

## OBJECT DETECTION

### Technical Field

- 5 The present invention relates to detecting objects in images. More particularly, the present invention relates to the assignment of bounding boxes around objects based on surrounding context and semantic information.

### Background

10

This background is provided for generally presenting the context of the disclosure. Contents of this background section are neither expressly nor implied admitted as prior art against the present disclosure.

- 15 Object detection is a fundamental subfield of computer vision that focuses on identifying and localizing objects in images and videos. This technique plays a critical role in AI-powered solutions for a variety of applications such as visual inspection, warehouse automation, inventory management, and security systems. For example, object detection can be used in defect detection to identify defects in manufactured products, in stock keeping unit detection for warehouse inventory management, and in people detection and face recognition for video surveillance systems. Object detection tasks are commonly performed using deep learning models. In this context, data plays a critical role in training the models to accurately detect objects.

25

- Access to information is often limited to select executives in many institutes and firms, making it difficult to collect sufficient data in the real world. For example, in healthcare, patient information is kept confidential for privacy reasons and may not be shared with researchers, which limits the amount of data available for making predictions.

30

This lack of data can result in inaccurate predictions due to the small sample size of available data. It is desirable, therefore, to provide a mechanism for object detection for improving object detection, in view of limited training data, when

compared with the prior art, or at least to provide a useful alternative to known methods.

### Summary

5 The system, and methods implemented thereby, disclosed herein provides a deep learning framework designed to boost the performance of object detection. The technology is versatile and can improve detection accuracy for a wide range of object detection tasks. It is particularly well-suited for scenarios where the  
10 available training samples are limited (e.g., relevant data may not be available or the collection process is difficult and time-consuming), as it can effectively leverage the available data to enhance the detection performance. Some embodiments incorporate both bounding box and semantic mask segmentation of objects as ground truth in the loss function, utilizing a bi-loss learning approach.  
15 By incorporating both types of ground truth information, the deep learning model is trained to focus not only on the surrounding context of the object but also on its specific characteristics. This method results in a more comprehensive understanding of the object when compared with prior art methods, leading to improved detection performance.

20 Disclosed is an object detection network training method comprising:  
receiving a plurality of input images, each showing an object and comprising a ground truth bounding box, and ground truth mask, for the object;  
for each image in the plurality of images:  
25 a) extracting features from said image using a neural network;  
b) applying one or more predicted bounding boxes to the image, based on the image features, using a bounding box generation network;  
c) calculating a combined loss comprising both:  
30 a bounding box regression loss between the one or more predicted bounding boxes and the ground truth bounding box for the object in the image;  
an instance coverage loss of the one or more predicted bounding boxes relative to ground truth mask; and

d) updating the bounding box generation network using the combined loss; and

e) repeating steps a) to d) until convergence, thereby producing a trained bounding box generation network; and

5 the object detection network comprising the trained bounding box regression network and being configured to determine a final bounding box based on the predicted boxes.

Also described herein is a method for assigning a bounding box to a further object  
10 in a further image, comprising:

receiving the further image; and

applying an object detection network trained in accordance with the method described above, to the further image, thereby producing the bounding box for the further object.

15

Also described herein is a system for training an object detection network, the system comprising:

memory;

at least one processor;

20 an object detection module comprising:

an image feature extractor network; and

the object detection network, the object detection network comprising:

a bounding box generation network; and

25 final bounding box predictor model,

the memory storing instructions that, when executed by the at least one processor, cause the system to:

obtain a plurality of input images, each showing an object and comprising a ground truth bounding box, and ground truth mask, for the object;

30 for each image in the plurality of images:

f) extract features from said image using the image feature extractor network;

g) apply one or more predicted bounding boxes to the image, based on the image features, using the bounding box generation  
35 network;

- h) calculate a combined loss comprising both:
- a bounding box regression loss between the one or more predicted bounding boxes and the ground truth bounding box for the object in the image;
  - an instance coverage loss of the one or more predicted bounding boxes relative to ground truth mask; and
- i) update the bounding box generation network using the combined loss; and
- j) repeat steps a) to d) until convergence, thereby training the bounding box generation network; and
- employ the final bounding box predictor model to determine a final bounding box based on the predicted boxes.

