

验证流程: (5 组) 帮助理解

```
>>> import numpy as np
>>> thre = 0.1
>>> lists = np.zeros((5, 5))
>>>
>>> lists[:, 0:4] = np.random.randint(100, size=[5, 4])
>>> lists[:, 4] = np.random.rand(5)
>>> print(lists)
[[35.      40.      54.      59.      0.35447849]
 [59.      88.      58.       1.      0.58816124]
 [42.      67.      49.      67.      0.55697901]
 [41.      55.      56.      16.      0.36570904]
 [ 3.      10.      11.       9.      0.38477632]]
>>> result_idx = []
>>> x1 = lists[:, 0]
>>> print(x1)
[35. 59. 42. 41.  3.]
>>> y1 = lists[:, 1]
>>> x2 = lists[:, 2]
>>> y2 = lists[:, 3]
>>> area = (x2 - x1 + 1) * (y2 - y1 + 1)
>>> print(area)
[ 400.  -0.    8. -608.   0.]
>>> scores = lists[:, 4]
>>> order = scores.argsort()[::-1]
>>> print(scores)
[0.35447849 0.58816124 0.55697901 0.36570904 0.38477632]
>>> print(order)
[1 2 4 3 0]
>>> order = scores.argsort()[::-1]
>>> print(order)
[1 2 4 3 0]
>>> print(len(order))
5
>>> idx = order[0]
>>> print(idx)
1
>>> result_idx.append(idx)
>>> print(result_idx)
[1]
>>> print()

>>> print(x1[idx])
59.0
```

```

>>> print(x1[order[1:]])
[42.  3. 41. 35.]
>>> iou_x1 = np.maximum(x1[idx], x1[order[1:]])
>>> print(iou_x1)
[59. 59. 59. 59.]
>>> iou_y1 = np.maximum(y1[idx], y1[order[1:]])
>>> iou_x2 = np.minimum(x2[idx], x2[order[1:]])
>>> iou_y2 = np.minimum(y2[idx], y2[order[1:]])
>>> print(iou_y2)
[1. 1. 1. 1.]
>>> print(iou_x2)
[49. 11. 56. 54.]
>>> w = np.maximum(0, iou_x2 - iou_x1 + 1)
>>> print(w)
[0. 0. 0. 0.]
>>> h = np.maximum(0, iou_y2 - iou_y1 + 1)
>>> print(h)
[0. 0. 0. 0.]
>>> iou_area = w * h
>>> print(iou_area)
[0. 0. 0. 0.]
>>> print(area[idx])
-0.0
>>> iou = iou_area / (area[idx] + area[order[1:]] - iou_area)
__main__:1: RuntimeWarning: invalid value encountered in true_divide
>>> print(area[order[1:]] - iou_area)
[  8.    0. -608.  400.]
>>> a=area[idx] + area[order[1:]] - iou_area
>>> print(a)
[  8.    0. -608.  400.]
>>> iou = iou_area / a
>>> print(iou)
[ 0. nan -0.   0.]
>>> print(np.where(iou < thre))
__main__:1: RuntimeWarning: invalid value encountered in less
(array([0, 2, 3]),)
>>> idx_iou = np.where(iou < thre)[0]
>>> print(idx_iou)
[0 2 3]
>>> idx_order = idx_iou + 1
>>> print(idx_order)
[1 3 4]
>>> print(order)
[1 2 4 3 0]

```

```

>>> order = order[idx_order]
>>> print(order)
[2 3 0]
>>> idx = order[0]
>>> print(idx)
2
>>> result_idx.append(idx)
>>> print(result_idx)
[1, 2]
>>> iou_x1 = np.maximum(x1[idx], x1[order[1:]])
>>> print(iou_x1)
[42. 42.]
>>> iou_y1 = np.maximum(y1[idx], y1[order[1:]])
>>> iou_x2 = np.minimum(x2[idx], x2[order[1:]])
>>> iou_y2 = np.minimum(y2[idx], y2[order[1:]])
>>> w = np.maximum(0, iou_x2 - iou_x1 + 1)
>>> print(w)
[8. 8.]
>>> h = np.maximum(0, iou_y2 - iou_y1 + 1)
>>>
>>> iou_area = w * h
>>> print(iou_area)
[0. 0.]
>>> iou = iou_area / (area[idx] + area[order[1:]] - iou_area)
>>> print(iou)
[-0.  0.]
>>> idx_iou = np.where(iou < thre)[0]
>>> print(idx_iou)
[0 1]
>>> idx_order = idx_iou + 1
>>> print(order)
[2 3 0]
>>> order = order[idx_order]
>>> print(order)
[3 0]
>>> idx = order[0]
>>> result_idx.append(idx)
>>> print(result_idx)
[1, 2, 3]
>>> iou_x1 = np.maximum(x1[idx], x1[order[1:]])
>>> iou_y1 = np.maximum(y1[idx], y1[order[1:]])
>>> iou_x2 = np.minimum(x2[idx], x2[order[1:]])
>>> iou_y2 = np.minimum(y2[idx], y2[order[1:]])
>>> w = np.maximum(0, iou_x2 - iou_x1 + 1)

```

```

>>>
>>> h = np.maximum(0, iou_y2 - iou_y1 + 1)
>>> iou_area = w * h
>>> print(iou_area)
[0.]
>>> iou = iou_area / (area[idx] + area[order[1:]] - iou_area)
>>> print(iou)
[-0.]
>>> idx_iou = np.where(iou < thre)[0]
>>> print(idx_iou)
[0]
>>> idx_order = idx_iou + 1
>>> print(order)
[3 0]
>>> order = order[idx_order]
>>> print(order)
[0]
>>> idx = order[0]
>>> print(idx)
0
>>> result_idx.append(idx)
>>> print(result_idx)
[1, 2, 3, 0]
>>> iou_x1 = np.maximum(x1[idx], x1[order[1:]])
>>> iou_y1 = np.maximum(y1[idx], y1[order[1:]])
>>> iou_x2 = np.minimum(x2[idx], x2[order[1:]])
>>>
>>> iou_y2 = np.minimum(y2[idx], y2[order[1:]])
>>> w = np.maximum(0, iou_x2 - iou_x1 + 1)
>>> h = np.maximum(0, iou_y2 - iou_y1 + 1)
>>> iou_area = w * h
>>> print(iou_area)
[]
>>> iou = iou_area / (area[idx] + area[order[1:]] - iou_area)
>>> print(iou)
[]
>>>
...
>>> idx_iou = np.where(iou < thre)[0]
>>> print(idx_iou)
[]
>>> idx_order = idx_iou + 1
>>> print(idx_order)
[]

```

```
>>> print(order)
[0]
>>> order = order[idx_order]
>>> print(order)
[]
>>> print(result_idx)
[1, 2, 3, 0]
>>> print(lists[result_idx])
[[59.      88.      58.      1.      0.58816124]
 [42.      67.      49.      67.      0.55697901]
 [41.      55.      56.      16.      0.36570904]
 [35.      40.      54.      59.      0.35447849]]
```

运行代码验证：【结果一致】

