Accurate Bounding Box Prediction for Single-Shot Object Detection

Xuan-Thuy Vo, *Graduate Student Member, IEEE*, and Kang-Hyun Jo, *Senior Member, IEEE*

***Abstract*—Accurate single-shot object detection is an extremely challenging task in real environments because of complex scenes, occlusion, ambiguities, blur, and shadow, i.e., these factors are called uncertainty problem. It leads to unreliable labeling of bounding box annotation and makes detectors arduous to learn bounding box localization. Previous methods viewed the ground truth box coordinates as a rigid distribution omitting localization uncertainty in real datasets. This article proposes a novel bounding box encoding algorithm integrated into the single-shot detector (BBENet) to consider the flexible distribution of bounding box localization. First, discretized ground truth labels are generated by decomposing each object's boundary into multiple boundaries. The new representation of ground truth boxes is more arbitrary and flexible to cover any case of complex scenes. During training, the detector directly learns discretized box locations instead of continuous domain. Second, the bounding box encoding algorithm reorganizes bounding box predictions to be more accurate. Furthermore, another problem in existing methods is inconsistency in estimating detection quality. The single-shot detection consists of classification and localization tasks, but the popular detectors consider the classification score as the final detection quality. Thus, it lacks localization quality and hinders the overall performance because both tasks have a positive correlation. To overcome this problem, BBENet introduces detection quality by combining the localization and classification quality to rank detection during nonmaximum suppression. The localization quality is computed based on how uncertain the predicted boxes are, which is a new perspective in detection literature. The proposed BBENet is evaluated on three benchmark datasets, i.e., MS-COCO, Pascal VOC, and CrowdHuman. Without bells and whistles, BBENet outperforms the existing methods by a large margin with comparable speed, achieving the state-of-the-art single-shot detector.**

***Index Terms*—Convolutional neural networks (CNNs), detection quality, localization quality, localization uncertainty, object detection.**

Manuscript received June 6, 2021; revised October 12, 2021; accepted December 19, 2021. Date of publication December 24, 2021; date of current version June 13, 2022. This work was supported by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF), Ministry of Education (MOE), under Grant 2021RIS-003. Paper no. TII-21-2384. *(Corresponding author: Kang-Hyun Jo.)*

The authors are with the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea (e-mail: xuanthuyhammer@gmail.com; acejo@ulsan.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3138336.

Digital Object Identifier 10.1109/TII.2021.3138336

I. Introduction

OBJECT detection is a fundamental task in computer vision research. This task has been widely used in many industrial applications, such as nondestructive defect detection [1], face detection [2], lung cancer detection [3], detection in cellular networks [4], copy–move forgery detection [5], and surveillance systems [6], [7].

Although object detection has achieved outstanding performance, it is still difficult to accomplish perfect detection. Specifically, detection in real environments is highly challenging because of uncertainty problem originated by crowded scenes, occlusion, ambiguities, blur, and shadow. Because uncertainty problem often appears in real datasets, it leads to challenging problems as follows.

1) *Ambiguous boundaries of objects are depicted:* All the previous detectors identify objects' location through rectangle bounding box form. Each object coordinate is determined based on object boundaries (top, right, bottom, and left boundaries). When creating ground truth (GT) bounding boxes, ambiguous boundaries make it challenging to identify object coordinates. Therefore, the definition of GT boxes is sometimes not reliable. For example, Fig. 1 depicts boundary ambiguities in several images. In the first image, the leg part of the person is partially occluded by the motorcycle, which is not annotated. The pixels belonging to the background class in the green circle of the second image are labeled as a truck class due to ambiguities, which is imprecise. The baseball glove class in the third image is unreliably labeled because most pixels of this class belong to background and person classes. The right boundary of the giraffe in the fourth image is not clear due to blur and ambiguities. In the fifth image, the horse class is occluded by the person and affected by shadow. All the factors affect box labeling and generate unreliable GT boxes.
2) *Ambiguous learning in object detection is identified:* Because object boundaries are not clear enough due to uncertainty, existing detectors can not know the exact object locations. As a result, detectors generate mislocalized and misclassified detections. Even though detectors produce highly overconfidence scores, the bounding box prediction does not satisfy high localization quality. Therefore, it directly degrades the detection performance.

The abovementioned problems hinder the learning ability of the model and prevent perfect localization in solving object detection.

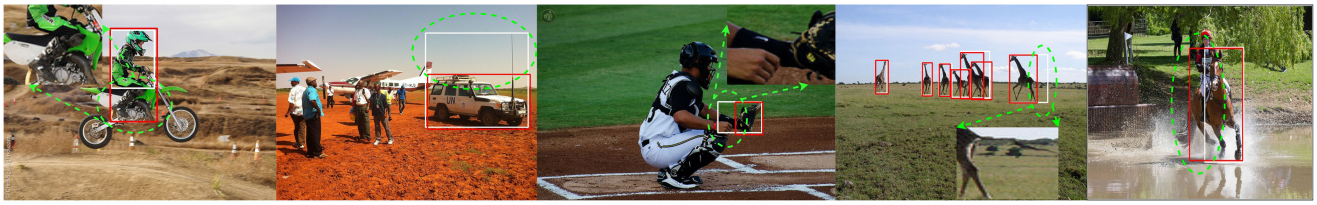

Fig. 1. Examples of the localization uncertainty in real datasets (e.g., MS-COCO dataset [8]) due to occlusion, ambiguity, blur, etc., are marked as green circles. The white boxes denote GT boxes. The red boxes indicate our predicted boxes learned under the discretized distribution of target locations.

However, conventional detectors [9]–[19] do not take localization uncertainty into consideration. The GT box representation of these methods can be regarded as Dirac delta distribution (e.g., does have any change in GT box process), which is too rigid and simple. Thus, these detectors fail to model localization uncertainty in the real world. Moreover, KL-Loss [20] and Gaussian YOLOv3 [21] were the first methods to solve localization uncertainty problem. Although both detectors model the bounding box predictions as Gaussian distributions, it is still too simple and symmetric to reflect the real distribution. In reality, the real distribution of the GT locations and predicted boxes is arbitrary and flexible.

To solve these limitations, this article presents the new representation of GT box coordinates with more reliability to reflect the real distribution and proposes a bounding box encoding algorithm to find optimal boxes from the prediction set. Both proposed components are used in the single-shot detector to reduce the uncertainty problem on detection performance, and this is called novel bounding box encoding algorithm integrated into the single-shot detector (BBENet). First, each GT box is discretized into multiple target boxes to recover the real locations of objects. The detector BBENet is learned under the discretized distribution of target boxes over the continuous domain. The new representation can produce more arbitrary bounding box predictions around truth target locations. Second, the bounding box encoding algorithm is introduced to reorganize predicted boxes to be well aligned to the boundaries of target locations. As shown in Fig. 1, the predicted boxes with red color are more credible and accurate.

