

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO**

-----*-----**

**BÁO CÁO GIỮA KÌ MÔN XỬ LÝ VÀ
PHÂN TÍCH HÌNH ẢNH**

Sinh viên thực hiện

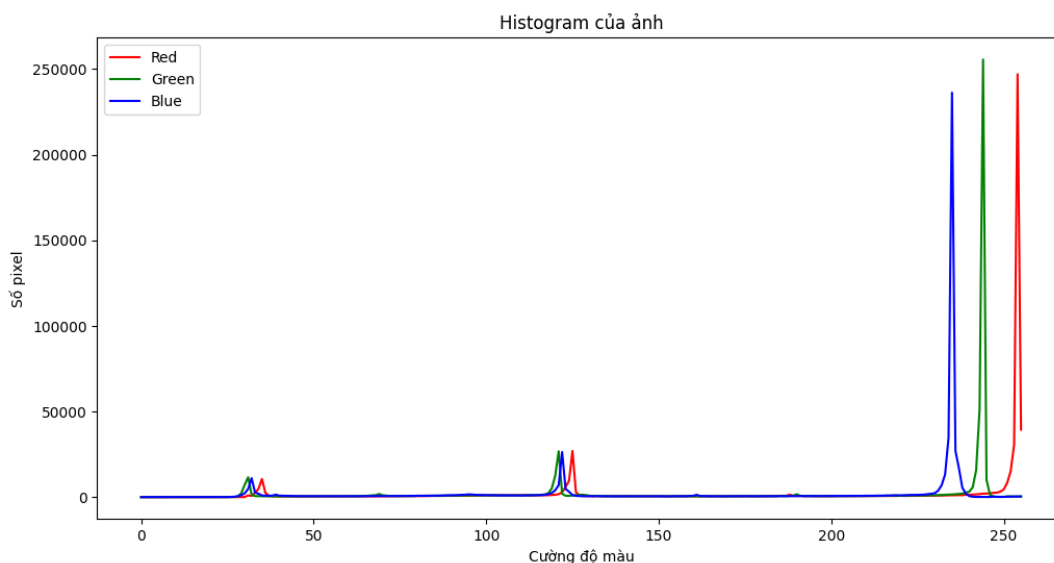
Tạ Nguyên Dũng - 22022546

I. Giới thiệu

Trong bài toán tìm đồ vật trong ảnh, mục tiêu là phát hiện một hoặc nhiều vật thể trong ảnh lớn bằng cách so sánh với một ảnh mẫu (template). Đây là một bài toán quan trọng trong thị giác máy tính, có ứng dụng trong nhận diện đối tượng, kiểm tra chất lượng sản phẩm, và hệ thống giám sát.

II. Phân tích

- Thông tin về ảnh:
 - Kích thước: 2048 x 1463 pixel
 - Số kênh màu: 3 (RGB)
- Mô tả nội dung ảnh
 - Ảnh là một tranh tìm đồ vật ẩn, với rất nhiều khuôn mặt trẻ em vẽ theo phong cách hoạt hình.
 - Bên phải là ảnh danh sách 15 đồ vật cần tìm gồm: kem, thuyền, xe hơi, bóng bay, thiên thần nhỏ, dưa hấu, nho, nơ, bánh ngọt, gấu bông, bóng tennis, dâu tây, thỏ và vịt cao su.
- Phân tích màu sắc:
 - Màu sắc chính: Ảnh có nhiều màu sắc rực rỡ, chủ yếu là các tông da (màu cam, vàng, nâu) cùng với màu sắc tươi sáng khác như đỏ, xanh và vàng.
- Phân bố histogram:
 - Histogram của ba kênh màu (đỏ, xanh lá, xanh dương) có nhiều giá trị 255, cho thấy có nhiều vùng sáng.
 - Màu sắc phân bố khá đồng đều, nhưng có nhiều điểm nhấn mạnh ở màu đỏ và xanh



- Bài toán:
 - Nhiệm vụ là tìm ra 15 đồ vật bị ẩn trong bức tranh
 - Có thể sử dụng kỹ thuật xử lý ảnh để tự động phát hiện các đối tượng ẩn bằng cách sử dụng các thuật toán tìm cạnh như Canny và tìm các hình dạng tương đồng (matching template) với danh sách đồ vật.

