

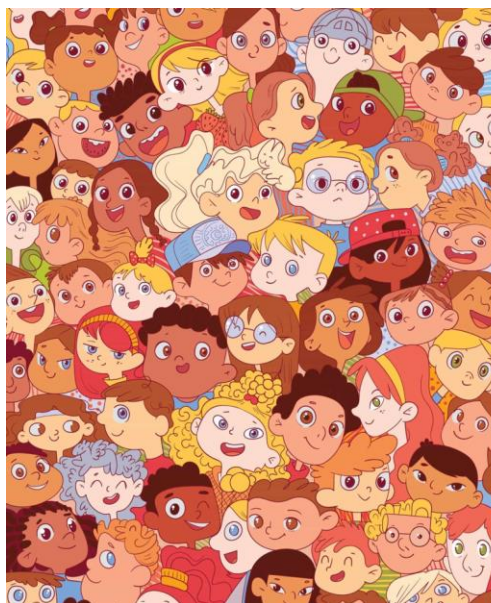
Template matching

Đăng Văn Khải - 22022550

Finding

Bài toán

Cho một hình ảnh lớn, phức tạp (**hình ảnh gốc**), chứa nhiều chi tiết gây rối mắt và một danh sách các đối tượng cần tìm. Yêu cầu: Xác định vị trí của mỗi đối tượng (template) trong hình ảnh gốc bằng phương pháp **template matching**.



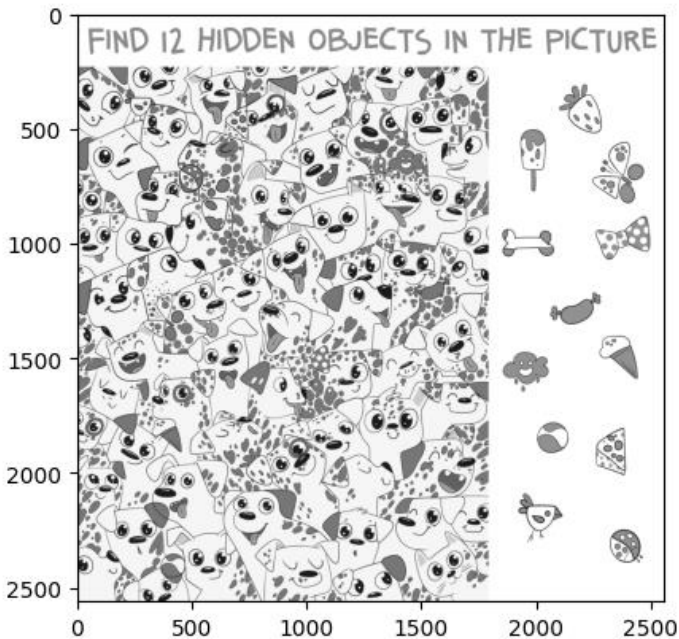
FIND
15
HIDDEN
OBJECTS
IN THE
PICTURE



Các bước làm

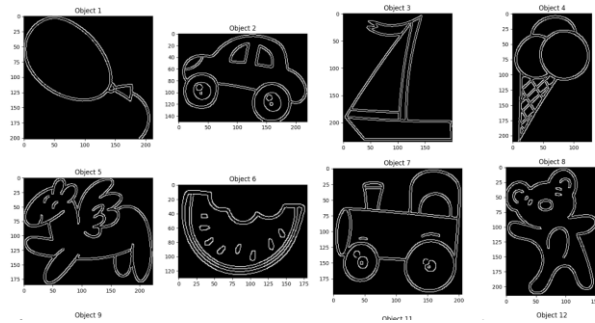
- Đọc hình ảnh: hình ảnh lớn và hình mẫu cần tìm.

- Chuyển về thang độ **xám**: Để giảm độ phức tạp và tăng tốc độ xử lý, cả hình ảnh lớn và hình mẫu đều được chuyển sang thang độ xám.