Single-shot object detectors require both classification and localization tasks to predict object categories and locations. Intersection of union (IoU)-aware [22] states that both tasks have a strong positive correlation. However, popular detectors, such as RetinaNet [9], Faster R-CNN [23], Reppoints [15], Foveabox [13], FreeAnchor [12], gradient harmonizing mechanism (GHM) [10], and FSAF [11] defined the classification score as the final detection quality without considering the localization quality. Hence, there is an inconsistency between classification and localization tasks, and it potentially decreases the detection performance. To address this problem, the proposed BBENet represents the detection quality by merging both localization quality and classification score to rank detection during inference. The localization quality is estimated from the reorganized bounding boxes, which can reflect the level of localization uncertainty, i.e., uncertainty score. For clear cases (nonocclusion and nonambiguities), the uncertainty score is low since detection networks precisely locate the coordinate of each object and otherwise. During nonmaximum suppression (NMS), the boxes have high uncertainty, filtered out because of low detection quality.

The main contributions of the proposed method are summarized as follows.

1) The new representation of GT bounding boxes is performed to cover truth boundaries of the object, generating more reliable boxes around real locations. The BBENet is trained under discretized probability distribution that has arbitrary characteristics suitable for real datasets.
2) The bounding box encoding algorithm with a simple and effective strategy produces accurate bounding box predictions by reducing the influence of localization uncertainty on the detection performance.
3) We estimate the localization quality by considering the predicted bounding box uncertainty. This is called uncertainty score, which is a new and different perspective in object detection literature. During testing, the uncertainty score and classification score are combined as a final detection quality to improve the representation of localization and classification tasks.
4) Extensive experiments on three benchmark datasets are conducted. Without bells and whistles, the proposed method achieves a state-of-the-art single-shot detector.

II. Related Works

A. Single-Shot Object Detection

The representative methods of single-shot detectors are single-shot detector (SSD) [24], RetinaNet [9] and its improvements [10]–[14], [16]–[18], [22], and FCOS [19]. SSD was the first deep learning-based object detection without region proposal generation. This network places anchor boxes with different scales and aspect ratios on each location of multiple feature maps, and then predicts object categories and box offsets for each anchor. After that, many researchers have attracted much attention to single-shot detectors due to their simpleness and high efficiency. RetinaNet proposed a simple and unified detector including five improved components.

1) Integrating feature pyramid networks (FPN) [25] into detection network to solve scale imbalance.

2) Placing nine anchor boxes on each location of feature pyramid to cover all the objects in images.
3) A simple IoU-based anchor assignment.
4) Designing classification and box regression subnets.
5) Introducing Focal loss to handle the class imbalance in training detectors.

Based on the baseline RetinaNet, there are many studies enhancing detectors' learning ability in many aspects. GHM [10] addressed gradient imbalance between easy and hard samples when training RetinaNet detector and proposed gradient harmonizing mechanism to balance the gradient contribution of each sample. Feature selective anchor-free (FSAF) [11] improved RetinaNet in two components: 1) adding anchor-free branch with online feature selection to classification and regression subnets and 2) based on the outputs of the anchor-free branch, FSAF defines the center region of each bounding box as positive samples. Instead of IoU-based anchor assignment, FreeAnchor [12] proposed an object–anchor matching mechanism via maximum likelihood estimation to train detectors. IoU-aware [22] attached an additional IoU prediction to the regression branch, leveraging the positive correlation between classification and localization tasks. FoveaBox [13] defined the positive area of each GT box to select negative and positive samples, eliminating anchor demand. Adaptive training sample selection (ATSS) [14] introduced adaptive training sample selection to dynamically separate negative and positive samples based on statistic distribution of IoU variable. YOLOF [17] investigated the redundant characteristics of the FPN in RetinaNet. Through empirical experiments, YOLOF only used one-level feature for detection and proposed uniform matching to balance the number of positive samples for each object. Fully convolutional one-stage object detector (FCOS) [19] considers anchor boxes as anchor points. Then, FCOS predicts distance offsets between the anchor point and object boundaries.

Unlike the existing methods, our proposed BBENet improves the baseline RetinaNet in four parts.

1) *Efficient anchor box design:* We only place one anchor box per location while RetinaNet places nine anchors per location, avoiding hyperparameter selections of the anchor box and high model complexity.
2) *Regression offset variables:* The BBENet considers the center of the anchor box as an anchor point and regresses distance offsets from the center during training.
3) *Localization Uncertainty:* We take localization uncertainty into consideration, while the previous methods considered single-shot detectors in different perspectives, such as anchor assignment [11]–[14], [16], [17], network-level [11], [17], [25]–[27], loss optimization [10], [12], [28], learned anchor boxes [29], and estimated correlation [22]. To the best of our knowledge, there is no prior work in literature improving RetinaNet concerning localization uncertainty.
4) *Estimated localization quality:* RetinaNet and state-of-the-art detectors [10]–[13], [16], [18] consider classification score as final detection quality, while our method joins both localization quality and classification score as final detection quality.

B. Localization Uncertainty

Although most of the state-of-the-art single-shot detectors [9]–[14], [16]–[19], [22], [24], [26]–[29] have achieved impressive performance in both accuracy and speed, they still do not take localization uncertainty into account. These methods viewed bounding box localization as Dirac delta distribution, which is too rigid and simple to model the arbitrary distribution of the target box locations in real datasets. As a result, these detectors generated mislocalized detections (false detections), thus it hampers the detection performance. KL-Loss [20] was the first method to solve the uncertainty problem in object detection research. This method proposed the new KL-Loss by modeling bounding box predictions and box targets as Gaussian and Dirac delta distribution. To form the final localization loss, KL-divergence is used to estimate the difference between two distributions: $\hat{\theta} = \text{argmin}_\theta \frac{1}{N} \sum \text{KL}(P(x)||\text{GT}(x))$, where $P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\hat{x})^2}{2\sigma^2}}$ is Gaussian distribution with regression variable $\hat{x}$ and standard deviation σ to indicate bounding box prediction, and $\text{GT}(x) = \delta(x - x_{\text{gt}})$ is Dirac delta distribution to address GT box locations x_{gt}. The KL-divergence function is considered as the new localization loss: $L_{\text{KL}} = \text{KL}(P(x)||\text{GT}(x)) \propto \frac{(x_{\text{gt}}-\hat{x})^2}{2\sigma} + \frac{1}{2}\log\sigma$. In addition to conventional boxes, a branch is added to the detection network to predict the standard deviation σ by using some convolutional layers. If σ is larger (i.e., sample contains ambiguous boundary), the loss for this object is lower. Even though KL-Loss achieves significant improvement, they have some drawbacks.

