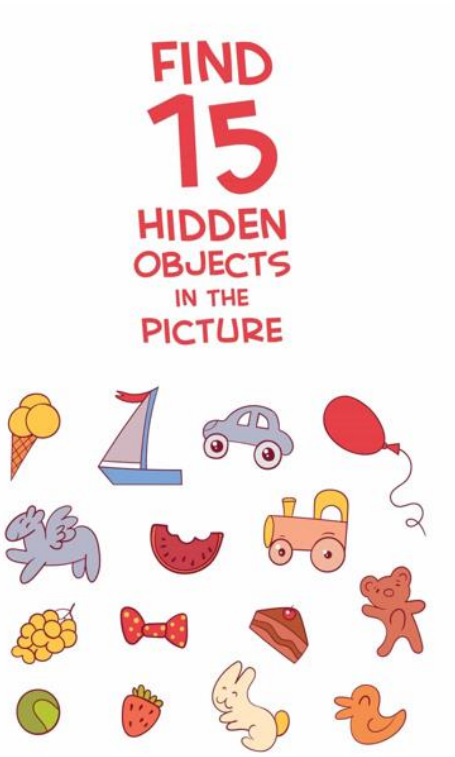
Họ và tên : Vũ Trung Hiếu - 22022515

Mục Lục

I. Giới thiệu

1. Bài toán

Sử dụng Template Matching để đếm về tìm kiếm các vật thể, đối tượng trong hình ảnh



2. Kỹ thuật sử dụng

Sản phẩm sử dụng 1 số kỹ thuật chính, hàm có sẵn trong thư viện cv2

- Template Matching: cv2.matchTemplate (normalized correlation coefficient)
- Edge Detection: cv2.canny
- Filtering: cv2.GaussianBlur, cv2.threshold

II. Counting

1. Quy trình chung

1. Load các ảnh, template và chuyển về dạng gray
2. Xử lý
    a) Edge detection , convert to binary image
    b) Flip, rotate template, resize
3. Template matching
4. Giảm số lượng bounding box

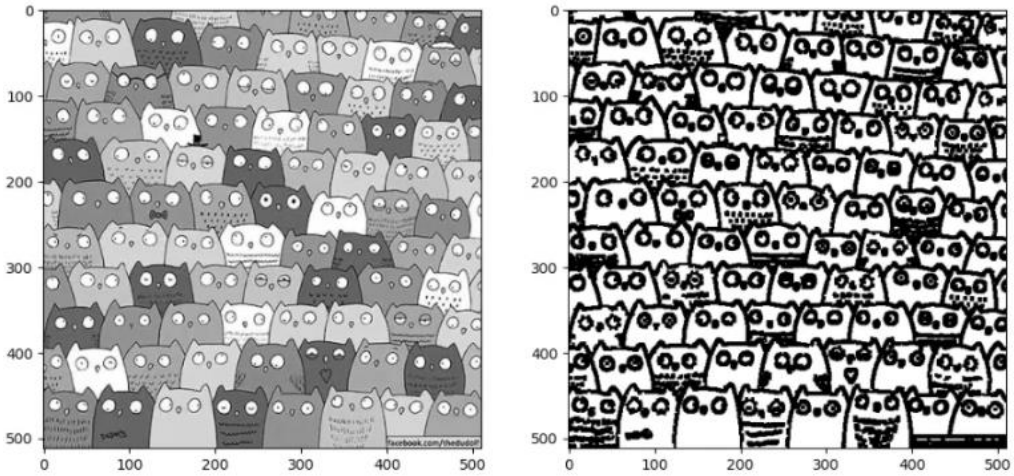
2. Xử lý

a) Edge detection, convert to binary image

Sử dụng cv2.canny nhằm phát hiện các cạnh

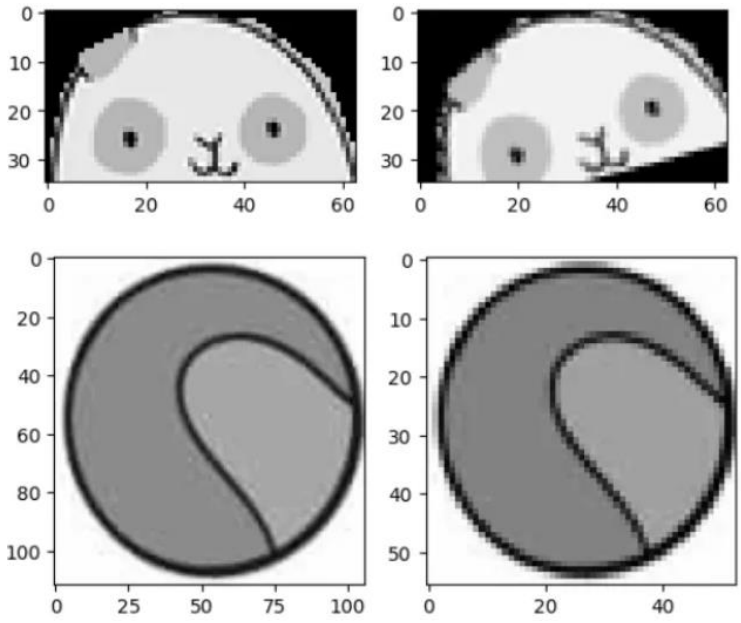
Sử dụng cv2.GaussianBlur để làm mượt các cạnh

