Confluence: A Robust Non-IoU Alternative to Non-Maxima Suppression in Object Detection

Andrew Shepley,
Greg Falzon, *Member, IEEE,*
and Paul Kwan, *Senior Member, IEEE,*

***Abstract*—This paper presents a novel alternative to Greedy Non-Maxima Suppression (NMS) in the task of bounding box selection and suppression in object detection. It proposes Confluence, an algorithm which does not rely solely on individual confidence scores to select optimal bounding boxes, nor does it rely on Intersection Over Union (IoU) to remove false positives. Using Manhattan Distance, it selects the bounding box which is closest to every other bounding box within the cluster and removes highly confluent neighboring boxes. Thus, Confluence represents a paradigm shift in bounding box selection and suppression as it is based on fundamentally different theoretical principles to Greedy NMS and its variants. Confluence is experimentally validated on RetinaNet, YOLOv3 and Mask-RCNN, using both the MS COCO and PASCAL VOC 2007 datasets. Confluence outperforms Greedy NMS in both mAP and recall on both datasets, using the challenging 0.50:0.95 mAP evaluation metric. On each detector and dataset, mAP was improved by 0.3-0.7% while recall was improved by 1.4-2.5%. A theoretical comparison of Greedy NMS and the Confluence Algorithm is provided, and quantitative results are supported by extensive qualitative results analysis. Furthermore, sensitivity analysis experiments across mAP thresholds support the conclusion that Confluence is more robust than NMS.**

***Index Terms*—Detectors, Feature Extraction, Neural Networks, Non-Maxima Suppression, Object Detection, Proposals, Confluence**

I. Introduction

ONE of the principle areas of research interest in computer vision is object detection. It has an extensive range of applications, in areas such as surveillance [1], autonomous vehicles and robotics [2], monitoring of livestock and wildlife [3], image and video analysis [4], amongst others. Since the advent of ImageNet [5] in 2009, data driven deep learning object detectors have been at the forefront of research, and have achieved previously unattainable levels of accuracy and recall.

The majority of Deep Convolutional Neural Networks (DCNN) based object detectors, including RetinaNet [6], Faster-RCNN [7] and Mask R-CNN [8] generate category independent region proposals of varying sizes and scales. A classification network assigns each proposal class specific confidence scores. Localization is improved via regression of proposals. These frequently converge to the same regions of interest (RoI), especially when an object detector is highly confident in the presence of an object within the RoI. This causes clustering of proposed bounding boxes around areas of interest within an image, as shown in Figure 1.

YOLOv3 [9] is based on a different paradigm. Rather than using sliding windows of varying sizes and scales, it partitions an image into sections, and assigns class-specific confidence scores to each section. Low confidence sections are eliminated, bounding boxes are generated from the higher confidence sections and NMS is used to select the optimal bounding box. This pipeline generates far fewer bounding boxes than region-based proposal networks, which renders it far more efficient.

Fig. 1. Top: Raw output from RetinaNet. Left: Confluence output. Bottom Right: Greedy NMS. Confluence embraces the tendency of DCNNs to return numerous bounding boxes within areas of interest. It uses this confluence to select a box superior to the maxima. Note the sub-optimal bounding box returned by NMS has a confidence of 96.4%, (woman and boy) while the two optimal bounding boxes returned by Confluence have confidence scores of 85.9% (woman) and 88.2% (boy). Despite their lower scores, they were selected by Confluence, as they fall within the zones of dense confluence. Also note Greedy NMS suppressed a true positive surrounding the girl due to its harsh overlap threshold. In contrast, Confluence returned one box for each object.

This work was funded by an Australian Research Scholarship.

It is then the role of Non-Maxima Suppression (NMS) to refine these detections by selecting optimal bounding boxes to represent each object, whilst suppressing false positives. It is imperative that true positives are not removed in this process. Figure 1 demonstrates the necessity of this step. It illustrates the raw output of RetinaNet, prior to the application of Greedy NMS, followed by the output of NMS. Greedy NMS is generally considered to be the preferred solution to resolve this problem, and is frequently used in state-of-the-art object detectors [10], [11], [12] including RetinaNet-ResNet50 [6], Yolo-V3 [9], Mask R-CNN [8], Faster R-CNN [7] and R-FCN [13].

II. Related Work

NMS has been an algorithm of significant importance in computer vision for approximately 50 years, being relied on for a plethora of tasks in both object detection [14], and other areas such as face recognition [15], keypoint detection [16], and edge detection [17]. Although NMS is used in both single and two-stage detectors, its effectiveness is impaired by its reliance on a hard-coded, arbitrary IoU threshold, and maxima confidence scores which may not be optimal. Various versions of NMS have been proposed in an attempt to meet the needs of modern object detectors. In face detection [18], bounding boxes were partitioned into clusters using an overlap threshold based on Intersect Over Union (IoU). IoU essentially computes the ratio of the overlapped area of two bounding boxes, over the union area shared by the two boxes. The average coordinates of each box in the cluster was then used to select an optimal box. This approach was adapted by [19] in a human detection task, to produce what is now known as Greedy NMS. Greedy NMS improved accuracy by first sorting candidate bounding boxes based on their confidence scores, from highest to lowest. It selects the bounding box with the highest confidence score, and then suppresses all bounding boxes which have an IoU above a predefined threshold, with the selected box. This process is repeated until there are no bounding boxes within the candidate set. It was demonstrated that this approach was superior, due to its greater average precision compared to other variants of NMS. It remains to this day the de-facto solution for generic object detection [14].

However, due to its reliance on IoU and individual confidence scores, Greedy NMS still suffers the same shortcomings as earlier versions of NMS. Recently, with the advent of powerful and accurate DCNN based object detectors, the sub-optimal performance of NMS has become a problem. Retention of false positives and removal of true positives in high occlusion settings imposes constraints on automated computer vision systems in difficult scenarios such as crowds [20], and impairs the adoption of computer vision in use cases where safety is imperative, such as embedded vision cameras in smart cars [21].

Consequently, researchers have endeavoured to improve NMS by adapting it, or proposing alternatives. These adaptations and alternatives can be generally classified as relying on clustering [22], [20], [23], spatial co-occurrence [24], [25], [26], [27], [28], hough transform methods [29], [30], [31], and end-to-end based learning approaches [10], [32], [33], [34], [35].

A. 'Soft' NMS Adaptations

A recent adaptation of NMS which has improved its performance is [14] which proposed Soft-NMS. Rather than suppressing highly-overlapped bounding boxes, their confidence scores are decreased. Like Greedy NMS, the bounding boxes are sorted based on their confidence scores. It then selects the highest scoring bounding box (M). Unlike Greedy NMS, bounding boxes which have a significant overlap with M are not immediately suppressed, instead their confidence scores are decayed. This work is extended by [36] by introducing variable voting within Soft-NMS. Localization confidence is improved by adjusting the location of M. This is achieved by assigning higher weights to bounding boxes that are more certain, and and have a high overlap. Likewise, bounding boxes with high variance, and a low IoU with M are assigned lower weights. Although this approach claimed to improve mAP, it is still heavily reliant on IoU, and is degraded by an additional tunable parameter to control variable voting.

B. Clustering Approaches

Many adaptations or alternatives to Greedy NMS rely on IoU based merging of clusters of bounding boxes around a region of interest. However, these approaches show inconsistent improvements, and a persistent dependence on IoU. An agglomerative clustering approach was applied by [22] to human detection using a poselet framework. Bounding boxes with an IoU above an arbitrary threshold were merged to eliminate false positives, and return one prediction box for a given object. This process was enhanced by the use of linear regression, and the calculation of a score for a cluster related to bounding boxes belonging to one object. However, this approach is heavily reliant on IoU, depends on 2D keypoint annotations, and is computationally expensive. In contrast, our approach only processes information returned by the object detector, and does no rely on IoU.