1) The distribution of GT locations is modeled as Dirac delta distribution that can not be well matched with Gaussian distribution.
2) Both used distributions are symmetric that fail to reflect the arbitrary distribution.
3) KL-Loss degenerates to L2-norm-based loss with predicted standard deviation $\sigma = 1$.

This leads to an imbalance in regression loss during training because L2 loss gives small errors for easy samples but larger errors for hard samples. To overcome problems (1) and (3) of KL-Loss, Gaussian YOLOv3 [21] models both predictions and targets as Gaussian distributions. However, the mentioned problem (2) opens challenging issues. To the best of our knowledge, there are two recent methods, KL-Loss and Gaussian YOLOv3, investigating localization uncertainty in the detection research.

Finally, the difference between our proposed method and conventional methods is summarized as follows.

1) The GT boxes are discretized into multiple target boxes to recover real boundaries of objects. The BBENet is trained with multiple targets to learn bounding box locations under discretized distribution over the continuous domain of KL-Loss and Gaussian YOLOv3. This new representation can solve the problem (2) of both KL-Loss and Gaussian YOLOv3.
2) Without creating new loss function such as KL-Loss and negative log likelihood (NLL) loss in Gaussian YOLOv3, we can reduce the influence of unclear boundaries on the detection performance.

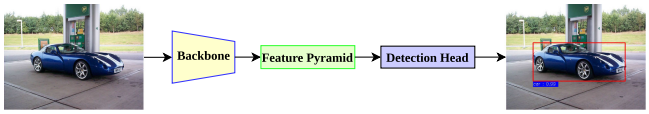

Fig. 2. Overall architecture of the BBENet consists of three parts: backbone network, feature pyramid, and detection head. The backbone network extracts the informative features from the input image. The feature pyramid constructs multilevel feature maps with different scales. The detection head predicts classification scores and bounding boxes for each object in the image.

3) The proposed bounding box encoding algorithm is integrated into the detection head to produce accurate boxes and uncertainty scores.

C. Localization Quality

Most of the state-of-the-art single-shot detectors [9]–[13], [16], [18], [24], [26], [27], [29] identified the predicted classification scores as final detection scores without considering localization quality. During NMS procedure, detectors use this score to rank detection. However, this step suppresses detection with high localization quality, degrading the overall performance since classification and localization have a strong positive correlation. In the literature, there are three methods, FCOS [19], ATSS [14], and IoU-aware [22], considering localization quality. FCOS and ATSS attached a new branch to the localization subnet to predict centerness score that is considered as localization quality. During inference, the classification score and centerness score are multiplied as a final detection quality to suppress low-quality predicted bounding boxes that are far from the center of objects. IoU-aware [R8] predicted the IoU score between the predicted box and GT box as localization quality. Both centerness and IoU scores do not reflect the localization uncertainty of bounding boxes. As a result, the detection with high IoU or centerness scores still does not satisfy accurate localization in some difficult cases.

Unlike existing methods, we introduce an uncertainty score as a new form of localization quality that is combined with classification scores to calibrate the box quality score.

III. Methodology

This section analyzes the general architecture of the single-shot object detection BBENet, the bounding box encoding algorithm in Section III-A, and uncertainty score prediction in Section III-B.

The overall architecture of the BBENet is shown in Fig. 2. The used backbone network is ResNet-50 [30], illustrated in Fig. 3. Inspired by FPN [25], the pyramid feature $\{P_3, P_4, P_5, P_6, P_7\}$ with various scales is also described in Fig. 3. As shown in Fig. 4, the detection head consists of the classification and localization branches, independently performing detection on each feature map of the pyramid. The classification branch classifies anchor boxes belonging to certain classes, i.e., outputting classification scores. The localization branch predicts distance offsets from the center of anchor boxes to multiple object boundaries. Then, the bounding box encoding algorithm

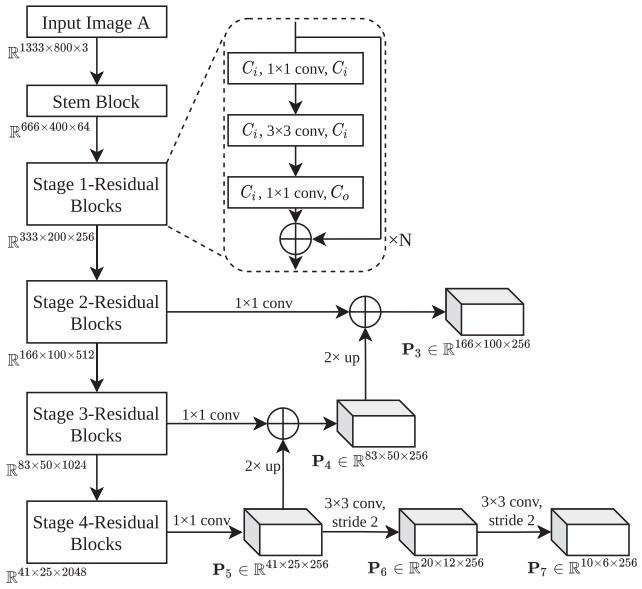

Fig. 3. Detailed network of the backbone and neck parts in BBENet. The stem block includes 7 × 7 convolution and 3 × 3 max-pooling with stride 2. C_i, C_o is the number of input and output channels. N denotes the number of residual blocks on each stage. $\{P_3, P_4, P_5, P_6, P_7\}$ is the feature pyramid with different scales.

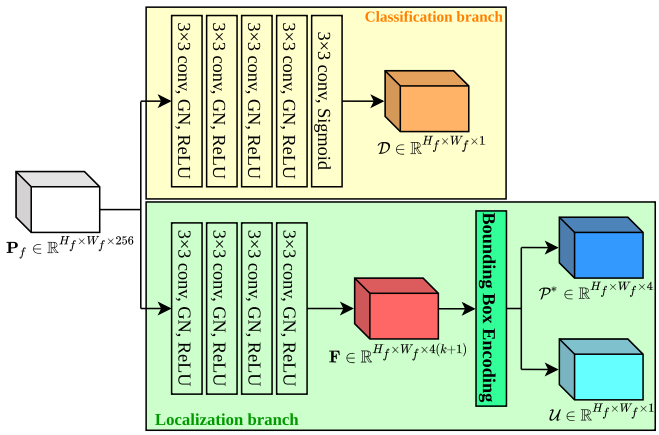

Fig. 4. Detailed network of the detection head in BBENet. P_f is the feature map in the feature pyramid with $f \in \{3, 4, 5, 6, 7\}$ and dimension $H_f \times W_f$ (height and width of tensor). k is the number of discretized boundaries on each direction of the target box, defined in Fig. 5. Each branch consists of four convolutional layers, where each layer contains 3 × 3 conv followed by group normalization (GN) and ReLU activation function.

aligns predicted bounding boxes learned under discretized representation of the GT box locations. Finally, the uncertainty score is estimated from the IoU set of aligned bounding boxes and the GT bounding box.