- `Edge_detection(image)`: trích xuất các đường viền của đối tượng trong cả hình ảnh gốc và template, từ đó giúp việc so khớp mẫu tập trung vào các biên(cạnh) thay vì toàn bộ thông tin pixel, làm tăng độ chính xác khi các đối tượng có thể thay đổi màu sắc hoặc ánh sáng.

```
def edge_detection(image):
    blur = cv2.GaussianBlur(image, (5,5), 0)
    edges = cv2.Canny(blur, 0, 50)
    return edges
```



- Template matching:
+ Ảnh template sẽ được so sánh với cửa sổ trượt tương ứng trên ảnh đầu vào bằng một công thức. Một số phương pháp so sánh được hỗ trợ trong opencv được cho ở hình dưới:

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	$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$

- + Trượt ảnh template trên ảnh đầu vào, một metric sẽ được tính toán tại mỗi pixel của ma trận đầu ra R . Ma trận R sẽ có kích

thước nhỏ hơn ảnh đầu vào (giống với kết quả khi thực hiện tích chập). Giá trị mỗi pixel của R biểu diễn mức độ trùng khớp của ảnh template với vị trí tương ứng trên ảnh đầu vào.

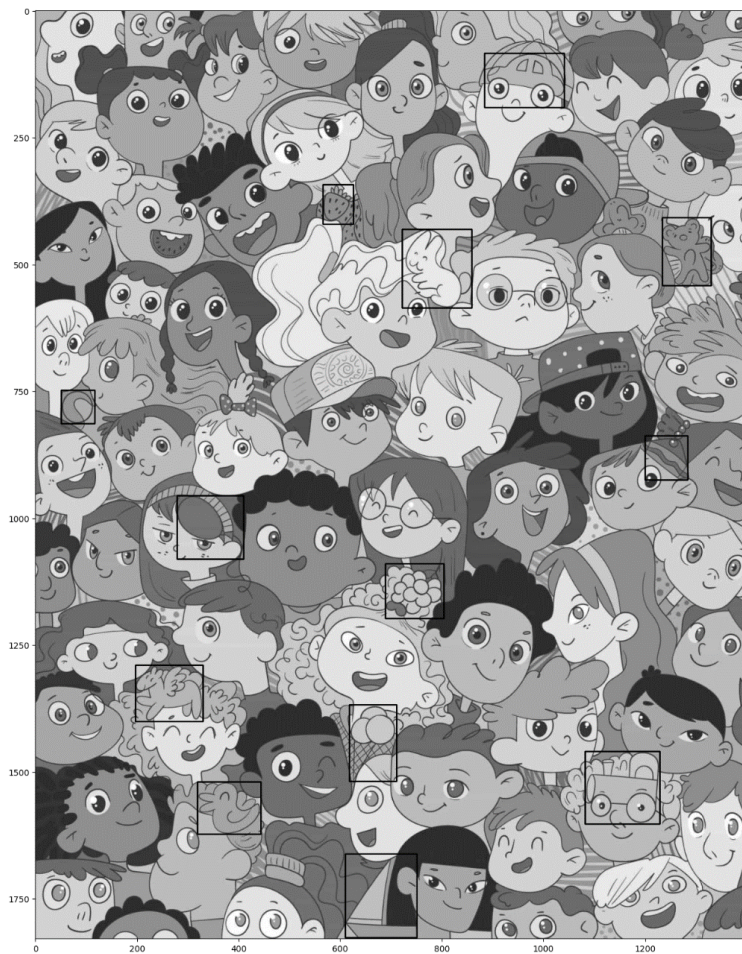
- + Ưu điểm: nhanh, đơn giản, Không tốn công làm data.

- + Nhược điểm: Template phải rất giống với vật thể trong ảnh cả về kích thước độ nghiêng, ... Nếu khác biệt quá lớn sẽ không phát hiện được.