Another approach which differs from NMS, yet relies on IoU was proposed by [20]. The overlap between each pair of detection boxes is computed using an unsupervised clustering method based on message passing, known as affinity propagation clustering (APC) [37]. APC groups proposed bounding boxes of the same class based on their IoU. It then selects exemplars for each cluster, returning the exemplars as the final detection boxes. This method performs well in object class detection, and has been used for multi-class detection in [23]. The issue of large scale object detection with many classes was addressed by [23]. It builds on the contribution of [20] but operates on inter-class rather than intra-class bounding boxes. It claims to improve results by using a similarity measure to cluster objects that are semantically similar. Again, this approach is reliant on IoU. It also addresses a different issue, that of optimisation of class labels rather than selection of an optimum bounding box within a given class.

An alternative to NMS is [38] which accumulates rather than suppresses bounding boxes to increase detection confidence.

The bounding boxes with maximum class scores are merged to produce a single high-confidence box. If a bounding is highly coherent, i.e. it has a high IoU with many other boxes, its confidence score is boosted. Bounding boxes which have low confidence scores, and low coherence are classified as false positives, and eliminated. This approach claims to be more robust than greedy NMS, however, it still depends on an arbitrary IoU overlap threshold to select final prediction boxes. This reduces its robustness to box size, and increases computational cost.

C. Spatial Co-occurrence

Co-occurrence is used in both [24] and [25] to prune false positives. Semantics and image information are employed to select optimal bounding boxes. In [25] spatial-object interactions are represented using learned statistics, allowing an IoU threshold to be adjusted based on the probability of an object being present within an image, at a given location. These approaches use NMS selectively, relying on spatial co-occurrence and contextual queuing to lower or increase the IoU threshold. This is an adaptation of, rather than an alternative to NMS.

This idea was applied by [26] to high-density crowds, suppressing bounding boxes based on estimations of crowd density. Similar concepts are seen in [27] and [28] in the context of pedestrian detection. In contrast, our proposed algorithm does not rely on image information, or semantics and is not hand-crafted to accommodate high occlusion. Rather, occlusion is handled using the natural coherence of intra-object bounding boxes returned by the object detector.

D. Hough Transform Approach

Some approaches aim to bypass NMS altogether by employing Hough voting. A Hough voting based detection framework was adopted by [29] using a tabu-inspired enumeration method. The inherent fragility of Hough voting based NMS was addressed by [30] by using a probabilistic framework to bypass the issue of multiple peak identification applied in a pedestrian detection context, claiming improvements in accuracy. This approach was also employed by [31] in a general random field setting to avoid the problem of reliance on an overlap threshold owing to high-density objects. However, in comparison to Greedy-NMS, these approaches result in sub-optimal performance [39]. In contrast, our approach does not rely on image information or labelling.

E. End-to-end Learning Based Approaches

Some approaches, such as [10] and [32] integrate NMS within the training pipeline. One notable approach is [10], which proposes GNet, a neural network that uses the concept of message passing between neighbouring bounding boxes, whereby changes in bounding box representations are learned based on the 'negotiations' between bounding boxes to decide which bounding box will represent which object. Although this approach only uses bounding boxes and scores as input, it is a highly complex network which requires significant amounts of training data. In contrast, our approach does not require any training, and can be easily incorporated into systems that currently use NMS, without the need for structural changes.

NMS is bypassed altogether in [33] which generates a sparse set of detections based on decoding of image content, claiming improvements in recall and more effective handling of occlusion. However, additional post-processing is still required to eliminate false positives on the boundaries of neighbouring predictions. A learning based framework is also used by [34] in the context of pedestrian detection. This approach uses a determinantal point process combined with individualness prediction scores to optimally select final detections. Detections are modelled using similarity with other detections. High scoring detections with low probabilities are then selected as final detections. The output of NMS is used by [35] to improve network training. Although these models are conceptually simpler due to their 'end-to-end' nature, and avoidance of post-processing, they generally achieve equivalent results to the traditional NMS approach.

F. NMS Attack

One recently researched shortcoming of NMS and its variants is its vulnerability to IoU based attack. An adversarial example attack was proposed by [21] to cause absolute failure of NMS within end-to-end object detection models including RetinaNet and YOLOv3. It successfully compromised the ability of NMS to filter redundant bounding boxes by manipulating the dimensions of bounding boxes overlaps. This attack resulted in a reduction in mean average precision to 0%, and an increase in the false positive rate to 99.9%. As noted by the authors, this form of attack would be fatal to the many autonomous devices such as self-driving cars, which rely on NMS. Our proposed algorithm is not susceptible to this attack as it does not rely on IoU or the size of bounding boxes at all.

III. Methodology

The proposed method is called Confluence. Our method derives its name from the confluence of bounding boxes returned by an object detector when an object is detected. Rather than treating the excessive proposals as a problem, Confluence embraces them as a way of identifying the most optimal bounding box. This is achieved by identifying the bounding box which is most confluent with the other bounding boxes, i.e. the box that best represents the collective intersect of the other boxes within the cluster.

Confluence is a two-staged algorithm which retains optimal bounding boxes, and removes false positives. Retention is achieved using a confidence weighted Manhattan Distance inspired proximity measure to evaluate bounding box coherence. The second stage involves removal of all bounding boxes which are confluent with the retained bounding boxes.

A. Manhattan Distance

The Manhattan Distance (MH) or L_1 norm, is the sum of the vertical and horizontal distances between two points

[40]. The MH between $u = (x_1, y_1)$ and $v = (x_2, y_2)$ can be represented by the following equation:

$$MH_{(u,v)} = | \ (x_1 - x_2) \ | + | \ (y_1 - y_2) \ | \quad (1)$$

It is a distinct characteristic of both traditional and modern DCNN-based object detectors to return large numbers of confident detections, forming clusters of coherent bounding boxes around locations of interest in an image.

We propose that the proximity P between any two bounding boxes can be represented by the sum of the MH between the upper left $u = (x_1, y_1), v = (x_2, y_2)$, and lower right $m = (p_1, q_1), n = (p_2, q_2)$ coordinate pairs as follows;

$$P_{(u,v,m,n)} = MH_{(u,v)} + MH_{(m,n)} \quad (2)$$

A diagrammatic representation of the proximity measure is provided by Figure 2.

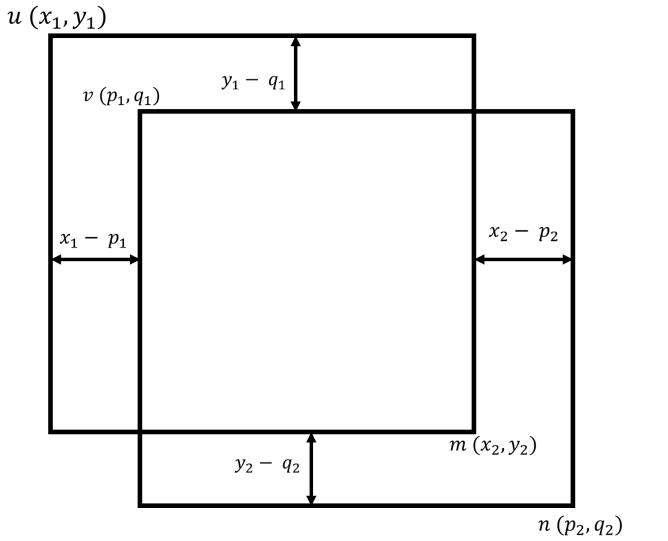

Fig. 2. Proximity calculation of a detection with coordinates x_1, y_1, x_2, y_2 with another detection represented by p_1, q_1, p_2, q_2

A small P value would denote highly confluent boxes, whilst a high P value would indicate boxes are not attributable to the same object - they may be somewhat overlapping, or completely disjoint. Thus, it can be inferred that if the P of a given box is measured against every other box in a set of bounding boxes, it will provide a measure of its confluence with every other box. This calculation will involve a large number of comparisons when there is a dense confluence of bounding boxes.

