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

Andrew J. Shepley, Greg Falzon, Member, IEEE, Paul Kwan, Senior Member, IEEE, and Ljiljana Brankovic

Abstract—Confluence is a novel non-Intersection over Union (IoU) alternative to Non-Maxima Suppression (NMS) in bounding box post-processing in object detection. It overcomes the inherent limitations of IoU-based NMS variants to provide a more stable, consistent predictor of bounding box clustering by using a normalized Manhattan Distance inspired proximity metric to represent bounding box clustering. Unlike Greedy and Soft NMS, it does not rely solely on classification confidence scores to select optimal bounding boxes, instead selecting the box which is closest to every other box within a given cluster and removing highly confluent neighboring boxes. Confluence is experimentally validated on the MS COCO and CrowdHuman benchmarks, improving Average Precision by 0.2–2.7% and 1–3.8% respectively and Average Recall by 1.3–9.3 and 2.4–7.3% when compared against Greedy and Soft-NMS variants. Quantitative results are supported by extensive qualitative analysis and threshold sensitivity analysis experiments support the conclusion that Confluence is more robust than NMS variants. Confluence represents a paradigm shift in bounding box processing, with potential to replace IoU in bounding box regression processes.

Index Terms—Computer vision, edge and feature detection, feature representation, image processing and computer vision, machine learning, confluence, non-maxima suppression, object detection, deep learning.

I INTRODUCTION

OBJECT detection algorithms allow objects of interest to be localized in images. One-to-many detectors return several candidate bounding boxes per object, each representing a slight variation in size and location. To obtain the final detection output, only one bounding box per object must be retained, and all other candidate bounding boxes must be removed.

Deep Convolutional Neural Networks (DCNNs) learn data-derived features [1], as illustrated by Fig. 2. They propose large clusters of bounding boxes, which congregate in regions likely to contain an object [2], [3], [4], [5], exemplified by the topmost image in Fig. 1. During the post-processing stage, most of these boxes are removed, leaving only one bounding box per object.

This task is usually performed by variants of the Non-Maxima Suppression (NMS) algorithm [6], [7]. NMS sorts the bounding boxes in descending order, using the classification confidence score attributed to each box by the DCNN. The box with the highest scoring box is considered the ‘optimal’ box for the first object and is retained. NMS then suppresses (sets to 0) or decays the confidence scores of all bounding boxes whose overlap with the ‘optimal’ box exceeds a user-defined Intersection over Union (IoU) threshold. This results in the removal of boxes that overlap heavily with the ‘optimal’ box. If there are any remaining bounding boxes, NMS selects the next highest scoring bounding box to represent the next object and repeats the suppression procedure. This process is repeated until the final detection set is attained – ideally, one bounding box per object.

NMS usually achieves acceptable performance in images where objects do not overlap each other much [8], [9]. However, when objects are occluded or overlap heavily such as in high density or crowded settings, reliance on IoU forces a trade-off...
between recall and precision [8], [9], [10], [11], [12]. This is because NMS is a heuristic algorithm that simply assumes that a high overlap between bounding boxes results in a high probability of one of the boxes being a duplicate [13]. Thus, to retain highly overlapping true positives, a higher IoU threshold must be used, leading to greater retention of false positives [14], [15], [16], [17], [18].

Another widely noted shortcoming of NMS is its poor localization accuracy occasioned by its sole reliance on the classification confidence score to select optimal bounding boxes [19], [20], [21], [22], [23], [24]. This score is a class label probability and does not effectively represent localization accuracy [24]. In other words, there is low correlation between localization accuracy and classification score [22]. Its usage by NMS means that NMS is susceptible to returning suboptimal bounding boxes, whilst suppressing better candidate boxes [21] as illustrated by Fig. 1. It should therefore not be used as the primary metric by which the optimal boxes are selected.

Although alternatives to and variants of NMS have been proposed [7], [25], [26], [27], [21], [28], [29], [30], [17], [31], [12], [18], [15], [32], [33], the majority share the inner mechanisms of NMS [34], namely reliance on IoU and confidence scores. The most commonly used variants of NMS are Greedy NMS (G-NMS) [6], which is widely regarded as the de-facto standard solution [35], [36], [10], [37], [9], [14], [38], [39], [40], [22], [35], [41], [42], [43], [44], [30] and the recently proposed Soft NMS (S-NMS) [7]. S-NMS aims to reduce the greedy suppression of true positives by G-NMS by decaying rather than eliminating the confidence scores of highly overlapping boxes as a continuous function of their overlap with the optimal box. S-NMS is being increasingly adopted to replace G-NMS in object detection pipelines [45], [46], [47], [48], [49], [50], [51], [52], [53], [54] however, it is limited by its ongoing reliance on IoU and the classification confidence score.

In this paper, we propose Confluence, a novel non-IoU alternative to NMS algorithms. The key contributions are threefold. First, we propose a proximity metric as an alternative to IoU in the suppression of false positives. Rather than using the IoU between bounding boxes to determine whether they represent the same object, we propose the confidence-weighted normalized pairwise Manhattan Distance [55] between corresponding bounding box coordinates to measure their coherence, and hence more accurately determine whether they point to the same object. Secondly, we propose Confluence NMS (NMS-C), a non-IoU variant of NMS, which retains bounding boxes using the classification confidence score, but suppresses false positives using the Confluence metric. Thirdly, we present Confluence, an algorithm which uses the proximity metric in both the selection of the optimal boxes, and the suppression of false positives. The effectiveness of Confluence and NMS-C is empirically validated against both G-NMS and S-NMS on the MS COCO [56] and CrowdHuman [57] datasets achieving gains of 2.3–3.8% in Average Precision and gains of 5.3–7.2% in Average Recall.

II RELATED WORKS

Although NMS has been an algorithm of significant importance in computer vision for over 50 years, its shortcomings are as widely recognized as its essential role in the object detection pipeline [10], [58], [39], [40], [35], [31]. This has given rise to many adaptations of NMS, and various alternatives.
Many alternatives and adaptations aim to limit or eliminate the bottleneck caused by NMS in the object detection pipeline [59] by reducing computational expense and improving efficiency [60], [28], [61], [62], [63]. However, improvements in efficiency are achieved at the expense of accuracy, as these methods do not address localisation accuracy or recall [64].

