



# Finding and counting report

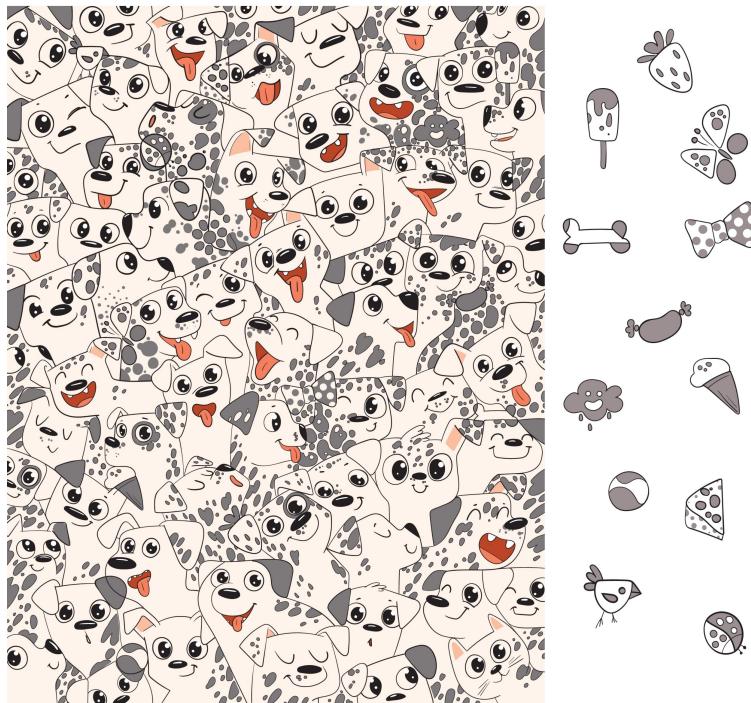
## 1. Giới thiệu

- Trong bài báo cáo này, em sẽ trình bày quy trình tìm kiếm và đếm đối tượng trong hình ảnh sử dụng các kỹ thuật xử lý ảnh trong OpenCV.
- Việc tìm kiếm và đếm đối tượng có ứng dụng rộng rãi trong nhiều lĩnh vực như giám sát, kiểm tra sản phẩm, và phân tích dữ liệu hình ảnh.
- Bài toán tìm kiếm vật thể
  - Input: ảnh đầu vào, ảnh vật thể cần tìm
  - Output: vị trí vật thể cần tìm kiếm
- Bài toán đếm số vật thể
  - Input: ảnh đầu vào, ảnh vật thể cần đếm
  - Output: số lượng vật thể đếm được

## 2. Bài toán tìm kiếm vật thể

- Dưới đây là 2 hình mẫu cho bài toán tìm kiếm vật thể

## FIND 12 HIDDEN OBJECTS IN THE PICTURE



Để tìm kiếm vật thể, em sử dụng phương pháp [Image Matching](#) được thiết kế để tìm kiếm và so khớp các đặc trưng giữa hai hình ảnh, sử dụng kỹ thuật SIFT (Scale-Invariant Feature Transform) và thuật toán FLANN (Fast Library for Approximate Nearest Neighbors) để xác định các cặp điểm tương đồng. Cụ thể phương pháp gồm các bước:

### 1. Tạo Bộ Trích Chọn Đặc Trưng:

- Sử dụng `cv2.SIFT_create()` để khởi tạo bộ trích chọn đặc trưng SIFT với số tầng octave được thiết lập là 10. SIFT giúp phát hiện các điểm đặc trưng mạnh mẽ trong hình ảnh, không bị ảnh hưởng bởi sự thay đổi về kích thước hoặc xoay.

## 2. Trích Suất Đặc Trưng:

- Sử dụng `detectAndCompute` để phát hiện và trích suất các điểm đặc trưng (keypoints) và mô tả (descriptors) từ cả hai hình ảnh (`img1` và `img2`).