A. Bounding Box Encoding Algorithm

Due to the uncertainty problem, ambiguous boundaries of objects and ambiguous learning are identified. Thus, we have to redefine the coordinates of GT box locations and design an algorithm to process the new representation of GT boxes. First, each boundary of the GT bounding boxes is discretized into

Algorithm 1: Bounding Box Encoding Algorithm.

Input:
$\mathcal{G}$ is a set of GT bounding boxes
F is a set of predicted bounding boxes
Output:
$\mathcal{P}^*$ is an optimal set of predicted bounding boxes
$\mathcal{U}$ is a set of uncertainty scores

1: $\mathcal{P}^* \leftarrow \varnothing; \mathcal{U} \leftarrow \varnothing;$
2: **for** $\mathcal{Q}_i \in F$ **do**
3: $\mathcal{T}_s \leftarrow \varnothing; \mathcal{L}_s \leftarrow \varnothing; \mathcal{B}_s \leftarrow \varnothing; \mathcal{R}_s \leftarrow \varnothing;$
4: **for** $q_{i,j} \in \mathcal{Q}_i$ **do**
5: $\mathcal{T}_s \leftarrow t_{i,j} \in q_{i,j};$
6: $\mathcal{L}_s \leftarrow l_{i,j} \in q_{i,j};$
7: $\mathcal{B}_s \leftarrow b_{i,j} \in q_{i,j};$
8: $\mathcal{R}_s \leftarrow r_{i,j} \in q_{i,j};$ // $(k+1)$ *right boundaries*
9: **end for**
10: $\mathcal{M}_i \leftarrow$ Recombination {Use (1)}
11: **for** $g \in \mathcal{G}$ **do**
12: $\mathcal{S}_g \leftarrow s^g_{i,c} = IoU(\mathcal{G}_g, \mathcal{M}_{i,c})$; {Use (2)}
13: $\sigma_g = Std(\mathcal{S}_g)$; {Use (6)}
14: $c^* = argmax(\mathcal{S}_g)$;
15: $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \mathcal{M}^g_{i,c^*}$; {Top-1 bounding box}
16: $\mathcal{U} \leftarrow \sigma_g$; {Uncertainty score}
17: **end for**
18: **end for**
19: **return** $\mathcal{P}^*, \mathcal{U}$

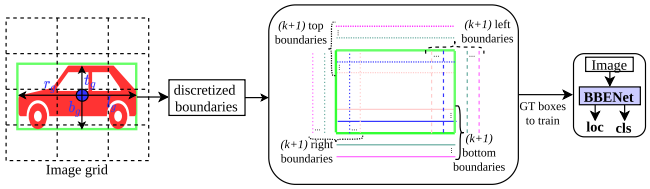

Fig. 5. Ground truth (GT) with the green box is discretized into $4(k+1)$ boundaries corresponding to $(k+1)$ target boxes to involve real boundaries of objects due to uncertainty. The BBENet is trained with multiple targets to learn bounding box locations under discretized probability distribution. $\{t_g, l_g, b_g, r_g\}$ are the coordinates of the original GT box.

$(k+1)$ boundaries to generate more reliable box coordinates. This procedure generates $(k+1)$ target boxes for each instance, as shown in Fig. 5. Second, the proposed BBENet with the new representation of GT labels obtains a dense set of foreground bounding boxes up to $(k+1)$ predictions for each pixel inside an instance A. However, the detector requires one bounding box per one object. Therefore, we propose the bounding box encoding algorithm to reduce the number of bounding boxes per location and keep the accurate ones. To perform that, the bounding box encoding algorithm reorganizes predicted bounding boxes to be more accurate and then finds the optimal box from the reorganized bounding box set for each instance. Because the detector does not know which boundaries are ambiguous, we reconstruct predicted boxes to form the relation between them, i.e., create accurate bounding box candidates as much as possible. The optimal box is sorted via ranking IoU scores, well aligned with object boundary. Algorithm 1 describes how the bounding box encoding operates.

Given an instance A, the set of predicted boxes are $F = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_n\}$, where n is the number of positive samples following [9] and[19]. For each pixel, the set of estimated boxes learned under improved distribution of box targets are $\mathcal{Q}_i = \{q_{i,0}, \ldots, q_{i,k}\}$, where k is the number of discretized boundaries on each direction of the bounding box. Following [19], the regressed distances of a box are $q_{i,j} = \{t_{i,j}, l_{i,j}, b_{i,j}, r_{i,j}\}$ corresponding to top, left, bottom, and right boundaries. Fig. 6 describes all components of the each set.

Boundary Selection: The first, lines 3–9 in Algorithm 1 arranges four groups of boundaries into four sets, e.g., a set of top boundary $\mathcal{T}_s = \{t_{i,0}, \ldots, t_{i,k}\}$, a set of left boundary $\mathcal{L}_s = \{l_{i,0}, \ldots, l_{i,k}\}$, a set of bottom boundary $\mathcal{B}_s = \{b_{i,0}, \ldots, b_{i,k}\}$, and a set of right boundary $\mathcal{R}_s = \{r_{i,0}, \ldots, r_{i,k}\}$. The structure of each set is analyzed in Fig. 6.

Recombination: The combination of the four sets is performed to reconstruct new bounding boxes described in the line 10 of Algorithm 1, defined as

$$\mathcal{M}_i = \{\mathcal{M}_{i,0}, \ldots, \mathcal{M}_{i,C}\} = \{\{t_{i,j}, l_{i,j}, b_{i,j}, r_{i,j}\} \mid t_{i,j} \in \mathcal{T}_s \; l_{i,j} \in \mathcal{L}_s, b_{i,j} \in \mathcal{B}_s, r_{i,j} \in \mathcal{R}_s \quad \forall j \in [0,k]\} \tag{1}$$

where $\mathcal{M}_i$ is the recombined bounding box from boundary sets for each pixel. Using all the boundaries of the four boundary sets to obtain bounding boxes as much as possible is an insightful way. However, in this strategy, the computational cost is too high and difficult to implement. To reduce the model complexity, the simple ranking is applied to sort the distances between the top boundary in each set and the top target boundary. If the ranking is highest, this boundary closest to the GT is fixed during the recombination step. For example, Fig. 7 states recombined boxes according to the fixed top boundary $t_{i,0}$ nearest GT.

