Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07469-x

ORIGINAL ARTICLE

Selection of object detections using overlap map predictions

Md Sohel Rana[1] · Aiden Nibali[1] · Zhen He[1]

Received: 29 October 2021 / Accepted: 24 May 2022

Abstract
Advances in deep neural networks have led to significant improvement of object detection accuracy. However, object detection in crowded scenarios is a challenging task for neural networks since extremely overlapped objects provide fewer visible cues for a model to learn from. Further complicating the detection of overlapping objects is the fact that most object detectors produce multiple redundant detections for single objects, which are indistinguishable from detections of separate overlapped objects. Most existing works use some variant of non-maximum suppression to prune duplicate candidate bounding boxes based on their confidence scores and the amount of overlap between predicted bounding boxes. These methods are unaware of how much overlap there actually is between the objects in the image, and are therefore inclined to merge detections for highly overlapped objects. In this paper, we propose an overlap aware box selection solution that uses a predicted overlap map to help it decide which highly overlapping bounding boxes are associated with actual overlapping objects and should not be pruned. We show our solution outperforms the state-of-the-art set-NMS bounding box selection algorithm for both the crowdHuman dataset and a sports dataset.

Keywords Object detection · Overlapping object detection · Overlap map · Pixel voting

1 Introduction

Object detection is an important research topic in the computer vision field with many real-life applications such as athlete detection for sport performance analysis [30], pedestrian detection for automated vehicles [18], intruder detection for security surveillance systems [24], concrete crack detection [7], and geospatial image analysis (a particularly challenging application of object detection to optical remote sensing [3]). With recent advancements in the field of deep learning, detection performance has improved to a great extent. However, object detection is still a challenging problem in crowded scenes where multiple objects tend to occlude each other. In such scenarios established detectors often fail to detect the overlapping objects.

In object detection tasks, an object detection is defined as an axis aligned bounding box in 2D image space. Almost all modern deep learning based approaches [4, 9, 11, 15, 18, 28] detect object bounding boxes in two distinct steps (shown in top half of Fig. 1a, labelled as overlap agnostic box selection). First, a model takes an image as input and predicts bounding boxes with associated confidence scores for all objects visible in the image. This initial set of predictions usually contains a lot of redundancy in the form of multiple, near-duplicate bounding boxes for each object. Next, a post processing step is performed where a selection algorithm is used to remove redundant bounding boxes. In this paper we call this post processing task the *bounding box selection* problem. Recent works achieve strong object detection accuracy by following this two-step process, using a deep convolutional neural network (CNN) for the initial prediction step and non-maximum suppression (NMS) for the final bounding box selection step [4, 11, 18, 28]. However, almost all existing bounding box selection algorithms still have a lot of trouble distinguishing between multiple redundant predictions for the same object and predictions

✉ Md Sohel Rana
M.Rana@latrobe.edu.au

Aiden Nibali
A.Nibali@latrobe.edu.au

Zhen He
Z.He@latrobe.edu.au

[1] Department of Computer Science, La Trobe University, Melbourne, Victoria 3086, Australia

Published online: 20 June 2022

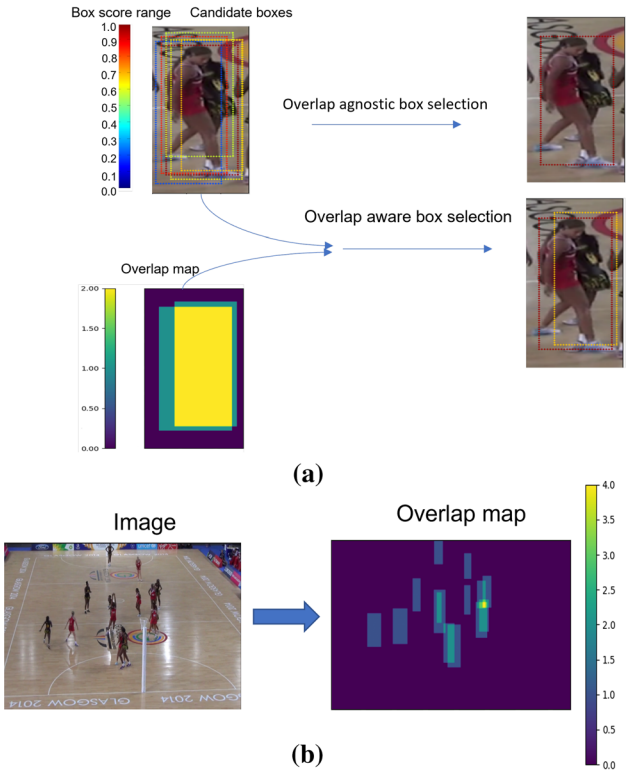

Fig. 1 **a** Overlap aware bounding box selection vs overlap agnostic bounding box selection. **b** The ground truth overlap map for a full image

which indicate the presence of distinct overlapping objects. This is because NMS is agnostic to the level of actual overlap between objects and must select boxes using only the predicted boxes and confidences.

NMS is the de facto standard for bounding box selection [4, 11, 18, 28]. NMS first sorts candidate bounding boxes based on their associated confidence scores. It then iteratively selects the highest confidence bounding box and discards all other boxes that highly overlap with that box. Overlap between boxes is measured using intersection over union (IoU) and compared against a threshold value when discarding boxes. However, NMS is an imperfect algorithm and can fail in scenarios where objects are highly overlapping. Figure 1 shows an example where NMS selects incorrect bounding boxes from candidate bounding boxes. Since all bounding boxes are overlapping more than the IoU threshold, NMS discards all but one of the bounding boxes despite there being two objects in the image. This is a hard limitation of NMS—we cannot simply resolve the issue by choosing a more permissive IoU threshold because that would result in also selecting redundant boxes, thus undermining the purpose of NMS. In this work, we address the limitation of existing overlap agnostic bounding box selection approaches like NMS by incorporating additional information explicitly representing the degree of object overlap in the selection process. In contrast to NMS, our method does not require an IoU threshold to be set and instead relies on a predicted overlap map for making such decisions. The predicted overlap map varies depending on the image used as input.

Our proposed bounding box selection algorithm selects bounding boxes based on a predicted overlap map generated from ground truth bounding box annotations. An overlap map is a 2D heatmap equal in size to the original image where each pixel of the overlap map determines the number of bounding boxes intersecting the pixel. Figure 1b shows an image and corresponding overlap map. For example, it shows the pixel value is three in the shared region of the three overlapping people. Since ground truth bounding boxes are not available at inference time, we create an overlap map prediction model to predict overlap maps. In particular, we use the semantic segmentation model, deeplabv3+ [1] as the backbone of our proposed overlap map model.

Once we have an overlap map, we need a bounding box selection algorithm that selects the bounding boxes by minimising the difference (via a cost function) between the overlap map produced by the selected bounding boxes and the overlap map predicted separately by our model. The exhaustive brute force solution involves considering the power set of candidate boxes, which has exponential complexity with respect to the number of boxes. This is generally not practical. Therefore, we propose a greedy bounding box selection algorithm called 'pixel voting' (PV) which is computationally feasible with respect to the number of candidate boxes. The pixel voting approach uses a heuristic score based on a candidate bounding box overlap map to guide the selection of candidate bounding boxes.

To evaluate the performance of our overlap-aware bounding box selection we conducted experiments on two datasets (the crowdHuman dataset [4] and a sports dataset) which contain frequently overlapping objects. Using a state-of-the-art prediction model [4], we produced candidate object detections. Separately, we trained a model to produce overlap maps from images. We then applied the pixel voting bounding box selection algorithm to select actual bounding boxes from duplicate predictions using overlap map guidance.

Our results on both person-detection datasets demonstrate that our proposed solution achieves better results than the existing set-NMS approach in terms of localisation-precision-recall (LRP) [22] detection metrics. We also show that our pixel voting bounding box selection accuracy improves drastically when ground truth overlap maps are used which shows the potential of our method to offer even better performance if a better overlap map prediction is developed.

In this paper, we make the following main contributions:

- We propose a novel approach for selecting candidate object bounding boxes that makes use of a predicted overlap map to disambiguate bounding boxes of truly overlapping objects from duplicate bounding boxes of the same object. This contrasts with most existing methods which use the amount of overlap between predicted bounding boxes and the confidence score for box selection decisions.
- In contrast to most existing bounding box selection algorithms, our method is specifically designed to work well for crowded scenes.
- We compare different overlap map prediction models to find the configuration which generates overlap maps with highest accuracy.
- We propose a fast overlap aware bounding box selection algorithm called 'pixel voting' that selects bounding boxes using the overlap map.
- We perform an extensive experimental study involving two different datasets to compare the performance of our pixel voting bounding box selection algorithm with the existing state-of-the-art set-NMS [4] based approach. The results show overlap-aware pixel voting outperforms the set-NMS approach.