## 3. Tìm Kiếm Các Cặp Đặc Trưng Tương Đồng:

- Sử dụng FLANN để tìm kiếm nhanh các cặp điểm đặc trưng tương đồng giữa hai tập hợp mô tả.
- `flann.knnMatch` được sử dụng để trả về hai đặc trưng gần nhất cho mỗi điểm trong `des1` và `des2`.

## 4. Lọc Các Ghép Cặp Tốt:

- Sử dụng tỉ lệ khoảng cách (distance ratio) để lọc ra các cặp ghép mà có khoảng cách giữa chúng đủ gần ( $m.distance < distance\_ratio * n.distance$ ). Chỉ các cặp tốt (good matches) sẽ được giữ lại.

## 5. Khoanh Vùng Đối Tượng Tương Ứng:

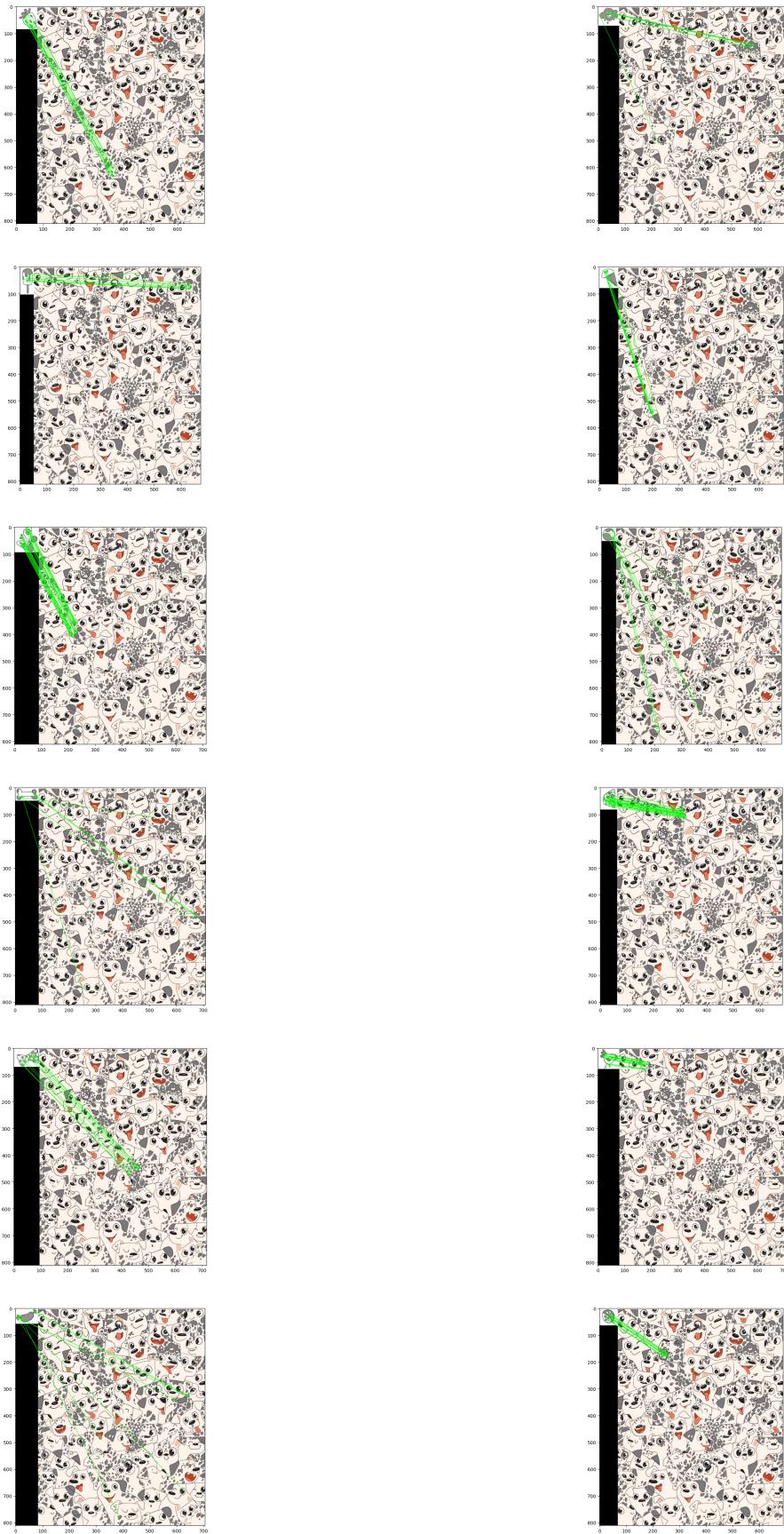
- Nếu số lượng ghép cặp tốt (`good`) vượt quá ngưỡng tối thiểu (`MIN_MATCH_COUNT`), ta sẽ thực hiện các bước sau:
  - Chuyển đổi các điểm đặc trưng trong ảnh nguồn và ảnh đích thành định dạng numpy.
  - Sử dụng `cv2.findHomography` để tìm phép biến đổi đồng dạng giữa các điểm đặc trưng được so khớp, cho phép xác định cách ảnh nguồn biến đổi để khớp với ảnh đích.