#### Brief description of the drawings

Some embodiments of the present invention will now be described by way of non-limiting example only, with reference to the accompanying drawings in which:

Figure 1 illustrates the comparative operation of the present invention when compared with prior art methods;

Figure 2, comprising images 2a, 2b and 2c, show an apple, golf club and a coastline with jetties, respectively, identified by a respective ground truth and predicted bounding boxes;

Figure 3 illustrates a method for training an object detection model and using the trained model in applying a bounding box around an object in a new image;

Figure 4 is a schematic of the YOLO object detection framework using the present bi-task learning paradigm to improve the accuracy and efficiency of object detection;

Figure 5, comprising images 5a to 5c, illustrates: in image 5a the difference between conventional IoU and BMIoU; in image 5b, the effectiveness of applying

BM loss on final prediction; and in image 5c, the effectiveness of using UBD loss on object prediction;

Figure 6, comprising images 6a and 6b, illustrates the framework for: in image 5 6a, a conventional bi-task learning approach; and in image 6b, the presently proposed approach;

Figures 7 to 10 show visualizations of sample evaluation results using three 10 training methods on Raccoon dataset; and

Figure 11 is a schematic of a system for performing the methods of Figure 3.

**Detailed description**

15 Disclosed are approaches to detecting objects in images, where only limited training data is available. The approaches use both the instance mask and bounding box of the object in an image as ground truth in the loss computation during model training. The motivation behind this concept is that, in the training process of conventional object detection, the error between two proposal boxes 20 and the ground truth box for a given object may be similar, but the semantic information contained in the two proposal boxes can vary significantly.

Figure 1 shows the past and current methods make use of the same dataset 100. The number of training samples in the dataset 100 may be low. Despite this, 25 present methods can effectively leverage the available data to enhance the detection performance. A neural network 102 detects image features in a known manner – e.g. first level features such as lines, curves and shading, second level features such as relationships (e.g. distance or relative position in an image) between first level features, and others. The neural network 102 may be the same 30 as that used in image feature extraction more generally, and will be understood by the skilled person. The method at the top of Figure 1 then involves applying bounding boxes to an image (104), based on context surrounding an object that is then detected 106. The method at the bottom involves applying both bounding boxes and semantic annotations to an image (108), to focus not only on the

surrounding context of the object but also on its specific characteristics, to enable object detection (110).

As shown in Figure 2, considering two proposed bounding boxes (herein interchangeably referred to as "predicted bounding boxes") of the objects in images 2a to 2c, solid outlined boxes are ground truth (GT) boxes of objects, dotted and dashed boxes are predicted boxes of objects. In image 2a, the apple is effectively an object with uniformly distributed semantic information. This shows predicted boxes with similar Intersection of Union (IOU) values tend to carry similar semantic information. Image 2b shows a golf club, being an object with non-uniformly distributed semantic information, demonstrating that the IOU value alone may not accurately represent the discriminative information of the predicted box. In image 2c, Aerial Maritime Drone photos illustrate how different prediction boxes can contain varying levels of semantic information for objects. The dashed lines and dotted lines boxes indicate the boundary of predicted objects.

The predicted (dotted and dashed) boxes have the similar overlap of around 0.5 with the solid outlined ground truth box. However, the semantic information included in each predicted box is different. The dotted box appears to contain more informative regions of the object than the dashed box. By training the deep learning model to focus not only on the contextual information surrounding the object but also on the unique characteristics of the object itself, the present object detection model is able to achieve a more comprehensive understanding of the object than prior art methods, leading to improved detection performance.

A method 300 for object detection is shown in Figure 3. The method 300 exhibits two phases – one phase is for training the bounding box generation model and the other phase for applying the trained bounding box generation model to a new image. During training, a plurality of images are received (step 302). The images may be received from a database of related images, in real-time from image capture using a camera or other device, or through any other means. To that end, "obtaining" or "receiving" images can include any mechanism for doing so. Moreover, to increase training data for particularly small training datasets, several approaches have been developed. These approaches intend to handle data scarcity in machine learning and include data augmentation techniques that

artificially expand the size of the dataset by creating new examples from existing ones (e.g., rotation, flipping, and scaling), transfer learning that leverages knowledge from pre-trained models on large datasets to learn new tasks, and few-shot learning that aims to train a model on a new task with limited labelled data  
5 by leveraging knowledge from related tasks that have more labelled data available. Other techniques for handling data scarcity include active learning that allows a model to select the most informative examples to annotate, generative models that can generate new data samples and data synthesis techniques that can create new samples based on existing ones. All of these approaches aim to  
10 help improve model performance when there are limited labelled data available. The present methodology can be used in addition to these techniques.

During the training phase, each image received at 302 shows an object and comprises a ground truth bounding box, and ground truth mask, for the object.  
15 The ground truth bounding box and ground truth mask may be manually applied or applied using an alternative machine learning algorithm.

Based on the input image received at 302, the feature extraction process is performed to extract relevant features from the image using a feature extractor  
20 neural network (304). The feature extractor neural network may be a convolution neural network, an edge feature detector, an autoencoder or other network.