Thus, a bounding box surrounded by a dense cluster of bounding boxes will be characterized by very low P values, in comparison with a bounding box which is not surrounded by competing bounding boxes, which could be correctly categorized as an outlier. In effect, this provides a measure of the object detector's confidence in the presence of an object at a given location. On this basis, we propose that that the bounding box b with the lowest intra-cluster P values represents the most confident detection for a given object.

Notably, this theory overcomes an issue faced by NMS and its alternatives - in situations where the highest scoring bounding box is sub-optimal in comparison with another lower scoring bounding box, NMS returns the sub-optimal bounding box, as illustrated by Figure 1. In contrast, the P measure allows for the bounding box which is most confluent with all other bounding boxes assigned a given object to be favoured, making it more robust.

B. Normalization

The theory outlined in the previous section operates effectively in circumstances where bounding boxes are of similar size. However, in practice, objects and their corresponding bounding boxes, will be of varying sizes. This poses a problem when regulating bounding box retention or removal using a hyper-parameter based on confidence-weighted P. This is because a trade-off between removing large false positives and retaining small true positives would need to be reached.

To overcome this issue, a normalization algorithm was used to scale the bounding box coordinates between 0 and 1, whilst preserving their relationship with each other. The normalization algorithm transforms each coordinate (x_i, y_i) as follows:

$$X = \{x_1, x_2, p_1, p_2\}$$
$$Y = \{y_1, y_2, q_1, q_2\}$$

$$norm(x_i, y_i) = \left(\frac{x_i - \min(X)}{\max(X) - \min(X)}, \frac{y_i - \min(Y)}{\max(Y) - \min(Y)} \right) \quad (3)$$

Normalization allows intra-object and inter-object bounding boxes to be distinguished by making the relationship between any 2 large intra-object bounding boxes comparable to any 2 small inter-object bounding boxes, as illustrated by Figure 3.

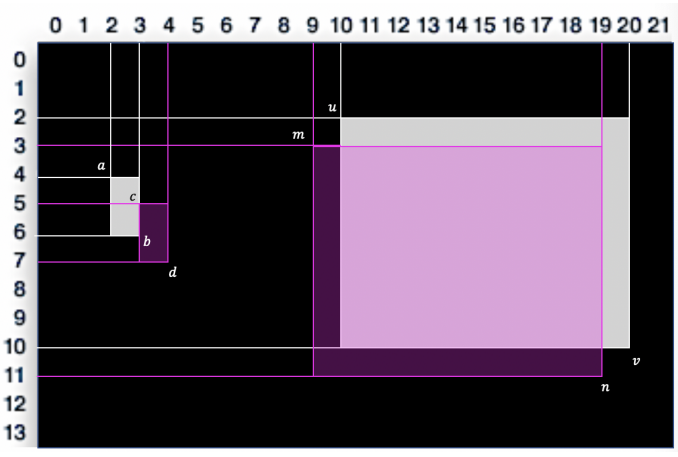

Fig. 3. Image co-ordinate plane illustrating the need to normalize bounding box co-ordinates.

It is apparent in Figure 3 that the two large bounding boxes on the right denote the same object. In contrast, the two small bounding boxes on the left denote two separate objects. However, when the P calculation is performed, the same value is obtained, as follows:

$$P_{(u,v,m,n)} = \mid (10-9) \mid + \mid (2-3) \mid + \mid (20-19) \mid + \mid (10-11) \mid = 4 \quad (4)$$

$$P_{(a,b,c,d)} = \mid (3-2) \mid + \mid (4-5) \mid + \mid (4-3) \mid + \mid (6-7) \mid = 4 \quad (5)$$

This poses the problem of distinguishing between bounding boxes belonging to the same or distinct objects. Normalization addresses this issue by retaining the overlap relationship between the bounding boxes, yet ensuring that any 2 bounding box relationships can be compared. For example, after applying equation (3) to each pair of bounding boxes illustrated in Figure 3, the coordinates belonging to the large intra-object bounding boxes are transformed from $u(10,2)$, $v(20,10)$, $m(9,3)$, $n(19,11)$ to $u(0.091,0)$, $v(1,0.889)$, $m(0,0.111)$, $n(0.909,1)$. When equation 2 is applied, it returns a value of 0.404. Similarly, the bounding boxes belonging to the small inter-object bounding boxes are transformed by normalisation from $a(2,4)$, $b(3,6)$, $c(3,5)$, $d(4,7)$ to $(a(0,0)$, $b(0.5,0.667)$, $c(0.5,0.333)$, $d(1,1))$. When equation 2 is applied, it returns a value of 1.666. Thus normalization allows the difference between intra-object and inter-object bounding boxes to be distinguished.

C. Intra-Cluster Retention and Removal

As all coordinate pairs are normalized between 0 and 1, any pair of intersecting bounding boxes will have a proximity value below 2. Thus, if the P value of any two bounding boxes is below 2, it is assumed that they belong to the same cluster, and therefore refer to the same object, or to one or more high density objects. Once clusters are identified, the optimal intra-cluster bounding box is found, by sorting the P values in ascending order. The bounding box with the smallest proximity at the nth position is taken to be the most confluent bounding box, and is retained.

Analysis of the intra-cluster gradient of P values then allows the most confluent bounding box to be selected. Graphing of P values allows visualization such that the difference between intra-object and inter-object bounding boxes is evident due to the blob-like nature of clustering, as shown by Figures 4 and 5. Each horizontal blob represents an object. Confluence selects the bounding box which best represents every other box within a given blob. Essentially, this means it selects a box within a data range characterized by a gradient approaching zero. This relationship is clear even in high density images, where distinct objects will be represented as distinct blobs.

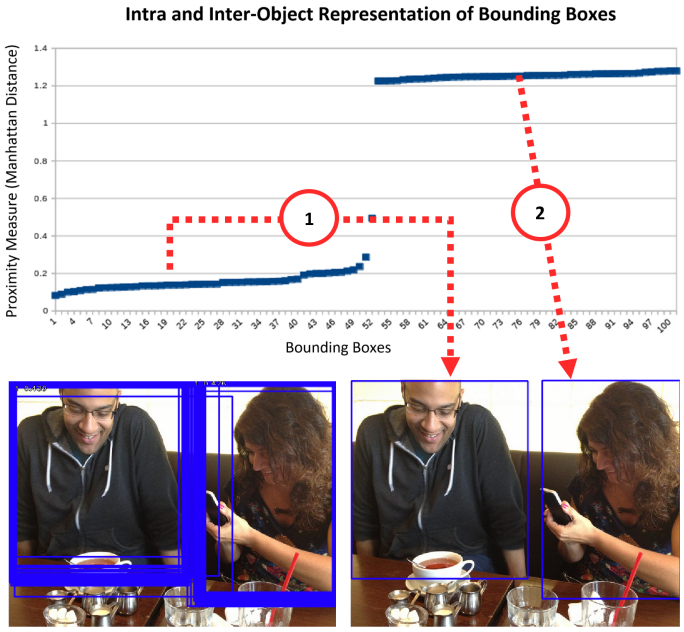

Fig. 4. Left: Raw RetinaNet output. Right: Confluence output. P values measured from a randomly chosen bounding box at $(0,0)$, with respect to every other bounding box in a set of approximately 100 bounding boxes. It is evident that normalizing the relationship between bounding boxes prior to calculation of P values allows the distinction between objects to be distinguished. Highly proximate values will cluster in a linear fashion, with a gradient close to 0.