2 Related works

2.1 Object detection approaches

Object detection has been studied by many researchers since the early age of computer vision research [5, 8, 32]. During the last few years, neural network based models [4, 15, 16, 27, 28] have improved detection performance significantly. Neural network based object detectors are mainly classified into two categories: one stage object detectors [19, 27, 33] and two stage object detectors [4, 15, 25, 28, 35].

One stage detectors [19, 27, 33] directly predict bounding boxes from the convolutional neural network feature maps. YOLO (You Only Look Once) [27] is one of the well-established single stage detection model that maps the grid cells in image space to bounding boxes. In particular, image feature maps are split across multiple grid cells and each grid cell is assigned fixed anchor boxes of varying shapes. During inference, the output head of the model expresses bounding box predictions relative to nearby grid cell anchor boxes.

Two stage object detectors are computationally expensive but provide better accuracy than one stage detectors. Faster R-CNN [28] is a pioneering two stage object detector widely used for many object detection applications. First, it uses convolutional neural networks (resnet-50 or resnet-100) as backbone to create feature maps from images. A set of region proposals are selected from the feature maps. In the next stage, region proposals are mapped into feature maps. The output consists of two classification heads and a regression head. The classification head determines the class of a region proposal and the regression head regresses the bounding box to the location with the highest classification score. Faster-RCNN often produces multiple predictions for a single object. Faster-RCNN uses the non-maximum suppression bounding box selection algorithm to prune replicated detections.

Most existing object detectors [4, 11, 18, 19, 27, 28, 33], regardless of whether they are one or two stage detectors, have difficulty detecting extremely overlapping objects since they use bounding box selection algorithms which aggressively prune overlapping bounding boxes under the assumption that they are duplicate detections of the same object. We consider the limitation of existing bounding box selection algorithms as an opportunity to address the problem of detection in highly overlapped situations.

2.2 Bounding box selection algorithms

The most popular bounding box selection algorithm is non-maximum suppression (NMS). However NMS is often overly aggressive at removing overlapping bounding boxes leading to false negative detections. Recently an extension of NMS called set-NMS [4] shows significant improvement in detecting multiple overlapping objects. Set-NMS works by using an object detection model which predicts sets of bounding boxes instead of individual bounding boxes. As a result, two objects which are overlapping will have separate bounding box predictions and the algorithm will either prune or keep both predicted bounding boxes belonging to the same set. This allows set-NMS to avoid erroneously pruning an object that highly overlaps with one another. set-NMS requires the size of the overlapping set to be predefined (the authors use a set size of 2). However, in practice there could be any number of overlapping boxes. The predetermined set size for set-NMS is not a hard limit in OLMM-c, since the set-NMS overlap map input is just a "hint" from which the actual overlap map is generated. Although one variant of our approach (OLMM-c) is dependent on the output of set-NMS and thus has the same limitations as set-NMS, our overall approach does not depend on set-NMS and thus does not require the number of overlapping bounding boxes to be predetermined. For example our models OLMM-a and OLMM-b are independent of set-NMS and thus do not have this limitation. Furthermore, in the future we could develop a better overlap map prediction algorithm that is built on top of a better segmentation algorithm (a hypothetical "OLMM-d")

which is independent of set-NMS. Additionally, performance can be improved further by also leveraging future state-of-the-art detection algorithms which do not require pre-determining the number of overlapping bounding boxes.

Songtao et al. [18] proposed a bounding box selection method called adaptive NMS that decays neighbouring bounding box scores rather than making a hard pruning decision as in standard NMS. They calculate each bounding box's density score based on maximum IoU overlap from remaining boxes. Then the NMS IoU threshold is chosen depending on the degree of individual box density. Jan et al. [11] proposes training a model to replace NMS. The trained model rescores the predicted candidate bounding boxes by reducing the scores of bounding boxes that are duplicates of the closest bounding box to the ground truth bounding box. They perform the rescoring by considering pairs of neighbouring boxes at once instead of scoring bounding boxes independently like most existing object detectors. As is the case with standard NMS, neither of these approaches are aware of overlapping objects in the input image as they select boxes.

Han et al. [12] propose an end-to-end object relation model that prunes duplicate bounding boxes without the need for any further post-processing. It uses an attention module that captures the relationships between objects based on their appearance and location. The relation module is used for predicting the initial set of bounding boxes and also used for duplicate removal. Recently Zixuan et al. [34] proposed a method that models each pedestrian using a beta distribution within the bounding box of the pedestrian. This allows higher probability to be assigned to the pixels of the pedestrian within the bounding box rather than assigning equal probability to the pixels in the bounding box. Their method performs bounding box selection using a method called BetaNMS which uses KL divergence between the beta distribution of pairs of bounding boxes to determine how likely two bounding boxes correspond to the same person. Both [12] and [34] box selection algorithms are tightly integrated with the detector itself making them incompatible with current and future alternative object detectors. In contrast, our box selection algorithm is decoupled from the object detector.

2.3 Image segmentation algorithms

The key component of our bounding box selection algorithm is the overlap map. The overlap map represents the number of objects overlapping each pixel of the image. Predicting an overlap map is a very similar problem to semantic segmentation where the task is to label each pixel in terms of which class it belongs to. Therefore most existing semantic segmentation solutions can be used to output the overlap map.

Semantic segmentation has improved significantly with the advancement of recent deep convolutional neural network architectures [20, 26, 29, 31]. Olaf et al. [29] segment medical images using a U-shaped network architecture that results in more precise localization of objects. Recently one of the best performing semantic segmentation methods is HRnet [31]. HRnet [31] uses multiple resolution model streams in parallel and combines the streams at different intermediate layers. However, this architecture is GPU memory intensive due to its multi-stream architecture. In contrast, Deeplabv3+ [1] is a popular semantic segmentation algorithm that is fast, uses much less GPU memory and gives near state-of-the-art accuracy. Therefore we use Deeplabv3+ [1] as the backbone of our proposed overlap map model. Deeplabv3+ employs encoder-decoder structure where it uses the ResNet model [10] as the backbone of the encoder. Deeplabv3+ encoder uses atrous convolution with different kernel size to capture scale invariant features. The output features with different atrous rates are combined and used in the input of the decoder and finally upsampled with additional convolutional layers.

Instance segmentation [9, 13] and panoptic segmentation [2, 21] problems are also related to our overlap map prediction problem. Instance segmentation is the task of detecting each distinct object of interest appearing in an image. Lei and Tai et al. [13] proposed a deep occlusion-aware instance segmentation model that uses different layers for occluding and occluded objects. This model requires annotation of both occluding and occluded objects explicitly. Recent research introduced panoptic segmentation, a new segmentation task by combining both semantic and instance segmentation. Existing work [2, 21] proposed panoptic segmentation that does per pixel instance classification without outputting the bounding box of the objects. In this task, objects are only represented as irregular shapes that are defined by the visible pixels.

3 Overlap aware bounding box selection

3.1 Overview of approach

Our aim is to propose a bounding box selection algorithm that removes duplicate bounding boxes to reduce false positive detections while retaining true positive bounding boxes even if they significantly overlap each other. Traditional approaches such as non-maximum suppression (NMS) only use two sources of information to remove redundant bounding boxes: confidence scores for each bounding box; and the IoU between bounding boxes which captures the degree of overlap between bounding boxes.

However this is not enough information for situations such as when one person is standing in front of another, thereby having a very similar bounding box with a high degree of overlap. In this situation a traditional NMS algorithm is likely to prune one of the true positive bounding boxes due to the high degree of overlap between the bounding boxes. To address this lack of information our solution predicts an overlap map that effectively tells our overlap-aware bounding box selection algorithm the degree of bounding box overlap for every location in the image.

Figure 2a shows an overview of our overlap-aware bounding box selection method. First an object detector is used to output candidate bounding boxes B. Then a box selection algorithm selects a subset of bounding boxes $B_s \subseteq B$ that best matches the ground truth bounding boxes. Instead of using the traditional NMS bounding box selection algorithm we use an overlap-aware bounding box selection algorithm that selects bounding boxes using a predicted overlap map. We train a fully-convolutional model to output an overlap map OL_m. The overlap map allows the model to determine if removing a bounding box will result in the correct number of bounding boxes overlapping a given region in the image. Our bounding box selection algorithm selects the subset B_s which minimizes the cost between the overlap produced from the predicted overlap map OL_m. A brute force solution would need to try every possible subset $B_s \subseteq B$ which has complexity $O(2^n)$ where $n = \mid B \mid$, the number of candidate boxes. This would not be a feasible solution. Instead we propose a new overlap-aware bounding box selection algorithm called 'pixel voting' which takes a greedy approach with lower computational complexity than the brute force solution. The idea behind pixel voting is to iterate through all boxes and find out if adding that bounding box reduces the overlap map costs. We describe pixel voting algorithm details in Sect. 3.4. We add two features to the pixel voting algorithm and detail it in Sect. 3.6, one of which allows us to trade off computational complexity for better accuracy. We describe the overlap map creator architecture in Sect. 3.2.