Alternatively, some methods aim to improve proposal refinement during training to provide NMS with better input, thus improving performance [11], [24]. Other methods aim to circumvent NMS or eliminate the need for it altogether [65], [66], [67], [68], [69], [70], [9] however, they involve significant changes to neural network architecture, and achieve inferior or competitive performance [66]. This means they cannot be adopted in state-of-the-art object detectors [3], [5] that rely on NMS.

Some methods are specific to video [31], [71], body-part [72] or 3D [25], [73], [74] applications and are not applicable to or evaluated on standard object detection tasks. Other methods achieve gains in performance when compared to G-NMS and S-NMS but rely on additional data such as object depth information [17], or a combination of pixel-based and amodal bounding boxes [12], and corresponding network architecture changes, which are often not available or able to be implemented in standard object detection pipelines.

Most variants of NMS involve minor conceptual differences, resulting in similar performance. Matrix NMS [64], implements S-NMS in parallel for instance segmentation. Significant gains in speed were achieved, and it did outperform G-NMS but it did not outperform S-NMS and still relies on IoU and classification confidence scores. Similarly, a neural network, GNet [10], uses message passing between neighboring 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 architectural changes.

IoU-NMS [22] aims to improve bounding box localization by embracing IoU as the “natural criterion for localization accuracy”. Initially, IoU-Net learns the IoU relationship between ground truth box and the candidate boxes, returning a localisation confidence score per box. IoU-NMS then uses the localization confidence instead of the classification confidence score in the NMS procedure [22]. Although IoU-NMS achieves higher Average Precision (AP) at higher IoU thresholds on several networks, it was not evaluated on standard benchmarks. Further, combining S-NMS with box-voting [75] returns similar results to IoU-NMS with reduced computational expense [21]. Notably, integration of IoU-Net in standard object detection pipelines is not easily achieved and requires retraining of object detectors, limiting its adoption.

Alternatively, a bounding box regression loss and a modified S-NMS dubbed SoftNMS is proposed, to improve bounding box localization [21]. It avoids reliance on the classification confidence score by assigning a localisation score to each box, which like IoU-NMS, is used to select the best box. The localisation score is however not sufficient to select the best box, with SoftNMS relying on application of G-NMS or S-NMS followed by mean averaging of localization confidence scores of clustered boxes, and calculation of standard deviation to measure the uncertainty of the estimated bounding box location. Although SoftNMS improved AP on MS COCO, its usage requires retraining of the object detector integrating the new loss function. It is also significantly more computationally expensive, and it only addresses the issue of poor localisation, and does not improve recall. Conversely, Confluence resolves both issues, by treating the strong coherence of bounding boxes as a measure of localization confidence without complex, computationally expensive calculations, retraining of networks or need of a localization loss.

Another limitation of NMS is its high sensitivity to the IoU threshold [37], [36], [35], which makes it difficult to achieve an optimal balance between retention of true positives and removal of false positives. This is particularly true for highly occluded settings such as crowds and has resulted in the development of crowd setting specific modifications of NMS. Rather than avoiding the use of IoU, these modifications are premised on trying to minimise the limitations of IoU in adequately representing true and false positives.

Adaptive NMS [36] proposes a dynamic per instance IoU threshold to achieve a better balance between retention of true positives and elimination of false positives, tailored to pedestrian detection. A learnable sub-network determines the IoU threshold, which changes depending on the density of the image. It achieved competitive results on the CityPersons and CrowdHuman benchmarks. Alternatively, handcrafted image descriptors of person silhouettes are exploited [35] to rescore candidate detections to improve performance of NMS on the PETS [76], COCO Person [56] and Okutama-Action [77] datasets. However, Adaptive NMS and handcrafted image descriptors [35] are significantly more computationally expensive than traditional NMS and not evaluated in general object detection tasks.

Similarly attribute-aware NMS [78] leverages semantic attributes such as density and diversity gained from a pedestrian-oriented attribute map to minimise false positives occasioned from using a high IoU threshold during NMS. Although it outperformed G-NMS, its reliance on a custom network designed specifically for crowd detection means it cannot be easily adopted in standard state-of-the-art object detection pipelines.

An adaptive threshold for NMS in video sequences [71] uses high confidence bounding boxes in key frames to improve detection in lower scoring frames. Although gains in AP were achieved, it is not useful for standard image-based object detection due to its reliance on multiple consecutive frames for threshold adaptation. In contrast, Confluence can be used in both image object detection studies and video-based object detection without retraining or additional image information, achieving significant gains in performance.
### III Methodology

Confluence derives its name from the highly confluent, aggregated clusters of bounding boxes returned by a neural network when an object is detected. Rather than treating the excessive proposals as problematic, Confluence embraces them as a way of identifying and retaining the bounding box which best represents the object location. The clustering can be interpreted as a collective vote on object location, where the box that best represents every other box is the optimal one. Bounding box confluence is also an effective way of removing those false positives that are confluent with the retained box.

Confluence is a recursive, two-staged algorithm which first retains an optimal bounding box, and then removes false positives that are confluent with it. Retention is achieved using a confidence weighted Manhattan Distance inspired proximity measure to evaluate bounding box coherence, enabling retention of the bounding box that best represents all boxes in a cluster. The second stage involves removal of all bounding boxes which are confluent with the retained bounding box. This process is repeated until all boxes have been processed.

#### A. Manhattan Distance

The Manhattan Distance (MD) or L1 norm, is the sum of the vertical and horizontal distances between two points \([55]\). The MD between the points \(u_i = (x_{u_i}, y_{u_i})\) and \(u_j = (x_{u_j}, y_{u_j})\) is shown by (1):

\[
MD(u_i, u_j) = |x_{u_j} - x_{u_i}| + |y_{u_j} - y_{u_i}|
\]  

(1)

Each bounding box \(b_i\) can be represented by two diagonally opposite corners. For example, \(b_i = (v_i, v_i)\) can be defined by the upper left corner \(v_i = (u_{x_i}, u_{y_i})\) and the lower right corner \(v_i = (x_{v_i}, y_{v_i})\).