After extraction, the extracted features are passed through a fully connected (FC) layer and bounding boxes regression is applied to generate proposal boxes with  
25 confidence scores (306).

The confidence score provides a level of certainty of object detection (i.e., how sure the model is that a given box contains an object), as well as the level of certainty of the bounding box prediction. The confidence score may be calculated  
30 as a product of the probability of a bounding box overlap with a detected object ( $\text{Pr}(\text{object})$ ), and the IoU value.  $\text{Pr}(\text{object})$  has a value of 0, when the bounding box is included within the background, and a value of 1, when the bounding box is included within the same area as the detected object.

Step 308, comprising sub steps 308a and 308b, generates a combined loss or "bi-loss" based on the bounding boxes predicted during bounding box regression and the ground truth bounding box. The bi-loss function enables more comprehensive semantic information about objects to be captured in the predictions made by the object detection model during training. In particular, the combined loss function leverages both the ground truth bounding box and ground truth semantic segmentation mask of an object in the loss function. The combined loss minimizes the error on learning two tasks: encourage proposal boxes to (1) be close to ground truth boxes of objects and (2) contain more semantic information (pixels on object) of object. The training framework for YOLO detector is illustrated in Figure 4 as an example. Given an input image 400, after feature extracting network 402 is applied, FC layer 404 and bounding box regression 406 on grid cells, proposal boxes with confidence score can be generated - the term "a confidence score" refers to the confidence that a particular bounding box aligns with the ground truth bounding box for the respective object. The segmentation masks over each object, and the bounding boxes defined around those objects, are output from the model (408).

Step 308a encourages predicted bounding boxes to be close to the ground truth boxes of the objects in the image. This is achieved by minimizing the distance between the proposal box and the ground truth box using a suitable distance metric such as mean squared error (MSE) or intersection over union (IoU). In some embodiments, bounding box regression loss is calculated between the one or more predicted bounding boxes and the ground truth bounding box for the object in the image - e.g. as a function of the distance and difference in area between each predicted bounding box and the ground truth bounding box for an image. The distance may be calculated based on the x and y coordinates of a corner of the box in the image. The area may be a length in pixels, or may be defined by the coordinates of the diagonally opposite corner of the image.

Bounding box regression (BBR) enables the optimization of object size and localization performance - this is critical to fast convergence (i.e., the predicted bounding box matching the ground truth bounding box, or approximating the ground truth bounding box to within a predetermined threshold - e.g., within a predetermined number of pixels from the ground truth bounding box, or having

an area within a predetermined percentage of that of the bounding box). The BBR loss function allows the model to iteratively adjust the predicted bounding boxes towards the ground truth boxes, which ultimately leads to improved object localization performance. To predict the bounding box, a loss function is used to

5 compute the error between the predicted and ground truth bounding boxes. Some loss functions, such as Smooth L1 (Huber) loss and Mean Square Error (MSE) loss, are effective in measuring the difference between predicted and ground truth bounding boxes.

- 10 As shown in Figure 4, applying BBR loss leads to predicted bounding boxes generated on objects, e.g., a first group of bounding boxes (solid outline) on "car", a second group of bounding boxes (dotted outline) on "person" and a third group of bounding boxes (dashed outline) on "horse". As object detector training progresses, the bounding boxes generated by the algorithm become increasingly accurate and
- 15 have a higher intersection over union (IoU) with the ground truth box of the objects.

In some embodiments, a smooth L1 loss is used as the bounding box regression loss, computed by:

20

$$L_{BBR} = \frac{1}{N_{box}} \sum_{i=1}^{N_{box}} \sum_{j \in \{x,y,w,h\}} smooth_{L1}(t_j^i - v_j) \quad (1)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (2)$$

- where (x, y) is the top-left of a box, w and h are width and height of the box. N<sub>box</sub> is the number of predicted boxes, t is a predicted box and v is the ground truth box. When objects have uniformly distributed semantic information (Figure 2, image 2a), a higher IoU value between the predicted bounding box and the ground truth box indicates that the predicted box contains more semantic information about the object. Moreover, predicted boxes with similar IoU values tend to carry
- 25 similar semantic information. With reference to image 2a, IoU values between the ground truth box (blue) and the two predicted boxes (dashed and dotted) are similar. This similarity in IoU values indicates that the dashed and dotted boxes contain similar semantic information of the apple, which contributes similarly to
- 30

the object recognition task. This is because apple is an object of semantic information uniformly distributed.

For objects with non-uniformly distributed semantic information, such as the golf club in image 2b, the IoU value alone may not accurately represent the discriminative information of the predicted box. In such cases, even if the IoU value between the dashed box and the ground truth box is lower than the value between the dotted box and the ground truth box, the dashed box may carry more discriminative information about the object.

