

Báo cáo bài tập Template Matching

Giới thiệu

Template Matching là một kỹ thuật quan trọng trong xử lý ảnh, được sử dụng để phát hiện và xác định vị trí các mẫu con (template) trong một hình ảnh lớn hơn (ảnh chính). Kỹ thuật này hữu ích trong nhiều bài toán thực tế như nhận diện đối tượng, kiểm tra chất lượng sản phẩm, và nhiều ứng dụng khác. Bài tập này nhằm sử dụng Template Matching để phát hiện và đếm số lượng đối tượng trong một hình ảnh bằng cách áp dụng nhiều bước xử lý như chuẩn hóa, thay đổi kích thước và góc độ mẫu, và kiểm tra trùng lặp để đảm bảo kết quả chính xác.

Phương pháp

1. Tìm kiếm mẫu con (template) trong ảnh chính:

- Các mẫu con được đọc và so khớp với ảnh chính thông qua hàm `cv.matchTemplate`. Phương pháp sử dụng hai chỉ số so khớp chính là `TM_CCORR_NORMED` và `TM_CCOEFF_NORMED` để đánh giá mức độ phù hợp giữa mẫu con và các vùng của ảnh chính.

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

2. Xử lý mẫu ảnh:

- Mẫu được xử lý thông qua bộ lọc cạnh (Canny edge detection) nhằm làm nổi bật các đường biên và loại bỏ các vùng không cần thiết. Việc làm này giúp tối ưu quá trình Template Matching bằng cách tập trung vào các đặc điểm cốt lõi của mẫu, giảm thiểu nhiễu.

3. Chuẩn hóa ảnh kernel và ảnh mục tiêu:

- Trước khi áp dụng Template Matching, cả ảnh chính và ảnh mẫu được chuẩn hóa bằng cách trừ đi giá trị trung bình và chia cho độ lệch chuẩn (standard deviation). Việc chuẩn hóa giúp loại bỏ các biến đổi ánh sáng và độ tương phản giữa ảnh và mẫu, làm tăng độ chính xác trong quá trình nhận diện mẫu.

4. So khớp mẫu theo tỷ lệ và góc độ:

- Để đảm bảo việc phát hiện mẫu ở các kích thước và góc độ khác nhau, mẫu được thay đổi kích thước và xoay góc. Điều này giúp hệ thống phát hiện các đối tượng dù chúng có sự thay đổi về tỷ lệ hoặc góc nhìn so với mẫu gốc.

5. Kiểm tra trùng lặp:

- Sau khi tìm kiếm các vị trí khớp với mẫu, một bước lọc được thực hiện để loại bỏ các vị trí trùng lặp quá gần nhau. Khoảng cách giữa các điểm khớp được tính toán, và những điểm nằm gần nhau quá mức sẽ được gộp lại thành một, giúp tăng độ chính xác trong quá trình đếm.