We propose a proximity measure \(P(b_i, b_j)\) between any two bounding boxes \(b_i = (u_i, v_i)\) and \(b_j = (u_j, v_j)\), represented by the sum of the MD between the upper left corners \(u_i = (x_{u_i}, y_{u_i})\) and \(u_j = (x_{u_j}, y_{u_j})\), and the lower right corners \(v_i = (x_{v_i}, y_{v_i})\) and \(v_j = (x_{v_j}, y_{v_j})\) of the two boxes as given by (2):

\[
P(b_i, b_j) = MD(u_i, u_j) + MD(v_i, v_j)
\]

[55]

\[
P(b_i, b_j) = |x_{u_j} - x_{u_i}| + |y_{u_j} - y_{u_i}| + |x_{v_j} - x_{v_i}|
\]

+ 

\[
|y_{v_j} - y_{v_i}|
\]  

(2)

A small \(P(b_i, b_j)\) value denotes highly confluent boxes \(b_i\) and \(b_j\), whilst a high \(P(b_i, b_j)\) value indicates boxes that are not attributable to the same object - they may be somewhat overlapping, or completely disjoint. A diagrammatic representation of \(P(b_i, b_j)\) is given in Fig. 3.

Let \(O(b_i)\) be a set of all boxes bounding the same object as box \(b_i\), such that \(O(b_i)\) does not include \(b_i\) itself. We define proximity \(P(b_i)\) of a box \(b_i\) as the mean value of the proximities of the box \(b_i\) to all the boxes in \(O(b_i)\):

\[
P(b_i) = \frac{1}{|O(b_i)|} \sum_{b_j \in O(b_i)} P(b_i, b_j)
\]  

(3)

A bounding box \(b_i\) surrounded by a dense cluster of bounding boxes will be characterized by very low \(P(b_i)\) values, when compared against a bounding box which is positioned further away from other boxes in the cluster. The latter could be correctly categorized as an outlier, or as suboptimal. 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 \(b_i\) with the lowest \(P(b_i)\), value represents the most confident detection for a given object.

Notably, this concept overcomes an issue faced by all variants that rely on the classification confidence score. 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 Fig. 1. In contrast, the \(P(b_i)\) measure allows for the bounding box \(b_i\) that is most confluent with all other bounding boxes assigned to a given object to be favored.

#### B. Normalization

The concept 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 vary significantly in size. Thus, when regulating bounding box retention or removal using a hyper-parameter based on \(P(b_i)\), a trade-off between removing large false positives and retaining small true positives needs to be reached. This is because small, disjoint boxes will often have similar values to that of large, highly overlapping boxes.

To overcome this issue, a pairwise normalization algorithm was used to scale the bounding box coordinates between 0 and 1. This preserves their relationship with each other, whilst allowing the same threshold to be used for small and large boxes, without any trade-off. The normalization algorithm transforms each coordinate of the boxes \(b_i\), and \(b_j\) as shown below.
Given boxes $b_i$, and $b_j$:

$$b_i = (u_i, v_i) = ((x_{u_i}, y_{u_i}), (x_{v_i}, y_{v_i}))$$

$$b_j = (u_j, v_j) = ((x_{u_j}, y_{u_j}), (x_{v_j}, y_{v_j}))$$

We define a set $X$ to comprise $x$-coordinates of the upper left and lower right corners of both boxes:

$$X = \{x_{u_i}, x_{u_j}, x_{v_i}, x_{v_j}\}$$

Similarly, we define set $Y$ to comprise coordinates of the upper left and lower right corners of both boxes:

$$Y = \{y_{u_i}, y_{u_j}, y_{v_i}, y_{v_j}\}$$

Then, we normalise the upper left corner of the box $b_i$ as:

$$(x_{u_i}, y_{u_i})_{\text{norm}} = \left(\frac{x_{u_i} - \min(X)}{\max(X) - \min(X)}, \frac{y_{u_i} - \min(Y)}{\max(Y) - \min(Y)}\right)$$

(4)

The other relevant corners of the boxes $b_i$, and $b_j$ are normalised in the same way. Then we have:

$$(b_i, b_j)_{\text{norm}} = ((x_{u_i}, y_{u_i})_{\text{norm}}, (x_{v_i}, y_{v_i})_{\text{norm}}), ((x_{u_j}, y_{u_j})_{\text{norm}}, (x_{v_j}, y_{v_j})_{\text{norm}})$$

(5)

The normalized proximity of the box $b_i$ is given by:

$$\frac{1}{|O(b_i)|} \sum_{b_j \in O(b_i)} P((b_i, b_j)_{norm})$$

For the sake of simplicity, we use $P(b_i)$ to denote both the proximity and the normalized proximity of the box $b_i$ and it will be clear from the context which on is the case.

In circumstances where the two corresponding pairs of horizontal (or vertical) borders are very closely aligned, $(1 - P(b_i))$ will have a very similar value to IoU. However, due to the independent normalisation of the vertical and horizontal borders, this similarity decreases significantly as the vertical and horizontal borders diverge. This also broadens the threshold range to 0–2, which may facilitate identification of false positives and allow for a larger range of acceptable threshold values.

C. Intra-Cluster Retention and Removal

As all bounding box pairs are normalized between 0 and 1, any pair of intersecting bounding boxes $(b_i, b_j)$ will have a $P(b_i, b_j)$ value below 2. However, empirical observation suggests that most clusters will be characterized by proximity values between 0 and 1.

In Figs. 4 and 5, each point on the horizontal axis represents a bounding box $b_i$, $1 \leq i \leq n$, while the vertical axis represents the proximity $P(b_0, b_i)$ between a randomly selected box $b_0$, and box $b_i$. To visualize the relationship between boxes, they were ordered so that $P(b_0, b_i) \leq P(b_0, b_2) \leq \cdots \leq P(b_0, b_n)$. This reveals the blob-like nature of $P(b_i, b_j)$ clustering, where each horizontal blob represents an object. $P(b_i, b_j)$ values generally lie between 0 and 1, with the optimal bounding box being represented within the flattest gradient of the blobs.

Thus, if the $P(b_i, b_j)$ value of any two bounding boxes is below the user defined Confluence threshold ($C_f$), 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. The optimal intra-cluster bounding box is found, by calculating the mean $P(b_i)$ value, using (3), of each box in the cluster. The bounding box with the lowest $P(b_i)$ is the most confluent and is retained.

All bounding boxes that are confluent with this chosen box are likely to be false positives. Thus, their classification confidence scores are either removed or decayed as a function of their confluence with the chosen box.