## 6. Xác Định Vị Trí Tương Ứng:

- Các điểm tương ứng ở bốn góc của ảnh nguồn sẽ được xác định và chuyển đổi sang vị trí tương ứng trong ảnh đích bằng cách sử dụng `cv2.perspectiveTransform`.

## 7. Hiển Thị Kết Quả:

- Kết quả sẽ được hiển thị bằng cách sử dụng `cv2.drawMatches` để vẽ các đường nối giữa các điểm đặc trưng tương đồng trên hai hình ảnh.
- Phương pháp này tìm được đủ 12 vật thể của hình thứ nhất, với vật thể thứ 2, phương pháp tìm đúng 9/15 vật thể

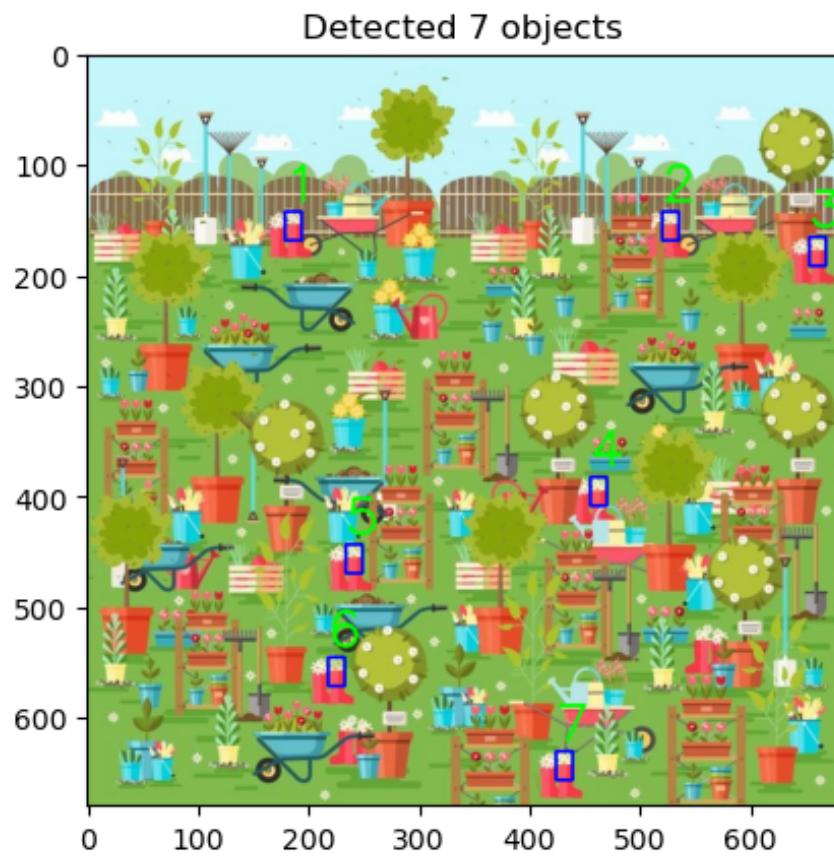


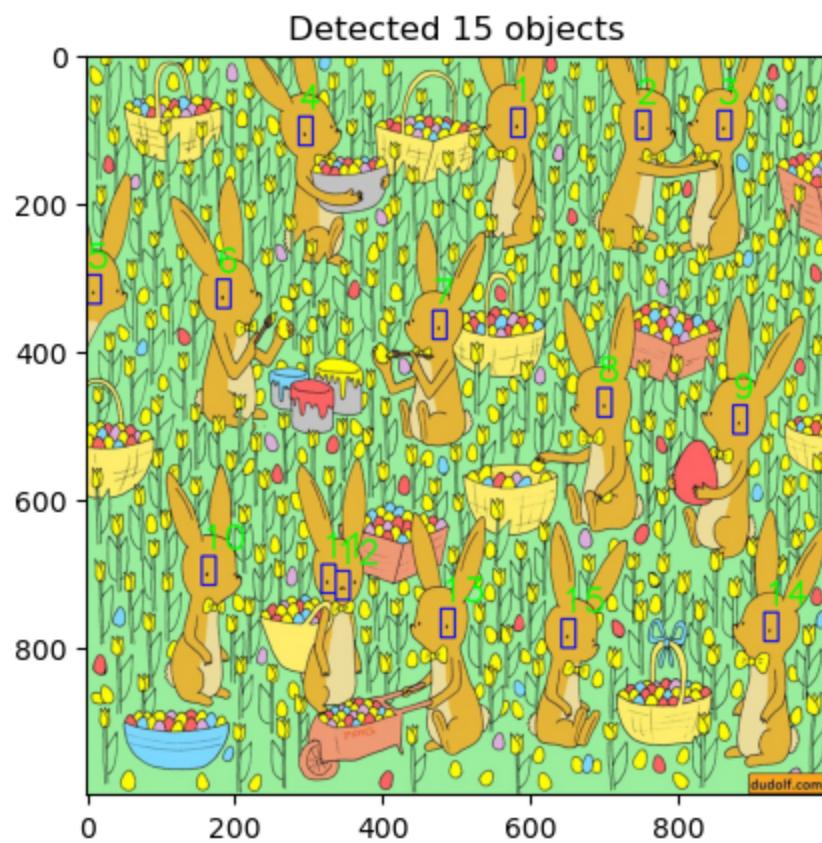


