

Báo cáo về bài toán đếm vật thể trong ảnh

Họ và tên sinh viên: Đinh Văn Sinh – 22022615

Link github: <https://github.com/sinh2206/Midterm-ImageSet>

## 1. Bài toán

Đếm các đôi ủng trong bức ảnh



## 2. Phương pháp

Dùng template – matching để đếm thông qua việc sử dụng các đôi ủng làm template

- Hàm `create_template` dùng để tạo 1 template từ ảnh đầu vào để tìm kiếm

```
def create_template(image, template, resize=0.7, threshold=200, new_color=100):
    obj = image[template[0]:template[1], template[2]:template[3]]

    # Thay đổi kích thước đối tượng
    # obj = cv2.resize(obj, None, fx=resize, fy=resize, interpolation=cv2.INTER_AREA)

    # Áp dụng threshold
    obj[obj > threshold] = new_color

    plt.imshow(obj, cmap='gray')
    plt.show()

    return obj
```

- Hàm `non_max_suppression` loại bỏ các bounding box trùng lặp hoặc chồng lấn nhau dựa trên ngưỡng chồng lấn (overlap threshold), giữ lại box có độ tin cậy cao nhất.

```
def non_max_suppression(bboxes, overlapThresh):
    if len(bboxes) == 0:
        return []

    if bboxes.dtype.kind == "i":
        bboxes = bboxes.astype("float")

    pick = []

    x1 = bboxes[:, 0]
    y1 = bboxes[:, 1]
    x2 = bboxes[:, 2]
    y2 = bboxes[:, 3]

    area = (x2 - x1 + 1) * (y2 - y1 + 1)
    idxs = np.argsort(y2)

    while len(idxs) > 0:
        last = len(idxs) - 1
        i = idxs[last]
        pick.append(i)

        # Tính phần chồng lấn (IoU) giữa box hiện tại và các box còn lại
        xx1 = np.maximum(x1[i], x1[idxs[:last]])
        yy1 = np.maximum(y1[i], y1[idxs[:last]])
        xx2 = np.minimum(x2[i], x2[idxs[:last]])
        yy2 = np.minimum(y2[i], y2[idxs[:last]])

        w = np.maximum(0, xx2 - xx1 + 1)
        h = np.maximum(0, yy2 - yy1 + 1)

        overlap = (w * h) / area[idxs[:last]]

        # Loại bỏ các box có overlap > overlapThresh
        idxs = np.delete(idxs, np.concatenate(([last], np.where(overlap > overlapThresh)[0])))
```