D. Confidence Score Weighting

The majority of NMS variants, including G-NMS and S-NMS rely on raw or decayed classification confidence scores returned by the object detector to select ‘optimal’ bounding boxes. In contrast, Confluence assesses the optimality of a given bounding box $b_i$ by comparing its $P(b_i)$ values, moderated by its confidence score, with competing bounding boxes. To achieve this, the $P(b_i)$ value is weighted by the classification confidence score $s_i$, as follows:

$$P_w(b_i) = P(b_i)(1 - s_i)$$

(6)

As $s_i$ is a value which lies between 0.01 and 1 (all classification confidence scores lie between 1-100%), this in effect provides a bias in favour of high confidence boxes by artificially reducing the value of $P(b_i)$ (Note that all bounding boxes with confidence scores below 0.01 are not considered). Conversely, the $P_w(b_i)$ 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 $P_w(b_i)$ values.
TABLE I

| Method     | Average Precision (area = all) (maxDets=100) | Average Precision (IoU: 0.5:0.95) (maxDets=100) | Average Recall (IoU: 0.5:0.95) (area=all) | Average Recall (IoU: 0.5:0.95) (maxDets=100) area = variable |
|------------|---------------------------------------------|-----------------------------------------------|---------------------------------------|-------------------------------------------------------------|
|            | Small | Medium | Large | Small | Medium | Large | 1 | 10 | 100 | 100 |
| RetinaNet  | 0.414 | 0.573  | 0.446 | 0.225 | 0.355  | 0.643 | 0.361 | 0.459 | 0.459 |
| S-NMS-L    | 0.423 | 0.584  | 0.464 | 0.230 | 0.385  | 0.644 | 0.361 | 0.496 | 0.497 |
| S-NMS-G    | 0.435 | 0.598  | 0.477 | 0.235 | 0.391  | 0.663 | 0.361 | 0.504 | 0.507 |
| C-NMS      | 0.437 | 0.600  | 0.481 | 0.235 | 0.390  | 0.662 | 0.361 | 0.507 | 0.512 |
| Confluence | 0.439 | 0.600  | 0.482 | 0.235 | 0.390  | 0.665 | 0.361 | 0.515 | 0.520 |
| G-NMS      | 0.493 | 0.678  | 0.541 | 0.306 | 0.531  | 0.655 | 0.621 | 0.621 | 0.621 |
| C-NMS-G    | 0.510 | 0.694  | 0.564 | 0.318 | 0.551  | 0.673 | 0.690 | 0.692 | 0.692 |
| C-NMS-G    | 0.506 | 0.686  | 0.560 | 0.315 | 0.546  | 0.667 | 0.691 | 0.693 | 0.694 |
| Confluence | 0.512 | 0.695  | 0.566 | 0.320 | 0.583  | 0.674 | 0.706 | 0.709 | 0.709 |
|            | 0.508 | 0.688  | 0.562 | 0.317 | 0.550  | 0.669 | 0.708 | 0.712 | 0.712 |
|            | 0.510 | 0.692  | 0.564 | 0.318 | 0.551  | 0.672 | 0.704 | 0.708 | 0.708 |

Performance of RetinaNet and detectoRS on the MS-COCO mini-val dataset with Greedy NMS (G-NMS), Soft-NMS linear (S-NMS-L), Soft-NMS Gaussian (S-NMS-G), Confluence-NMS (C-NMS) and Confluence. The Greedy and Soft NMS variants were tested with an IoU threshold of 0.3, while an MD threshold of 0.7 was used for Confluence and Confluence NMS. The Confluence algorithms significantly outperform the IoU-based G-NMS and S-NMS in both Average Precision (AP) and Average Recall (AR) metrics.

This algorithm is based on the principle that a powerful classifier can be constructed by using the sum of weaker individual classifiers [79, 80]. Each individual \( P_w(b_i) \) 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 Tables I, II, and III suggest that this is a reliable means to accurately identify true positives, whilst effectively minimizing false positives. This allows achievement of optimal precision and recall values.

E. Implementation

The pseudocode outlining Confluence is provided by Algorithm 1. Both Confluence and C-NMS were implemented in Python and are freely available on GitHub.¹

The main steps are as follows (bounding boxes are in accordance with the raw output of the object detector):

1. We compare each bounding box \( b_i \) in the input set \( I \) to all the other boxes (line 4):
   a) For each box \( b_j \) we compute the normalized coordinates of the corners of \( b_i \) and \( b_j \) by computing \((b_i, b_j)_{norm}\) using (5) (line 7).
   b) We then use (2) to calculate the normalized proximity \( P((b_i, b_j)_{norm}) \) between the two boxes (line 8).
   c) If \( P((b_i, b_j)_{norm}) \) is below the user-defined threshold \( C_t \), the normalized proximity \( P \) of the box \( b_i \) is incremented by \( P((b_i, b_j)_{norm}) \) (line 10) and \( b_j \) is added to the set of neighbors of \( b_i \), i.e., \( N[b_i] \) (line 11).
   d) Once \( b_i \) has been compared against every other bounding box, (6) is applied to calculate its weighted proximity value \( P_w \), which is then added to the dictionary \( C \) (lines 15–16).
2. Once all boxes in \( I \) have been processed, the optimal bounding box \( b_m \) with the lowest \( P_w \) is identified (line 19).
3. \( b_m \) is added to the final detections set \( D \) and removed from the dictionary \( I \); all bounding boxes that have been identified as neighboring \( b_m \) are suppressed, as they are deemed to locate the same object (line 20).

Fig. 5. Bottom left: Raw RetinaNet output, comprised of approx. 130 boxes. Bottom right: Confluence output. Even when objects are close together or overlapping, Confluence is capable of clearly distinguishing between intra-object and inter-object bounding boxes. For detailed information on how the graph was generated, see Fig. 4 caption.

¹[Online]. Available: github.com/ashep29/confluence
4. The while loop at line 18 repeats this process until \( I \) is empty, and all bounding boxes locating objects are returned in \( D \).

The computational complexity of each go through of the outer for loop in the Confluence algorithm is \( O(n) \), where \( n \) is the size of the input set of bounding boxes. This is due to the calculation of the normalized proximity score of the given box. As this measure is computed for each bounding box, we go through the outer for loop \( n \) times and thus Confluence has an overall computational complexity of \( O(n^2) \).