3.2 Overlap map prediction model

The overlap map prediction can be framed in a very similar way to the semantic segmentation problem. We label each pixel with the number of ground truth bounding boxes that overlap that pixel, an automatic process which does not require any further manual annotation. The overlap map can be represented as either a single-channel image (for regression) or one-hot encoded as a multi-channel image

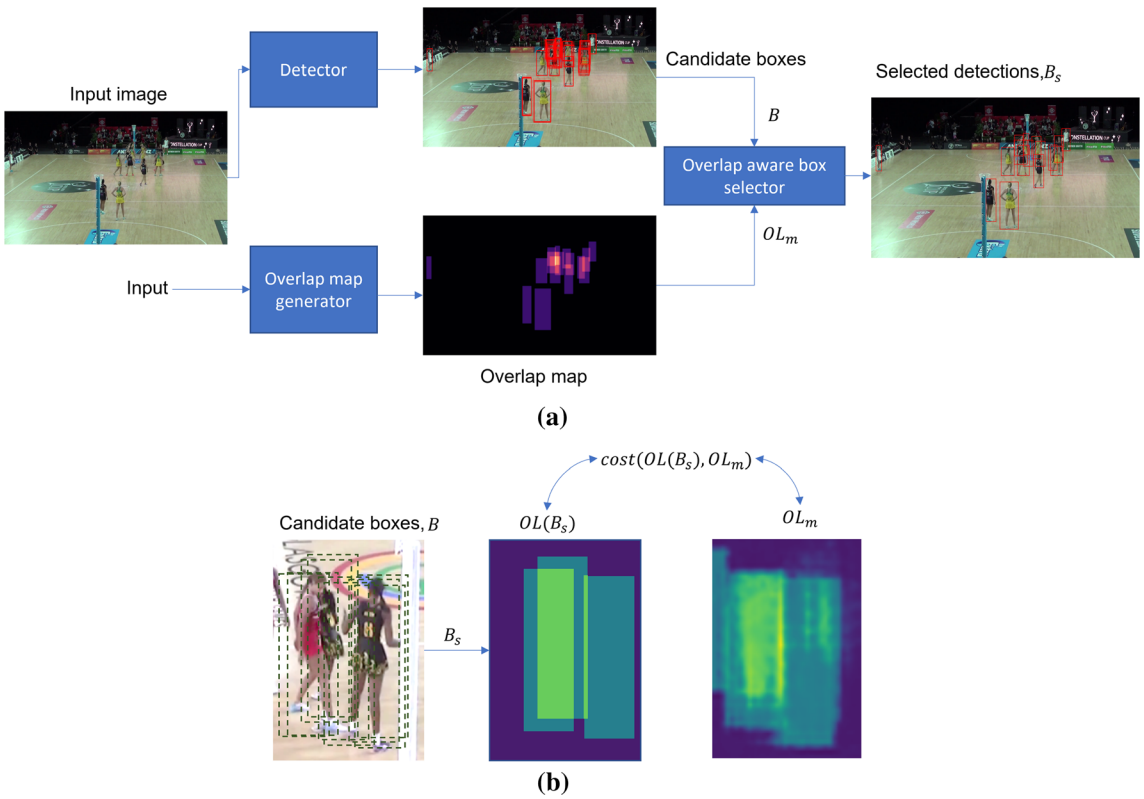

Fig. 2 **a** Overall architecture of our overlap map based bounding box selection. **b** Visualisation of selected boxes (B_s) overlap map and model produced overlap map OL_m

(for classification). In this paper, we frame the problem as a regression problem (using the former overlap map representation) as preliminary experimentation indicated that this produced more accurate results. This may be due to the ability for regression to better handle distribution imbalance in the output. Another advantage of using the regression approach is that the number of predicted overlapping bounding boxes is unbounded. Due to the fact that the overlap map prediction problem is very similar to the semantic segmentation problem we leverage the established deeplabv3+ architecture [1] designed for semantic segmentation to solve our overlap map prediction problem. We modify deeplabv3+ by squeezing the final classification layer to output a *WXH* dimensional tensor where *WXH* is the height and width of the image and each pixel represents the number of objects shared by that pixel.

We propose three different solution configurations for solving the overlap map prediction problem. The configurations differ on what they take for input, as shown in Fig. 3. The first solution configuration shown in Fig. 3a just takes the image $X_I \in R^{3XWXH}$ as input and outputs $OL_m \in R^{WXH}$ where W and H represent width and height of the image, respectively. We call this model configuration OLMM-a. The second solution configuration (shown in Fig. 3b) takes both the image $X_I \in R^{3XWXH}$ and the overlap map created from all of the candidate bounding boxes $X_B \in R^{WXH}$ as input. We call this configuration OLMM-b. The detection overlap map and set-NMS overlap map are simply "images" from the bounding boxes outputted by the detector and from set-NMS, respectively. Each pixel value represents the number of boxes overlapping it. Taking the candidate overlap map as input gives the model extra information on the location of the candidate bounding boxes predicted by the object detection model.

The third solution configuration (shown in Fig. 3c) takes the same input as OLMM-b and adds the overlap map $X_S \in R^{WXH}$ created from the output of the set-NMS bounding box selection algorithm as input and outputs $OL_m \in R^{WXH}$. This allows the model to focus its resources on using the other two sources of information to fine tune the overlap map created from the output of set-NMS. We use a skip connection by adding set-NMS bounding box produced overlap map X_S with the overlap map model output. This skip connection preserves the initial set-NMS overlap map and allows the model to refine the map by adding to or subtracting from it. We call this solution OLMM-c. Note this approach can utilize the output of any

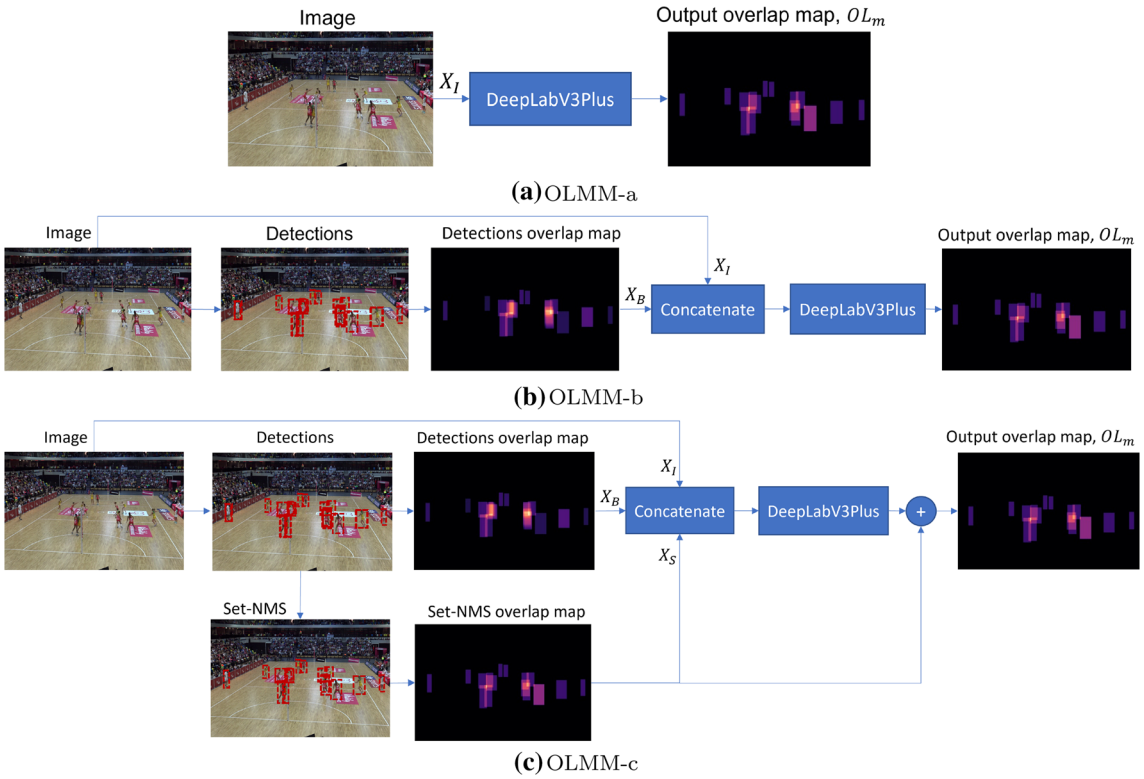

Fig. 3 OLMM-a (Fig. 3**a** shows the model that takes the only image as input. OLMM-b (Fig. 3**b** shows the model that takes both the input image and an overlap map created from detector outputted candidate bounding boxes as input. OLMM-c (Fig. 3c) shows the model that takes, the input image, the candidate boxes overlap map and the overlap map created from the setnms selected bounding boxes

other bounding box selection algorithm instead of set-NMS.