- Hàm `counting_object` đếm số lượng vật thể trong ảnh bằng cách so khớp template (template matching) và áp dụng NMS để loại bỏ kết quả trùng lặp.

```

def counting_object(img_finding, img_finding_rgb, templates, methods, threshold=0.15, overlapThresh=0.5):
    methods_name = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR',
                    'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']

    fig, ax = plt.subplots(1, 1, figsize=(10, 8))
    fig.suptitle('Count Pair Boot')

    for idx, method in enumerate(methods):
        rectangles = []

        for template in templates:
            template_canny = cv2.Canny(template, 100, 200)
            h, w = template.shape[:2]

            img_canny = cv2.Canny(img_finding, 100, 200)

            result = cv2.matchTemplate(img_canny, template_canny, method)

            loc = np.where(result >= threshold)

            for pt in zip(*loc[::-1]):
                bottom_right = (pt[0] + w, pt[1] + h)
                rectangles.append([pt[0], pt[1], bottom_right[0], bottom_right[1]])

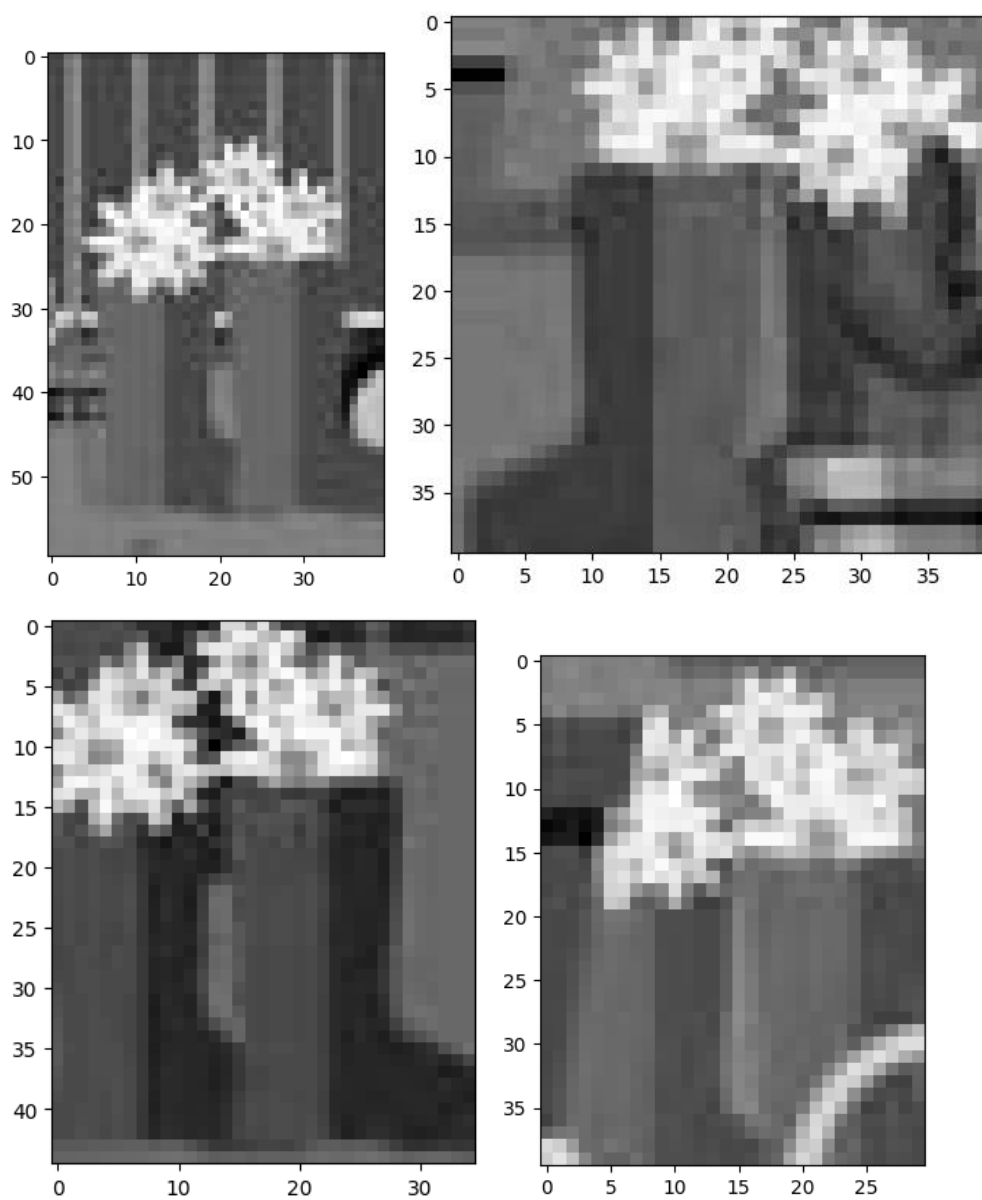
        rectangles = np.array(rectangles)

        if len(rectangles) > 0:
            nms_rectangles = non_max_suppression(rectangles, overlapThresh)
        else:
            nms_rectangles = []

        result_img = img_finding_rgb.copy()
        for (x1, y1, x2, y2) in nms_rectangles:
            cv2.rectangle(result_img, (x1, y1), (x2, y2), (0, 0, 255), 5)

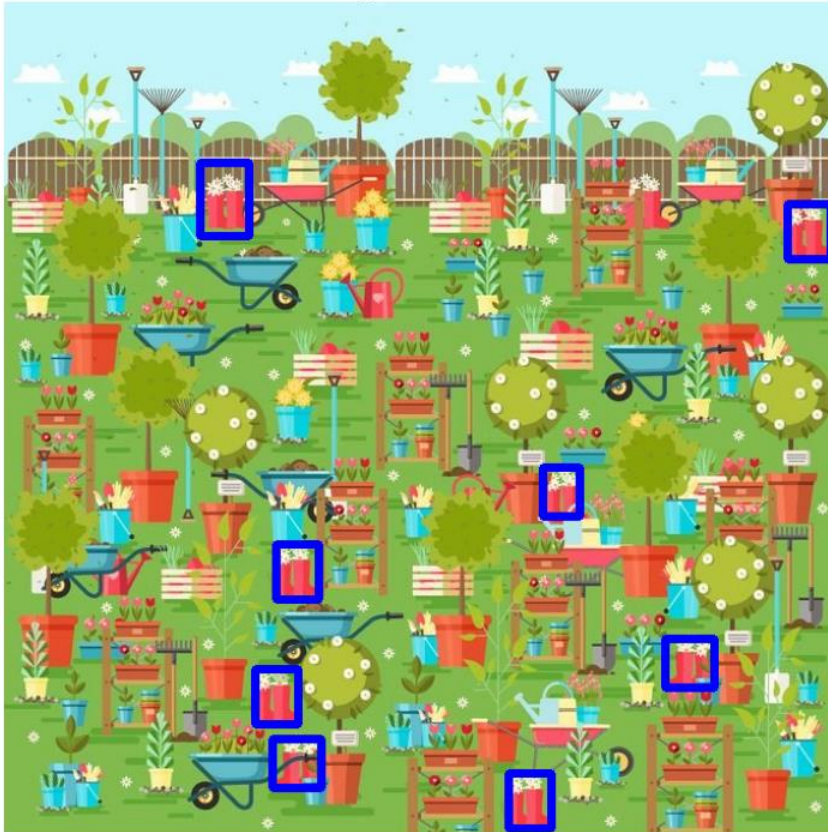
```

- Sử dụng 4 template do ứng xuất hiện ở nhiều vị trí bị một số vật khác che mất một phần(do phương pháp này rất nhạy nên dễ dẫn đến sai lệch)



### 3. Kết quả

Count Pair Boot  
Method: cv2.TM\_CCOEFF  
Objects found: 8



Tìm được : 8

Đếm bằng mắt : 9

Tỉ lệ : ~ 90%