Sử dụng cv2.threshold để chuyển hình ảnh về dạng binary (chỉ có 2 giá trị điểm ảnh: 0 và 255)



b) Flip, Rotate và Resize

Xây dựng các hàm để thay đổi template như lật, xoay, và thay đổi kích thước



3. Template Matching

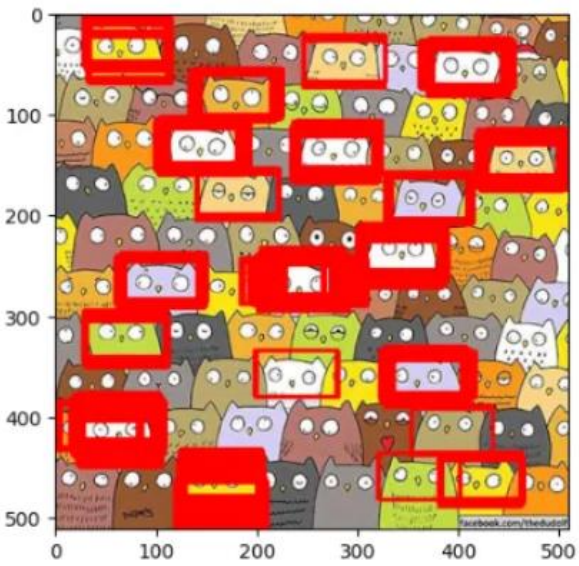
Dùng các công cụ xử lý đã được xây dựng (Flip, Rotate, Resize), thực hiện thay đổi các template. Sử dụng Normalized Cross Correlation Coefficient để phát hiện đối tượng

```
res = cv2.matchTemplate(image_gray, templates_gray[1], cv2.TM_CCOEFF_NORMED)
threshold = sorted(res.flatten())[int(0.99 * len(res.flatten()))]
loc = np.where(res >= threshold)
x, y = loc[::-1]
res_image = image_rgb.copy()
w, h = templates_gray[0].shape[::-1]
for bb in zip(*loc[::-1]):
    cv2.rectangle(res_image, bb, (bb[0] + w, bb[1] +h), (0,0,255), 2)
plt.imshow(res_image[:, :, ::-1])
```

4. Giảm số lượng bounding box

Nhận thấy có nhiều hình ảnh (đối tượng) ko phát hiện được, đi kèm với đó sự trùng lặp của các bounding box

⇒ phải kết hợp nhiều template và loại bỏ các bounding box trùng nhau



Thuật toán:

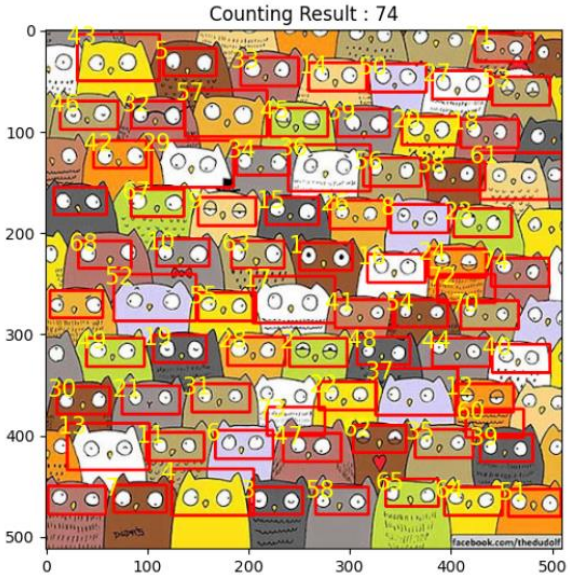
- Bước 1: Tạo 1 list các bounding box (bboxes) với các phần tử có dạng (score, x, y, w, h)
  Trong đó
  - score: giá trị ở trong matching map
  - x, y: tọa độ
  - w, h: kích thước bounding box
- Bước 2: Thực hiện tính toán lại score của các bounding box: Lấy trung bình của score và normalized square error
  Do khi hình ảnh được chuyển về dạng binary (chỉ chứa các điểm ảnh có giá trị 0 và 256), việc sử dụng template matching để tìm kiếm vật thể gây ra việc phát hiện

