

Báo cáo bài tập week5

Phát hiện đối tượng trong ảnh bằng OpenCV

Giới thiệu

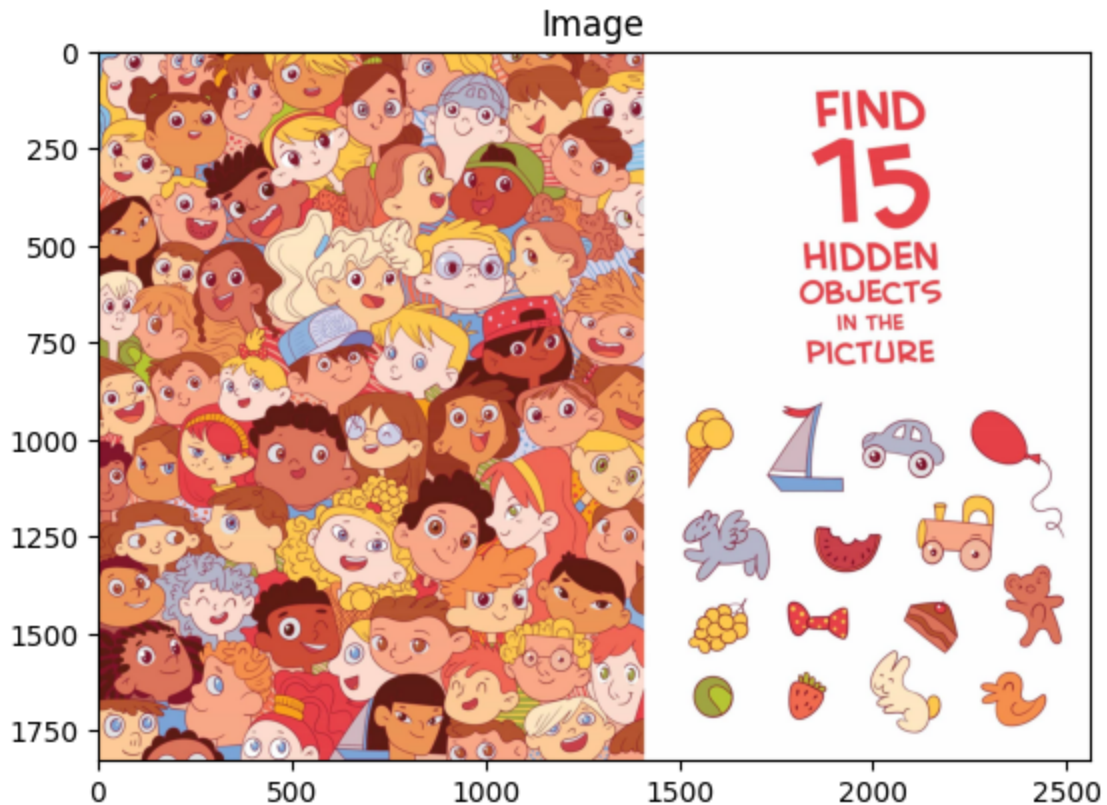
Bài toán phát hiện đối tượng trong ảnh yêu cầu tìm vị trí của một đối tượng nhất định trong một bức ảnh lớn hơn. Trong báo cáo này, chúng ta sẽ sử dụng thư viện OpenCV để phát hiện các đối tượng bằng phương pháp **template matching** (so khớp mẫu), với các bước chính gồm chuyển đổi ảnh sang thang xám, nhị phân hóa và phát hiện biên.

Mô tả phương pháp

1. Chuẩn bị ảnh và mẫu:

- Đầu tiên, chúng ta đọc ảnh đầu vào và hiển thị nó bằng thư viện `matplotlib`. Sau đó, một mẫu đối tượng cần tìm (ví dụ: bóng tennis hoặc quả dâu tây) được cắt từ ảnh gốc dựa trên tọa độ x, y và kích thước w, h .

```
image = cv2.imread(image_path)
template_tennisball = image[y:y+h, x:x+w]
```



2. Nhị phân hóa ảnh mẫu:

- Mẫu đối tượng sau khi cắt sẽ được chuyển đổi sang ảnh xám và tiếp tục nhị phân hóa (binary) bằng phương pháp ngưỡng (`thresholding`) để dễ dàng xử lý và so sánh.

```
gray_template_tennisball = cv2.cvtColor(template_tennisball, cv2.COLOR_BGR2GRAY)
_, thresh_template = cv2.threshold(gray_template_tennisball, 120, 255, cv2.THRESH_BINARY)
```

Original Template



Binary Template



3. Chuyển đổi ảnh gốc:

- Tương tự, ảnh gốc cũng được chuyển sang dạng ảnh xám và nhị phân hóa để phục vụ quá trình so khớp mẫu.