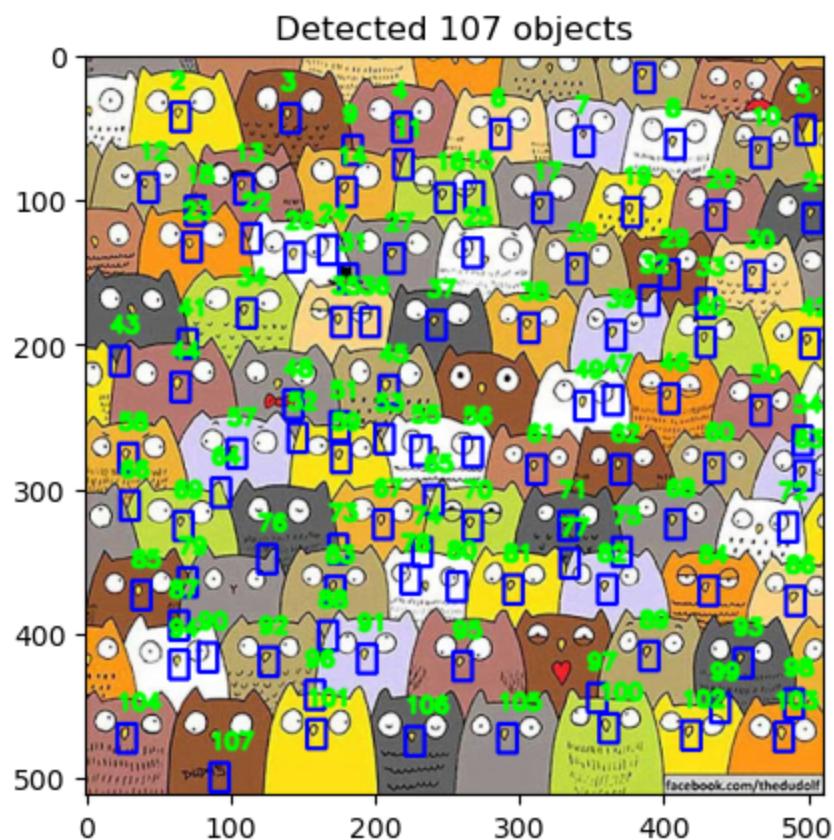
Tóm lại, **SIFT** có nhiều ưu điểm hơn Template Matching khi đối mặt với các điều kiện thay đổi về kích thước, góc xoay, ánh sáng và nhiễu. Template Matching đơn giản và nhanh hơn trong các trường hợp cơ bản, nhưng SIFT lại vượt trội khi cần độ chính xác và sự linh hoạt trong các tình huống phức tạp. Với bài toán này, có thể sử dụng Template Matching với các kích thước template khác nhau, tuy nhiên sẽ tốn thời gian nhiều hơn và không thích ứng tốt nếu có các thay đổi về góc, kích thước.