III. Phương pháp tiếp cận

- Phương pháp tiếp cận bao gồm các bước sau:
 - Sử dụng các bức hình đồ vật ở bên phải làm template mẫu để tìm đồ vật trong bức ảnh

- Chuyển ảnh về thang độ xám (GRAYSCALE):

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray_template = cv2.cvtColor(template, cv2.COLOR_BGR2GRAY)
```

- Áp dụng Canny Edge Detection:

```
# Áp dụng Canny Edge Detection
edges_image = cv2.Canny(gray_image, 50, 150)
```

- Sử dụng Template Matching:

```
result = cv2.matchTemplate(edges_image, edges_template, cv2.TM_CCOEFF_NORMED)
```

- Sử dụng Dynamic Size cho Template:

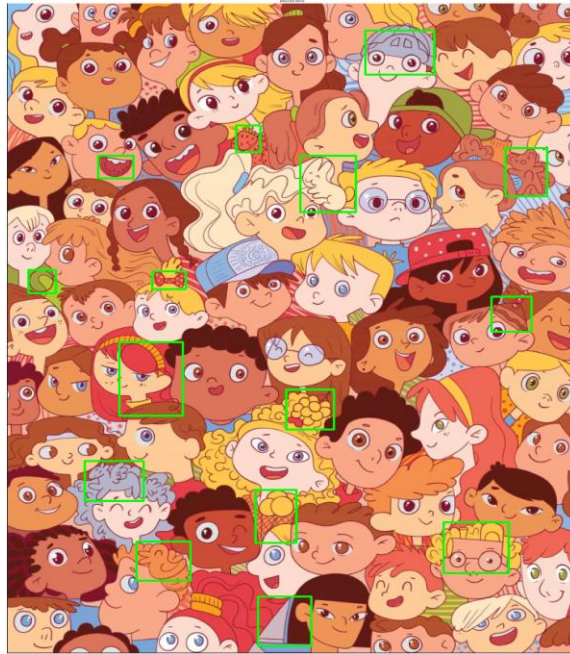
```
# Duyệt qua các tỷ lệ
for scale in np.arange(scale_range[0], scale_range[1], scale_step):
    resized_template = cv2.resize(gray_template, None, fx=scale, fy=scale, interpolation=cv2.INTER_LINEAR)
    edges_template = cv2.Canny(resized_template, 50, 200)
```

- Tùy chỉnh scale range từ 0.5 đến 1, scale_step = 0.01 cho thấy hiệu quả tốt nhất trong việc tìm kiếm.

IV. Giải thích, lập luận tại sao phương pháp hiệu quả

- Đưa ảnh về grayscale giúp loại bỏ thông tin màu không cần thiết, giúp thuật toán xử lý nhanh hơn.
- Canny Edge Detection tập trung vào biên, giúp tìm đối tượng chính xác hơn trong môi trường có ánh sáng thay đổi.
- Template Matching với tìm kiếm đa tỷ lệ giúp phát hiện đối tượng ngay cả khi kích thước của nó thay đổi.
- Việc kết hợp các phương pháp này giúp tăng độ chính xác của việc tìm kiếm vật thể trong ảnh phức tạp.

V. Đánh giá kết quả



- Nhận xét về phương pháp tìm đồ vật ẩn bằng Template Matching và Canny Edge Detection
 - **Ưu điểm:**
 - Phương pháp đã tìm được chính xác 15/15 vật cần phát hiện trong bức ảnh với tốc độ khá nhanh.
 - Dễ triển khai – Sử dụng OpenCV với `cv2.matchTemplate()` và Canny Edge Detection giúp tìm kiếm hình mẫu một cách nhanh chóng mà không cần mô hình phức tạp.
 - Không cần dữ liệu huấn luyện – Khác với các phương pháp deep learning, phương pháp này chỉ cần ảnh mẫu mà không cần quá trình huấn luyện phức tạp.
 - Hiệu quả với các mẫu rõ ràng – Nếu đối tượng cần tìm có viền sắc nét và không bị che khuất quá nhiều, phương pháp này hoạt động khá tốt.
 - Nhanh chóng trên ảnh nhỏ – Khi ảnh có kích thước vừa phải, phương pháp này có thể chạy nhanh mà không tiêu tốn nhiều tài nguyên.
 - **Nhược điểm:**
 - Nhạy cảm với kích thước và góc quay – Nếu đồ vật trong ảnh lớn hoặc nhỏ hơn nhiều so với mẫu, hoặc bị xoay góc, phương pháp có thể không tìm thấy chính xác.
 - Không xử lý tốt khi có nhiễu hoặc che khuất – Nếu vật thể bị che một phần hoặc pha trộn với nền phức tạp, kết quả có thể không chính xác.
 - Phụ thuộc vào tham số Canny Edge Detection – Việc chọn ngưỡng thích hợp cho Canny rất quan trọng. Ngưỡng quá cao hoặc quá thấp có thể bỏ sót hoặc làm nhiễu việc tìm kiếm.
 - Không tối ưu cho ảnh có nhiều màu sắc và độ sáng thay đổi – Template Matching chủ yếu hoạt động tốt trên ảnh grayscale, điều này khiến nó kém hiệu quả khi làm việc với hình ảnh có độ tương phản kém.