

Template Matching for Image Processing

Ngô Văn Kiệt - 22022643

Github

April 1, 2025

1 Ý tưởng

Trong bài toán này, chúng ta thực hiện phát hiện các đối tượng từ một tập hợp hình ảnh mẫu đã được cắt ra trước đó. Chúng ta sử dụng thuật toán **Template Matching** để tìm vị trí của các đối tượng trong ảnh chính.

2 Phương pháp sử dụng

2.1 Template Matching

Template Matching là kỹ thuật tìm kiếm và so khớp một mẫu nhỏ (template) trong một ảnh lớn. Trong bài này, ta sử dụng phương pháp `cv2.matchTemplate()` với **Squared Difference Normalized (TM_SQDIFF_NORMED)**.

$$R(x, y) = \frac{\sum_{i,j} (T(i, j) - I(x + i, y + j))^2}{\sum_{i,j} T(i, j)^2} \quad (1)$$

Trong đó:

- $R(x, y)$ là giá trị tương quan tại vị trí (x, y) .
- $T(i, j)$ là giá trị pixel của template tại vị trí (i, j) .
- $I(x + i, y + j)$ là giá trị pixel của ảnh nguồn tại vị trí $(x + i, y + j)$.
- Giá trị $R(x, y)$ càng nhỏ thì mức độ khớp càng cao.

2.2 Multi-Scale Template Matching

Vì kích thước của đối tượng có thể thay đổi, ta thực hiện tìm kiếm trên nhiều tỷ lệ khác nhau:

- Chuyển đổi ảnh về ảnh xám (Grayscale).
- Tạo Binary Mask bằng `cv2.threshold()`.
- Thử nhiều tỷ lệ khác nhau bằng `cv2.resize()`.
- Tìm vị trí tốt nhất với `cv2.minMaxLoc()`.
- Vẽ Bounding Box lên ảnh kết quả.

2.3 Intersection over Union (IoU)

Intersection over Union (IoU) là một chỉ số dùng để đo mức độ chồng lấn giữa hai hình chữ nhật đại diện cho các kết quả tìm kiếm. Công thức tính IoU như sau:

$$IoU = \frac{A_{\text{intersection}}}{A_{\text{union}}} \quad (2)$$

Trong đó:

- $A_{\text{intersection}}$ là diện tích giao nhau của hai vùng.
- A_{union} là diện tích hợp nhất của hai vùng.

Nếu $IoU > 0.5$, coi như hai vùng này là trùng lặp và loại bỏ một trong số đó.

3 Kết quả thực nghiệm

Template Matching giúp xác định được các vùng có sự xuất hiện của template với độ chính xác cao. Việc sử dụng IoU để lọc trùng lặp giúp giảm số lượng kết quả bị sai và cải thiện độ chính xác của thuật toán.