It is important to note that the use of the predicted overlap map for making bounding box selection decisions relies on the training data matching the testing conditions. Hence in this work we assume the training and testing data are drawn from the same overall data distribution. This is a necessary pre-condition for almost all machine learning based computer vision solutions, including the object detection models that we are seeking to improve. Although we agree that our approach will only work under this assumption, our results show that as long as this assumption is met it will work for different datasets (i.e. Sports and CrowdHuman [4]).

Conceptually, one might suppose that a lack of spatial correlation between input visual evidence and the expected rectangular shapes of overlap map predictions would severely hinder prediction accuracy. However, there is a precedent for this kind of prediction being made (e.g. CornerNet [14] is able to locate bounding box coordinates for non-rectangular objects using a spatial heatmap). Furthermore, our results show that it is indeed possible for the extended receptive field of a convolutional neural network model to form predictions that are rectangular in shape and indicate the number of overlapping objects by using surrounding visual evidence.

3.2.1 Loss function

We use the L2 distance between the ground truth overlap $OL_{gt} \in R^{WXH}$ and the predicted overlap map $OL_m \in R^{WXH}$ to compute the loss. This is the loss shown by Eq. 1:

$$loss = \sum (OL_m - OL_{gt})^2 \tag{1}$$

3.3 Overlap-aware box selection problem

In our overlap-aware approach we use candidate bounding box (the predicted bounding boxes from the detection model) positions $B \in R^{nX4}$ as input to the bounding box selection algorithm where n is the number of boxes. The task is to select a subset of the candidate bounding boxes $B_s \subseteq B$ which are closest to the ground truth bounding boxes. We first use a model m to predict an overlap map $OL_m \in R^{WXH}$. An overlap map is effectively a 2D image with width W and height H equal in size to the original image where each pixel in the overlap map has a value equal to the number of objects overlapping that pixel (i.e. the number of bounding boxes which contain that pixel). For a particular set of selected bounding boxes B_s we can derive another overlap map $OL_s \in R^{WXH}$ and compare it with OL_m.

The overlap-aware box selection problem is thus defined as selecting the optimal B_s which produces an overlap map OL_s that minimizes the overlap map error (OME) with respect to OL_m as defined by the equation as follows:

$$OME(OL_s, OL_m) = \mid \sum_{i}^{n} OL_m(i) - OL_s(i) \mid \tag{2}$$

Where i is the i^{th} pixel in the overlap map and n is the total number of pixels in the image.

Note a brute force solution to the above problem would require trying every subset of B and computing the cost in Eq. 2. This would be prohibitively expensive to compute. Therefore we have developed a greedy framework for solving the overlap-aware box selection problem which is computationally feasible.

3.4 Greedy overlap-aware box selection framework

In this section, we present a greedy framework for overlap-aware box selection. The framework iteratively selects one bounding box at a time to add into B_s. First all remaining bounding boxes are sorted according to a heuristic score that represents how desirable it is to include a bounding box in B_s. Then the top scoring bounding box is considered for inclusion into the B_s when inclusion reduces OME cost defined in Eq. 2. The algorithm terminates when none of the boxes can help to reduce OME. Algorithm 1 shows the pseudo code for the greedy overlap-aware box selection framework.

Algorithm 1 Pseudo code for greedy overlap-aware box selection

initialization
  Set the selected bounding boxes $B_s = \emptyset$
  Set the remaining bounding boxes $B_r = B$
repeat
  Let $B_r^* =$ the subset of boxes in B_r that can lower OME
  Let $b =$ the box in B_r^* with highest heuristic score
  Move b from B_r to B_s
until no box in B_r can lower OME

A naive choice for the heuristic score is to simply use OME itself. This would effectively mean during each iteration you would always pick the bounding box that reduces *OME* the most. We call this the absolute error (*AE*) heuristic since *OME* is computed using the absolute error. Equation 3 below shows how *AE* can be computed for a bounding box b currently under consideration:

$$AE(OL_s, OL_m) = \left| \sum_{i \in b} OL_m(i) - OL_{sb}(i) \right| \quad (3)$$

Where i is a pixel that belongs to the bounding box b currently under consideration and $OL_{sb}(i)$ is the overlap map created from adding b to all the bounding boxes selected in previous iterations. Note just computing the absolute error within b is enough since the absolute error in the rest of the overlap map for the selected bounding boxes are unchanged with the inclusion of b.

The *AE* heuristic may seem like a wise choice since your overall objective is to find the B_s that minimizes *OME*. However, choosing the bounding box that reduces *OME* the most in the iteration is only a locally optimal solution, it may very well lead to a worse global solution when considering the entire set B_s as a whole. Upon further analysis it becomes evident that AE will tend to select larger bounding boxes which cover large areas of the overlap map which have not yet been covered adequately, even when smaller boxes more closely match the overlap map. Case 1 in Fig. 4 shows this situation. Notice in this case *AE* would select the red bounding box since it covers most of the area of the one-value overlap map regions (green regions) without covering much of the zero-value overlap map region (blue background). However, in the example it would be better to select the two smaller white bounding boxes since they fit tighter around the nonzero value overlap regions. A potential simple way to overcome *AE*s bias towards selecting larger bounding boxes is to use a heuristic score that computes the change in the normalized absolute error (*NAE*) where the normalization function simply divides *AE* by the number of pixels of the bounding box under consideration. Equation 4 below shows how *NAE* can be computed for a bounding box b currently under consideration:

$$NAE(OL_s, OL_m) = \left| \sum_{i \in b} OL_m(i) - OL_{sb}(i) \right| / |N_b| \quad (4)$$

Where $|N_b|$ is the number of pixels in the bounding box b. The other terms have the same meaning as Eq. 3.

The NAE heuristic would tend to select smaller bounding boxes than AE. For example in case 1 of Fig. 4, NAE would end up selecting the two smaller bounding boxes which in this case would be the best bounding boxes to select. However, there are other situations where it is

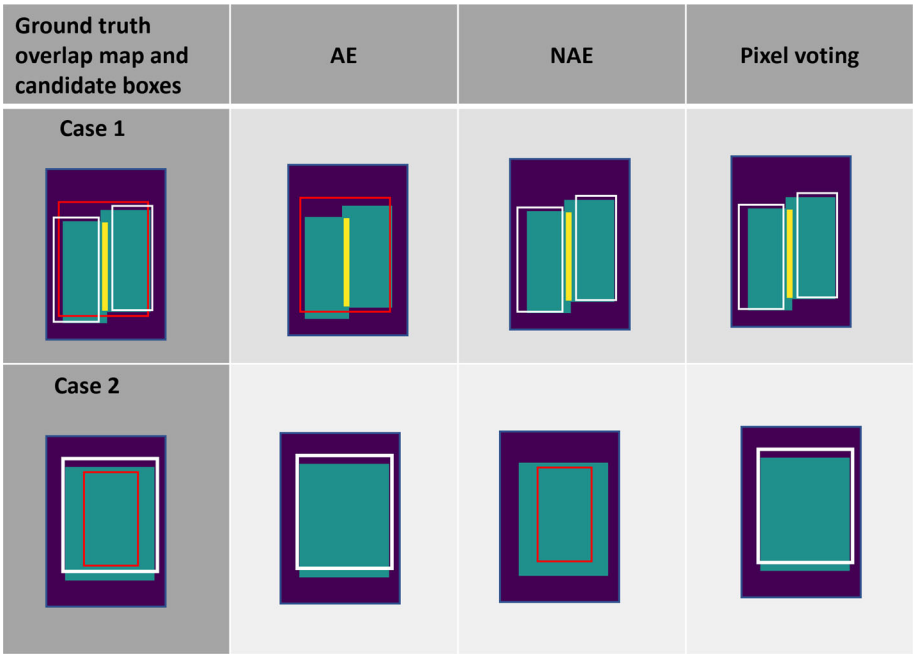

Fig. 4 This table illustrates how well the greedy box selection framework works using the three different heuristic scores of *AE*, *NAE* and pixel voting. The first column in the table shows the overlap map for the ground truth bounding boxes where the blue, green and yellow region represents regions with zero, one and two overlapping ground truth bounding boxes. The red and white boxes represent the candidate bounding boxes. In each case the white bounding boxes are the best bounding boxes for selection and the red bounding boxes should be avoided. Columns 2 to 3 show which of the candidate bounding boxes is selected when using each of the three heuristic scores

more desirable to select larger bounding boxes. Case 2 of Fig. 4 shows an example where it would be better to select the larger white bounding box, but NAE would instead select the smaller red bounding box.