Although the computational expense of Confluence is not significant due to the recursive reduction in the size of the set of bounding boxes, it may still be suboptimal for applications that prioritize speed. Thus, we also propose the non-IoU NMS algorithm Confluence-NMS (C-NMS), with computational complexity of \( O(mn) \), where \( m \) is the number of boxes and \( n \) is the number of objects to be identified. Although asymptotically C-NMS may still require \( O(n^2) \) steps, in practice, it is likely to perform better than the Confluence algorithm.

Like other NMS variants, C-NMS retains the bounding box with the maxima score, however its performance in minimizing false positives whilst maximizing true positives is significantly improved. This is because it relies on Confluence to suppress false positives and retain true positives, improving both recall and precision. Algorithm 2 provides pseudo-code illustrating C-NMS.

C-NMS operates as follows:

1. The bounding box \((b_m)\) in \( I \), with the highest confidence score is retained as the best box to represent the first object in the image (line 3).
2. \( b_m \) is reserved in the final detections set \( D \) and removed from \( I \) (line 4).
3. \( (b_m, b_i)_{\text{norm}} \) is computed for each remaining box \((b_i)\) (line 6).
4. Next, the normalized proximity \( P((b_m, b_i)_{\text{norm}}) \) is calculated (line 7).
5. If \( P((b_m, b_i)_{\text{norm}}) \) is less than or equal to the user defined Confluence threshold \( C_t \) (line 8), \( b_i \) is removed from \( I \) (line 9).
6. This process is repeated until all boxes have been retained in \( D \) or suppressed.

Both Confluence and C-NMS have been implemented such that bounding box suppression can be achieved linearly or using Gaussian weighting. This is inspired by the use of Gaussian weighting by [7] to improve recall by down-sampling rather than suppressing high-scoring bounding boxes that overlap with optimal boxes.

### IV DATASETS, ALGORITHMS AND EVALUATION

Experimental results presented in this paper were collected on the publicly available 2017 MS-COCO mini validation (mini-val) dataset [56] and 2018 CrowdHuman [57] validation datasets. These datasets were chosen to demonstrate Confluence on widely recognized standard and high-density, high-occlusion benchmarks.

The COCO mini-val set contains 5000 images of 80 classes. The CrowdHuman validation set contains 4370 images of people in highly crowded settings, with an average of 23 humans per image [57]. The AP calculations were obtained using the COCO-style evaluation metrics via the standard COCO API, using default settings including 100 maximum detections per image.

The G-NMS and S-NMS algorithms used were implemented by [21] and are publicly available on their GitHub repository. These algorithms were evaluated using three widely used NMS-reliant object detectors; RetinaNet-ResNet50 [5], detectoRS (HTC-R101) [82] and Mask-RCNN (HRNet) [3]. We used a Mask R-CNN implementation (and associated tools) to evaluate the performance of our algorithms.
Algorithm 2. Confluence NMS.

Input: \( I(b, s) = \{b_0 : s_0, \ldots, b_n : s_n\} \), \( C_I \) is a dictionary mapping a box \( b_i \) to its corresponding weighted confidence score \( s_i \). \( C_t \) is the Confluence threshold (optimal range: 0.6–0.9)

Output: The final detection set \( D \)

begin:
1: \( D \leftarrow \{\} \)
2: while \( I \neq \text{empty} \) do
3: \( b_m \leftarrow \arg\max(I) \)
4: \( D \leftarrow D \cup b_m, I \leftarrow I - b_m \)
5: for \( b_i \in I \) do
6: Compute \( (b_m, b_i)_\text{norm} \)
7: Compute the normalised proximity \( P((b_m, b_i)_\text{norm}) \)
8: if \( P((b_m, b_i)_\text{norm}) < C_t \) then
9: \( P \leftarrow P + P((b_m, b_i)_\text{norm}) \)
10: \( N[b_i] \leftarrow N[b_i] \cup b_j \)
11: end if
12: end for
13: if \( P((b_m, b_i)_\text{norm}) < C_t \) then
14: \( I \leftarrow I - b_i \)
15: end if
16: end while
17: end for
18: end while
19: \( D \leftarrow D \cup b_m, I \leftarrow I - b_m \)
20: end for
21: \( I \leftarrow I - N[b_m] \)
22: end while
23: return \( D \)

achieved by Confluence across variations in the \( C_t \) threshold. Sensitivity analysis was achieved using RetinaNet, applied to the MS-COCO mini-val dataset.

V RESULTS

In this section, we provide performance results on both MS-COCO mini-val and CrowdHuman datasets. In Table I we compare the performance of Confluence and C-NMS against G-NMS and S-NMS on the MS-COCO mini-val dataset using RetinaNet and detectorRS. The Confluence \( C_t \) threshold was set to 0.7.

Both Confluence and C-NMS improve object detector performance on the AP@0.5:0.95 calculation. When compared against G-NMS and S-NMS linear (S-NMS-L), Confluence achieves gains in AP of up to 1.9% while C-NMS outperformed G-NMS and S-NMS-L by up to 2.3%. Similarly, at the PASCAL VOC AP metric of IoU@0.5, Confluence outperforms G-NMS and S-NMS-L by up to 1.7% while C-NMS outperforms G-NMS and S-NMS-L by up to 2.7%. Notable improvements in AR are also achieved by Confluence and C-NMS, with gains of 1.4–5.3% at variable max detections. Confluence and C-NMS also outperform G-NMS and S-NMS-L across all object sizes, improving AR by 1.3–9.3% and AP by up to 2.2%.

Gaussian confidence score decaying is more computationally expensive, and only results in gains in AP and AR when a very low confidence threshold is used, reducing its practical applicability. Regardless, we provide results for C-NMS-G, which outperforms S-NMS-G by 0.2–0.4% on the AP@0.5:0.95 metric and 0.2% on the PASCAL VOC 0.5% AP metric. Improvements of up to 1.9% in AR on the max detections – variable metric, and up to 2.7% on objects of varying sizes were made. These improvements are significant for the MS-COCO dataset and evaluation metric.

pretrained model) published by [81], which is publicly available on GitHub.\(^3\)

The Mask-RCNN model used was trained by [81] on the CrowdHuman training set as described in [81]. The RetinaNet and detectorRS implementations and pretrained models used were obtained from GitHub.\(^4\) These models were trained on the MS COCO 2017 dataset as outlined in the respective repositories. We did not conduct further model training, instead selecting publicly available pre-trained models in all cases. We simply replaced the default G-NMS module with the S-NMS implementation provided by [21], and our implementations of Confluence and C-NMS, gathering all results using default settings.