IoU Computation and Assignment: For each GT bounding box $g \in \mathcal{G}$, the proposed algorithm computes the IoU scores between the GT g and recombined bounding box $\mathcal{M}_{i,c}$ in line the 12 of Algorithm 1. For one reconstructed bounding box, the IoU score is computed as follows:

$$s^g_{i,c} = \text{IoU}(\mathcal{G}_g, \mathcal{M}_{i,c}) = \frac{I(\mathcal{G}_g, \mathcal{M}_{i,c})}{\text{Union}(\mathcal{G}_g, \mathcal{M}_{i,c})} \tag{2}$$

where $I(\mathcal{G}_g, \mathcal{M}_{i,c})$, $\text{Union}(\mathcal{G}_g, \mathcal{M}_{i,c})$, and $\text{IoU}(\mathcal{G}_g, \mathcal{M}_{i,c})$ compute the intersection area, union area, and the intersection of the union between the GT $\mathcal{G}_g$ and reconstructed box $\mathcal{M}_{i,c}$, respectively. $I(\mathcal{G}_g, \mathcal{M}_{i,c})$ is defined as

$$I(\mathcal{G}_g, \mathcal{M}_{i,c}) = (\min(t_{i,c}, t_g) + \min(b_{i,c}, b_g)) \times (\min(l_{i,c}, l_g) + \min(r_{i,c}, r_g)). \tag{3}$$

The function $\text{Union}(\mathcal{G}_g, \mathcal{M}_{i,c})$ is calculated as

$$\text{Union}(\mathcal{G}_g, \mathcal{M}_{i,c}) = (t_{i,c} + b_{i,c})(l_{i,c} + r_{i,c}) + (t_g + b_g)(l_g + r_g) - I(\mathcal{G}_g, \mathcal{M}_{i,c}) \tag{4}$$

where $\{t_g, l_g, b_g, r_g\}$ and $\{t_{i,c}, l_{i,c}, b_{i,c}, r_{i,c}\}$ are the top, left, bottom, and right boundaries of the GT bounding box and recombined box, respectively. The reconstructed bounding box has a low IoU score, and it is not selected since its boundaries

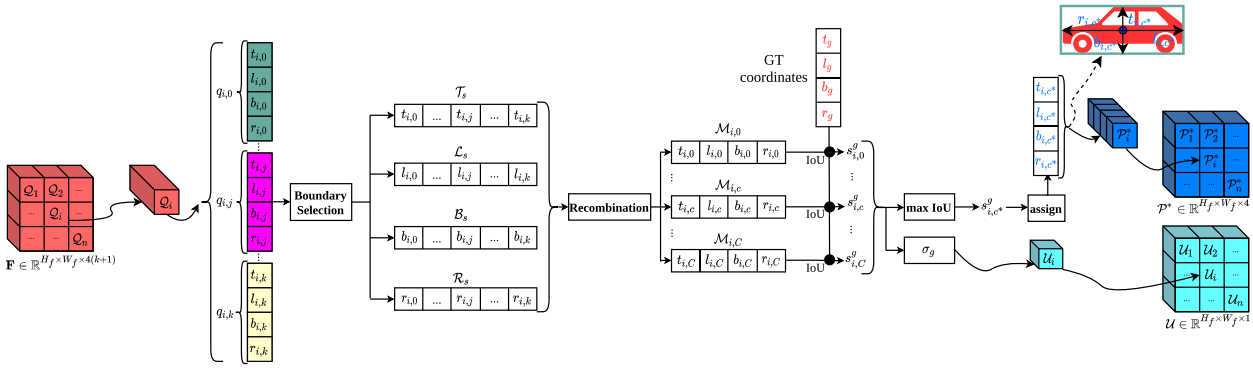

Fig. 6. Flow diagram of the bounding box encoding algorithm for selecting the optimal bounding box. Each pixel $\mathcal{Q}_i$ in the feature $\mathbf{F}$ regresses $(k+1)$ bounding boxes, each bounding box is $q_{i,j} = \{t_{i,j}, l_{i,j}, b_{i,j}, r_{i,j}\}$ with $j \in [0, k]$. $\mathcal{T}_s$, $\mathcal{L}_s$, $\mathcal{B}_s$, and $\mathcal{R}_s$ are the set of top, left, bottom, and right boundaries. IoU indicates the intersection of union of the recombined box $\mathcal{M}_{i,c}$ and the original GT box (red color). Then, the bounding box with highest rank is considered as the final output. σ_g is the uncertainty score of the IoU set $\{s^g_{i,0}, \ldots, s_{i,C}\}$, where C is the number of recombined bounding boxes. Finally, the feature $\mathcal{P}^*$ and $\mathcal{U}$ are computed for every pixel inside the feature $\mathbf{F}$.

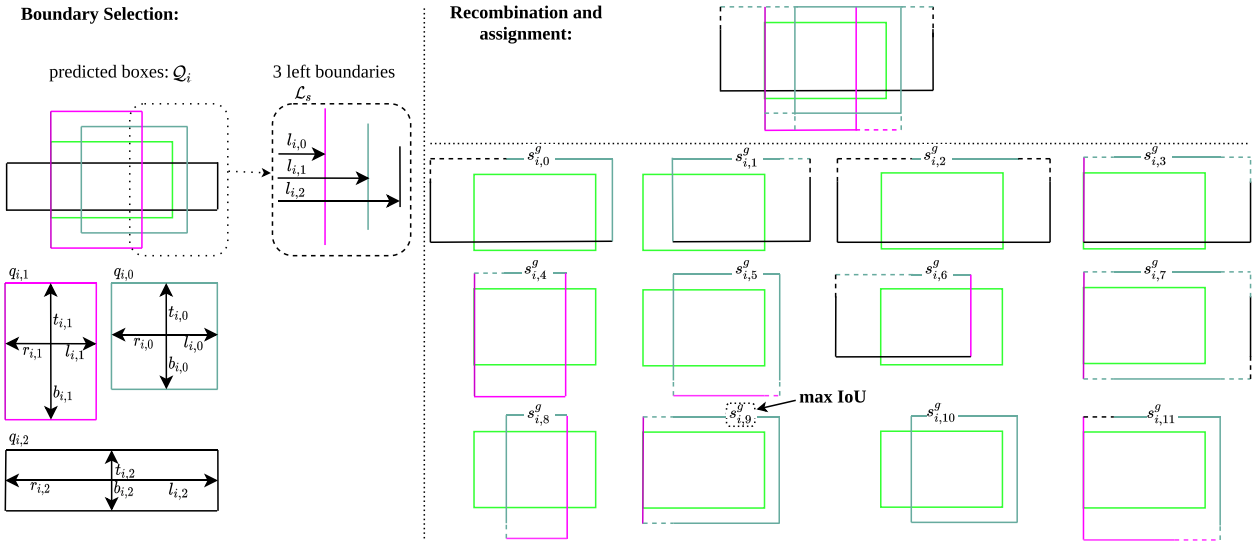

Fig. 7. Toy example with $k = 2$ to describe box coordinates, boundary selection, and recombination. The original GT box is denoted by the green box. Other colors indicate predicted bounding boxes. The number of recombined boxes is $C = 12$ for simplicity. The $s^g_{i,9}$ with the highest score is assigned as the final result.

