

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



BÁO CÁO PROJECT GIỮA KÌ

Đề tài

ĐẾM ĐỐI TƯỢNG TRONG HÌNH ẢNH

Sinh viên: **Ngô Huy Hoàn - 22022590**

Link github : [Object-Counting-in-Images](https://github.com/ngohuyhoan/Object-Counting-in-Images)

Học phần : Xử lý và phân tích hình ảnh (AIT3002# 2)

Giảng viên: **TS. Trần Quốc Long**

I. Giới thiệu

- Bài báo cáo này trình bày phương pháp đếm đối tượng trong hình ảnh dựa trên các kỹ thuật xử lý ảnh với thư viện OpenCV.
- Việc tìm kiếm và đếm đối tượng trong hình ảnh có ứng dụng rộng rãi trong nhiều lĩnh vực như giám sát, kiểm tra sản phẩm, và phân tích dữ liệu hình ảnh.
- Mô tả bài toán :
Input: Ảnh đầu vào, ảnh vật thể cần đếm
Output: Tổng số lượng vật thể đếm được trong ảnh

II. Cơ sở lý thuyết

1. Template Matching

- Template Matching là kỹ thuật tìm kiếm vị trí của một mẫu ảnh nhỏ (template) trong ảnh lớn hơn bằng cách so sánh từng vùng của ảnh gốc với template.
- Nguyên lý:
 - Template được "trượt" qua từng pixel của ảnh gốc.
 - Tính toán độ tương đồng giữa template và vùng ảnh đang xét bằng các phương pháp thống kê.
- Ứng dụng:
 - Phát hiện đối tượng có kích thước và hướng nhìn cố định.
 - Áp dụng trong kiểm tra chất lượng sản phẩm, nhận diện biển số xe, v.v.

2. Phương Pháp Normalized Cross-Correlation(TM_CCOCOEFF_NORMED)

- Phương pháp này đo độ tương quan **chuẩn hóa** giữa template và ảnh gốc, giúp giảm ảnh hưởng của thay đổi độ sáng.
- Công thức :

intensity normalization

$$\hat{f} = \frac{f - \bar{f}}{\sqrt{\sum (f - \bar{f})^2}}$$
$$\hat{g} = \frac{g - \bar{g}}{\sqrt{\sum (g - \bar{g})^2}}$$

$$NCC(f,g) = C_{fg} (\hat{f}, \hat{g}) = \sum_{[i,j] \in R} \hat{f}(i,j) \hat{g}(i,j)$$

3. Tiền xử lý ảnh

- Chuyển ảnh sang grayscale:
- Giảm kênh màu từ RGB (3 kênh) thành 1 kênh xám, giúp giảm độ phức tạp tính toán.
- Lý do: Template Matching không yêu cầu thông tin màu sắc nếu đối tượng có thể nhận diện qua cấu trúc hình học.

4. Xác định ngưỡng (Thresholding)

- Ngưỡng (THRESHOLD):
 - Là giá trị quyết định một vùng ảnh được coi là khớp với template.
 - Ví dụ:
 - THRESHOLD=0.84
 - THRESHOLD=0.84 nghĩa là chỉ các vùng có độ tương quan $\geq 84\%$ mới được chọn.
 - Cân bằng giữa Precision và Recall:
 - Ngưỡng cao \rightarrow Giảm false positives nhưng tăng nguy cơ bỏ sót đối tượng.
 - Ngưỡng thấp \rightarrow Phát hiện nhiều đối tượng nhưng dễ nhiễu.

5. Xử lý kết quả trùng lặp

- Do Template Matching có thể trả về nhiều vùng trùng lặp cho cùng một đối tượng, cần áp dụng lọc nhiễu:
- Kiểm tra khoảng cách tối thiểu (min_distance):
- Sử dụng khoảng cách Euclid để loại bỏ các điểm gần nhau:

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Nếu khoảng cách giữa hai điểm nhỏ hơn min_distance, chỉ giữ lại điểm có độ tin cậy cao hơn.