- + Như vậy, phải thực hiện scale nhiều kích cỡ để có thể phù hợp với kích thước trong ảnh lớn

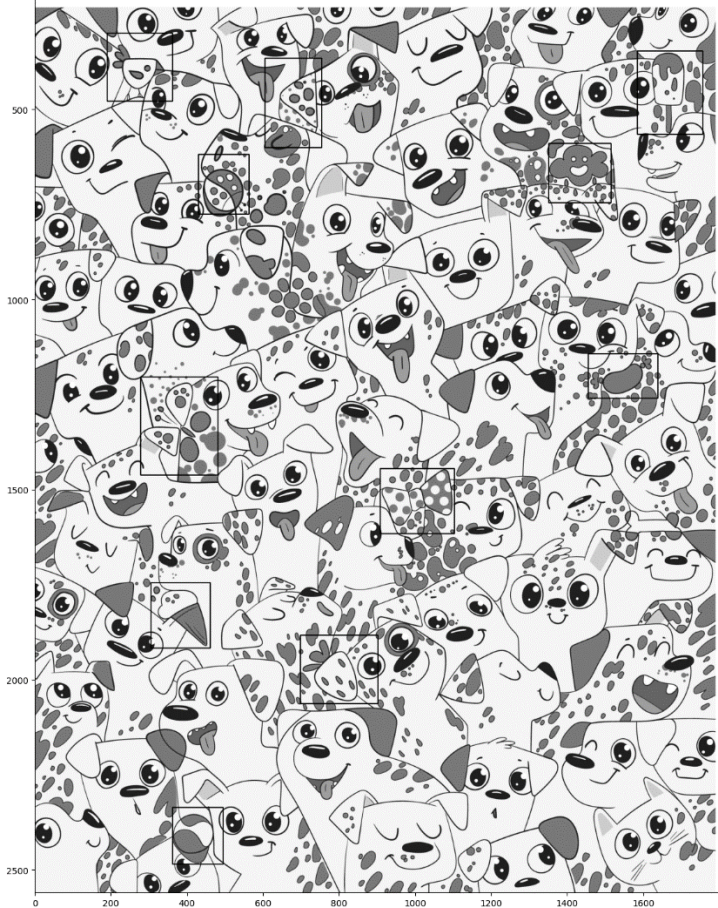
Kết quả

- + Tìm được 13/15 templates trong hình ảnh 1:



+ Tìm được 11/12 templates trong hình ảnh 2:

FIND 12 HIDDEN OBJECTS IN THE



Counting

Bài toán

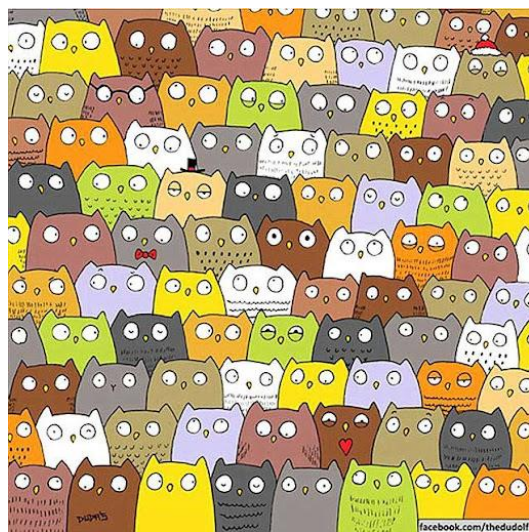
Cho một hình ảnh lớn, phức tạp (**hình ảnh gốc**), có template được lặp lại nhiều lần trong ảnh. Yêu cầu: Xác định số lượng template trong hình ảnh gốc bằng phương pháp **template matching**.