Under step 308b, which may occur prior to, concurrently with, or after step 308a, instance coverage loss (IDL) encourages the predicted boxes to contain more semantic information (i.e., more pixels on the object) of the objects in the image. This is achieved by minimizing the error between the predicted box and the ground truth mask of the objects in the image using a suitable loss function such as binary cross-entropy. Both of these tasks are combined into a bi-task learning framework to train the object detection model.

IDL differs from instance segmentation loss (e.g., such as the binary cross-entropy loss or the Dice loss). Segmentation loss is the error between the predicted binary mask and the ground truth binary mask. It represents the extent to which semantic information of an object is embodied, with the key distinction being that the application of instance coverage loss and instance segmentation loss resulting in the prediction of a box for the object and a mask for the object. The instance coverage extent can be reflected in two aspects. Firstly, each box should contain as high a percentage as possible of pixels of the object. Thus, the instance coverage loss is determined, at least in part, by calculating a box mask intersection over union (BMIoU) between each predicted bounding box and the ground truth mask. Secondly, all boxes should be uniformly distributed on the object to ensure that all instances are covered in at least one box. The instance coverage loss is thus, at least in part, based on, for each object, a distribution of all said predicted bounding boxes around the object.

Regarding the bounding boxes containing as high a percentage as possible of instances of the object, the accuracy of an instance representation within a box

can be determined by calculating the BMIoU between the predicted bounding box and the ground truth mask. This measure is used to compute the loss, called BM loss,  $L_{BM}$ , which can be expressed as follows:

$$L_{BM} = 1 - \frac{\sum_{i=1}^{N_{box}} BMIoU(B_i, M_{GT})}{N_{box}} \quad (3)$$

$$BMIoU(B, M) = \frac{A_B \cap A_M}{A_B} \quad (4)$$

where  $N_{box}$  is the number of predicted boxes on an object,  $B_i$  depicts the  $i$ th predicted box and  $M_{GT}$  is the ground truth mask of the object,  $A_B$  is the area of box  $B$  and  $A_M$  is area of mask  $M$ . Equation (4) demonstrates that computation of BMIoU differs from the conventional IoU calculation, which involves finding the intersection of two areas and dividing it by their union. The reason for using BMIoU to measure the extent of embodied semantic information of a predicted box is that most predicted boxes only contain parts of an object. The final box of an object is obtained using the maximum and minimum coordinates over all boxes to define the corners of the final bounding box, or by applying non-maximum suppression (NMS) on all predicted boxes.

This difference is reflected in Figure 5. Image 5a illustrates the difference between conventional IoU and BMIoU in determining whether a predicted box contains sufficient semantic information to be useful in defining a bounding box over the object. The majority of boxes may be excluded (assuming the IoU threshold is 0.5 by default). When the confidence score is set to a fixed value (e.g., 0.2), it becomes apparent that predicted box 4 is highly prone to being mis-predicted in subsequent training if conventional IoU is used as the measurement metric. In contrast, if we use BMIoU as the measurement metric, all four boxes will contribute to the final boundary box prediction of the object. This approach provides a more accurate representation of the object than using only three boxes.

The box mask (BM) provides higher sensitivity to changes in predicted bounding boxes. This enables more fine-grained detection since the limit at which variations have negligible effect on regression values used to update the bounding box generation (regression) model is smaller. This is demonstrated in image 5b. The left hand image shows the predicted boxes with IoU performed in a traditional

manner. The boxes predicted under BMIoU (right hand side of image 5b) shows that fewer "non-object" instances are contained within the predicted boxes. As such, the final predicted bounding box BOX2 of the right hand image much more closely approximates the boundary of the object 200 than the final predicted bounding box BOX2 of the left hand image.

As mentioned with reference to step 308b, the instance coverage loss is also based on uniformity of distribution of all predicted bounding boxes around an object in an image. The uniformity of box distribution (UBD) can be determined by calculating the number of pixels that are included in the boxes for each instance and then computing the distribution loss,  $L_{UBD}$  is as:

$$L_{UBD} = 1 - \frac{\sum_{i=1}^{N_{box}} \sum_{j \in (A_B \cap A_M) \text{ and } j \in B_{ins}} B_{ins} + 1}{\sum_{i \in A_M} M_{ins}} \quad (5)$$

where  $M_{ins}$  is the number of pixels of object, and  $B_{ins}$  is number of object pixels in all predicted boxes.

Image 5c illustrates the relevance of the distribution loss on overall prediction accuracy. In the distribution of boxes in the left hand picture the right bottom portion 202 of the object is not covered by any box and in fact falls outside the final bounding box BOX2. In contrast, the distribution on the right is more uniform around the object. This results in more accurate final prediction boundary BOX2, containing the object 200 in full.