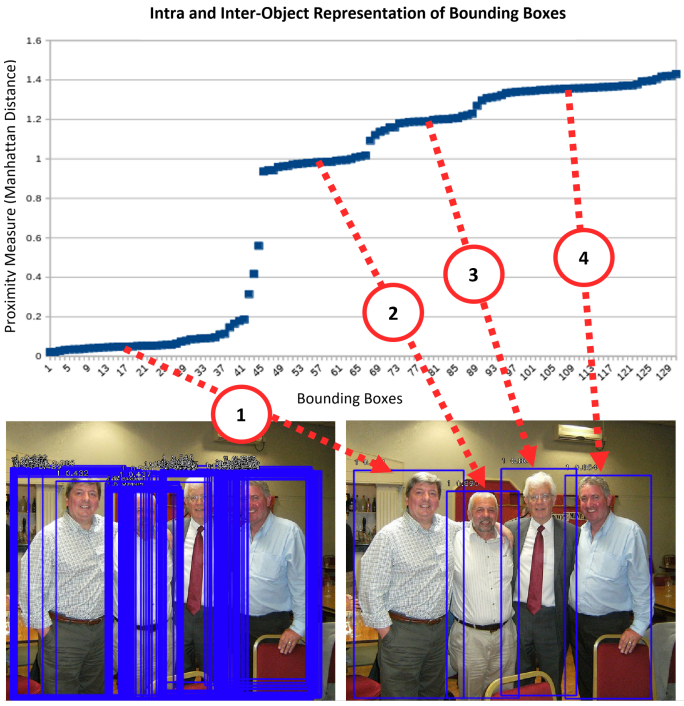

Fig. 5. Left: Raw RetinaNet output. Right: Confluence output. Even when objects are at close proximity, the Confluence algorithm is capable of clearly distinguishing between intra-object and inter-object bounding boxes. Note, one bounding box from each horizontal section in the graph is selected to represent each person in this image.

Once the most confluent bounding box is selected, all intra-cluster bounding boxes with a proximity value below a predefined threshold are removed. This process is repeated recursively until all bounding boxes have been processed.

D. Confidence Score Weighting

NMS uses a single confidence score returned by the object detector as the sole means by which an 'optimal' bounding box is selected. In contrast, Confluence assesses the optimality of a given bounding box b by considering both its confidence score c and its P values with competing bounding boxes. Weighted Proximity WP is achieved by dividing the P of b by its confidence score as follows:

$$WP_{(u,v,m,n)} = \frac{P_{(u,v,m,n)}}{c} \tag{6}$$

As c is a value which lies between 0.05 and 1, this in effect provides a bias in favour of high confidence boxes by artificially reducing the value of the WP (Note that all bounding boxes with confidence scores below 0.05 are not considered). Conversely, the WP value of low confidence boxes will be greater. This increases the likelihood of a high confidence box being selected, as bounding boxes are chosen based on small WP values.

This algorithm is based on the principle that a powerful classifier can be constructed by using the sum of weaker individual classifiers [41], [42]. Each individual WP value is a weak classifier on its own, but when these weak classifiers are collectively interpreted, they provide a powerful means to classify a bounding box as either confident - via high confluence, or not confident, via disparate positioning with respect to other bounding boxes. In essence, this provides a vote of confidence by the object detector on which bounding box best represents every other bounding box assigned to an object. Our experimental results presented in Table 1 suggest that this is a reliable means to accurately identify true positives, whilst effectively minimizing false positives. This allows achievement of optimal mAP and recall values.

E. Implementation

The Confluence Algorithm was implemented in Python. The pseudo-code is outlined in Algorithm 1. Code will be made available on GitHub. The main steps of the Confluence algorithm are:

1) Algorithm 1, line 1: Variables B_F, S_F and C_F are sets created to store the bounding boxes and corresponding scores and class labels which will be returned to be drawn on the image.
2) Algorithm 1, line 2: The algorithm loops through each class separately, which gives it the ability to handle multi-class object detection. For each class, it selects n bounding boxes, each representing one object.
3) Algorithm 1, line 4: The variables b_s, and s_s are defined to temporarily store bounding boxes, and their corresponding scores while the optimal box for the class being processed is selected.

Algorithm 1 Confluence

Input: B = {b_1, .., b_N}, S = {s_1, .., s_N}, C = {c_1, .., c_N}, M_D, P_S
B is the list of initial bounding boxes
S is the set of corresponding detection scores
C is the set of corresponding class predictions
M_D is the Manhattan Distance threshold hyper-parameter (experimental results indicate optimal range is from 0.3-0.7)
I_P is the size of the image

begin:

1: $B_F, S_F, C_F \leftarrow \{\}, \{\}, \{\}$
LOOP Process
2: **for** c_i in C **do**
3: **while** $B \neq empty$ **do**
4: $b_s, s_s \leftarrow \{\}, \{\}$
5: $optimalConfluence \leftarrow I_P$
6: **for** b_i, s_i in B, S **do**
7: $confluence(b_i) \leftarrow 0$
8: $\forall b \in B$ except $b = b_i$
9: $nb, nb_i, \leftarrow normalize(b, b_i)$
10: $P \leftarrow proximity(nb, nb_i)$
11: **if** $P < 2$ **then**
12: $confluence \leftarrow confluence \cup Proximity * s_i$
13: **end if**
14: **if** $confluence < optimalConfluence$ **then**
15: $optimalConfluence \leftarrow confluence$
16: $b_s, s_s \leftarrow b_i, s_i$
17: **end if**
18: **end for**
19: $B_F, S_F, C_F \leftarrow B_F \cup b_s, S_F \cup s_s, C_F \cup c_i$
20: $B, S \leftarrow B - b_s, S - s_s$
21: $\forall b \in B$
22: $P \leftarrow proximity(b, b_s)$
23: **if** $P < M_D$ **then**
24: $B, S \leftarrow B - b, S - s$
25: **end if**
26: **end while**
27: **end for**
28: **return** B_F, S_F, C_F

4) Algorithm 1, line 5: The variable $optimalConfluence$ is initialized to the size of the image.
5) Algorithm 1, lines 6-18: The algorithm then loops through all the bounding boxes b_i, comparing each bounding box to every other bounding box in the set B.
6) Algorithm 1, lines 9-10: Coordinate relationships are normalized, followed by the proximity calculation.
7) Algorithm 1, line 11: As previously mentioned, if the proximity calculation value is below 2, the bounding boxes are disjoint, and will be treated as separate objects. This condition restricts the P value for b_i to values below 2.
8) Algorithm 1, lines 12-16: An optimal bounding box is selected via convergence on the minimum confidence

weighted confluence value.

9) Algorithm 1, lines 19-20: Once the optimal bounding box is selected, it is added to B_F, to be returned as a final detection, along with its corresponding class, and confidence value, and removed from the sets B, S.
10) Algorithm 1, lines 21-24: Subsequently, all bounding boxes in B which have a proximity with the optimal bounding below the predefined M_D hyper-parameter, are removed.
11) Steps 3-10 are performed recursively until all bounding boxes have been processed.

The computational complexity of each step of Confluence is $O(N)$, where the input set of bounding boxes is N. This is due to the calculation of the normalized proximity score. As this measure is computed for each bounding box against every other bounding box, Confluence has an overall computational expense of $O(N^2)$, which is the same as Greedy NMS. Due to the reduction in the size of the set of bounding boxes by recursion, computational time is not significant.

IV. Datasets, Models and Evaluation Metrics

Experimental results presented in this paper were collected on the publicly available 2017 MS-COCO mini validation (minival) datset [43] and 2007 PASCAL VOC [44] datasets. These datasets were chosen as they were not included in the model training sets. The COCO minival set contains 5000 images and 80 classes. The PASCAL VOC train/test/validation dataset includes a total of 9,963 images containing 20 classes, and 24,640 annotated objects. These datasets are used extensively to train and evaluate deep learning object detection models [6], [9], [8]. The mAP calculations for the COCO minival were obtained using the COCO-style evaluation metrics using the standard COCO API. The results on Pascal VOC were obtained using the 2012 Pascal mAP calculation outlined in [45].

Evaluation of Confluence in comparison to Greedy-NMS was achieved using three state-of-the-art object detectors; RetinaNet-ResNet50 [6], Yolo-V3 [9], and Mask R-CNN [8]. We did not conduct model training, instead selecting publicly available pre-trained models in all cases. The models made available by their authors, were trained on the MS-COCO 2017 training set using default configurations outlined in their respective papers. They were evaluated using both Pascal VOC and the COCO API. As discussed by [46], the classes within the Pascal VOC dataset are a subset of MS-COCO. It was therefore inferred that the MS-COCO trained models were sufficiently sophisticated to obtain test results on the Pascal VOC dataset, without further training or fine-tuning. Thus no training was necessary to evaluate Confluence against Greedy NMS.

