

# Finding and counting report

- 1. Thông tin
- 2. Giới thiệu
- 3. Tìm kiếm đối tượng
- 4. Đếm số lượng đối tượng
  - a. Phương pháp sử dụng
  - b. Quy trình:
  - c. Kết quả thu được

## 1. Thông tin

- Người viết: Nguyễn Hoàng Vũ
- Mã sinh viên: 22022502

## 2. Giới thiệu

- Trong báo cáo này, tôi sẽ trình bày quy trình giải bài toán tìm kiếm và đếm số lượng của một đối tượng cho trước trong hình ảnh sử dụng thư viện OpenCV Python.
- Bài toán tìm kiếm đối tượng:
  - Input: ảnh của đối tượng cần tìm kiếm, ảnh cần tìm đối tượng
  - Output: vị trí của đối tượng trong ảnh cần tìm
- Bài toán đếm số lượng:
  - Input: ảnh của đối tượng cần tìm kiếm, ảnh cần đếm số lượng
  - Output: số lượng lần xuất hiện của đối tượng trong ảnh

## 3. Tìm kiếm đối tượng

## 4. Đếm số lượng đối tượng

### a. Phương pháp sử dụng

- Template matching
- Non-maximum suppression.

## b. Quy trình:

- Đọc ảnh ở dạng RGB và chuyển sang grayscale.

```
img_bgr = cv2.imread(img_name + '.jpg')
img = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)
img_template_bgr = cv2.imread(img_name + '_template.jpg')
img_template = cv2.cvtColor(img_template_bgr, cv2.COLOR_BGR2G
```

- Sử dụng hàm `cv2.matchTemplate` để so sánh hình ảnh mẫu với các vùng khác nhau trong hình ảnh chính, thu được một ma trận thể hiện mức độ tương khớp giữa mẫu với các vị trí trong ảnh chính.

```
res = cv2.matchTemplate(img, img_template, cv2.TM_CCOEFF_NORM
```

- Dựa trên một giá trị ngưỡng `threshold` cho trước, lọc ra các vùng có độ chính xác cao, đồng thời tạo một danh sách hình chữ nhật tương ứng với các vùng khớp với ảnh mẫu:

```
loc = np.where(res >= threshold)
w, h = img_template.shape[::-1]
boxes = [[x, y, x + w, y + h] for x, y in zip(*loc[::-1])]
```

- Sử dụng phương pháp Non-maximum suppression để loại bỏ các hình chữ nhật quá gần và đè lên nhau:

```
scores = [res[y, x] for x, y in zip(*loc[::-1])]
indices = cv2.dnn.NMSBoxes(boxes, scores, threshold, 0.5)
```

- Vẽ các hình chữ nhật đã lọc ra lên bức ảnh ban đầu:

```
for i in indices.flatten():
    x, y, u, v = boxes[i]
    cv2.rectangle(img_bgr, (x, y), (u, v), (0, 0, 255), 2)
```

## c. Kết quả thu được

- Phương pháp đã sử dụng hoạt động khá tốt với bốn hình ảnh trừ ảnh các con chuột, điều này có thể do mẫu được lựa chọn không đủ tốt.