The IDL is then the sum of the  $L_{BM}$  and  $L_{UBD}$ :

$$L_{IDL} = L_{BM} + L_{UBD} \quad (6)$$

Incorporating IDL into training of the object detection model (particularly the bounding box generation model), the predicted object bounding boxes show greater uniformity in distribution around the object, resulting in a greater amount of semantic information being captured.

The combined loss, or bi-loss, for minimizing error in the object detection process is then determined by:

$$L_{total} = L_{BBR} + L_{IDL} \quad (7)$$

5

Thus, the output of step 308 is  $L_{total}$ .

During the detector training process, the model adjusts its weights and biases by minimizing the combined loss – i.e., the loss provided by the loss function as  
 10 feedback to the predictions in the immediately previous training iteration. As a result, the model can quickly identify the approximate location of an object (achieved by minimizing BBR), and then refine the boundary box to include more semantic information about the object (achieved by minimizing IDL).

15 Under step 310,  $L_{total}$  is then back-propagated to the object detection model to update the bounding box generation for a next iteration (per repeat step 312). Steps 302 to 310 (for single image at a time), or steps 304 to 310 (for multiple images received on performance of step 302) are repeated until convergence. Convergence may occur when the final bounding box (predicted at step 314) is  
 20 the same as the ground truth bounding box of an input image. Alternatively, convergence may occur when the bounding box is within a predetermined distance of the ground truth bounding box for the image, and/or has an area within a predetermined threshold of that of the ground truth bounding box. This may be determined using non-maximum suppression (NMS). Non-maximum suppression  
 25 is a technique used mainly in object detection that aims at selecting the best bounding box out of a set of overlapping boxes. In object detection task, NMS is normally used to obtain final object bounding box prediction. NMS may involve selecting a single bounding box – e.g., a bounding box that overlaps or contains all other bounding boxes, or a bounding box that contains the greatest proportion  
 30 of pixels of all other bounding boxes.

When a further image is received (step 302) – i.e., an image that was not in the training set of images – the trained bounding box generation model of the object detection model can then be used for a new image (step 316) and the final

bounding box is determined based on the bounding boxes predicted by the trained bounding box generation model.

5 The combined loss therefor refines the allocation of bounding boxes such that they contain more semantic information, are more regularly distributed around the object in an image, and closely approximate the ground truth bounding box. The combined loss makes use of the BMIoU metric, that is designed to promote predicted boxes to contain more semantic information of object, and the UBD metric, that is designed to measure the degree of uniformity in the distribution of semantic information across all predicted boxes.

10 The bi-task learning process, being learning from the combined loss, focuses on two tasks that share the same output, which is the regression of proposal or predicted boxes. The goal of the first task is to generate candidate bounding boxes for objects in an image. The second task is to refine the proposal boxes to accurately localize the objects. These two tasks will generally be performed using a neural network architecture, which minimizes two losses, i.e., the bounding box regression loss, and the instance coverage loss. The final output of this process is a set of predicted object bounding boxes, that can be used for tasks such as object detection, tracking, and classification.

As set out above, the present method differs from conventional bi-task learning. With reference to Figure 6, conventional bi-task learning simultaneously solves two different learning tasks using a shared representation, with a two-dimensional output space, as shown in image 6a. Each task generates a separate loss based on a separate output. In contrast, the bi-task learning approach used herein refers to a learning paradigm enhances the performance of one output by simultaneously learning two tasks at an equivalent level, as depicted in image 6b.

30 Figure 11 shows a system 1100 for implementing the method 300. The system 1100 may be any computing system such as a laptop, handheld device such as a smartphone, a distributed system or any other appropriate system. The system 1100 comprises memory 1102 storing program code or instructions 1004 that is executable by the processor or processors 1106 to cause the system 1100 to perform the method 300. To that end, system 1100 receives the images for

training, or the image for object detection using the trained bounding box generation model, through I/O interface 1122 from a database 1116 over network 1118. The network may be wireless or hardwired, or a combination thereof.

- 5 During training, the images are fed to object detection module 1120 that runs an image feature extractor network 1108 over the images, to detect image features in the images. The image feature extractor network will also be applied during use of the trained model.
- 10 After image features are extracted, the object detection network 1110 of the object detection module 1120 runs bounding box regression network 1112 over the image features, and the final bounding box predictor 1114 applies a final bounding box of the image or images. During training, the combined loss is calculated relative to the ground truth bounding box and ground truth
- 15 segmentation mask, to update the bounding box generation network 1112. During use of the trained model, the combined loss is not calculated and, instead, the final bounding box is predicted and output for further processing.

