

REPORT COUNTING OBJECTS IN IMAGES (RABBIT)

LÊ TRUNG HIẾU - 22022576

(link code: https://github.com/hieu7404/counting_rabbit)

1. Giới thiệu

Phương pháp sử dụng trong bài này là **Template Matching** kết hợp với **Intersection over Union (IoU)** để đếm số lượng thỏ trong ảnh. Phương pháp này giúp xác định các vùng trong ảnh có chứa đối tượng tương tự với mẫu template đã chọn, đồng thời loại bỏ các kết quả trùng lặp để đảm bảo độ chính xác.

2. Phương pháp

2.1 Template Matching

Template Matching là một kỹ thuật xử lý ảnh dùng để tìm một mẫu nhỏ (template) trong một ảnh lớn hơn (search image). Phương pháp **TM_SQDIFF_NORMED (Template Matching with Normalized Squared Difference)**: là một biến thể của Template Matching dựa trên sự khác biệt bình phương chuẩn hóa. Công thức tính toán:

$$R(x, y) = \sum_{i,j} (T(i, j) - I(x + i, y + j))^2$$

Trong đó:

- $T(i, j)$ là giá trị pixel tại vị trí (i, j) của template.
- $I(x + i, y + j)$ là giá trị pixel tại vị trí $(x + i, y + j)$ của ảnh gốc.
- $R(x, y)$ là giá trị của phép so sánh tại vị trí (x, y) .

Giá trị $R(x, y)$ càng nhỏ thì mẫu càng khớp với vùng ảnh.

2.2 Intersection over Union (IoU)

Sau khi thực hiện Template Matching, ta có thể thu được nhiều vị trí khớp nhau gần nhau, dẫn đến việc đếm sai số lượng thỏ. Do đó, cần sử dụng **Intersection over Union (IoU)** để lọc các vị trí trùng lặp.

2.2.1 Công thức IoU

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Trong đó:

- A và B là hai bounding box của template tại hai vị trí khác nhau.
- $|A \cap B|$ là diện tích phần giao giữa hai bounding box.
- $|A \cup B|$ là diện tích hợp nhất của hai bounding box.

Nếu $IoU > 0.5$, hai bounding box được coi là trùng lặp và chỉ giữ lại một trong số đó.