Performance measurement was conducted on both the COCO minival and Pascal VOC datasets. Default RetinaNet settings including an NMS suppression threshold of 0.3, and detector confidence of 0.05 were used [6]. The detector confidence used in the YOLOv3 model was preset to 0.3. This was lowered to 0.05 to maximize recall. YOLOv3 and Mask-RCNN also used default NMS thresholds of 0.3. In calculating the mAP using the COCO API, maximum detections per image was set to 100, which is the default setting for the COCO evaluation metric. The default IoU of 0.5 was used in the calculation of the PASCAL VOC mAP.

V. Results

In this section, we provide performance results on both MS-COCO minival and PASCAL VOC 2007 datasets. Sensitivity analysis results are presented to demonstrate the robustness of Confluence, by examining changes in performance of Greedy NMS and Confluence across variations in the overlap threshold parameter using the COCO minival dataset.

1) MS-COCO: In Table 1 we compare the performance of Confluence against Greedy NMS on MS-COCO minival using RetinaNet, YOLOv3, and Mask-RCNN. The NMS overlap threshold hereafter referred to as O_T was set to 0.3 in accordance with the default object detector settings. M_D was set to 0.45.

It is clear that Confluence improves object detector performance across all three networks, particularly on the more stringent AP@0.5:0.95 mAP calculation. We achieve improvements of 0.7%, 0.7%, and 0.3% on each RetinaNet, YOLOv3, and Mask-RCNN. Gains in recall are more significant with improvements of 2.3%, 2.5%, and 1.4% on RetinaNet, YOLOv3, and Mask-RCNN, which is significant for the MS-COCO dataset and evaluation metric.

2) PASCAL VOC: The same experiments were performed on the PASCAL VOC 2007 dataset, with results provided in Table 2. The PASCAL VOC 2012 evaluation metric [44] was used. Similarly to MS-COCO, Confluence achieved gains in performance of 0.68% and 0.63% on RetinaNet and YOLOv3. It was interesting to note that for classes which frequently involve high occlusion, we achieved the greatest increase in mAP. For example, the person class was improved from 81.85% using Greedy NMS, to 84.13% using Confluence.

These improvements in performance were consistent across both proposal based object detectors such as RetinaNet and Mask-RCNN, and linear function based detectors such as YOLOv3. Furthermore, Confluence was engineered to take the same parameters as Greedy NMS, which means it can easily be integrated into existing object detectors with minimal modifications.

A. Sensitivity Analysis

This subsection presents the results of sensitivity analysis of the O_T in comparison to M_D using the COCO mini-val on RetinaNet. Both parameters were varied, and the AP@0.5:0.95 was calculated, see Figure 6. Note that the optimal range for NMS lies between 0.3-0.7, while the optimal threshold for Confluence ranges from 0.5-0.8. The reason for this is because a large O_T threshold means only heavily overlapping bounding boxes are removed. As the value of O_T decreases, the number of bounding boxes removed increases. In contrast, high M_D values indicate non-proximity of bounding box borders, while low M_D values indicate highly confluent bounding boxes. Thus the higher the M_D threshold, the more bounding boxes

METHOD	AP 0.5:0.95	AP @ 0.5	AP small	AP medium	AP large	Recall @ 10	Recall @ 100
RetinaNet-50 [6] + NMS	32.3	48.0	15.3	35.2	46.0	39.4	39.6
RetinaNet-50 [6] + Confluence	**33.0**	**48.3**	**15.6**	**36.2**	**46.6**	**41.5**	**41.9**
YOLOv3 [9] + NMS	35.8	63.9	17.2	39.4	53.6	42.2	43.4
YOLOv3 [9] + Confluence	**36.5**	**64.2**	**17.8**	**40.5**	**54.0**	**43.8**	**45.9**
Mask R-CNN [8] + NMS	30.9	48.1	16.2	35.7	43.4	38.0	38.9
Mask R-CNN [8] + Confluence	**31.2**	**48.2**	**16.4**	**36.1**	**43.6**	**39.1**	**40.3**

TABLE I
Results on MS-COCO on the 2017 validation set (5K images) for RetinaNet, YOLOV3, and Mask R-CNN. All results were collected with NMS threshold of 0.3 overlap compared with Confluence at a M_D threshold of 0.45

METHOD	AP@0.5	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
RetinaNet [6] + NMS	74.26	86.18	78.66	72.15	60.24	58.57	80.81	81.37	90.35	54.43	82.04	54.26	85.07	86.68	84.10	78.68	48.94	70.71	70.22	85.75	76.06
RetinaNet [6] + Conf	74.94	87.44	80.67	73.22	59.98	59.52	81.51	82.03	91.02	55.07	82.69	52.93	86.04	87.33	83.08	80.78	49.82	72.25	69.81	86.58	76.10
YOLOv3 [9] + NMS	77.57	88.93	83.44	72.79	57.90	67.39	87.13	82.50	88.07	66.23	75.51	66.97	87.33	88.97	85.40	81.85	54.24	73.39	74.19	88.88	80.38
YOLOv3 [9] + Conf	78.20	90.14	84.68	73.22	58.13	67.95	87.56	83.29	88.90	67.04	76.84	67.89	86.99	89.42	85.83	84.13	55.09	74.43	72.92	89.16	80.41
Mask R-CNN [8] + NMS	75.66	88.03	80.67	75.51	60.16	67.86	79.37	73.76	89.91	58.79	80.87	55.13	86.64	86.17	84.39	83.05	52.63	78.73	67.78	88.53	75.23
Mask R-CNN [8] + Conf	75.53	88.28	80.53	75.36	60.17	67.58	80.32	72.49	87.67	58.85	81.26	55.57	85.55	87.38	85.26	84.20	51.85	79.19	66.97	87.49	74.69

TABLE II
Results on Pascal VOC across 20 classes (9963 images) for RetinaNet, YOLOv3, and Mask R-CNN. All results were collected with NMS threshold of 0.3 overlap compared with Confluence at a M_D threshold of 0.45

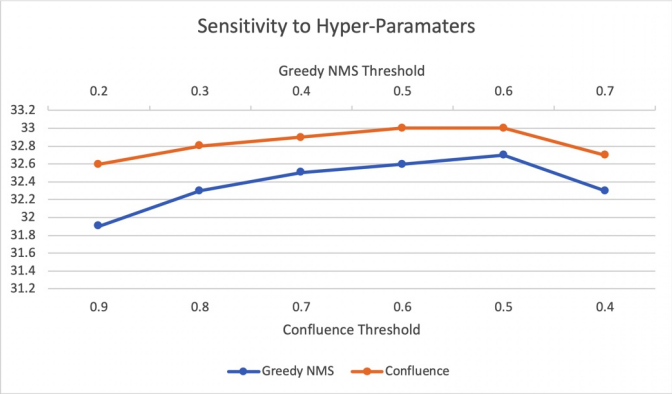

Fig. 6. RetinaNet sensitivity to Greedy NMS and Confluence hyper-parameters.

are removed. This is why the optimum threshold value for Confluence is higher than that used by NMS.

Performance by both algorithms tends to decrease outside these ranges. Note that the variation in AP for Confluence within its optimal range is more stable than NMS, with variation in AP for Confluence being 0.2%, while variation in AP for Greedy NMS is 0.4%. This means that the Confluence algorithm threshold is less sensitive to fluctuations in levels of object density and occlusion, which makes it more robust. As shown by Figure 6 performance of Confluence remains approximately 0.5% better than NMS at all times, even at the optimal NMS threshold.