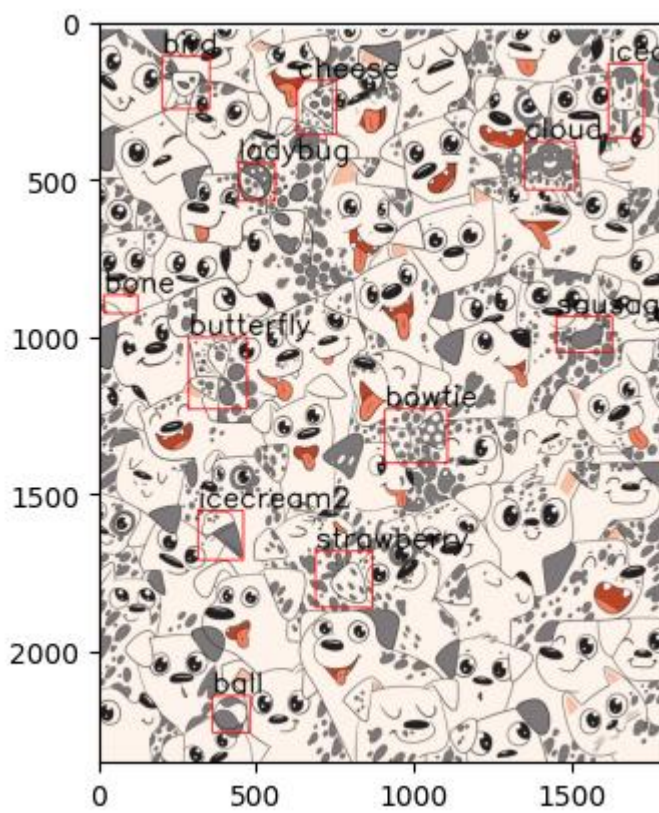
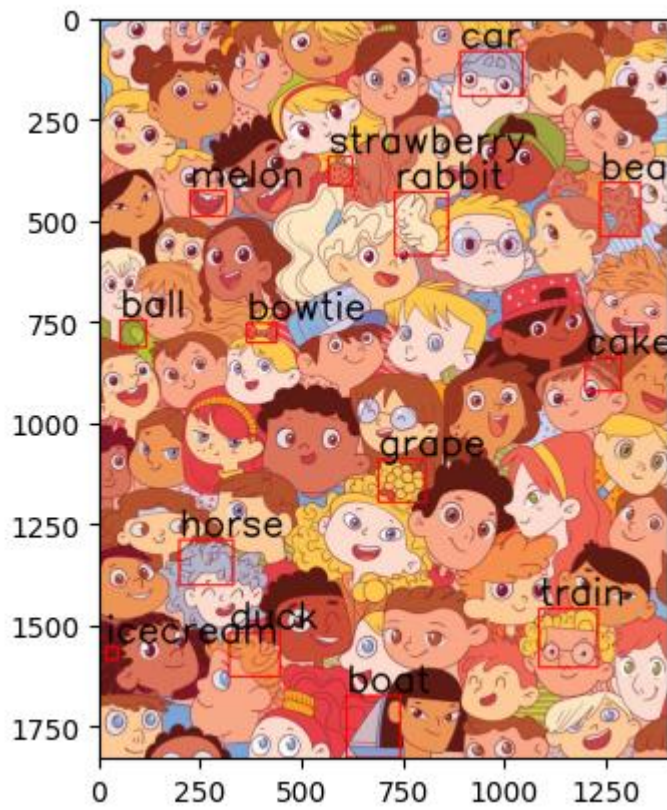
6. Đếm số lượng đối tượng:

- Sau khi loại bỏ các vùng trùng lặp, số lượng đối tượng phát hiện được sẽ được tính toán. Kết quả cuối cùng hiển thị số lượng đối tượng được phát hiện trong hình ảnh, đồng thời các vị trí phát hiện được sẽ được vẽ lên ảnh.

Kết quả

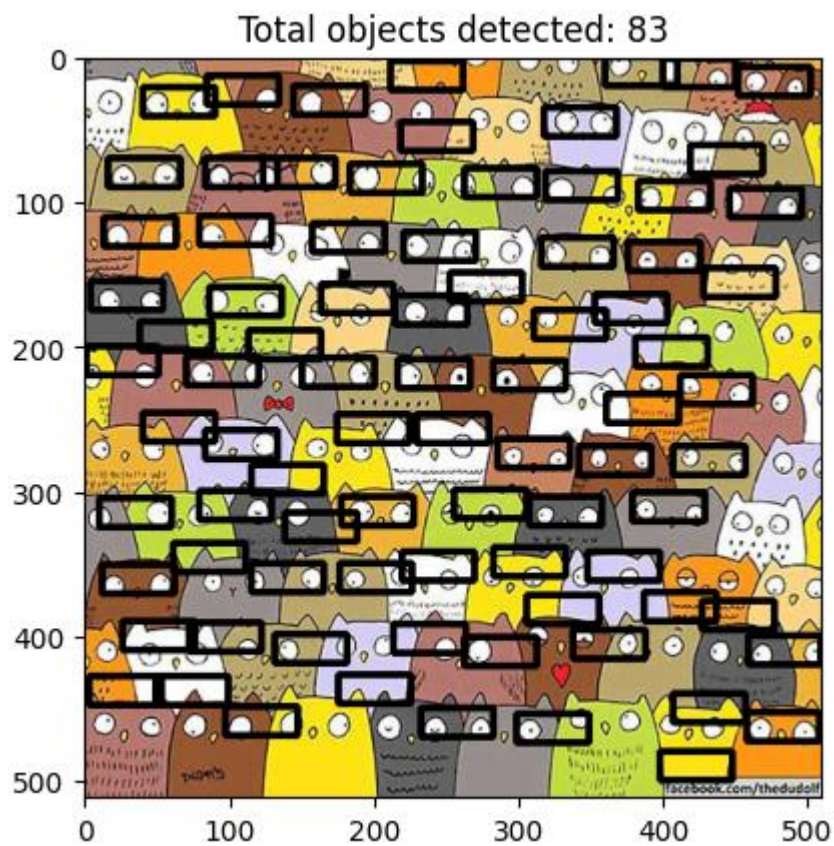
Dưới đây là hình ảnh kết quả của bài toán sau khi áp dụng các phương pháp đã nêu:

Phát hiện đối tượng

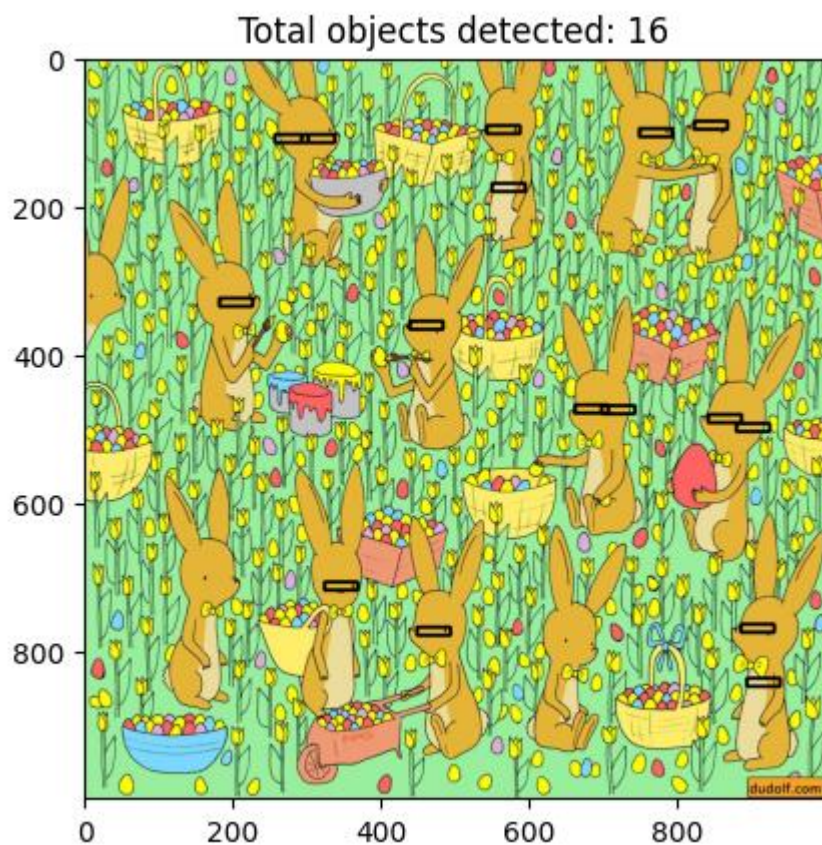


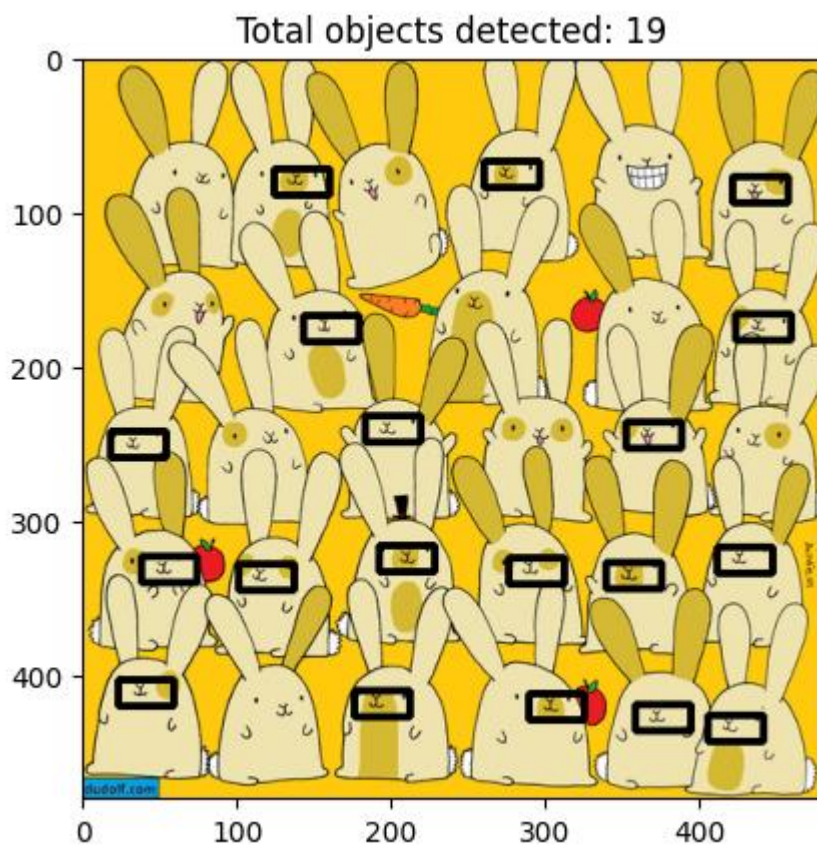
Bộ phát hiện hoạt động rất tốt

Đếm đối tượng có độ chính xác không cao do chỉ dùng 1 kernel và các đối tượng có phần khác nhau