Threshold sensitivity analysis was also conducted to demonstrate the robustness of the Confluence threshold \((C_t)\), by examining changes in AP attained by G-NMS and S-NMS over variations in IoU threshold in comparison to change in AP

\(^3\)[Online]. Available: github.com/hasanirtiza/Pedestron

\(^4\)[Online]. Available: github.com/tzzyt/keras-retinanet.github.com/openmllab/mmddetection
Both parameters were varied incrementally by 0.1. The optimal threshold for G-NMS and S-NMS on MS-COCO lies between 0.3-0.6. This is because a high IoU threshold results in only highly overlapping bounding boxes being removed, resulting in a high number of false positives but greater recall. In contrast, a lower IoU threshold removes more bounding boxes, reducing recall yet minimising false positives.

The optimal range for Confluence and C-NMS lies between 0.5 and 0.8. This is because a high $C_t$ value indicates low proximity of bounding box borders, while lower values indicate greater bounding box confluence. Thus, the higher the $C_t$ threshold, the more bounding boxes are removed. This is why the optimum threshold value for Confluence and C-NMS is higher than that used by the IoU dependent G-NMS and S-NMS.

Performance by both algorithms tends to decrease outside these ranges. Note that the variation in AP for Confluence and C-NMS within its optimal range is more stable than G-NMS and S-NMS, with variation in AP for $C_t$ being 0.3% while variation in AP for IoU is 0.6%. This means that the $C_t$ threshold is less sensitive to fluctuations in object density and occlusion, which makes it more robust. Furthermore, as shown by Fig. 6 performance of Confluence and C-NMS always remains approximately 1.5% better than G-NMS and S-NMS, even at the optimal IoU threshold.

VI DISCUSSION

This section will relate the quantitative results presented in Section V to a qualitative comparison of Confluence, C-NMS and the IoU-based G-NMS and S-NMS. It will explain how and why Confluence returns optimal bounding boxes, using qualitative data to demonstrate why coherence of bounding box borders is a more appropriate metric than IoU to use in bounding box selection and suppression in object detection. We will also discuss the possible reasons why C-NMS outperforms Confluence, and provide insight into possible future work to further improve the performance and applicability of the Confluence algorithm.

A. Qualitative Comparison of Confluence Against IoU-Based G-NMS and S-NMS

A fundamental problem with IoU based suppression of bounding boxes is the elimination of true positives in high density images. Once the highest 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 Figs. 1 and 7, high IoU will often result in the suppression of detections denoting true positives.

The raw, unfiltered object detector output is illustrated in Figs. 1 and 7 at a confidence threshold of 5%. It is evident by the thick confluence of proposals that all objects are detected and localized correctly. Thus, the aim of the post-processing stage is to maximize precision by selecting an optimal detection to represent each true positive, without lowering recall by suppressing true positives. IoU variants of NMS, such as G-NMS and S-NMS do not achieve this when applied to these images. Their
reliance on the maxima confidence score causes them to return suboptimal bounding boxes, while their IoU dependency causes them 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.

1) Improved Bounding Box Selection: Although most NMS variants interpret the bounding box with the highest confidence score within a cluster as the optimal bounding box, there are many instances where the highest scoring box is not the optimal bounding box. 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. A few examples of these situations are provided in Figs. 8 and 9.

Rather than simply selecting the highest scoring bounding box, Confluence selects the bounding box which is the most coherent with every other bounding box in the cluster. 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: The reliance of NMS on the classification confidence score as the sole means by which an optimal bounding box is selected reduces recall. For example, Fig. 10 illustrates how G-NMS selects the highest confidence box (87.7% confidence, shown in red) 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 G-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 both bounding boxes locating 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, G-NMS suppresses a bounding box which has a high IoU with a 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, Fig. 11 shows the output of G-NMS, and Confluence on the same image. G-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, the Confluence algorithm uses areas of confluence returned by the object detector to determine whether a second object is present,
Fig. 9. The optimal box does not always have the highest confidence score. Often, a bounding box which is too large is returned in instances where objects, such as people, are at close proximity.

Fig. 10. Reduction in Recall. Note the poor selection by G-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).

4) Improved Recall: Confluence achieves better recall than G-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 are most evident when objects are occluding each other. For example, when a person is standing in front of another person, as shown in Figs. 1, 7, and 10. 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 G-NMS on all tested object detectors.

Fig. 11. The optimal bounding box for the background giraffe on the far left (yellow) is suppressed due to IoU, which forces G-NMS to return a sub-optimal bounding box.

One shortcoming of the use of Confluence to remove false positives is its tendency to retain those false positives which are not confluent with other boxes. This can be seen in Fig. 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 these (uncommon) cases, NMS has higher precision, as it harshly removes any highly overlapping bounding boxes.

B Future Work

Our experimental results strongly indicate that Confluence and C-NMS are superior alternatives to IoU-based variants of NMS. However, in comparison to C-NMS, Confluence tends to consistently underperform. This is likely due to the confidence-score ranking of bounding boxes by the mAP calculator. [43], [44], [9]. mAP results are collected using parameters that favour true positive retention (at the expense of including more false positives) because suppression of true positives is penalized more harshly than failure to suppress false positives. Thus, parameters such as a higher NMS IoU threshold, lower Confluence threshold, and a very low confidence score (5%) are used to maximise the number of boxes (even those with low confidence scores), thus maximising retention of true positives.

Although this does result in the retention of many false positives, it does maximise the mAP score. Unfortunately for Confluence, this sometimes results in the mAP calculation misconstruing which bounding box was selected by Confluence as the ‘optimal’ bounding box. For example, if a sub-optimal box, that would in practice be excluded, 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 instead of the confidence score to rank bounding boxes in mAP calculations. This would overcome the disadvantage faced by Confluence, enabling more effective evaluation of the extent of improvements in both bounding box 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
batter AP and AR values achieved by Confluence algorithms over IoU algorithms, as well as lower threshold sensitivity could be the larger range of values for normalised proximity of boxes over Intersection over Union. For overlapping boxes, normalised proximity ranges from 0 to 2, while IoU ranges from 1 down to 0. Therefore, the normalised proximity values of a given set of boxes are more spread out that corresponding IoU values, which may facilitate identification of false positives and allow for a larger range of acceptable thresholds.