Furthermore, Table 3 provides results for sensitivity analysis across variations in both O_T and M_D at increasing mAP IoU thresholds. This aims to demonstrate that Confluence consistently returns more accurate bounding boxes, which better fit the ground truth annotations. For example, at AP@0.8-0.95, Confluence achieves AP values ranging between 17.4-17.6% while Greedy NMS achieves AP values between 16.7-16.9%. This trend is consistent at all mAP threshold values. This supports the previously discussed qualitative results.

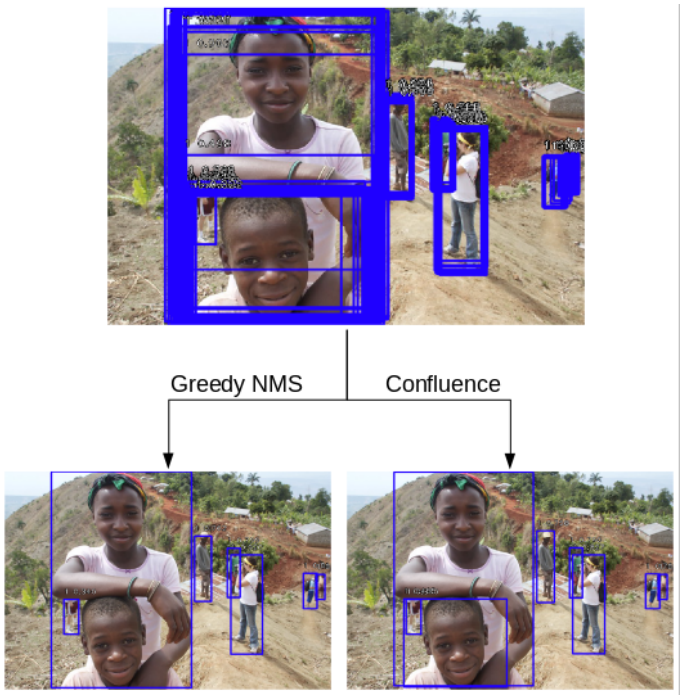

Fig. 7. Top: RetinaNet output. Left: Greedy NMS. Right: Confluence output. Note Greedy NMS labelled the two people in the foreground as one person, whilst Confluence distinguishes them as two separate objects due to the two areas of bounding box confluence.

VI. Discussion

This section will relate the quantitative results presented in the previous section to qualitative comparison of Greedy NMS

O_T	AP@0.50:0.95	AP@0.60:0.95	AP@0.70:0.95	AP@0.80:0.95	M_D	AP@0.50:0.95	AP@0.60:0.95	AP@0.70:0.95	AP@0.80:0.95
0.2	31.9	28.3	23.4	16.7	**0.9**	32.6	29.0	24.2	17.5
0.3	32.3	28.6	23.6	16.8	**0.8**	32.8	29.1	24.3	17.5
0.4	32.5	28.7	23.7	16.9	**0.7**	32.9	29.2	24.3	17.6
0.5	32.6	28.8	23.8	16.8	**0.6**	33.0	29.3	24.4	17.6
0.6	32.7	28.9	23.8	16.8	**0.5**	33.0	29.3	24.4	17.5
0.7	32.3	28.7	23.8	16.8	**0.4**	32.7	29.1	24.3	17.4

TABLE III
Results of sensitivity analysis on RetinaNet across varying COCO mAP thresholds, and variations in NMS O_T and Confluence M_D.

and Confluence. We will explain how and why Confluence returns optimal bounding boxes, using qualitative results to demonstrate why we believe the Manhattan Distance metric is a more appropriate metric to use in bounding box selection and suppression in object detection and localization. We will also provide insight into possible future work to further improve the performance and usability of the Confluence algorithm.

A. *Qualitative Results*

A fundamental issue with IoU based suppression is the elimination of true positives in high density images. Once a high confidence bounding box b is selected, any detection with a sufficiently high overlap with b is removed. In situations where objects are occluded by other objects of the same class, for example, when a person occludes another person as shown in Figures 1 and 7, NMS will often suppress detections denoting true positives.

Figures 1 and 7 show the raw unfiltered output returned by RetinaNet, at a confidence threshold of 0.0%. It is evident by the thick confluence of proposals that all objects are detected. Thus, the aim of NMS is to maximize precision by selecting an optimal detection to represent each true positive, without lowering recall by suppressing true positives. NMS does not achieve this when applied to these images. Its reliance on the maxima confidence score causes it to return sub-optimal bounding boxes, while its IoU dependency causes it to suppress true positives. In contrast, Confluence uses the heavy cluster of bounding boxes as an indicator of the presence of an object, thus returning one bounding box per cluster. This results in both higher recall and precision.

Confluence rewards high confluence, weighted by confidence, interpreting clustering as a vote of confidence by the neural network on the likeliest location of an object in an image. Thus, perhaps the most effective means by which Confluence can be evaluated is by qualitative data. Figures 8, 9, 10, 11 and 12 illustrate qualitative results using images from the COCO dataset. An N_T of 0.3 is used to compare NMS against Confluence, as this produces visually optimal results. Confluence has a M_D of 0.6. RetinaNet was used to generate detections.

1) Improved Bounding Box Selection: As previously discussed, object detectors return many bounding boxes, and their corresponding scores in locations where the probability of an object being present is high. Although selecting the highest confidence score within a cluster of bounding boxes is generally a reliable method by which an optimal bounding box can be selected, there are many instances where the highest score is not the optimal bounding box. A few examples of these situations are provided in Figures 8 and 9.

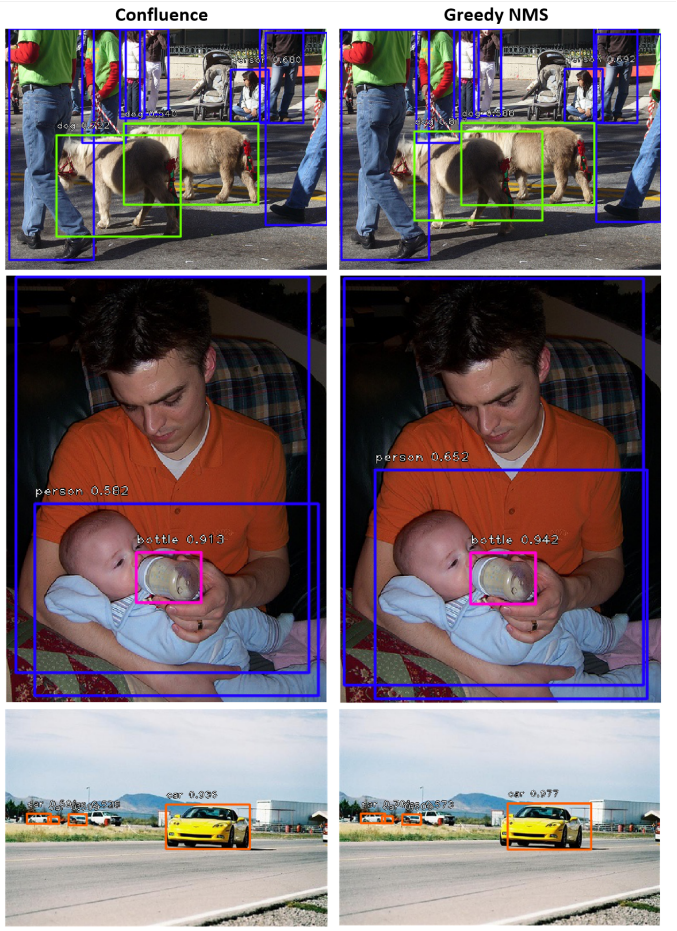

Fig. 8. The highest scoring box is not always optimal. Top row: Confluence at M_D of 0.6. Bottom row: NMS at O_T of 0.3. Even when objects are not occluded, or highly proximate, the highest scoring box is not always optimal. This is true for all classes.

It was interesting to see that many of these instances involved people, at close proximity, such as those shown in Figure 9. NMS simply returns the highest scoring box, even if it is too large. In contrast, Confluence has the capacity to return a more accurate box by taking advantage of the confluence of tighter fitting bounding boxes around each person.

