

Chương 9: CHEAT SHEET – NUMPY + IOU + NMS

NUMPY ARRAY

1. import numpy as np: Nhập thư viện NumPy (viết tắt là np).
2. np.array(...): Tạo một NumPy Array từ một List Python.
3. np.zeros(n): Tạo Array 1D với n phần tử là số 0.
4. array_name.shape: Thuộc tính trả về hình dạng (kích thước/số chiều) của Array.
5. np.append(a1, a2): Nối hai Array, trả về Array mới.
6. np.insert(a, i, v): Thêm giá trị v vào vị trí i, trả về Array mới.
7. np.delete(a, i): Xóa phần tử tại vị trí i, trả về Array mới.
8. a_data.sort(): Sắp xếp các phần tử Array tại chỗ (in-place).
9. a_data * scalar: Phép nhân từng phần tử (Element-wise) với số vô hướng.
10. np.min(a) / np.max(a): Tìm giá trị nhỏ nhất / lớn nhất trong Array.
11. np.sum(a): Tính tổng các phần tử trong Array.
12. a_data.reshape(...): Thay đổi hình dạng (shape) của Array, trả về Array mới.
13. np.where(condition): Trả về chỉ số (index) của các phần tử thỏa mãn điều kiện.

IOU – INTERSECTION OVER UNION

1. Định dạng Box

Box có dạng:

[x_min, y_min, x_max, y_max]

2. Các bước tính IoU

1. Tính diện tích box 1
2. Tính diện tích box 2
3. Tìm tọa độ vùng giao:
 - o $xA = \max(x1_min, x2_min)$
 - o $yA = \max(y1_min, y2_min)$
 - o $xA = \min(x1_max, x2_max)$
 - o $yA = \min(y1_max, y2_max)$
4. Tính chiều rộng và chiều cao của vùng giao
5. Diện tích giao = $w * h$
6. Diện tích hợp = $\text{area1} + \text{area2} - \text{intersection}$
7. $\text{IoU} = \frac{\text{intersection}}{\text{union}}$

NMS – NON-MAXIMUM SUPPRESSION

1. Mục đích

Loại bỏ các bounding box bị trùng lặp.

Giữ lại box tốt nhất (score cao nhất).

2. Thuật toán NMS – Tóm tắt

1. Chọn box có **confidence cao nhất**, đưa vào danh sách kết quả.
2. Tính IoU của box này với **tất cả box còn lại**.
3. Loại bỏ các box có IoU lớn hơn threshold.
4. Lặp lại cho đến khi không còn box nào.