are far from the target box. Thus, the optimal bounding box is assigned to be well aligned to the GT box in lines 14 and 15 of Algorithm 1. This representation ensures the assigned bounding box is close to the target location, which aims to reduce localization uncertainty on the detection performance. As shown in Fig. 7, the $s^g_{i,9}$ achieves the highest IoU score, the 9th reorganized box is assigned to the final form toward more accurate estimation.

B. Uncertainty Score as Localization Quality

The proposed BBENet outputs the uncertainty score to represent the localization quality for each box in the line 16 of Algorithm 1. For one GT, the set $\mathcal{S}_g$ contains IoU scores of recombined bounding boxes. Inspired by KL-Loss [20], the distribution of IoU variables follows the normal distribution. Accordingly, the mean and standard deviation of the statistic characteristics of reorganized boxes are computed as

$$\mu_g = \frac{1}{C}\sum_{c=0}^{C} \text{IoU}(\mathcal{G}_g, \mathcal{M}_{i,c}) \tag{5}$$

$$\sigma_g = \sqrt{\frac{1}{C}\sum\nolimits_{c=0}^{C} \left[\text{IoU}(\mathcal{G}_g, \mathcal{M}_{i,c}) - \mu_g\right]^2} \tag{6}$$

where C is the number of recombined boxes for each GT ($C = 12$ in Fig. 7). $\text{IoU}(\mathcal{G}_g, \mathcal{M}_{i,c}) = s^g_{i,c}$ is the intersection of the union between the GT and recombined box. Because the GT is not provided during inference, the bounding box with the highest confidence score is considered as the reference to compute the IoU score.

The BBENet obtains the standard deviation σ_g as an uncertainty score. The final detection quality includes the classification and uncertainty scores, calculated in line 3 of Algorithm

Algorithm 2: NMS With Uncertainty Score.

Input:
$\mathcal{P}^* = \{\mathcal{P}_1^*, \ldots, \mathcal{P}_n^*\}$ is the list of optimal bounding boxes
$\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ is the list of classification scores
$\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_n\}$ is the list of uncertainty scores
α is the IoU threshold
Output:
$\mathcal{O}$ is the list of final bounding boxes
$\mathcal{H}$ is the list of detection quality scores

$\mathcal{O} \leftarrow \{\}; \mathcal{H} \leftarrow \{\};$
for $\mathcal{D}_i \in \mathcal{D}$ and $\mathcal{U}_i \in \mathcal{U}$ **do**
  $\mathcal{H} \leftarrow \mathcal{H}_i = \mathcal{D}_i \times (1 - \mathcal{U}_i)$; *// detection quality scores*
end for
while $\mathcal{P}^* \neq$ empty **do**
  $m \leftarrow \text{argmax}(H)$;
  $\mathcal{Y} \leftarrow \mathcal{P}_m^*$;
  $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{Y}; \mathcal{P}^* \leftarrow \mathcal{P}^* - \mathcal{Y}$;
  for $\mathcal{P}_i^* \in \mathcal{P}^*$ **do**
    if $\text{IoU}(\mathcal{Y}, \mathcal{P}^*) \geqslant \alpha$ **then**
      $\mathcal{P}^* \leftarrow \mathcal{P}^* - \mathcal{P}_i^*; \mathcal{H} \leftarrow \mathcal{H} - \mathcal{H}_i$;
    end if
  end for
end while
return $\mathcal{O}, \mathcal{H}$

2. If the standard deviation σ_i is larger (the box i contains ambiguous boundaries), the value $(1 - \mathcal{U}_i)$ becomes smaller. After multiplying, the detection score is smaller. It means that this box has low localization quality. Thus, the box i has a low ranking, and it will potentially be suppressed because the boundary of the bounding box is farther from the object's boundary (this box is considered as a mislocalized box). If we do not suppress it, this box becomes false positive (FP). It directly decreases the detection performance even the detector produces a high classification score; otherwise σ_i is lower, which means that this box has high localization quality. The value $(1 - \mathcal{U}_i)$ becomes larger. Thus, the detection score $\mathcal{H}_i$ becomes larger. This box is sorted as high ranking (high detection quality). Therefore, our method produces high detection quality and suppresses the uncertainty box correctly. As illustrated in Fig. 8, our quality scores under the uncertainty problem are greater than the nonlocalization quality method.

IV. Experiments

A. Dataset

The proposed method is conducted on the challenging MS-COCO benchmark [8] to evaluate the effectiveness of the bounding box encoding algorithm and uncertainty score as localization quality. This dataset contains 115 k images for training and 5 k validation images for performing the ablation study. Comparisons with previous works are measured on the test − dev with 20 k images. All detection results are evaluated by evaluation code [8], with the standard metrics, e.g., average precision (AP), AP at different IoU thresholds (AP^{50}, AP^{75}), and AP at across scales (AP^S, AP^M, AP^L).

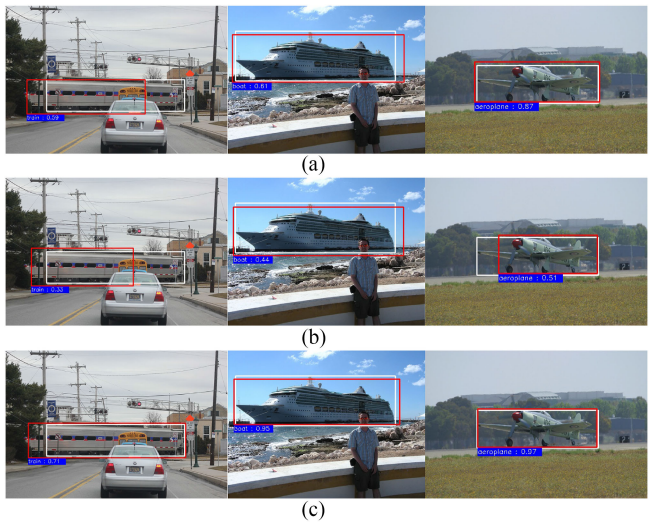

Fig. 8. Qualitative results of (a) RetinaNet [9], (b) ATSS [14], and (b) our BBENet on MS-COCO validation set with uncertainty cases. The white boxes denote GT labels. Red boxes denote the box prediction. Each score indicates detection quality.