Another obvious choice for the heuristic score is the confidence score for a given candidate bounding box. This is similar to the approach of NMS where only candidate bounding boxes above a confidence threshold are considered. However, this approach does not work well due to two properties of bounding box detectors: (1) Redundant detections are produced for each person, and (2) Detections (including redundancies) for unobscured people tend to have higher confidence scores than detections for obscured people. Consider the case where two people are highly overlapped in the image. In such a case if we select boxes based on confidence we would be unlikely to ever select a detection from the obscured person due to the multiple high-confidence redundant detections from the unobscured person.

Our greedy overlap-aware box selection shown in Algorithm 1 has a computational complexity of $O(n^2 log(n))$. This is due to the need to sort the remaining set of candidate bounding boxes in terms of the heuristic score when selecting each object.

3.5 Pixel voting heuristic (votemap)

As motivated in the previous section, there is a need to find a scoring function that better captures the overall benefit of selecting a bounding box versus selecting multiple alternative bounding boxes. Therefore we developed the pixel voting heuristic that computes a *votemap* from all remaining boxes and uses this to score each individual box. The *votemap* assigns higher scores to currently inadequately covered regions of OL_m for which few alternative bounding boxes can cover. For example in case 2 of Fig. 4 the pixel voting heuristic would prefer to select the white bounding box since this box contains many pixels that indicate overlap and are not covered by any alternative bounding box (red bounding box). In contrast for case 1 of Fig. 4 the pixel voting heuristic would not select the larger red bounding box despite it covering a larger region of nonzero overlap map. This is because in this case the *votemap* heuristic can tell that the two smaller white bounding boxes can cover the nonzero overlap map regions better (i.e. better alternatives exist).

The pixel voting heuristics creates a 2D *votemap* using a heuristic function that assigns higher values to regions according to two considerations. 1) Regions where the already selected bounding boxes overlap map OL_s values are much lower than the model predicted overlap map OL_m. These are the regions that already selected bounding boxes do not adequately cover. 2) Regions where the remaining bounding boxes overlap map OL_r has low values. These are regions where the remaining bounding boxes do not cover well (i.e. there are few alternatives to choose from). For example, in the case that there is only one remaining bounding box that can cover a region it is important that the bounding box is selected earlier since there are no alternative bounding boxes that can cover that region. Equation 5 below shows the formula used to compute pixel i of the *votemap*:

$$votemap(i, OL_m, OL_{sb}, OL_r) = (OL_m(i) - OL_{sb}(i))/OL_r(i) \tag{5}$$

The pixel voting heuristic score (PV) for a bounding box b is then computed using Eq. 6 shown below:

$$PV(OL_s, OL_m, OL_r) = \sum_{i \in b} votemap(i, OL_m, OL_{sb}, OL_r)/ \mid N_b \mid \tag{6}$$

Where $\mid N_b \mid$ is the number of pixels in bounding box b. Note the regions of the *votemap* which are not covered by any remaining bounding boxes are undefined. However, this is not a problem since we only compute votemap values for the remaining bounding boxes since these are the only bounding boxes we can potentially select next. Figure 5 illustrates the process of computing the *votemap* using an example set of input overlap maps (OL_m, OL_{sb}, OL_r).

The pixel voting heuristic algorithm just uses the greedy overlap-aware box selection algorithm described in Sect. 3.4 but with the heuristic score for a given bounding box computed using Eq. 6.

3.6 Improved pixel voting

In the standard pixel voting algorithm, at each iteration we select the bounding box with the highest *votemap* score among the remaining bounding boxes B_r that lowers *OME* cost. As mentioned in the previous section the *votemap* heuristic score better captures the overall benefit of selecting a box compared to using the *AE* heuristic score, despite *AE* more closely matching our final objective cost function (*OME*). We will now consider an extension to pixel voting that uses the *votemap* score and *OME* cost together to select the bounding box, which is somewhat of a hybrid between pixel voting and *AE*. In particular we first select the top K bounding boxes B_K according to the *votemap* score. Second, we select the bounding box with the lowest *OME* cost among all bounding boxes in B_K. We call this top-K pixel voting. In our experiments we use a K value of 6 which we found gives best performance.

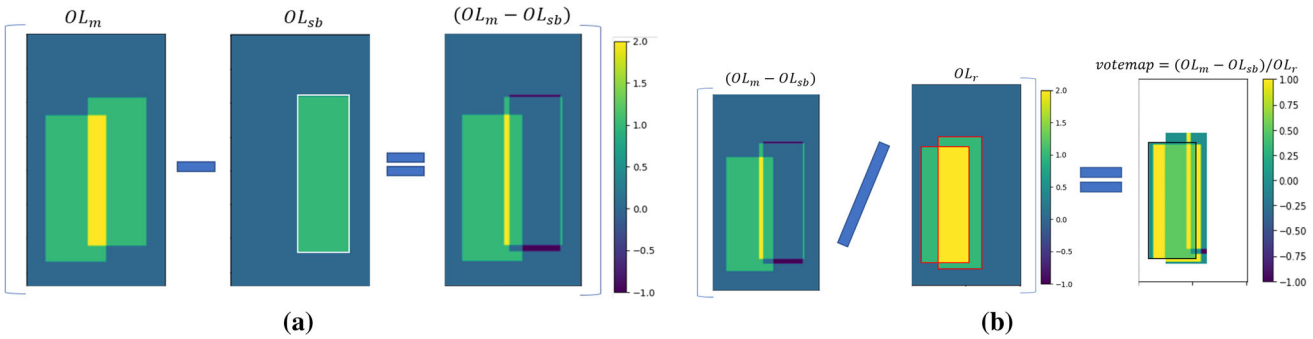

Fig. 5 *votemap* calculation from given overlap map OL_m, selected boxes overlap map OL_s and remaining boxes overlap map OL_r. (White, red and black bounding box represents selected boxes, remaining boxes and highest score votemap box respectively)

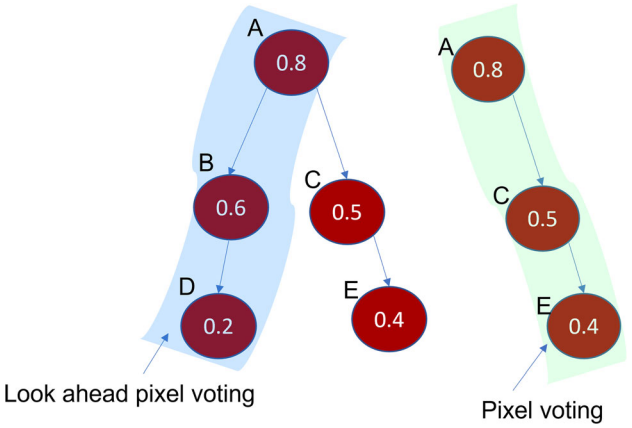

Fig. 6 Each circle represents a selected candidate bounding box and the number in the circle represents the *OME* if the box is selected. The look ahead pixel voting selects boxes *A*, *B*, *D* which results in a lower OME than greedy pixel voting which selects boxes *A*, *C*, *E*

Although standard pixel voting and top-*K* pixel voting consider the number of alternative bounding boxes, they are still ultimately greedy algorithms in that they select one bounding box at a time without fully calculating the impact on future selections. However, when selecting a bounding box *b* we may be able to achieve a lower final *OME* cost if we also consider multiple alternative selections. This is because selecting *b* may look sub-optimal now but when you consider subsequent selections the combination may lead to lower final OME cost. Therefore we extend top-*K* pixel voting to conduct a more thorough search by performing a best-first tree search with node expansion limited to *W*, and set $K = W$. This approach, which we call "look-ahead pixel voting", enables deep look ahead at each iteration when deciding which box to select at the cost of computational complexity. Figure 6 shows an OME cost comparison between greedy pixel voting and the look ahead pixel voting approach. Here, each circle represents a selected box and the number of the circle represents *OME* cost when that box is included on the selection list. After selecting box *A*, greedy pixel voting selects box *C* as it reduces the *OME* cost more than *B*. However, *D* reduces cost most if *B* was selected. Therefore, if we calculate the cost for both branches (*A*,*B*,*D* and *A*,*C*,*E*) and select the branch with minimum *OME* cost, we could achieve a more optimal solution. However, branching look-ahead adds a significant amount of computational overhead to pixel voting box selection which can be controlled by the parameter *W*.

4 Experimental setup

4.1 Dataset

We evaluated the various bounding box selection algorithms using a sports video dataset and the crowdHuman public dataset [4]. The sports video consisted of 53 short video clips which we split into 37 training and 16 testing video clips collected by the Australian Institute of Sport. These video clips are taken from 8 individual sports. Table 1 shows the total video clips for each individual sport. Each video clip had approximately 750 frames. The crowdHuman dataset contained 19370 images split into 15000 training and 4370 test images. The crowdHuman

Table 1 Individual sports video clip for train and test set

Sports name	Train set video clip	Test set video clip
Hockey	16	6
Netball	8	3
AFL	3	1
Basketball	3	2
Rugby	3	1
Beach volleyball	2	1
Cricket	1	1
Soccer	1	1
Total frames	27487	11936

dataset has more overlapping bounding boxes than the sports dataset. Therefore these two datasets allow us to evaluate the performance of different bounding box selection algorithms in both relatively lower and higher overlap situations.