Rather than simply selecting the highest scoring bounding box, Confluence selects the bounding box which not only

Fig. 9. The optimal box does not always have the highest confidence score. Left: Confluence at M_D of 0.6. Right: NMS at O_T of 0.3. Often, a bounding box which is too large is returned in instances where objects, such as people are at close proximity

has a relatively high score, but is also the most coherent with every other bounding box in the cluster. As outlined in the Methodology section, it achieves this by measuring the Manhattan Distance between bounding boxes, weighted by their confidence scores. Consequently, if a highly confluent bounding box is a better representation of the object, it will be returned by Confluence despite having a lower confidence score.

This improvement in bounding box selection is most evident when the object detector is a proposal based DCNN, such as RetinaNet, which has a tendency of returning very dense confluence around objects.

2) Improved Bounding Box Selection Improves Recall: It was interesting to note that due to the use of the maxima by NMS as the sole means by which an optimal bounding box is selected, recall was damaged. For example, Figure 10 illustrates how NMS selects a highest confidence box (87.7% confidence, shown in red), apparently to locate the boy, but this bounding box is sub-optimal in comparison to the bounding box selected by Confluence (82.1% confidence). Due to NMS's selection of the maxima, it suppressed the bounding box allocated to the man standing behind the boy (shown in yellow), as they share a high IoU, thus reducing recall. In contrast, Confluence retains a bounding box located both the man and boy.

3) Suppression of False Positives via Manhattan Distance Improves Accuracy: It was observed that due to suppression of false positives via IoU, in some situations, Greedy NMS suppresses a bounding box which has a high IoU with a

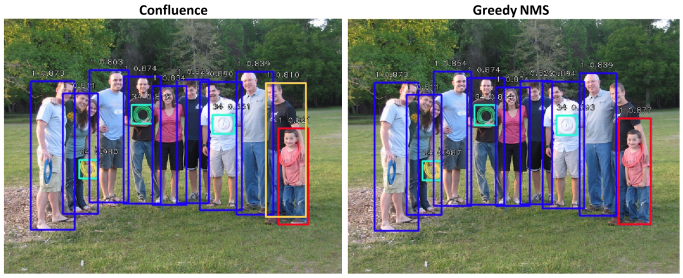

Fig. 10. Reduction in Recall. Note the poor selection by NMS of a bounding box to locate the boy on the far right (red box), and the suppression of the bounding box allocated to the man standing behind him (yellow box).

higher confidence box, even if it correctly locates a second object. NMS is then forced to select a sub-optimal bounding box to locate the second object, due to suppression of the optimal box. For example, Figure 11 shows the output of NMS, and Confluence on the same image. Note that NMS suppresses the optimal, high confidence (66%) box allocated to the giraffe in the background due to its high IoU with the high confidence (94.8%) box allocated to the giraffe in the foreground. It is then forced to return a low confidence (50.7%), sub-optimal bounding box for the giraffe in the background. This issue can be rectified by increasing the NMS IoU overlap threshold, however this comes at a cost - the number of false positives returned increases significantly. In contrast, Confluence uses areas of confluence returned by the object detector to determine whether or not a second object is present, thus making it more robust to this issue. It has the capacity to return optimal bounding boxes without returning false positives.

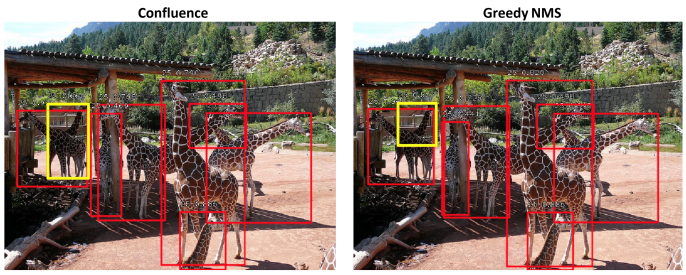

Fig. 11. Left: Bounding boxes returned by Confluence. Right: Bounding boxes returned by Greedy NMS. Note that the optimal bounding box for the background giraffe on the far left (yellow) is suppressed due to IoU, which forces NMS to return a sub-optimal bounding box.

4) Improved Recall: Confluence achieves better recall than Greedy NMS because bounding box removal is based on the coherence of bounding box borders with each other, rather than the extent of their overlap. The benefits of this approach is most evident when objects are occluding each other, for example, when a person is standing in front of another person, as shown in Figures 1, 7 and 12. Although the smaller bounding box is a subset of the larger box, its borders are not sufficiently coherent to be removed by Confluence. However, they share a large overlap, which means the lower confidence box is removed. This suggests that Confluence is more robust to high occlusion, and explains why its recall is higher than

Greedy NMS on all tested object detectors.

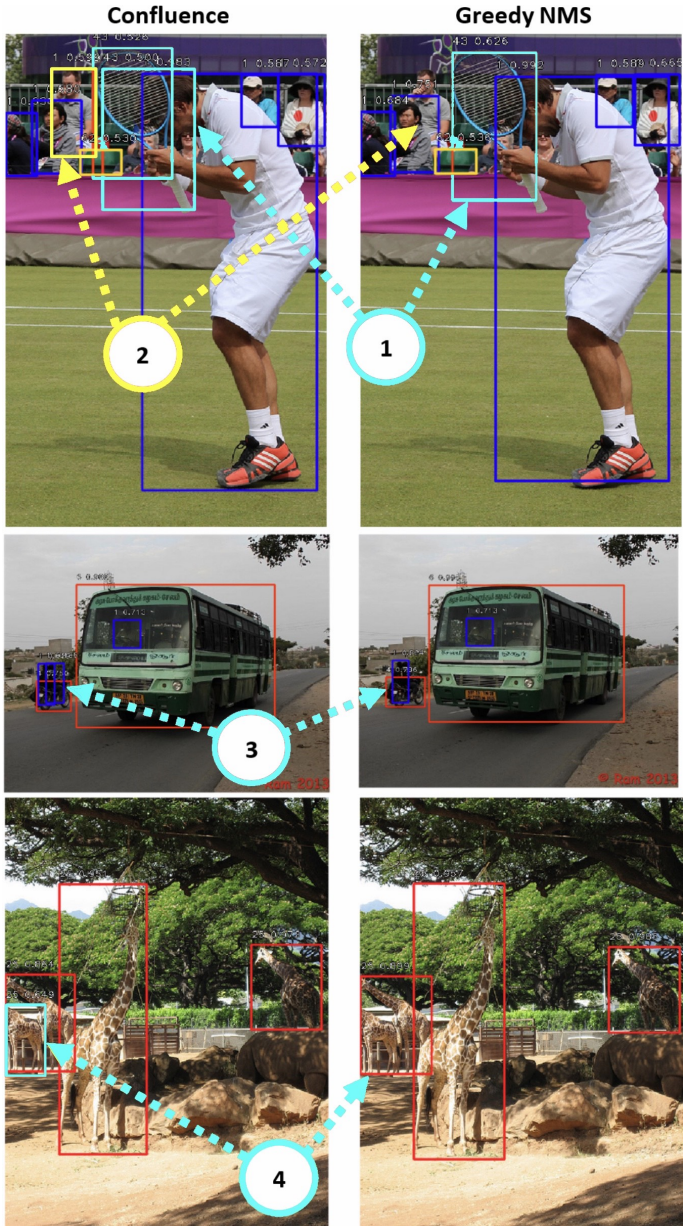

Fig. 12. Left: Confluence. Right: Greedy NMS. Notice that the giraffe numbered 4 is a subset of the bounding box allocated to the giraffe in the background. It is removed by NMS, yet retained by Confluence, accounting for its higher recall. This also occurs in crowds, as shown by 2, and where objects are occluding each other, as shown by 3.

One shortcoming of the use of Confluence to remove false positives is its tendency to retain false positives which are not confluent with other boxes in a given cluster. This can be seen in Figure 12. The tennis racket is surrounded by two bounding boxes rather than one (see annotation 1), as the two boxes are not confluent. In situations like this, the object detector is not confident, which causes it to return spurious bounding boxes around an object. In such cases, NMS has higher precision, as it harshly removes any highly overlapping bounding boxes.