- Performance of the present combined or bi-loss methodology was evaluated using
- 20 public datasets with varying sample sizes. The evaluation showed a performance improvement of 2% over existing methodologies. More notably, significant performance gains were attained on smaller datasets, with a 6% increase in detection accuracy on Raccoon and an 8% improvement on the ultra-small Aerial Maritime Drone dataset, shown in Figure 7.

- 25 In particular, effectiveness of the present bi-loss training approach was tested on four datasets: Pascal VOC, Raccoon, Blood Cell Count and Detection (BCCD), and Aerial Maritime Drone (AMD). The Pascal VOC dataset consists of 17,125 images (21 classes) with bounding box annotations, but only 2,913 of those images have
- 30 object mask annotations required for our bi-loss training. Thus, the evaluation was conducted on a subset of 2,913 images, split into 1,464 for training and 1,469 for testing. For the Raccoon dataset (single class), the 196 images that included bounding box information were used. Object mask annotations were manually produced to enable bi-task loss training. The training dataset comprised 150
- 35 images, with the remaining images being used for testing. The BCCD dataset

comprises 364 cell images (containing WBC-white blood cells, RBC-red blood cells, and Platelets) with bounding box annotations. The model was trained using bi-loss training, with manually annotated object masks for these images. The training dataset comprised 255 images and 109 images were used for testing. The AMD dataset includes 49 images (consisting of docks, boats, lifts, jetskis, and cars), each with bounding box annotations and manually generated mask annotations. 30 images were used for training and the remaining images for testing.

To evaluate performance, the Average Precision (AP) was calculated. A threshold value of 0.5 (AP50) and 0.75 (AP75) was used for evaluating the models. AP50 measures the number of objects that can be correctly detected, while AP75 has more stringent requirements on the bounding box position. The performance of one-stage and two-stage object detectors (such as YOLO and Faster R-CNN) on various datasets was evaluated using three training approaches: (1) using only BBR loss, (2) using only IDL loss, and (3) using both BBR and IDL losses. The evaluation results, reported in terms of AP50 (left column) and AP75 (right column), are presented in Table 1 for Pascal Voc, and Table 2 for Raccoon, BCCD, and Aerial Maritime Drone datasets, respectively.

|             | BBR    |        | IDL    |        | BBR + IDL |        |
|-------------|--------|--------|--------|--------|-----------|--------|
| YOLO        | 0.6097 | 0.4885 | 0.3351 | 0.2019 | 0.6283    | 0.5309 |
| Faster RCNN | 0.6246 | 0.4905 | 0.3475 | 0.2226 | 0.6416    | 0.5542 |

Table 1 – evaluation results of comparison on Pascal Voc dataset. For each training approach, results in left and right columns are for AP50 and AP75.

The results in Table 1 show that using both BBR and IDL losses together leads to better performance compared to using either loss alone on Pascal Voc dataset. These results verify the effectiveness of the proposed bi-loss learning approach. The reason why using only BBR loss resulted in a lower AP than using both BBR and IDL losses is because applying the IDL loss can help to increase the number of instances predicted within the boxes and also promote a more uniform distribution of the boxes on objects. When using only the IDL loss, a significant degradation in average precision was observed, which can be attributed to the fact that instance-related losses focus primarily on local information while ignoring

global information. As a result, the training process for the detector may take a longer time to converge or even lead to divergence. The results show that utilizing a global loss to quickly identify the coarse position of an object, followed by the application of a local loss to refine its position, can significantly improve the training of the detector.

The performance gap between using the combined BBR and IDL loss, and using only BBR or IDL, increases when the evaluation metrics shift from AP50 to AP75. The reason for this is that by minimizing the IDL loss, the detector is not only able to detect more objects, but also able to make more precise predictions by fitting the object tightly. The visualization samples in Figure 8 demonstrate that, in the first and third rows, the proposed training approach is able to detect more objects (such as the person on the right in the first image, and the person and horse in the foreground of the third image) compared to using either BBR or IDL alone. In the second and last rows, while using BBRL alone is able to detect both the person and train, the proposed training approach is able to detect tighter boxes around these objects, resulting in more accurate detection.

|         |             | BBR    |        | IDL    |        | BBR + IDL |        |
|---------|-------------|--------|--------|--------|--------|-----------|--------|
| Raccoon | YOLO        | 0.9094 | 0.7922 | 0.4535 | 0.3060 | 0.9461    | 0.8509 |
|         | Faster RCNN | 0.9204 | 0.8176 | 0.4588 | 0.3127 | 0.9498    | 0.8542 |
| BCCD    | YOLO        | 0.6240 | 0.4870 | 0.3022 | 0.1833 | 0.6792    | 0.5654 |
|         | Faster RCNN | 0.6311 | 0.5056 | 0.3154 | 0.1921 | 0.6948    | 0.5993 |
| AMD     | YOLO        | 0.7614 | 0.5945 | 0.6863 | 0.5457 | 0.8446    | 0.7255 |
|         | Faster RCNN | 0.7788 | 0.5992 | 0.6889 | 0.5626 | 0.8494    | 0.7374 |

Table 2. Evaluation results comparison on Raccoon, BCCD and AMD. For each training approach, results in left and right columns are for AP50 and AP75.