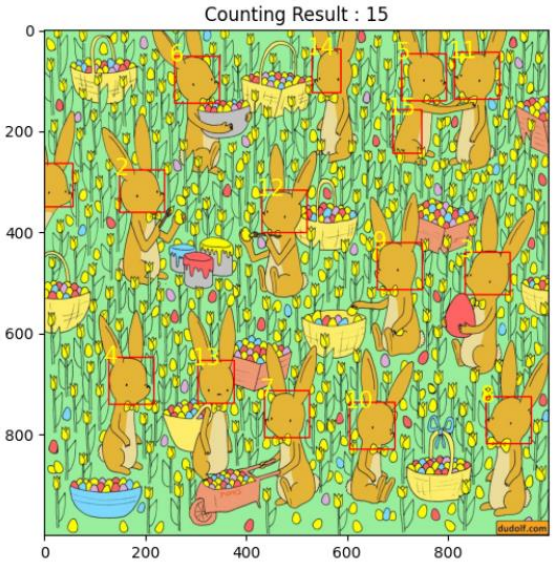
1 số đối tượng có màu khác. Vì vậy cần thực hiện tính toán để giảm score của các đối tượng bị lệch đã được nêu trên.

- Bước 3: Sắp xếp bboxes, lấy phần trăm hoặc số lượng phần tử nhất định (ví dụ: lấy 100 bounding box có chỉ số cáo nhất)

- Bước 4: Lấy dần các bounding box có điểm số cao, khi xét 1 bounding box mới sẽ kiểm tra xem có bị trùng lặp với các bounding box đã được xác định

5. Kết quả



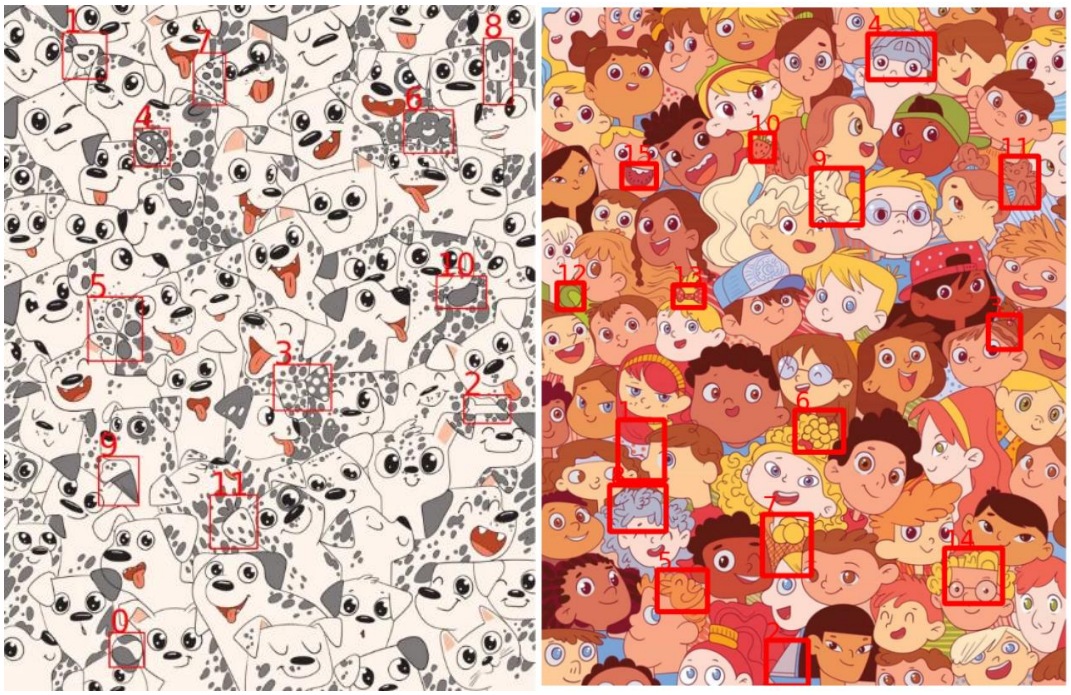


III. Finding

1. Thuật toán

- Load hình ảnh và các template
- Với mỗi loại template
  - Thực hiện resize template theo tỷ lệ từ 0.5 đến 1 (0.5, 0.55, 0.6, …1)
  - Sử dụng Template Matching với phương pháp normalized cross correlation
  - Tìm kiếm tỷ lệ resize thu được điểm số cao nhất
  - Đánh giấu vị trí trong hình ảnh tương đương với điểm số cao nhất đó
- Vẽ hình ảnh với các bounding box của các vật thể cần tìm kiếm

2. Kết quả


1
8
7
6
4
10
5
3
2
9
11
0
4
10
15
9
11
12
13
3
6
7
14
5