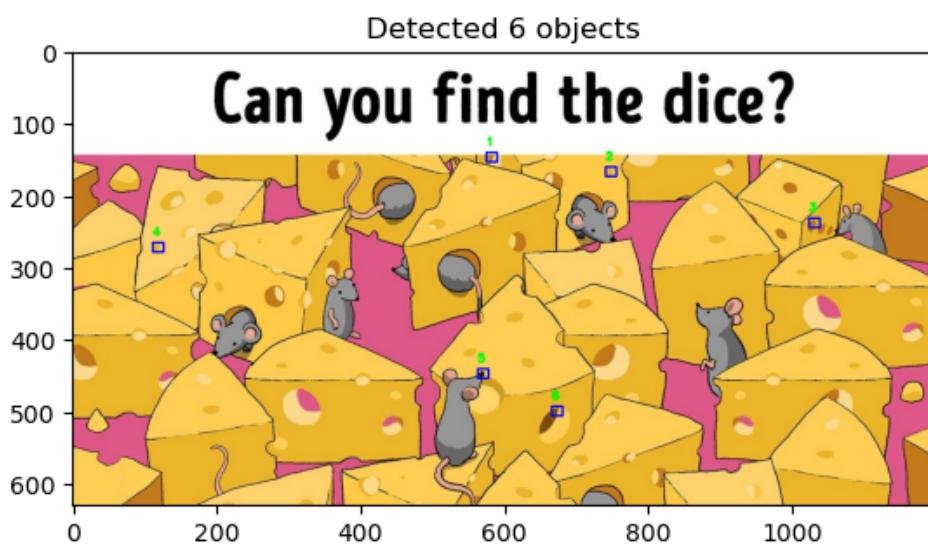
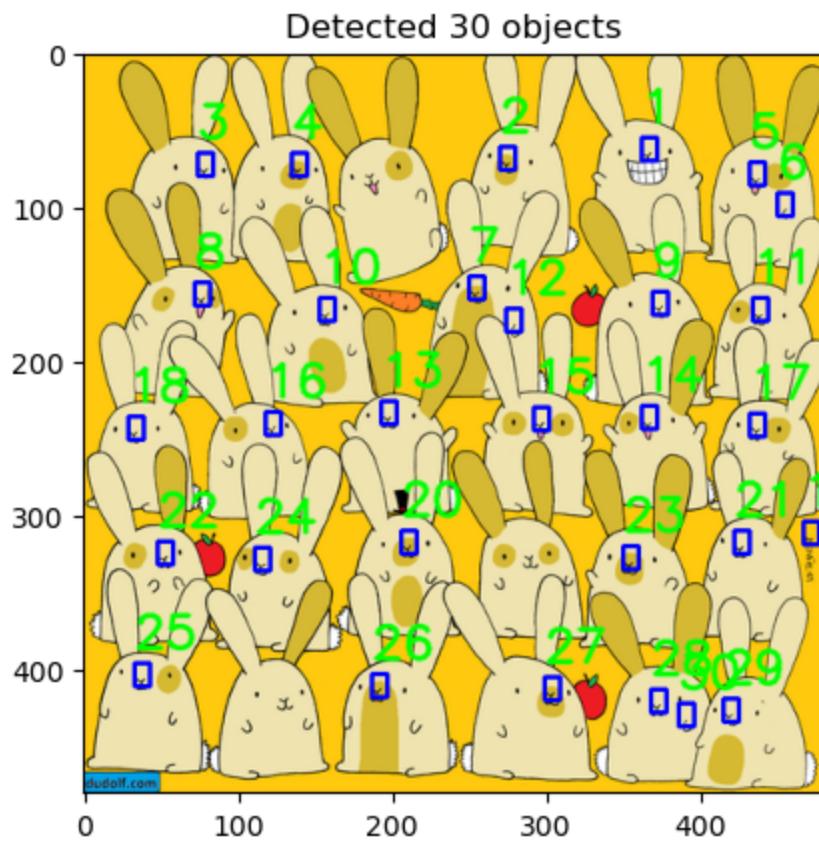
### 3. Bài toán đếm vật thể

- Trong bài toán này, phương pháp **template matching** được sử dụng để phát hiện các đối tượng trong hình ảnh dựa trên mẫu (template). Quá trình bắt đầu bằng việc chuyển đổi cả hình ảnh chính và mẫu thành ảnh xám (grayscale) để giảm độ phức tạp tính toán. Sau đó, hàm `cv2.matchTemplate` được áp dụng để so sánh mẫu với các vùng khác nhau trong hình ảnh chính, trả về một ma trận kết quả biểu thị mức độ khớp giữa mẫu và từng vị trí trong ảnh.
- Dựa trên giá trị ngưỡng (**THRESHOLD**), các vị trí khớp được xác định. Để tránh việc đếm nhiều lần các đối tượng quá gần nhau, hàm `is_too_close` kiểm tra khoảng cách giữa các điểm phát hiện và loại bỏ các điểm quá gần nhau. Khi phát hiện được một đối tượng mới, hình chữ nhật bao quanh đối tượng đó được vẽ và đánh số thứ tự lên hình ảnh.
- Cuối cùng, ảnh kết quả với các đối tượng được khoanh vùng và đánh số được hiển thị bằng thư viện Matplotlib, cho biết tổng số đối tượng đã phát hiện.
- Điểm quan trọng nhất của bài toán này là việc chọn template sao cho tất cả các đối tượng cần đếm đều match.
  - Kết quả: Phương pháp này đếm tốt trên hầu hết các hình ảnh, riêng với hình ảnh con chuột, phương pháp hoạt động chưa tốt do không có template phù hợp chung cho các mẫu.









Trên đây là bài báo cáo của em cho project finding and counting. Rất mong nhận được sự đánh giá và góp ý của thầy để bài làm của em được hoàn thiện hơn. Em xin chân thành cảm ơn!