4.2 Evaluation metrics

The average precision (*AP*) metric and its multi-class extension (*mAP*) are commonly used for evaluating object detection models. *AP* evaluates the recall and precision of predictions over a range of confidence thresholds by calculating the area under a precision-recall graph, based on the assumption that selected bounding boxes can be meaningfully ranked by confidence. This metric makes sense for evaluating algorithms that can have their precision/recall trade-off calibrated by adjusting a confidence threshold, but has limited utility in the case where the algorithm does not rely on a confidence threshold. Additionally, it does not evaluate the realistic need to set a confidence threshold based on the training set prior to evaluation on the test set. In practice, an object detector must preset a specific confidence threshold prior to its use. *AP* effectively ignores this aspect of the algorithm—since it evaluates across all possible confidence thresholds, it does not test how well we have selected a particular threshold. What we desire is a metric which evaluates how well a concrete set of predicted boxes matches the ground truth, with the selection of the confidence threshold (if applicable) being considered a part of the algorithm under evaluation.

In contrast to *AP*, the recently proposed Localization-Recall-Precision (*LRP*) [22] set of metrics can evaluate detector performance for a given confidence threshold. Specifically *LRP* evaluates a detector in terms of localisation ($LRP_{LocComp}$), recall (LRP_{FNComp}) and precision (LRP_{FPComp}) errors. The $LRP_{LocComp}$ (IoU localisation) metric is a sum of intersection over union error on all true positive bounding boxes. The LRP_{FNComp} (recall) metric is a measure of the proportion of false negative detections. The LRP_{FPComp} (precision) metric is a measure of proportion of false positive detections. The overall *LRP* error summarizes these three error components ($LRP_{LocComp}$, LRP_{FNComp}, LRP_{FPComp}) by first computing a weighted sum of the components. Then the weighted sum is divided by a normalising constant z where the constant is a summation of true positive, false positive and false negative (shown in equation 1 of [22]). Weight terms w_{IoU},w_{FP} and w_{FN} are the number of true positives, number of detections, number of ground truth boxes respectively. These weight terms are used to control each error component in *LRP* calculation. Given ground truth bounding boxes X and detections Y_s, *LRP* error is calculated using Eq. 7:

$$\begin{aligned}\mathrm{LRP}(X, Y_s) = \frac{1}{z}(w_IoU \cdot \mathrm{LRP}_{LocComp}(X, Y_s) \\ + w_FP \cdot \mathrm{LRP}_{FPComp}(X, Y_s) + w_FN \cdot \mathrm{LRP}_{FNComp}(X, Y_s))\end{aligned} \tag{7}$$

To mimic real-world deployment we set the confidence score threshold based on the best performance achieved on the training dataset and estimate the error for the corresponding thresholds on the test dataset. Another benefit of being able to measure performance at a specific confidence threshold is that we can draw a graph of *LRP* performance as the threshold is varied. Sect. 5.3 results of such an experiment.

We evaluate the overlap map model performance with a metric called mean-square-error (*MMSE*). *MMSE* is a metric that calculates the sum of mean-square-error for each image and then calculates the mean across all images of error. Equation 8 shows *MMSE* error for n images where OL^igt, OL^i_m is ground truth overlap map and model produced overlap of the i-th image, respectively:

$$MMSE = \sum_{i=1}^{n} MSE(OL^i_{gt}, OL^i_m)/n \tag{8}$$

4.3 Implementation details

Our overlap map prediction model uses the deeplabv3+ [1] semantic segmentation architecture with a resnet34 [10] backbone pretrained on imagenet [6] data. The model can be configured in any of the three ways described in Sect. 3.2 to perform overlap map prediction. We use the RAdam [17] optimizer with an initial learning rate of 2.5e-4 to train the model. In overlap map model training, we stopped early to avoid overfitting (based on validation loss). We implemented, trained and tested our deep learning models using the PyTorch deep learning software framework [23]. We used GPU servers with Nvidia RTX 2080Ti GPUs to conduct our experiments.

During training we oversample images with more densely overlapping bounding boxes, since they are under represented within the dataset. For each training image, we determine the maximum number of overlapping objects by finding the highest-value pixel in the ground truth overlap map. After counting the number of training images residing at different highest overlapped pixel values, we found that both datasets have fewer images with extreme overlap. Therefore, this distribution skewed towards a small number in terms of highest overlapped pixel value. We calculate the probability for different ranges of highest overlapped pixel values (shown in Eq. 9). We use oversampling

$weight = \frac{1}{PB_{m-n}}$ when the highest overlapped pixel value resides between m and n:

$$PB_{m-n} = \frac{\text{Number of images with highest overlap in range } m \text{ to } n}{\text{total images}} \tag{9}$$

We tried three different solution configurations for the overlap map prediction model as was described in Sect. 3.2. For the OLMM-b model we used the overlap map created by candidate bounding boxes with confidence score greater than 0.01. For OLMM-c the model takes the overlap map created from the selected boxes from set-NMS as additional input. The confidence threshold for set-NMS is tuned by finding the best *LRP* score on the training set. Experimental results in Sect. 5.2 show that the OLMM-c gives the lowest *MMSE* hence we use this configuration as our default method for predicting the overlap map. We round fractional pixel values to their closest integer value which converts unrepresentative fractional pixels to their actual pixel classes.

We perform two additional steps to reduce the number of bounding boxes inputted into the overlap aware box selection algorithms. (1) We merge extremely overlapped bounding boxes into a single bounding box. (2) We cluster the candidate bounding boxes into separate non-overlapping clusters and apply overlap aware bounding box selection on each cluster separately. Note that boxes belonging to the same cluster may overlap. Figure 7 shows these two steps.

The box merging step is a simple trick which we found to improve results in practice. We observed that, in many situations, extremely overlapped object detector predictions would be individually worse than the mean of those predictions. So, in order to both improve results and speed up box selection (by reducing the number of candidate boxes) we merge redundant boxes together. The boxes are merged by iteratively considering each box and merging all boxes with it that are within greater than an overlap threshold. This is recursively applied until none of the remaining merged bounding boxes overlap with any bounding box by the overlap threshold. In our experiments we set an overlap merging threshold of 0.85 intersection over union (IoU) for the sports datatset and 0.75 IoU for the crowdHuman dataset. Importantly, this threshold is much higher than standard NMS and is unlikely to merge the bounding boxes of distinct objects.

We can further speed up overlap aware bounding box selection by splitting the candidate bounding boxes into separate clusters and then processing each cluster separately. Since the clusters do not overlap each other at all this second step does not change the resultant set of selected bounding boxes compared to performing the bounding selection on all the boxes at once.

We thoroughly explored the heuristic scoring functions used in the overlap aware bounding box selection algorithm and their associated parameters in our sports dataset with ground truth overlap map as input (see Sect. 5.1 for the details of the experiment). It was evident that look-ahead pixel voting ($W > 1$) performs better than pixel voting without look-ahead ($W = 1$). We set $W = 2$ to keep computation overhead within an acceptable time limit. Some images of crowdHuman dataset have a very large number of candidate bounding boxes which makes look-ahead pixel voting impractical. For this reason we set

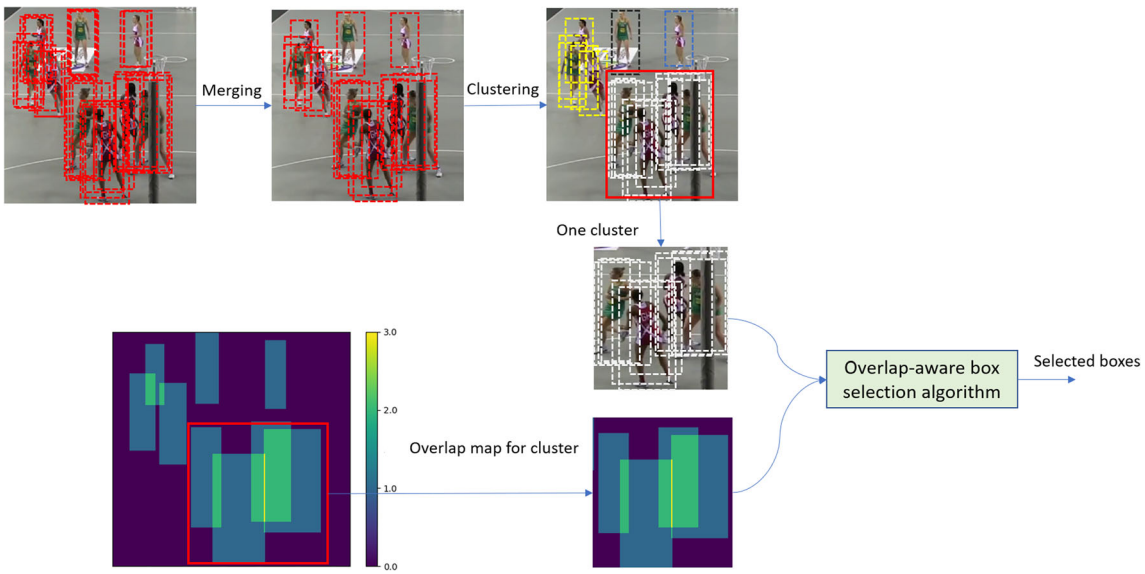

Fig. 7 Our proposed overlap-aware box selection steps with candidate bounding box merging and clustering

$W = 1$, $K = 6$ for the crowdHuman dataset and $K = W = 2$ for the sports dataset.