B. Implementation Details

All experiments were implemented by the deep learning Pytorch framework. The proposed network adopts the stochastic gradient descent as an optimizer with a weight decay of 0.0001 and momentum of 0.9. The BBENet was trained for 12 epochs with a batch size of 8 on a GPU NVIDIA Titan, Cuda 10.2, and CuDNN 7.6.5. Specifically, the learning rate began at 0.0025 and decreased ten times at epoch 8 and epoch 11. Following common settings [9], [14], [19], the input image is resized to 1333×800.

As shown in Fig. 3, the proposed network employs the backbone ResNet [30] pretrained on ImageNet for feature extraction. The initialized weights of the added convolution layers in the feature pyramid and detection head were filled from the normal distribution. For anchor box settings, the proposed method only places one square anchor box per location to avoid computational overhead. Note that the offset prediction is the distance value between the center of the anchor box and boundaries.

During training, the number of encoded boundaries on each side of the bounding box was adopted to k=16 for all experiments because the performance achieves accuracy and speed balance at this value. Similar to previous works with FCOS [19] and ATSS [14], the training loss is defined as

$$L = \lambda_1 L_{\text{cls}}(\mathcal{D}_i, \hat{\mathcal{D}}_i) + \lambda_2 L_{\text{loc}}(\mathcal{P}_i^*, \mathcal{G}) \tag{7}$$

where L_{cls} is the Focal loss [9] for the classification task, focusing on hard samples and downweighting the contribution of a large number of easy negative samples. $\mathcal{D}_i$, and $\hat{\mathcal{D}}_i$ denote the classification score and class label. L_{loc} is the GIoU loss [31] for

TABLE I
COMPARISON WITH STATE-OF-THE-ART SINGLE-SHOT DETECTORS ON MS-COCO *TEST-DEV* SET

Method	Backbone	Schedule	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	#param	GFLOPs	FPS
RetinaNet [9]	ResNet-50	1×	36.9	56.2	39.3	20.5	39.9	46.3	37.74M	250.34	19.0
FCOS [19]	ResNet-50	1×	36.9	56.7	39.3	20.6	39.5	46.0	32.02M	209.71	22.7
FoveaBox [13]	ResNet-50	1×	37.0	56.7	39.1	20.3	40.0	45.6	36.19M	215.80	24.1
GHM [10]	ResNet-50	1×	37.5	56.4	39.9	21.1	39.8	47.0	37.74M	250.34	3.3
FSAF [11]	ResNet-50	1×	37.5	57.1	39.8	20.4	39.9	47.0	36.19M	215.80	12.3
AugFPN [26]	ResNet-50	1×	37.5	58.4	40.1	21.3	40.5	47.3	-	-	-
YOLOF [17]	ResNet-50	1×	37.7	56.9	40.6	19.1	42.5	53.2	43.88M	104.76	**30.2**
Sparse RCNN [29]	ResNet-50	1×	37.9	56.0	40.5	20.7	40.0	53.5	106.07M	156.43	17.2
RepPoints [15]	ResNet-50	1×	38.3	**59.2**	41.3	21.9	41.5	47.2	36.62M	198.94	17.5
NAS-FCOS [27]	ResNet-50	1×	38.5	57.3	41.3	21.8	42.3	50.7	38.84M	204.82	13.3
FreeAnchor [12]	ResNet-50	1×	38.9	57.8	41.6	21.3	41.5	49.0	37.74M	250.34	18.4
ATSS [14]	ResNet-50	1×	39.6	58.2	42.9	**23.3**	42.4	48.5	32.07M	214.67	19.7
BBENet (Ours)	ResNet-50	1×	**40.0**	58.6	**43.2**	22.4	**43.2**	**49.9**	32.22M	217.91	21.5
RetinaNet [9]	ResNet-101	1×	39.0	58.6	41.7	21.9	42.2	49.3	56.74M	330.21	15.0
FCOS [19]	ResNet-101	1×	39.2	59.0	41.9	22.1	42.1	49.0	50.96M	289.58	17.3
FoveaBox [13]	ResNet-101	1×	38.9	58.7	41.5	21.7	42.4	48.1	55.19M	295.67	17.4
GHM [10]	ResNet-101	1×	39.5	58.6	42.0	22.2	42.4	50.1	56.74M	330.21	4.4
FSAF [11]	ResNet-101	1×	39.7	59.5	42.5	21.9	42.4	50.1	55.19M	295.67	10.8
YOLOF [17]	ResNet-101	1×	39.8	59.4	42.9	20.5	44.5	54.9	62.82M	184.63	**19.6**
RankDetNet [18]	ResNet-101	1×	40.0	59.7	43.2	21.9	43.0	50.6	-	-	-
Pseudo-IoU [16]	ResNet-101	1×	40.4	59.5	40.9	23.7	44.9	51.4	-	-	-
RepPoints [15]	ResNet-101	1×	40.9	**62.0**	44.0	23.3	44.0	51.5	55.62M	278.82	13.7
FreeAnchor [12]	ResNet-101	1×	40.7	59.7	43.7	22.1	43.5	51.5	56.74M	330.21	14.9
Dr. Retina [28]	ResNet-101	1×	41.7	60.9	44.8	23.5	44.9	53.1	-	-	-
ATSS [14]	ResNet-101	1×	41.9	60.6	45.6	**24.6**	45.1	51.8	51.06M	294.55	12.3
BBENet (Ours)	ResNet-101	1×	**42.2**	61.1	**45.7**	23.4	**45.8**	**53.3**	51.21M	297.79	16.3
ATSS [14]	ResNet-101-ms	2×	43.6	62.1	47.4	26.1	47.0	53.6	51.06M	294.55	12.3
BBENet (Ours)	ResNet-101-ms	2×	44.8	63.6	48.6	27.1	48.4	55.1	51.21M	297.79	**16.2**
ATSS+BBENet(Ours)	ResNet-101-ms	2×	**45.6**	**63.4**	**49.6**	**28.4**	**50.2**	**60.0**	51.26M	302.75	14.2
ATSS [14]	ResNeXt-32×4d-101-ms	2×	45.1	63.9	49.1	27.9	48.2	54.6	50.70M	298.56	8.9
BBENet (Ours)	ResNeXt-32×4d-101-ms	2×	45.9	64.6	50.0	28.2	49.3	56.3	50.85M	301.70	**13.0**
ATSS+BBENet(Ours)	ResNeXt-32×4d-101-ms	2×	**48.4**	**66.5**	**52.3**	**29.3**	**53.0**	**64.5**	50.90M	306.76	10.4

The bold values used to indicate the best performance among detectors.

the localization task. $\mathcal{P}_i^* = \{t_{i,c^*}, l_{i,c^*}, b_{i,c^*}, r_{i,c^*}\}$ and $\mathcal{G}$ denote the optimal bounding box selected in Algorithm 1 and GT box, respectively. Following ATSS [14], the balance terms λ_1 and λ_2 are 1.0 and 2.0, respectively.