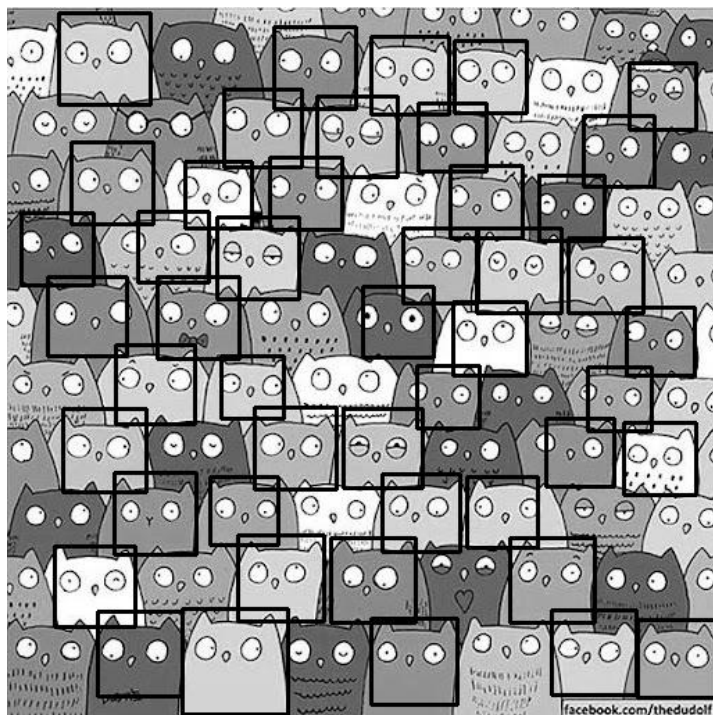
Các bước làm

- Tương tự như cách làm trên, ảnh lớn và template sẽ được chuyển về ảnh xám, để giảm khối lượng tính toán. Sau đó tiếp tục chuyển được trích xuất ra các feature cạnh (edge)
- Xác định ngưỡng(threshold) để xác định một kết quả khớp tốt.
- Thay đổi kích thước mẫu(template)
- Tương tự thực hiện template matching của template với ảnh lớn như trên
- Non-maximum suppression (NMS) là thuật toán giúp loại bỏ các phát hiện trùng lặp hoặc chồng lấn quá nhiều. Tính diện tích các hình chữ nhật được xác định thông qua bước trên. Xác định tỉ lệ chồng lấn, nếu 2 template chồng lấn \geq **overlapThresh**, thì sẽ bị loại bỏ.
- Thay đổi mức scale, threshold, overlapThresh để được kết quả tốt nhất.

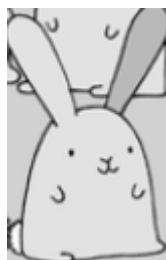
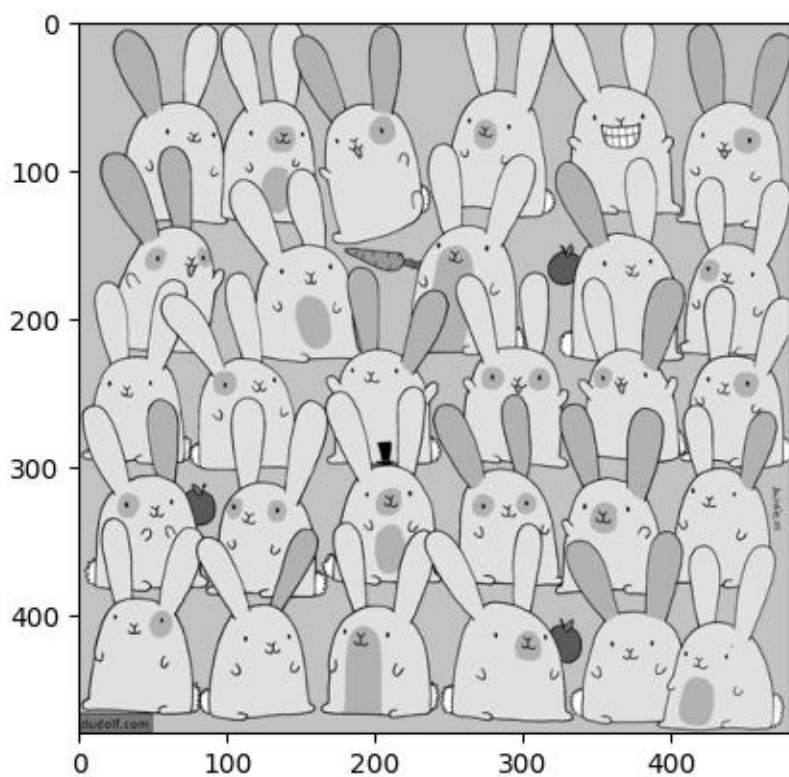
Kết quả

