

Bài tập tuần 4

Đoàn Nhật Bình

Ngày 13 tháng 10 năm 2024

1 Giới thiệu về Template Matching

Template matching cho phép chúng ta phát hiện vật thể trong ảnh đầu vào bằng cách sử dụng một ảnh tham chiếu (template) chứa vật thể cần phát hiện. Cách thuật toán hoạt động:

- Trượt ảnh template trên ảnh đầu vào (giống tích chập 2D)
- Ảnh template sẽ được so sánh với cửa sổ trượt tương ứng trên ảnh đầu vào.

1.1 Template Matching trong OpenCV

OpenCV cung cấp sẵn hàm `cv2.matchTemplate()` cho thuật toán template matching. Hàm này nhận vào 3 tham số chính:

- Ảnh đầu vào chứa vật thể cần nhận diện
- Ảnh template
- Cách tính output

Nếu hình ảnh đầu vào có kích thước ($W \times H$) và hình ảnh mẫu có kích thước ($w \times h$), hình ảnh đầu ra sẽ có kích thước là $(W-w+1, H-h+1)$. Sau khi nhận được kết quả, ta có thể sử dụng hàm `cv2.minMaxLoc()` để tìm giá trị tối đa/tối thiểu. Lấy nó làm góc trên cùng bên trái của hình chữ nhật và lấy (w, h) làm chiều rộng và chiều cao của hình chữ nhật. Hình chữ nhật đó sẽ là vùng mẫu cần tìm.

Một số phương pháp so sánh được hỗ trợ trong opencv:

cv2.TM_SQDIFF	$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$
cv2.TM_SQDIFF_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$
cv2.TM_CCORR	$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$
cv2.TM_CCORR_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$
cv2.TM_CCOEFF	$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$ <p>where,</p> $T'(x', y') = T(x', y') - 1/(w \cdot h) \cdot \sum_{x'', y''} T(x'', y'')$ $I'(x + x', y + y') = I(x + x', y + y') - 1/(w \cdot h) \cdot \sum_{x'', y''} I(x + x'', y + y'')$
cv2.TM_CCOEFF_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$

Hình 1: Một số phương pháp

2 Bài toán tìm vật thể

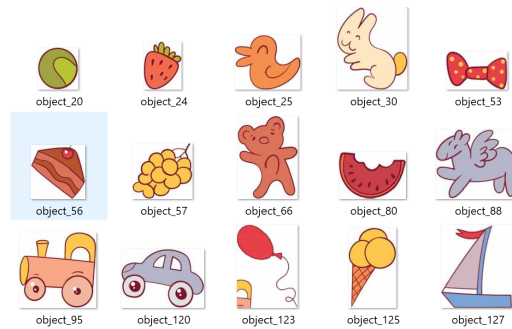
Sử dụng template matching để tìm vật thể có trong ảnh.

2.1 Tạo các ảnh template

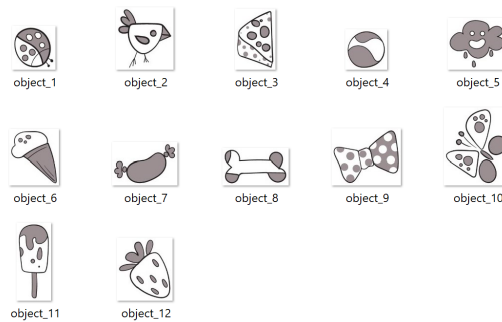
Biến các vật thể cần tìm thành các ảnh template: mỗi vật thể một ảnh Ban đầu, ảnh gốc được chia làm ảnh chứa các vật thể cần tìm và ảnh đầu vào để nhận diện. Ảnh chứa các vật thể được xử lý qua các bước sau:

- Chuyển đổi sang ảnh xám (grayscale)
- Tách vật thể và nền trong ảnh bằng ngưỡng (threshold)
- Tìm các đường bao (contours) xung quanh các vật thể trong ảnh. Mỗi đường bao sẽ tương ứng với một vật thể
- Cắt các vật thể có trong ảnh và lưu lại thành từng ảnh riêng biệt

Khó khăn: Ảnh các vật thể có thể chồng lên nhau, đòi hỏi phải tìm đúng đường bao (countours). Ngoài ra, đối tượng có thể có khoảng hở giữa các bộ phận. Ví dụ như hình con bướm, dẫn đến cắt nhầm thành 2 ảnh với 2 vật thể.



Hình 2: Ảnh vật thể 1



Hình 3: Ảnh vật thể 2

2.2 Áp dụng template matching

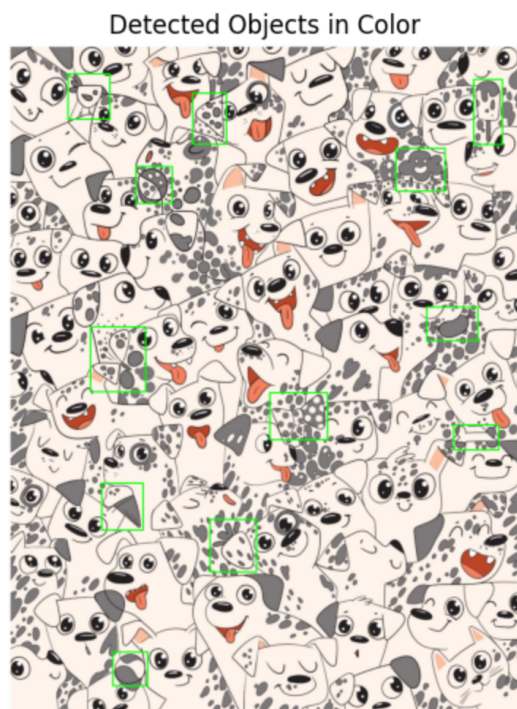
Dùng các kĩ thuật kết hợp với template matching để tìm vật thể:

- Áp dụng Canny Edge Detection trên cả ảnh gốc và ảnh mẫu (template) để phát hiện các cạnh của đối tượng. Kỹ thuật này làm nổi bật các đường viền rõ ràng trong ảnh, giúp dễ dàng so khớp mẫu.
- Thay đổi kích thước mẫu để tăng khả năng phát hiện đối tượng.
- Hàm cv2.minMaxLoc cho biết vùng nào của ảnh gốc khớp tốt nhất với mẫu hiện tại. Vị trí và kích thước của mẫu tốt nhất sẽ được lưu để vẽ khung.
- Vẽ khung chữ nhật xung quanh vùng đối tượng phát hiện được trong ảnh gốc.

Phần cài đặt tại đây: [Github](#)



Hình 4: Ảnh kết quả 1



Hình 5: Ảnh kết quả 2

3 Bài toán đếm số lượng vật thể

Sử dụng template matching để đếm số lượng vật thể trong ảnh.

3.1 Hướng tiếp cận

Nhận diện toàn bộ vật thể:

- Giảm nhiễu và phát hiện cạnh: sử dụng các hàm `cv2.GaussianBlur`, `cv2.Canny`
- Tìm các đường viền: sử dụng `cv2.findContours`
- Đếm số lượng đường viền và đưa ra kết quả

Ta có thể nhận thấy kết quả trả về không chính xác do đã tính cả mắt và mũi của mèo. Vì vậy, cách tiếp cận này không khả thi.

Nhận diện dựa trên đặc trưng:

- Tìm các template là đặc trưng của từng ảnh. Ví dụ như ảnh mèo, đặc trưng nhận dạng là mũi (cắt bằng tay).
- So khớp mẫu với từng kênh màu của ảnh đầu vào sử dụng phương pháp `TM_SQDIFF_NORMED` của hàm `cv2.matchTemplate`
- Vẽ hộp ở các vị trí có kết quả so khớp nhỏ hơn hoặc bằng ngưỡng
- Tính toán khoảng cách các hộp và gộp các hộp gần nhau thành một.

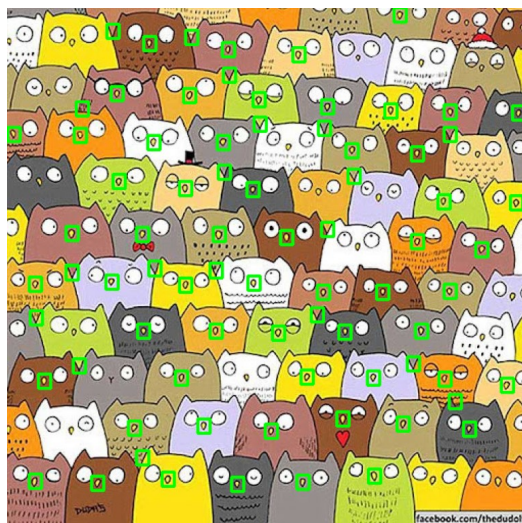


Hình 6: Nhận diện cả con mèo

Kết quả bài counting:

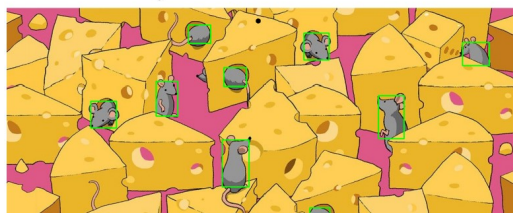


Hình 7: Templates



Hình 8: Cat result

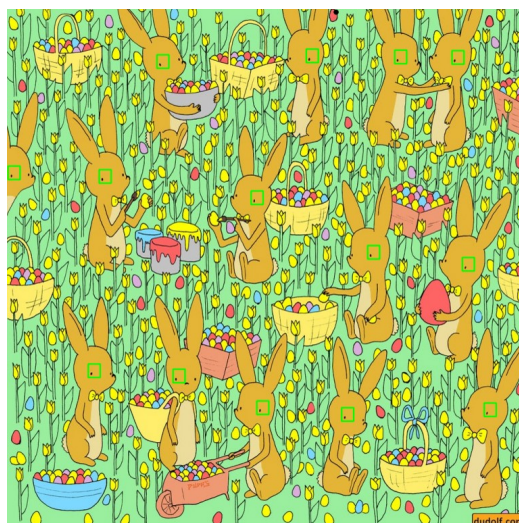
Can you find the dice?



Hình 9: Mouse result



Hình 10: PairBoot result



Hình 11: Rabbit result

4 Tham khảo

1