Figure 7 illustrates a visualization of sample evaluation results using three training methods on Raccoon. The results in Table 2 above are consistent with those in Table 1, indicating that using both BBR and IEL losses for training the detector leads to better performance on all three datasets. Additionally, visualization samples of detection on Raccoon, BCCD, and AMD are shown in Figures 7, 9, and 10, respectively. The visualizations demonstrate that the new detector training

paradigm not only reduces missed detections but also achieves more precise predictions around objects.

Applying the bi-loss requires the availability of object ground truth mask annotations. While datasets such as MS COCO and Pascal VOC typically provide these annotations, some datasets may not. In such cases, additional human labelling work may be required to generate the necessary annotations.

There are several use cases for the proposed bi-loss training method including, but not limited to:

1. Medical Imaging: In the field of medical imaging, there is often a limited amount of data available due to privacy concerns and the difficulty of obtaining high-quality medical images.
2. Satellite Imagery: Object detection in satellite imagery can be challenging due to the large scale of the images and the need for high-resolution imagery to detect small objects. This can lead to a lack of sufficient labelled data for some object detection tasks in satellite imagery.
3. Defect Detection: In manufacturing, defects can be relatively rare compared to normal or non-defective examples. As a result, it can be challenging to obtain a sufficient number of labelled samples of defective products for use in deep learning. Additionally, labelling data for defects can be a time-consuming and costly process, which can further contribute to the scarcity of defect samples.

The methods presented herein afford object detection methods and systems that incorporate both bounding box and semantic mask segmentation of objects as ground truth in a training dataset for a loss function that incorporates a Bounding Box Regression Loss for optimization of object size and localization performance, and Instance Coverage Loss that takes into consideration the following factors:

- all proposal bounding boxes should be uniformly distributed on an object of interest, and
- each proposal bounding box has a high percentage of coverage of the object of interest.

It will be appreciated that many further modifications and permutations of various aspects of the described embodiments are possible. Accordingly, the described

aspects are intended to embrace all such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

Throughout this specification and the claims which follow, unless the context  
5 requires otherwise, the word "comprise", and variations such as "comprises" and  
"comprising", will be understood to imply the inclusion of a stated integer or step  
or group of integers or steps but not the exclusion of any other integer or step or  
group of integers or steps.

10 The reference in this specification to any prior publication (or information derived  
from it), or to any matter which is known, is not, and should not be taken as an  
acknowledgment or admission or any form of suggestion that that prior publication  
(or information derived from it) or known matter forms part of the common  
general knowledge in the field of endeavour to which this specification relates.

15

**Claims**

1. An object detection network training method comprising:  
receiving a plurality of input images, each showing an object and comprising  
5 a ground truth bounding box, and ground truth mask, for the object;  
for each image in the plurality of images:  
k) extracting features from said image using a neural network;  
l) applying one or more predicted bounding boxes to the image,  
based on the image features, using a bounding box generation  
10 network;  
m) calculating a combined loss comprising both:  
a bounding box regression loss between the one or more  
predicted bounding boxes and the ground truth bounding box  
for the object in the image;  
15 an instance coverage loss of the one or more predicted  
bounding boxes relative to ground truth mask; and  
n) updating the bounding box generation network using the  
combined loss; and  
o) repeating steps a) to d) until convergence, thereby producing a  
20 trained bounding box generation network; and  
the object detection network comprising the trained bounding box  
regression network and being configured to determine a final bounding box  
based on the predicted boxes.
- 25 2. The method of claim 1, wherein the bounding box regression loss is based  
on, for each image:  
a distance between each predicted bounding box and the ground truth  
bounding box for the respective image; and  
an area of each predicted bounding box relative to an area of the ground  
30 truth bounding box.
3. The method of claim 1 or 2, wherein the instance coverage loss is based  
on, for each object, a distribution of all said predicted bounding boxes  
around the object.

