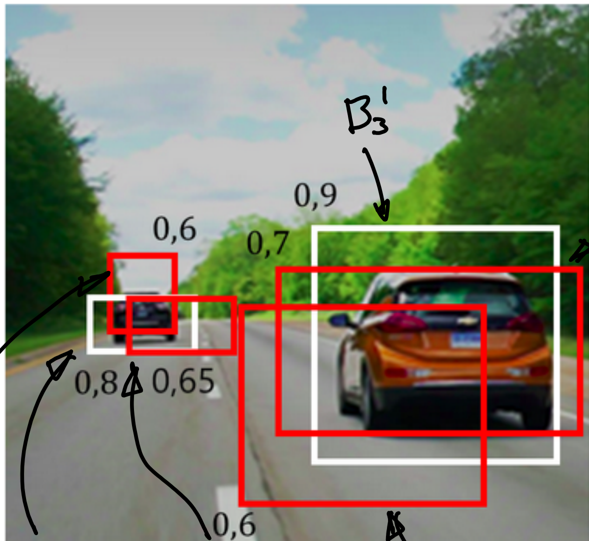
Handout On Bounding Box Selection Algorithm

71L

B_3'

B_2'



$(B_1, 0.6)$ $(B_3, 0.8), (B_2, 0.65)$ B_1'

$B: (B_1; 0.6), (B_2, 0.65), (B_3, 0.8)$

Given Image with 2 Objects, multiple Bounding Boxes. $B = \{(B_1, 0.6), (B_2, 0.65), (B_3, 0.8); (B_1', 0.6), (B_2', 0.7), (B_3', 0.9)\}$, e.g. (B_i, C_i)

Sol: Step1. Pick the B_i or B_i' with the Highest confidence Level, so

$(B_3', C_3') = (B_3', 0.9)$

Step2. Find IOU.

$IOU_{B_3'B_1} = \phi$, Keep B_1 in the collection

Similarly, $IOU_{B_3'B_2} = IOU_{B_3'B_3} = \phi$, Keep B_2, B_3.

Since $IOU_{B_3'B_2'} = \Delta > IOU_{B_3'B_1'} = \Delta - \Delta'$

Delete B_2' from the collection.

$IOU_{B_3'B_1'} \neq \phi$, But Smaller Confidence

$C_1' < C_3'$, so B_3' is selected And placed in the final collection, And Delete B_1'.

$B_{NMS} = \{(B_3', C_3')\}$

Step3, Select B_i or B_i' with the next Highest Confidence C_i or C_i', from Updated Collection

$B = \{(B_1, C_1), (B_2, C_2), (B_3, C_3)\}$

So $(B_3, 0.8) = (B_3, C_3)$ is selected.

Step4. Compute $IOU_{3,1} = \Delta$, $IOU_{3,2} = \Delta + \Delta'$

$\because IOU_{3,2} > IOU_{3,1} \rightarrow$ Discard B_2,

And $\because C_1 < C_3$, Delete C_1, place B_3 into the final selection

$B_{nms} = \{(B_3', C_3'), (B_3, C_3)\}$, update the collection

$B = \{\phi\}$. Done //

Appendix A. Pseudo Code from Another Reference

https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
    $B_{nms} \leftarrow \emptyset$ — **Initialize empty set**
    for $b_i \in B$ **do** => **Iterate over all the boxes**
        $discard \leftarrow \text{False}$ — **Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded**
        for $b_j \in B$ **do** — **Start another loop to compare with b(i)**
            if $\text{same}(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then** — **If both boxes having same IOU**
                if $\text{score}(c, b_j) > \text{score}(c, b_i)$ **then**
                    $discard \leftarrow \text{True}$ — **Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.**
        if not $discard$ **then** — **Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.**
            $B_{nms} \leftarrow B_{nms} \cup b_i$
    return B_{nms} — **Do the same procedure for remaining boxes and return the final list**

(END)