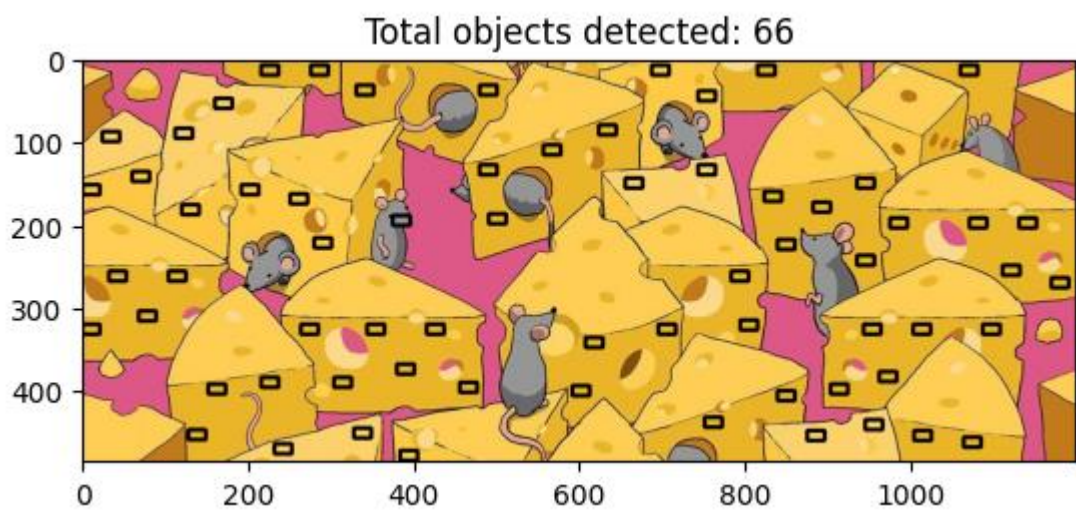


Hai ảnh dùng mắt làm kernel

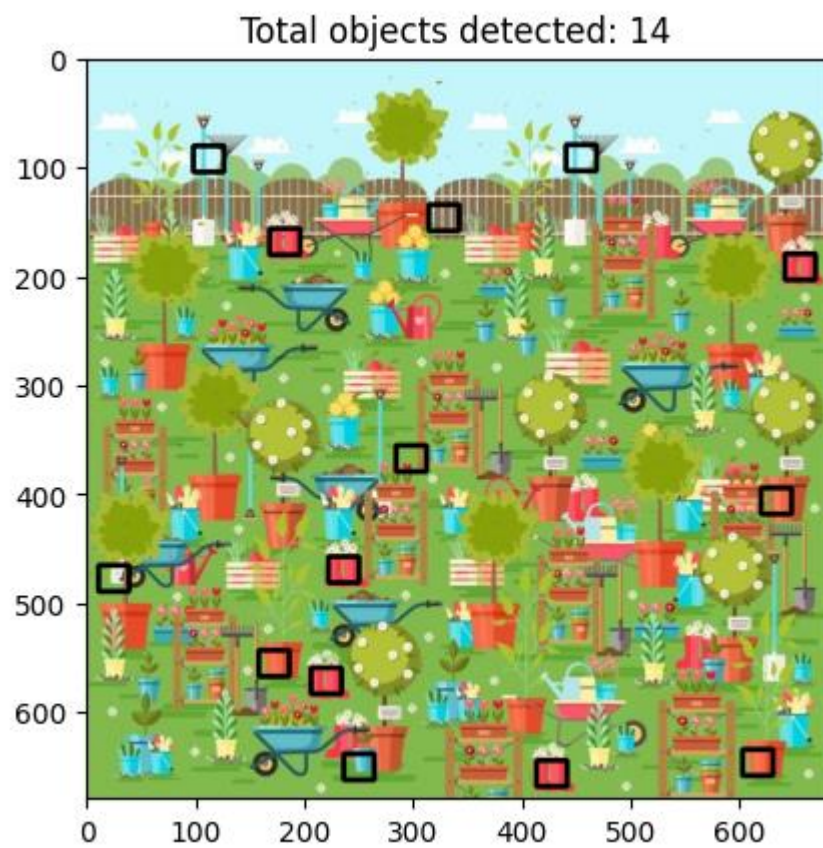




Ảnh trên cũng dùng kernel là mắt



Ảnh trên dùng mũi chuột và kết quả không tốt



Kernel của ảnh này là phần giữa của đôi giày