In our experiments, we chose the confidence score threshold that outputs optimal *LRP* for set-NMS algorithm (based on training data). Then boxes above that confidence score threshold are used as input of our box selection algorithm.

5 Experimental results

We conducted multiple experiments to assess the performance of our overlap-aware bounding box selection algorithm against the state-of-the-art overlap-agnostic bounding box selection algorithm, set-NMS [4]. Firstly, we conduct experiments comparing the use of different heuristic scoring functions for greedy overlap-aware box selection. We also find the best settings to use with our pixel voting box overlap aware selection algorithm. Secondly, we find the best overlap map generation model. Finally, we compare our algorithm using the best pixel voting and overlap map generation model against set-NMS using both the sports and crowdHuman data sets.

5.1 Performance comparison of different overlap aware box selection algorithms

In this section, we evaluate the performance of different overlap aware box map selection heuristics described in Sect. 3.4. In addition we report the results from different pixel voting setups. We conducted this experiment on the sports test data sets using the ground truth overlap map. Table 2 shows the results of the experiment. The results show that the normalized absolute error (*NAE*) heuristic score produces the best LRP performance compared to the other naive heuristic scoring functions (absolute error (*AE*) and confidence score). This is mainly due to the lower false negative rate from using the NAE heuristic scoring function. As we discussed in Sect. 3.4, *NAE* tends to select smaller bounding boxes which would in turn favour picking multiple bounding boxes to cover a region rather than a single large bounding box. Thus *NAE* is likely to pick more boxes and hence result in a lower false negative rate. However, all variants of our pixel voting heuristic outperform all other overlap aware box selection algorithms for all *LRP* metrics. The results show look-ahead pixel voting achieves the best result overall. However, due to the high computation cost of using $W = 2$, we use pixel voting heuristics with $K = 6$ and $W = 1$ for the crowdHuman dataset where a larger number of candidate bounding boxes are generated than the sports dataset.

5.2 Evaluation of overlap map models

In this section we compare different overlap map generation models in terms of the mean absolute error between the predicted and ground truth overlap map. We conduct this experiment on both the sports and crowdHuman datasets. We compare three deeplabev3+ based models which differ based on the input type, as described in Sect. 3.2. Table 3 shows the performance of different overlap models for both dataset. The results show OLMM-c outperforms both OLMM-a and OLMM-b models for both datasets. This indicates that the inclusion of input overlap maps generated from predicted bounding boxes integrates prior learned bounding boxes defined features which improves overlap output. Additionally, skip connection in OLMM-c keeps the set-NMS prior at the output which also improves the overlap map prediction. Therefore, we choose OLMM-c as our overlap map model used for the remainder of the experiments.

5.3 Overall performance evaluation of bounding box selection algorithms

In this section, we compare our overlap-aware pixel voting box selection algorithms using predicted overlap maps against two baseline algorithms in terms of *LRP* error metrics and precision-recall graphs. The two baseline box selection algorithms are the commonly used NMS [28] algorithm and state-of-the-art set-NMS [4] algorithm. We also demonstrate the highest benefit that can be gained from pixel voting by using an overlap map created from ground truth bounding boxes. Table 4 shows the results comparing NMS, set-NMS and our box selection

Table 2 Comparison of different overlap aware box selection algorithms and settings for the pixel voting algorithm on the sports dataset

Selection algorithm	LRP	$LRP_{LocComp}$	LRP_{FPComp}	LRP_{FNComp}
Confidence score	0.2926	0.1246	0.0179	0.04106
AE	0.2903	0.1220	0.0119	0.0505
NAE	0.2696	0.1162	0.0153	0.0342
Pixel voting, K=1, W=1	0.2520	0.1095	0.0097	0.0333
Pixel voting, K=6, W=1	0.2519	0.1098	0.0087	0.0333
Pixel voting, K=W=2	**0.2374**	**0.1035**	**0.0067**	**0.0320**

Table 3 *MMSE* error of different overlap map model described in Sect. 3.2 (Bold font used to highlight best performing overlap model configuration)

Overlap map model input	Dataset	Mean of MSE error (MMSE)
OLMM-a	CrowdHuman	0.30384
OLMM-b	Crowdhuman	0.30037
OLMM-c	Crowdhuman	**0.29330**
OLMM-a	Sports	0.01578
OLMM-b	Sports	0.01367
OLMM-c	Sports	**0.01206**

Table 4 Comparison of baseline algorithms (NMS and set-NMS) with optimal confidence score threshold and our pixel voting using the same optimum confidence score threshold (Bold font used to highlight the best performing bounding box selection algorithm that does not use ground truth overlap map)

Bounding box selection algorithm	Dataset	*LRP*	$LRP_{LocComp}$	LRP_{FPComp}	LRP_{FNComp}
NMS [28]	Sports	0.32028	0.11934	0.04172	0.07106
Set-NMS [4]	Sports	0.31187	0.11816	**0.04118**	0.06264
PV	Sports	0.30817	0.11543	0.04300	0.06262
Look ahead PV	Sports	**0.30615**	**0.11506**	0.04220	**0.06147**
PV (ground truth overlap map)	Sports	0.26432	0.10709	0.00454	0.05977
NMS [28]	CrowdHuman	0.57950	0.18869	**0.11592**	0.25898
Set-NMS [4]	CrowdHuman	0.56429	0.18622	0.12408	**0.22995**
PV	CrowdHuman	**0.55999**	**0.18192**	0.11918	0.23691
PV (ground truth overlap map)	CrowdHuman	0.51192	0.17952	0.07222	0.19051

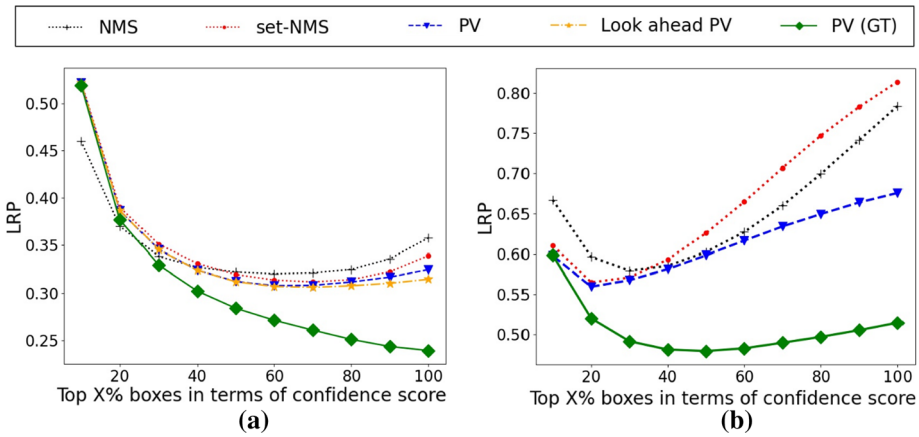

Fig. 8 LRP error for varying confidence thresholds using the sports and crowdHuman datasets. (Lower LRP is better). **a** Sports dataset results. **b** crowdHuman dataset[1]. [N.B. Confidence threshold values are selected based on the percentage of bounding boxes retained after the threshold is applied]

algorithms. The results show our pixel voting box selection algorithm outperforms baseline algorithms for both datasets in terms of unbiased LRP error. Pixel voting (PV) and look ahead pixel voting surpasses set-NMS in terms of LRP_{FNComp} error and underperforms set-NMS in terms of LRP_{FPComp} error in sports dataset. In contrast, our approach outperforms set-NMS in terms of LRP_{FPComp} and underperforms in terms of LRP_{FNComp} error in the crowdHuman dataset. For both datasets, Pixel voting outperforms NMS in terms of all LRP components except the LRP component LRP_{FPComp} of CrowdHuman dataset. Pixel voting using ground truth overlap map as input significantly outperforms all other bounding box selection algorithms for all *LRP* components for both datasets. This indicates there is still significant potential for our approach to yield better results if we can develop a more accurate overlap map prediction model.