35

4. The method of claim 3, wherein the instance coverage loss is determined, at least in part, by calculating a box mask intersection over union (BMIoU) between each predicted bounding box and the ground truth mask.
5. The method of claim 4, wherein the instance coverage loss is based on, for each object, a percentage of each respective predicted bounding box that comprises an instance of the object.
6. The method of claim 5, wherein the instance coverage loss is based on, for each object, uniformity of the distribution of all said predicted bounding boxes around the object.
7. The method of claim 6, wherein uniformity of the distribution of all said proposed bounding boxes (UBD) is determined by calculating a number of pixels that are included in the boxes for each instance and then computing the distribution loss as:

$$L_{UBD} = 1 - \frac{\sum_{i=1}^{N_{box}} \sum_{j \in (A_B \cap A_M) \text{ and } j! \in B_{ins}} B_{ins} + 1}{\sum_{i \in A_M} M_{ins}}$$

- where  $N_{box}$  is the number of proposed bounding boxes on an object,  $A_B$  is the area of box  $B$ ,  $A_M$  is area of mask  $M$ ,  $M_{ins}$  is the number of pixels of object, and  $B_{ins}$  is number of object pixels in all predicted boxes.
8. The method of any one of claims 1 to 7, wherein the object detection model is configured to determine the final bounding box by applying non-maximum suppression on all predicted boxes.
9. A method for assigning a bounding box to a further object in a further image, comprising:  
receiving the further image; and  
applying an object detection network trained in accordance with the method of any one of claims 1 to 8, to the further image, thereby producing the bounding box for the further object.

10. A system for training an object detection network, the system comprising:
- memory;
  - at least one processor;
  - an object detection module comprising:
    - 5 an image feature extractor network; and
    - the object detection network, the object detection network comprising:
      - a bounding box generation network; and
      - final bounding box predictor model,
  - 10 the memory storing instructions that, when executed by the at least one processor, cause the system to:
    - obtain a plurality of input images, each showing an object and comprising a ground truth bounding box, and ground truth mask, for the object;
    - for each image in the plurality of images:
      - 15 p) extract features from said image using the image feature extractor network;
      - q) apply one or more predicted bounding boxes to the image, based on the image features, using the bounding box generation network;
      - 20 r) calculate a combined loss comprising both:
        - a bounding box regression loss between the one or more predicted bounding boxes and the ground truth bounding box for the object in the image;
        - an instance coverage loss of the one or more predicted
        - 25 bounding boxes relative to ground truth mask; and
      - s) update the bounding box generation network using the combined loss; and
      - t) repeat steps a) to d) until convergence, thereby training the bounding box generation network; and
    - 30 employ the final bounding box predictor model to determine a final bounding box based on the predicted boxes.

11. The system of claim 10, wherein the bounding box regression loss is based on, for each image:

a distance between each predicted bounding box and the ground truth bounding box for the respective image; and  
 an area of each predicted bounding box relative to an area of the ground truth bounding box.

5

12. The system of claim 10 or 11, wherein the instance coverage loss is based on, for each object, a distribution of all said predicted bounding boxes around the object.

10

13. The system of claim 12, wherein the instance coverage loss is determined, at least in part, by calculating a box mask intersection over union (BMIOU) between each predicted bounding box and the ground truth mask.

15

14. The system of claim 13, wherein the instance coverage loss is based on, for each object, a percentage of each respective predicted bounding box that comprises an instance of the object.

20

15. The system of claim 14, wherein the instance coverage loss is based on, for each object, uniformity of the distribution of all said proposed bounding boxes around the object.

25

16. The system of claim 15, wherein uniformity of the distribution of all said proposed bounding boxes (UBD) is determined by calculating a number of instances that are included in the boxes for each instance and then computing the distribution loss as:

$$L_{UBD} = 1 - \frac{\sum_{i=1}^{N_{box}} \sum_{j \in (A_B \cap A_M) \text{ and } j \in B_{ins}} B_{ins} + 1}{\sum_{i \in A_M} M_{ins}}$$

30

where  $N_{box}$  is the number of proposed bounding boxes on an object,  $A_B$  is the area of box  $B$ ,  $A_M$  is area of mask  $M$ ,  $M_{ins}$  is the number of pixels of object, and  $B_{ins}$  is number of object pixels in all predicted boxes.

17.The system of any one of claims 10 to 16, wherein the object detection model is configured to determine the final bounding box by applying non-maximum suppression on all predicted boxes.

5 18.The system of any one of claims 10 to 17, being used for assigning a bounding box to a further object in a further image, the system comprising memory and at least one processor, the memory storing instructions that, when executed by the at least one processor, cause the system to  
10 determine that the bounding box generation module is trained, receive a new image and apply the object detection model to the new image to generate a final bounding box for the new image.

15