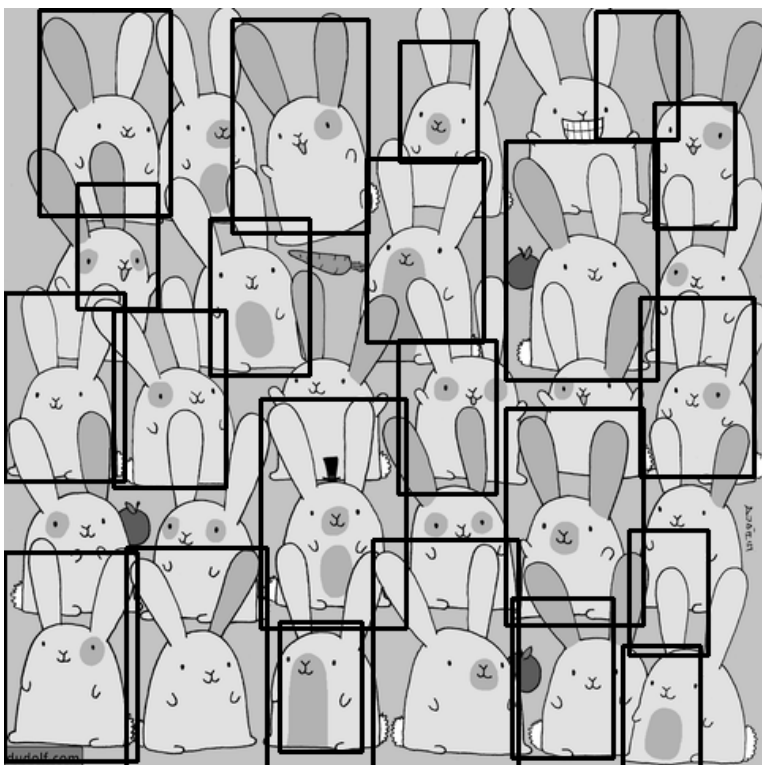
- Hình ảnh 1: **Đếm được 54 / 84 con mèo:**
+ Nhận xét: có thể thấy được kết quả khá tốt, vì với template lựa chọn là con mèo không lông, mắt mở to thì đã phát hiện được nhiều con mèo có đặc điểm tương tự mặc dù vẫn có lỗi sai và thiếu.





- Hình ảnh 2: **Tìm được 18/29 chú thỏ**
+ Nhận xét: Vì template được lấy là chú thỏ hoàn chỉnh (không bị che bởi vật thể khác), nên những chú thỏ được khoanh vùng (hình chữ nhật) tốt sẽ không bị che lấp quá nhiều.





- Hình 3: **Tìm được 6/14 con thỏ**
+ Nhận xét: trong 14 con thỏ có 1 con không hoàn chỉnh (thân không xuất hiện trong ảnh) nên khó có thể nhận diện vì template được lấy là một chú thỏ hoàn chỉnh. Trong 6 con thỏ được phát hiện thì tất cả con thỏ thì đều có đặc điểm rất giống với template (quay sang phải, 2 tai thẳng). Vì vậy có thể thấy phương pháp Template matching chỉ phù hợp với điều kiện là template rất giống với những vật thể cần tìm trong ảnh.