Finally, Confluence can be seamlessly integrated within currently used object detectors without modifications or training, making it an attractive alternative to NMS variants in object detection. Confluence represents a paradigm shift away from the heavily researched and used IoU towards a more robust boundary distance metric, which may be used to replace IoU in other applications, ranging from regression modules to localization losses and AP calculators.

**REFERENCES**

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.
[4] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6517–6525.
[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020.
[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2005, pp. 886–893.
[7] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS — improving object detection with one line of code,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 5562–5570.
[8] P. Jackson and B. Obara, “Avoiding over-detection: Towards combined object detection and counting,” in Artificial Intelligence and Soft Computing, Berlin, Germany: Springer, 2017, pp. 75–85.
[9] N. Kim, D. Lee, and S. Oh, “Learning instance-aware object detection using determinantal point processes,” Comput. Vis. Image Understanding, vol. 201, Jan. 01, 2020, Art. no. 103611.
[10] J. Hosang, R. Benenson, and B. Schiele, “Learning non-maximum suppression,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6469–6477.
[11] D. Rukhovitch, K. Sofiuk, D. Galeev, O. Barinova, and A. Konushin, “IterDet: Iterative scheme for object detection in crowded environments,” in Structural Syntactic Statistical Pattern Recognition: Joint IAPR International Workshops, Berlin, Germany: Springer, 2021, pp. 344–354.
[12] N. Gähler, N. Hanselmann, U. Franke, and J. Denzler, “Visibility Guided NMS: Efficient boosting of amodal object detection in crowded traffic scenes,” in Proc. Workshop Mach. Learn. Auton. Driving, 2020.
[13] N. O. Salscheider, “FeatureNMS: Non-maximum suppression by learning feature embeddings,” in Proc. 25th Int. Conf. Pattern Recognit., 2021, pp. 7848–7854.
[14] Y. Liu, L. Liu, H. Rezatofighi, T.-T. Do, Q. Shi, and I. Reid, “Learning pairwise relationship for multi-object detection in crowded scenes,” 2019, arXiv:1901.03796.
[15] Y. Song, Q.-K. Pan, L. Gao, and B. Zhang, “Improved non-maximum suppression for object detection using harmony search algorithm,” Appl. Soft Comput., vol. 81, 2019, Art. no. 105476.
[16] J. Yan, H. Wang, M. Yan, D. Wenhui, X. Sun, and H. Li, “IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery,” Remote Sens., vol. 11, 2019, Art. no. 286.
[17] D. Wang, X. Chen, H. Yi, and F. Zhao, “Improvement of non-maximum suppression in RGB-D object detection,” IEEE Access, vol. 7, pp. 144134–144143, 2019.
[18] D. Hema and S. Kannan, “Evaluating the robust non-maximum likelihood using the combined linear and nonlinear function in NMS for object detection,” in Proc. 3rd Int. Conf. Intel. Sustain. Syst., 2020, pp. 940–944.

[19] L. Fu, J. Zhang, and K. Huang, “Mirrored non-maximum suppression for accurate object localization,” in Proc. 3rd IAPR Asian Conf. Pattern Recognit., 2015, pp. 051–055.

[20] S. Qiu, W. Gongjian, Z. Deng, J. Liu, and Y. Fan, “Accurate non-maximum suppression for object detection in high-resolution remote sensing images,” Remote Sens. Lett., vol. 9, pp. 238–247, 2018.

[21] Y. He, X. Zhang, M. Savvides, and K. Kitani, “Softer-NMS: Rethinking bounding box regression for accurate object detection,” 2018. [Online]. Available: github.com/facebookresearch/detection

[22] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, “Acquisition of localization confidence for accurate object detection,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 784–799.

[23] W. Ma, K. Li, and G. Wang, “Location-aware box reasoning for anchor-based single-shot object detection,” IEEE Access, vol. 8, pp. 129300–129309, 2020.

[24] J. Zeng, J. Xiong, X. Fu, and L. Leng, “ReFPPN-FCOS: One-stage object detection for feature learning and accurate localization,” IEEE Access, vol. 8, pp. 225052–225063, 2020.

[25] A. Kumar, G. Brazil, and X. Liu, “GrooMed-NMS: Grouped mathematically differentiable NMS for monocular 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8969–8979.

[26] N. Salscheider, “FeatureNMS: Non-maximum suppression by learning feature embeddings,” in Proc. 25th Int. Conf. Pattern Recognit., 2021, pp. 7848–7854.

[27] I. Cortés, J. Beltrán, A. de la Escalerla, and F. Garcia, “SinaNMS: Non-maximum suppression with siamese networks for multi-camera 3D object detection,” in Proc. IEEE Intell. Veh. Symp., 2020, pp. 933–938.

[28] A. Herout, A. Herout, M. Hradiš, M. Hradiš, P. Zemˇci, “EmNS: Early non-maxima suppression: Speeding up pattern localization and other tasks,” Pattern Anal. Appl., vol. 15, no. 2, pp. 121–132, 2012.

[29] H. Huang, Y. Feng, M. Zhou, B. Qiang, J. Yan, and R. Wei, “Receptive field fusion RetinaNet for object detection,” J. Circuits Syst. Comput., vol. 30, 2021, Art. no. 2150184.

[30] J. Li and S. Ghosh, “Quantum-sot QUBO suppression for accurate object detection,” in Computer Vision – ECCV 2020, Berlin, Germany: Springer, 2020, pp. 158–173.

[31] W. Han et al., “Seq-NMS for video object detection,” 2016, arXiv:1602.08465.

[32] M. Zhao, K. Ning, S. Yu, and L. Lint, “Quantizing oriented object detection network via outlier-aware quantization and IoU approximation,” IEEE Signal Process. Lett., vol. 27, pp. 1914–1918, 2020.

[33] Z. Lin, Q. Wu, S. Fu, S. Wang, Z. Zhang, and Y. Kong, “Dual-NMS: A method for autonomously removing false detection boxes from aerial images object recognition results,” Sensors, vol. 19, 2019, Art. no. 4691.

[34] D. Oro, C. Fernández, X. Martorell, and J. Hernando, “Work-efficient parallel non-maximum suppression kernels,” Comput. J., vol. 65, pp. 773–782, 2022.

[35] C. Syneomidis, I. Mademlis, N. Nikolaidis, and I. Pitas, “Improving neural non-maximum suppression for object detection by exploiting interest-point detectors,” in Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process., 2019, pp. 1–6.