3. Cài đặt

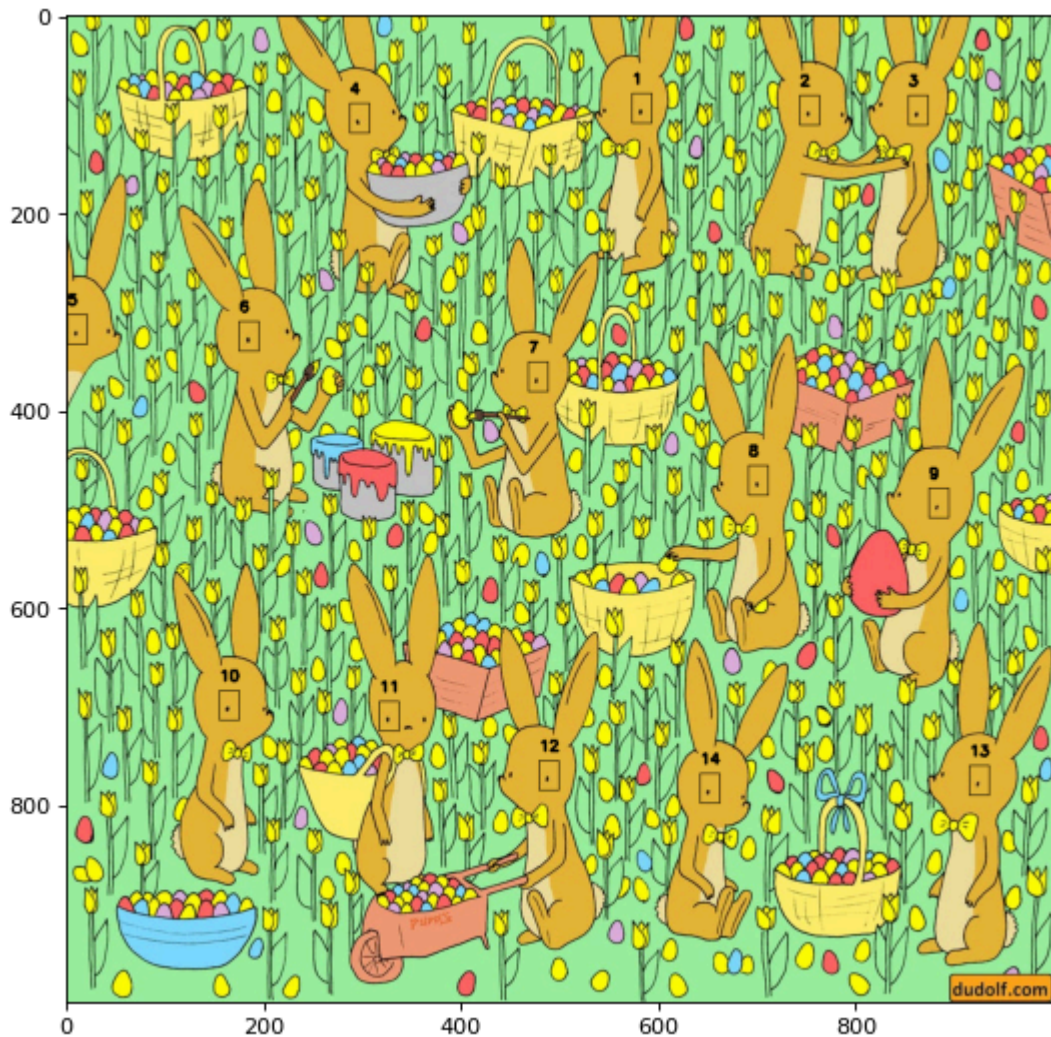
Chương trình được cài đặt bằng Python với OpenCV. Gồm các bước:

- **Tiền xử lý ảnh:** Đọc ảnh, chuyển ảnh về dạng grayscale, chỉ lấy kênh Blue.
- **Cắt template:** Chọn một vùng nhỏ trong ảnh chứa một con thỏ làm template (ở đây ta chọn mắt của thỏ).
- **Thực hiện Template Matching:** Sử dụng `cv2.matchTemplate` để tìm tất cả vị trí khớp với template trong ảnh.
- **Lọc kết quả bằng IoU:** Xác định vùng giao nhau giữa các bounding box và loại bỏ những vị trí trùng lặp.
- **Vẽ bounding box lên ảnh và đếm số lượng thỏ.**

4. Kết quả

Sau khi chạy thuật toán trên ảnh `rabbit.jpeg`, ta thu được **14 con thỏ** sau khi đã loại bỏ các vùng trùng lặp.

Hình ảnh kết quả:



5. Kết luận

Bài toán đếm thỏ bằng Template Matching kết hợp với IoU đã cho kết quả chính xác với số lượng thỏ trong ảnh. Tuy nhiên, phương pháp này có một số hạn chế:

- **Nhạy cảm với thay đổi kích thước và góc quay:** Nếu kích thước của thỏ thay đổi quá nhiều so với template, thuật toán có thể không hoạt động tốt.
- **Dễ bị ảnh hưởng bởi nhiễu nền:** Nếu nền ảnh phức tạp, có thể xuất hiện các false positive.
- **Không hoạt động tốt với hình ảnh có nhiều biến thể:** Nếu hình ảnh có nhiều dạng thỏ với tư thế khác nhau, việc sử dụng chỉ một template có thể không đủ.

Trong tương lai, có thể cải tiến phương pháp này bằng cách sử dụng **Deep Learning (CNNs, YOLO, Faster R-CNN)** để phát hiện và đếm số lượng thỏ một cách chính xác hơn.