B. Future Work

Our experimental results strongly indicate that Confluence is a superior alternative to Greedy NMS and its variants in both multi-class and single-class object detection. However the extent of its improvement over Greedy NMS may have been obscured by the way that the mAP calculation processes bounding boxes. Bounding boxes are ranked by confidence, rather than high IoU ratio [43], [44], [9]. Consequently, if a sub-optimal box has a higher confidence score than a superior lower confidence box, the sub-optimal bounding box will be chosen by the mAP calculator, and the superior box chosen by Confluence will be ignored. This degrades the overall mAP score. Thus future work could encompass the development of a confluence score to be used in ranking bounding boxes in mAP calculations. This would overcome the disadvantage faced by Confluence arising from confidence ranking when evaluated using the COCO and Pascal benchmarks. This would enable more effective evaluation of the extent of improvements in both bounding retention and removal.

Furthermore, the hyper-parameter used by Confluence could be learned to optimize performance in any object detector. Another interesting area of research could be investigation of the applicability of the Manhattan Distance principles used by Confluence as a possible non-IoU alternative to Jaccard Index based regression modules, and mAP calculators.

VII. Conclusion

We have proposed a superior non-IoU alternative to Greedy NMS which does not rely on IoU or the maxima confidence score in bounding box retention and suppression. Results collected on RetinaNet, YOLOv3 and Mask-RCNN evaluated using both the MS-COCO dataset/mAP and PASCAL VOC dataset/mAP suggest it consistently outperforms Greedy NMS, Furthermore, sensitivity analysis of variations in thresholds indicates that it is more robust to occlusion. Finally, it can be seamlessly integrated within currently used object detectors without modifications or training, making it a viable alternative to Greedy NMS in object detection tasks.

Acknowledgment

This research was conducted with the support of an Australian Research (RTP) scholarship.

References

[1] A. Walha, A. Wali, and A. M. Alimi, "Moving object detection system in aerial video surveillance," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, A. Kasinski, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham: Springer International Publishing, 2013, pp. 310–320.

[2] A. Teichman and S. Thrun, "Practical object recognition in autonomous driving and beyond," in *Advanced Robotics and its Social Impacts*, Oct 2011, pp. 35–38.

[3] G. Falzon, C. Lawson, K.-W. Cheung, K. Vernes, G. Ballard, P. Fleming, A. Glen, H. Milne, A. Mather-Zardain, and P. Meek, "Classifyme: a field-scouting software for the identification of wildlife in camera trap images," 2019. [Online]. Available: https://doi.org/10.1101/646737

[4] K. Risha and A. C. Kumar, "Novel method of detecting moving object in video," *Procedia Technology*, vol. 24, pp. 1055 – 1060, 2016, international Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017316303267

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. [Online]. Available: http://arxiv.org/abs/1409.0575

[6] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 06, pp. 1137–1149, jun 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 03 2017.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[10] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6469–6477.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[13] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 379–387. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157139

[14] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms — improving object detection with one line of code," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 5562–5570.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[16] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key.net: Keypoint detection by handcrafted and learned CNN filters," *CoRR*, vol. abs/1904.00889, 2019. [Online]. Available: http://arxiv.org/abs/1904.00889

[17] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. C-20, no. 5, pp. 562–569, May 1971.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec 2001, pp. I–I.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1.

[20] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham: Springer International Publishing, 2015, pp. 290–306.

[21] D. Wang, C. Li, S. Wen, S. Nepal, and Y. Xiang, "Daedalus: Breaking non-maximum suppression in object detection via adversarial examples," *CoRR*, vol. abs/1902.02067, 2019. [Online]. Available: http://arxiv.org/abs/1902.02067

[22] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 168–181.

[23] D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell, "Spatial semantic regularisation for large scale object detection," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2003–2011.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman, "Using the forest to see the trees: Exploiting context for visual object detection and localization," *Commun. ACM*, vol. 53, no. 3, pp. 107–114, Mar. 2010. [Online]. Available: http://doi.acm.org/10.1145/1666420.1666446

[25] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 1–12, Oct 2011. [Online]. Available: https://doi.org/10.1007/s11263-011-0439-x

[26] M. Rodriguez, I. Laptev, J. Sivic, and J. Audibert, "Density-aware person detection and tracking in crowds," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2423–2430.

[27] W. Ouyang and X. Wang, "Single-pedestrian detection aided by multi-pedestrian detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3198–3205.

[28] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 58–69, Oct 2014. [Online]. Available: https://doi.org/10.1007/s11263-013-0664-6

[29] P. Kontschieder, S. Rota Bulò, M. Donoser, M. Pelillo, and H. Bischof, "Evolutionary hough games for coherent object detection," *Comput. Vis. Image Underst.*, vol. 116, no. 11, pp. 1149–1158, Nov. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2012.08.003

[30] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using hough transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1773–1784, Sep. 2012.

[31] P. Wohlhart, M. Donoser, P. M. Roth, and H. Bischof, "Detecting partially occluded objects with an implicit shape model random field," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 302–315.

[32] Li Wan, D. Eigen, and R. Fergus, "End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 851–859.

[33] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2325–2333.

[34] D. Lee, G. Cha, M.-H. Yang, and S. Oh, "Individualness and determinantal point processes for pedestrian detection," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 330–346.

[35] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," *CoRR*, vol. abs/1607.03476, 2016. [Online]. Available: http://arxiv.org/abs/1607.03476

[36] Y. He, X. Zhang, M. Savvides, and K. Kitani, "Softer-nms: Rethinking bounding box regression for accurate object detection," *CoRR*, vol. abs/1809.08545, 2018. [Online]. Available: http://arxiv.org/abs/1809.08545

[37] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007. [Online]. Available: https://science.sciencemag.org/content/315/5814/972

[38] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2014.

[39] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 702–709.

[40] S. Craw, *Manhattan Distance*. Boston, MA: Springer US, 2017, pp. 790–791. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_511

[41] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 80–91. [Online]. Available: http://doi.acm.org/10.1145/279943.279960

[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," vol. 1, 02 2001, pp. I–511.

[43] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.

[45] ——, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates,

Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

Andrew Shepley is a PhD candidate at the University of New England (UNE), Australia. He received a BSc. Hons at the University of New South Wales (UNSW), a Grad. Degree Information Science at UNSW, and a Grad Dip Education at UNE. He currently holds the position of Software Engineer/Computer Vision Expert at UNE. He has over 10 years experience as an educator and executive with the NSW Department of Education. His research interests include development of algorithms to improve the accuracy and efficiency of deep learning systems and artificial intelligence.

Gregory Falzon is a Senior Lecturer in Computational Science at the University of New England in Armidale, Australia. He holds a PhD in biomedical imaging, and a Grad Dip Education and BSC (Hons I) from the University of New England. Postdoctoral experience spans diverse areas of statistical engineering applied to the agricultural, biomedical and ecological fields. He is the Intelligent Systems Theme Leader within the University of New England Smart Farm and is a Principal Investigator on several externally funded research projects exploring edge computing and sensor networks. Current research interests include data science, IoT devices, machine vision, robotics and their applications, particularly those relevant to agriculture.

Paul Kwan is a Professor of Computer Science at The University of New England in Armidale, Australia. He has a PhD in Advanced Engineering Systems, specialized in Intelligent Interaction Technologies, from The University of Tsukuba in Japan, a BSc and an MSc degree, both in Computer Science, from Cornell University and University of Arizona in the United States. His research fields include Artificial Intelligence, Computer Vision, Computational Modelling, Image Processing and Machine Learning, and their applications to digital agriculture, human and animal biometrics, computer simulation and analysis of animal diseases and invasive pests spread. He has been a Member of Australian Computer Society since 2006, a Senior Member of Association of Computing Machinery since 2008, and a Senior Member of IEEE since 2010.