III. Thực nghiệm

1. Dữ Liệu

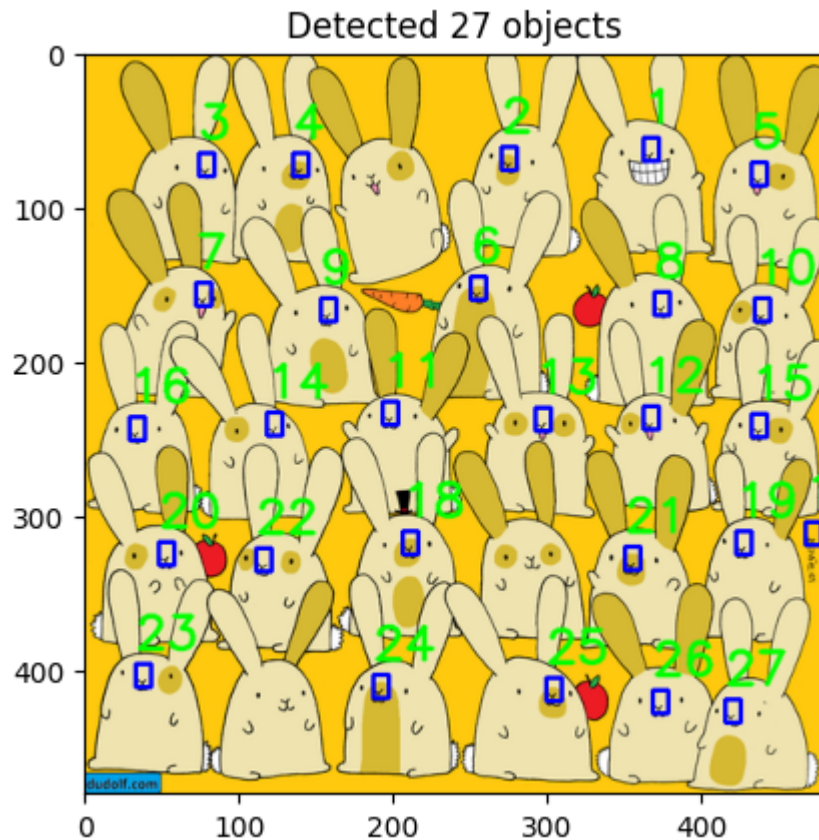
- **Ảnh đầu vào:**
 - Ảnh chứa nhiều đối tượng cần đếm (ví dụ: giày, chai nước, linh kiện điện tử).
 - Độ phân giải đa dạng (từ 640x480 đến 1920x1080).
- **Template:**
 - Ảnh mẫu của đối tượng cần tìm kiếm, được cắt từ ảnh gốc hoặc chuẩn bị riêng.
 - Kích thước nhỏ hơn ảnh gốc .

2. Quy Trình Thực Hiện

- Thư viện sử dụng
 - opencv2
 - matplotlib.pyplot
 - numpy
- Tiền xử lý ảnh
 - Sử dụng ảnh rabbit2.png
 - Chuyển ảnh và template sang ảnh xám bằng `cv2.COLOR_BGR2GRAY`
- Template matching
 - Sử dụng phương pháp `cv2.TM_CCOEFF_NORMED`
 - Áp dụng ngưỡng để lọc kết quả
- Lọc kết quả trùng lặp
 - Kiểm tra khoảng cách Euclid giữa các điểm
 - Loại bỏ điểm có khoảng cách $< \text{min_distance}$
- Hiển thị kết quả
 - Vẽ bounding box màu xanh (BGR: (255,0,0))
 - Đánh số thứ tự màu xanh lá (BGR: (0,255,0))

3. Kết Quả

- Tham số sử dụng : `THRESHOLD=0.62, min_distance=32, font_scale=1`
- Kết quả hình ảnh :



IV. Kết luận

Trong project này , tôi đã triển khai phương pháp đếm đối tượng trong hình bằng phương pháp Template matching sử dụng thư viện OpenCV. Phương pháp này đã nhận diện và đếm số lượng đối tượng xuất hiện trong hình ảnh.