Figure 8 shows *LRP* error results for varying the confidence threshold used to prune the candidate bounding boxes inputted into the box selection algorithms. The threshold was varied such that the top x percent of candidate bounding boxes (by confidence score) were used as input to the box selection algorithms. In this experiment we calculate the combined *LRP* error, precision and recall for varying values of confidence threshold. This allows us to

[1] We were unable to include look-ahead PV results for the CrowdHuman dataset due to the high computational complexity of look-ahead PV and the large number of bounding boxes in CrowdHuman.

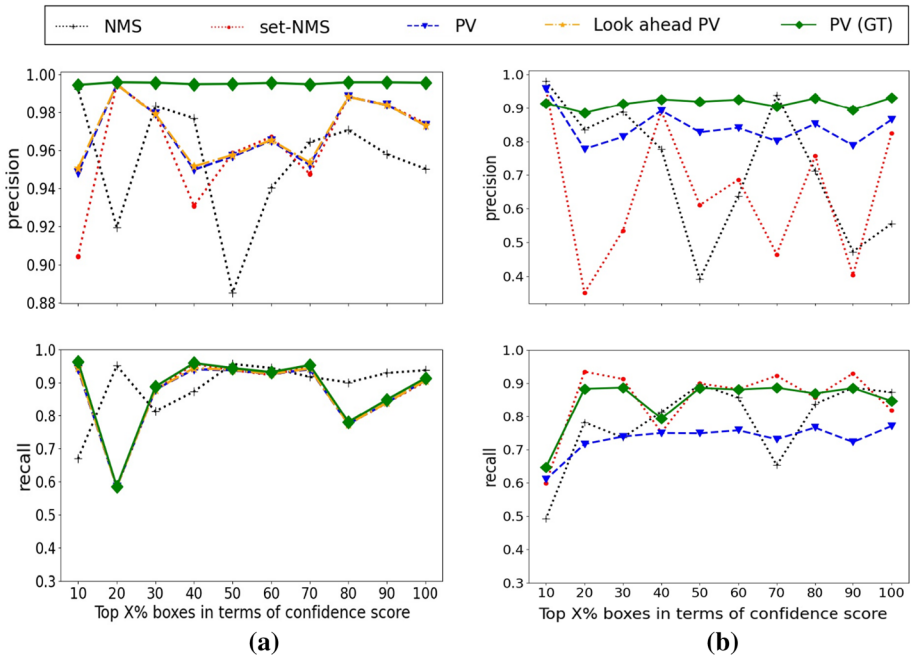

Fig. 9 Precision and recall graphs for varying confidence thresholds. **a** Sports dataset results. **b** crowdHuman dataset results[1]. [N.B. Confidence threshold values are selected based on the percentage of bounding boxes retained after the threshold is applied]

observe how well each algorithm is able to trade off precision against recall. The results show pixel voting outperforms both NMS and set-NMS in terms of *LRP* for almost the entire range of confidence score thresholds. The sports dataset results show that looking ahead pixel voting (look ahead PV) marginally improves pixel voting (PV). Pixel voting with ground truth overlap map (PV (GT)) outperforms the other algorithms by quite a large margin for the entire range of confidence score thresholds.[1]

Figure 9 shows precision-recall graphs for different amounts of input bounding boxes with sports and crowdHuman dataset respectively. The results show that our pixel voting bounding box selection algorithms outperform both NMS and set-NMS for precision. This is particularly apparent for the crowdHuman dataset where the precision of both NMS and set-NMS drops significantly at different points along the graph. In contrast, pixel voting's precision stays consistently high throughout the graph. Our overlap map model uses set-NMS output to refine set-NMS which make a generalized per pixel decision considering all data sets candidate box density. Therefore, pixel voting is consistent and performs better in terms of precision. Pixel voting performs only slightly worse than set-NMS for recall performance on the sports dataset but loses out to set-NMS much more for the crowdHuman dataset. We hypothesize that this is a result of the overlap map model failing to learn about situations where many objects overlap, as these are uncommon in the dataset. Therefore, it may be beneficial to train the overlap model with more examples depicting extreme overlaps to improve the overlap map output and ultimately improve recall. Overall, compared to set-NMS, pixel voting makes a better trade-off between precision versus recall since it outperforms set-NMS by a much larger margin in terms of precision compared to the amount it loses out in recall.

Pixel voting with ground truth overlap maps significantly outperforms the other algorithms in precision and slightly outperforms both NMS and set-NMS for recall. This result suggests that a better overlap map model which more accurately mimics the ground truth overlap maps could drastically improve results.

6 Conclusions

This paper presented a new approach for selecting candidate bounding boxes based on a predicted overlap map whose pixel values represent the number of objects overlapping the pixel. This contrasts with existing solutions which do not use this critically important information. We developed a novel greedy bounding box selection algorithm called pixel voting which selects bounding boxes based on the overlap map. Experiments showed that pixel voting outperformed the conventional NMS [28] and state-of-the-art set-NMS [4] bounding box selection algorithm on the commonly used crowdHuman dataset and a sports dataset. For future work, we want to explore over overlap map prediction models, since the results show a more accurate overlap map prediction model has the potential to significantly improve the overall detection accuracy.

Acknowledgements We are grateful to Dr. Stuart Morgan and the Australia Institute of Sport (AIS) for providing PhD scholarship

support. We also want to thank the AIS for providing the annotated sports dataset used in this project.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 801–818
2. Cheng B, Collins MD, Zhu Y, Liu T, Huang TS, Adam H, Chen LC (2020) Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12475–12485
3. Cheng, G., Han, J.: A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing 117, 11–28 (2016)
4. Chu X, Zheng A, Zhang X, Sun J (2020) Detection in crowded scenes: one proposal, multiple predictions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12214–12223
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), IEEE, vol. 1, pp 886–893
6. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, IEEE, pp 248–255
7. Fan W, Chen Y, Li J, Sun Y, Feng J, Hassanin H, Sareh P (2021) Machine learning applied to the design and inspection of reinforced concrete bridges: resilient methods and emerging applications. In: Structures, Elsevier, vol. 33, pp 3954–3963
8. Girshick RB, Felzenszwalb PF, McAllester D (2012) Discriminatively trained deformable part models, release 5
9. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
11. Hosang J, Benenson R, Schiele B (2017) Learning non-maximum suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4507–4515
12. Hu H, Gu J, Zhang Z, Dai J, Wei Y (2018) Relation networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3588–3597
13. Ke L, Tai YW, Tang CK (2021) Deep occlusion-aware instance segmentation with overlapping bilayers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4019–4028
14. Law H, Deng J (2018) Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp 734–750
15. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
16. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
17. Liu L, Jiang H, He P, Chen W, Liu X, Gao J, Han J (2019) On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265
18. Liu S, Huang D, Wang Y (2019) Adaptive nms: Refining pedestrian detection in a crowd. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6459–6468
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision, Springer, pp 21–37
20. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030
21. Mohan, R., Valada, A.: Efficientps: Efficient panoptic segmentation. International Journal of Computer Vision 129(5), 1551–1579 (2021)
22. Oksuz K, Cam BC, Akbas E, Kalkan S (2018) Localization recall precision (lrp): A new performance metric for object detection. In: Proceedings of the European conference on computer vision (ECCV), pp 504–519
23. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (Eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
24. Pérez-Hernández F, Tabik S, Lamas A, Olmos R, Fujita H, Herrera F (2020) Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. Knowledge-Based Systems 194:105590
25. Qiao S, Chen LC, Yuille A (2021) Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10213–10224
26. Ranftl R, Bochkovskiy A, Koltun V (2021) Vision transformers for dense prediction. arXiv preprint arXiv:2103.13413
27. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28, 91–99 (2015)
29. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 234–241
30. Shih, H.C.: A survey of content-aware video analysis for sports. IEEE Transactions on Circuits and Systems for Video Technology 28(5), 1212–1231 (2017)
31. Sun K, Zhao Y, Jiang B, Cheng T, Xiao B, Liu D, Mu Y, Wang X, Liu W, Wang J (2019) High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514
32. Viola P, Jones M (2001) 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, IEEE, vol. 1, pp I–I
33. Wang CY, Yeh IH, Liao HYM (2021) You only learn one representation: unified network for multiple tasks. arXiv preprint arXiv:2105.04206
34. Xu Z, Li B, Yuan Y, Dang A (2020) Beta r-cnn: Looking into pedestrian detection from another perspective. Adv Neural Inform Process Syst 33:19953-63
35. Zheng A, Zhang Y, Zhang X, Qi X, Sun J (2022) Progressive end-to-end object detection in crowded scenes. arXiv preprint arXiv:2203.07669