Original Image



Binary Image



4. Phát hiện đối tượng bằng Template Matching:

- Sau khi đã có mẫu và ảnh gốc nhị phân, chúng ta áp dụng phương pháp so khớp mẫu bằng hàm `cv2.matchTemplate`. Để phát hiện đối tượng ở nhiều kích thước khác nhau, mẫu được thay đổi tỷ lệ trong khoảng từ 0.5 đến 1.0, và kết quả so khớp sẽ được kiểm tra bằng một ngưỡng xác định (threshold).

```

for scale in np.linspace(0.5, 1.0, 100):
    resized_template = cv2.resize(thresh_template, (0, 0), fx=scale, fy=scale)
    result = cv2.matchTemplate(binary_image, resized_template, cv2.TM_CCOEFF_NORMED)
    loc = np.where(result >= threshold)
    for pt in zip(*loc[::-1]):
        cv2.rectangle(image, pt, (pt[0] + int(w * scale), pt[1] + int(h * scale)), (0, 0, 255))

```

Matching areas with threshold 0.6

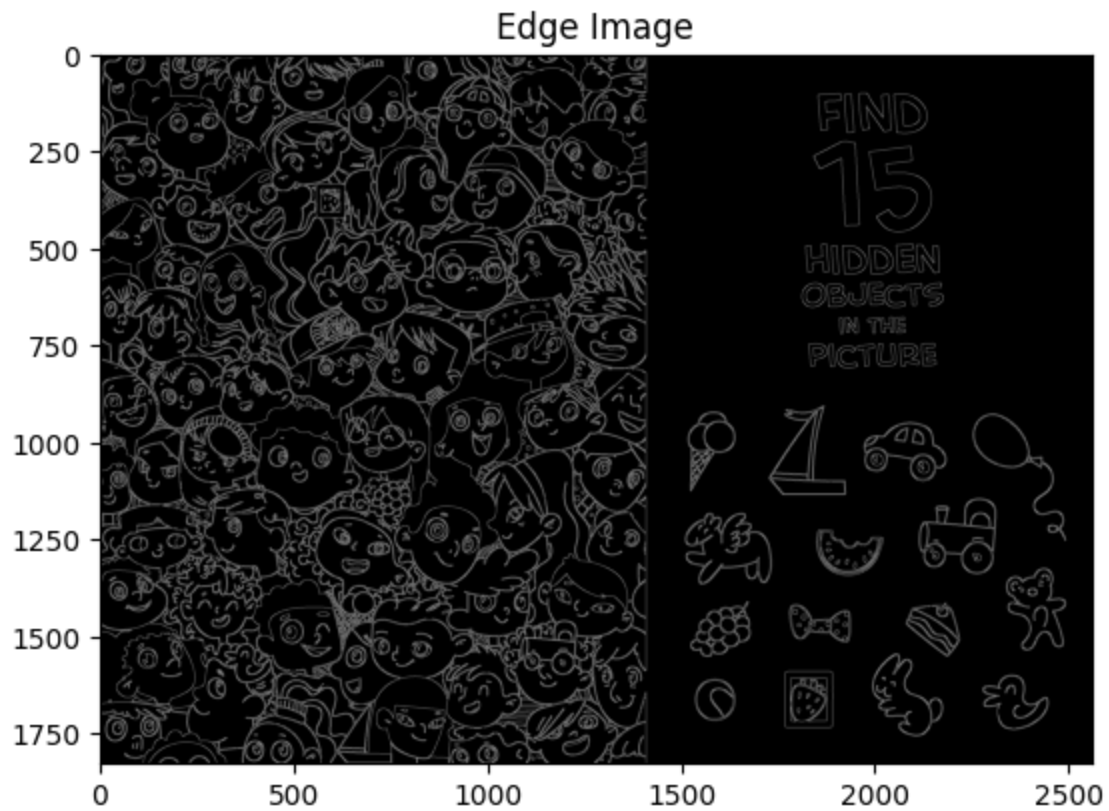


Cách tiếp cận khác: Thay vì chuyển sang nhị phân, ta sử dụng phát hiện cạnh

Phát hiện biên cạnh (Edge Detection):

- Ngoài phương pháp nhị phân hóa, chúng ta còn sử dụng phương pháp phát hiện biên cạnh (Canny Edge Detection) để so khớp các biên của đối tượng. Kỹ thuật này giúp tăng độ chính xác trong các trường hợp đối tượng có hình dạng rõ ràng.

```
edges_image = cv2.Canny(gray_image, 100, 200)
edges_template_tennisball = cv2.Canny(gray_template_tennisball,
```



Original Template



Edge Template



Matching areas with threshold 0.4