[36] S. Liu, D. Huang, and Y. Wang, “Adaptive NMS: Refining pedestrian detection in a crowd,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6452–6461.

[37] H. Teng, H. Lu, M. Ye, K. Yan, Z. Gao, and Q. Jin, “Applying of adaptive threshold non-maximum suppression to pneumonia detection,” in Proc. Int. Conf. Bio-Inspired Comp.: Theories Appl., 2020, pp. 518–528.

[38] L. Cai et al., “MaxpoolNMS: Getting rid of NMS bottleneck in two-stage object detectors,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9348–9356.

[39] Z. Liu, T. Zheng, G. Xu, Z. Yang, H. Liu, and D. Cai, “TFTNeXt for real-time object detection,” Neurocomputing, vol. 433, pp. 59–70, 2020.

[40] Z. Shi, H. Gao, Y. Tang, H. Zheng, S. Kang, and Y. Liu, “Universal optimization strategies for object detection networks,” Int. J. Pattern Recognit. Artif. Intell., vol. 35, no. 05, pp. 2155005, 2021.

[41] J. Wang, X. Yin, L. Wang, and L. Zhang, “Hashing-based non-maximum suppression for crowded object detection,” 2020, arXiv:2005.11426.

[42] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: Real-time instance segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9156–9165.

[43] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: Dynamic and fast instance segmentation,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., Red Hook, New York, NY, USA: Curran Associates, 2020.

[44] E. Park and A. Berg, “Learning to decompose for object detection and instance segmentation,” 2015, arXiv:1511.00449.

[45] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” 2019, arXiv:1904.07850.

[46] P. Sun, Y. Jiang, E. Xie, Z. Yuan, C. Wang, and P. Luo, “What makes for end-to-end object detection?”, in Proc. 38th Int. Conf. Mach. Learn., 2021, pp. 9934–9944.

[47] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, “End-to-end object detection with fully convolutional network,” 2020, arXiv:2012.03544.

[48] Q. Zhou, C. Yu, S. Chen, Z. Wang, and H. Li, “Object detection made simpler by eliminating heuristic NMS,” 2021, arXiv:2101.11782.
Andrew J. Shepley received the BSc honours degree from the University of New South Wales (UNSW), the Graduate Diploma in information science from the UNSW, the Graduate Diploma in education from the UNE, and the PhD degree in computer science from UNE. He is a lecturer and researcher with the University of New England (UNE), NSW, Australia. His research experience is centered on implementing algorithms to improve the accuracy and efficiency of deep learning systems and artificial intelligence. His current research involves the development of biometrics for individual recognition.

Paul Kwan (Senior Member, IEEE) received the BSc degree in computer science from Cornell University, the MSc degree in computer science from the University of Arizona, USA, and the PhD degree in advanced engineering systems, specialized in intelligent interaction technologies from the University of Tsukuba, in Japan. He is a professor of computer science with the University of New England in Armidale, Australia. His research interests include artificial intelligence, computer vision, computational modeling, 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 Science Society since 2006, a senior member of Association for Computing Machinery since 2008.

Gregory Falzon (Member, IEEE) received the Graduate Diploma Education and BSc (Hons I) from the University of New England, and the PhD degree in biomedical image analysis from the University of New England. He is associate professor with the Precision Agriculture Systems, Flinders University, Adelaide, Australia. He also holds adjunct associate professor status and was previously a senior lecturer in computational science with the University of New England in Armidale, Australia. Major professional achievements include the 2020 ATSE ICM Agrifood Innovation Award for advancing AI in agriculture; President’s Medal, ASCCT; Science and Innovation Award, Australian Government Department of Agriculture, and a solid track-record of industry funding. Current research spans the development of novel technologies especially machine vision systems and their applications to agriculture.

Ljiljana Brankovic received the graduate degree in electrical engineering from the University of Belgrade, Serbia, and the PhD degree in computer science from the University of Newcastle, NSW, Australia. She is a professor in computational science with the University of New England (UNE), Australia. Her main research interests include algorithms, cybersecurity and privacy, and graph theory and combinatorics. She is a fellow of the Institute of Combinatorics and its Applications and a life member of the Combinatorial Mathematics Society of Australasia. From 2007 to 2020, she served as an Australian Computer Society Chair of the National Committee for Computer Security.

[70] S. Prokudin, D. Kappler, S. Nowozin, and P. Gehler, “Learning to filter object detections,” in Lecture Notes in Computer Science, vol. 10496, Cham, Switzerland: Springer, 2017.

[71] M. Jiang et al., “Inference adaptive thresholding based non-maximum suppression for object detection in video image sequence,” in Proc. 3rd Int. Conf. Innov. Artif. Intell., 2019, pp. 21–27.

[72] J. Gu, C. Lan, W. Chen, and H. Han, “Joint pedestrian and body part detection via semantic relationship learning,” Appl. Sci., vol. 9, 2019, Art. no. 752.

[73] Q. Xie et al., “Vote-based 3D object detection with context modeling and SOB-3DNMS,” Int. J. Comput. Vis., vol. 129, pp. 1857–1874, 2021.

[74] X. Shi, Z. Chen, and T.-K. Kim, “Distance-normalized unified representation for monocular 3D object detection,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 91–107.

[75] S. Gidaris and N. Komodakis, “Object detection via a multi-region and semantic segmentation-aware CNN model,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1134–1142.

[76] J. Ferryman and A. Ellis, “PETS2010: Dataset and challenge,” in Proc. IEEE 7th Int. Conf. Adv. Video Signal Based Survell., 2010, pp. 143–150.

[77] M. Barekatain et al., “Okutama-action: An aerial video dataset for concurrent human action detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 2153–2160.

[78] J. Zhang et al., “Attribute-aware pedestrian detection in a crowd,” IEEE Trans. Multimedia, vol. 23, pp. 3085–3097, 2020.

[79] R. E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” Mach. Learn., vol. 37, no. 3, pp. 297–336, 1999.

[80] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2001, pp. 1–1.

[81] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao, “Generalizable pedestrian detection: The elephant in the room,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11323–11332.

[82] S. Qiao, L. Chen, and A. Yuille, “DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 10208–10219.

[83] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proc. Comput. Vis. – ECCV, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds, Cham, Germany, 2014, pp. 818–833.