During inference, the input image is forwarded to the network that outputs classification scores, bounding box regression, and uncertainty scores. At the NMS step, the classification and uncertainty scores are combined to rank the detection.

V. RESULTS ON MS-COCO

A. Comparison With State-of-The-Art Methods

The results of the proposed BBENet were evaluated on the MS-COCO *test-dev* set and compared with state-of-the-art single-shot object detections listed in Table I. Note that all experiments used the input image size of 1333 × 800. The bold font denotes the best performance among detectors with the same backbone and learning schedule (1×, 2× means the model is trained for 12 epochs and 24 epochs). Here, ms indicates multiscale training. We use two scales: 1333 × 480 and 1333 × 800 of the input image to train the model.

For the backbone ResNet-50 [30], the proposed BBENet outperforms the popular RetinaNet [9] and FCOS [19] by a large margin, e.g., 3.1% AP. Moreover, the proposed method runs at 21.5 frames per second (FPS), faster than RetinaNet with 19.0 FPS. Because the number of anchor boxes per location is one for the BBENet and nine for RetinaNet, the computation cost is fewer 9× than RetinaNet. Compared with FCOS, the parameter of the BBENet only increases 0.2 M (millions) while the speed reduces from 22.7 to 21.5 FPS. Note that the proposed network considers RetinaNet and FCOS as the baseline. Accordingly, the proposed BBENet improves the detection performance of the baseline without affecting the speed of the network. Remarkably, our result achieved 40% AP, surpassing all state-of-the-art detectors, such as FoveaBox [13] at 37% AP, GHM [10] at 37.5% AP, AugFPN [26] at 37.5% AP, YOLOF [17] at 37.7% AP, Sparse RCNN [29] at 37.9% AP, RepPoints [15] at 38.3% AP, NAS-FCOS [27] at 38.5% AP, FreeAnchor [12] at 38.9% AP, and the strong detector ATSS [14] at 39.6% AP. As expected, the AP at IoU=0.75 of the BBENet is larger than the baseline RetinaNet and FCOS by 3.9% AP. Therefore, our method performs well at high IoU thresholds, i.e., more accurate detections.

Similarly, the performance of the proposed method also surpasses the baselines and other single-shot detectors with the same backbone network ResNet-101 and learning schedule 1×. By using the stronger backbone ResNeXt-32×4d-101 and multiscale training strategy, the BBENet achieves the best result 45.9% AP at 13 FPS that outperforms other detectors by a large margin without bells and whistles. Additionally, we insert the advanced anchor assignment ATSS to BBENet. Our detection performance achieves 48.4% AP that establishes the new state-of-the-art single-shot detector.

TABLE II
INVESTIGATION OF DIFFERENT VALUES OF k

k	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
4	25.3	50.0	21.8	16.3	26.4	32.3
8	39.3	57.7	42.4	22.2	42.8	51.8
12	39.7	58.0	42.9	22.4	43.7	52.0
16	39.8	58.1	42.7	22.6	43.9	52.3
20	39.7	57.8	42.9	23.0	43.1	52.1

TABLE III
EFFECTS OF EACH COMPONENT ON MS-COCO VALIDATION SET

Method	AP	AP^{75}	#param	GFLOPs	FPS
Baseline	36.6	38.8	32.07	214.67	22.7
Baseline+BBEAlg	37.9	40.8	32.22	217.91	21.5
Baseline+BBEAlg+σ	39.8	42.7	32.22	217.91	21.5

TABLE IV
COMPARISON OF UNCERTAINTY METHOD AND LOCALIZATION QUALITY

Method	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
KL-Loss [20]	39.2	57.6	42.5	21.2	41.8	52.5
IoU-aware [22]	36.9	56.1	40.1	20.9	40.0	46.0
Centerness [14]	39.4	57.6	42.8	23.6	42.9	50.3
BBENet (Ours)	39.8	58.1	42.7	22.6	43.9	52.3

TABLE V
RESULTS OF DIFFERENT DETECTORS ON MS-COCO VALIDATION SET

Method	AP	AP^{75}	#param	GFLOPs	FPS
RetinaNet [9]	36.5	39.1	37.74	250.34	19.0
RetinaNet*	36.3	38.6	32.07	214.67	22.7
RetinaNet*+BBEAlg	39.9	42.7	32.22	217.91	21.5
FCOS [19]	36.6	38.8	32.02	209.71	22.7
FCOS+BBEAlg	39.8	42.7	32.22	217.91	21.5
ATSS [14]	39.4	42.8	32.07	214.67	19.7
ATSS+BBEAlg	40.7	44.0	32.27	222.87	17.6

B. Ablation Studies

1) Hyperparameter k: Several experiments are conducted to investigate the robustness of the proposed BBENet to the value k (i.e., number of discretized boundaries on each direction of the target box). As shown in Table II, the value $k \in \{4, 8, 12, 16, 20\}$ is selected to train the model.

The results of the proposed method are very sensitive to the changes in k. Specifically, the result achieves 25.3% AP at k=4 and 39.3% AP at k=8. The small k=4 decreases the accuracy because too few encoded boundaries correspond to too few recombined bounding boxes, causing coarse boundary prediction. When the value k increases from 8 to 20, the performance is more stable, i.e., a slight increase in the accuracy from 39.3% AP to 39.7% AP. Accordingly, our method chooses $k = 16$ for all experiments because the model achieves accuracy and speed balance at this value.

2) Effect of Each Component: This section analyzes the performance of each component on the MS-COCO validation set. The results shown in Table III are as follows.

1) *Baseline:* The type inherits the flexible FCOS [19] and anchor-based RetinaNet [9] as the simplest version of BBENet.
2) *Baseline+BBEAlg:* The bounding box encoding algorithm 1 (BBEAlg) toward accurate prediction is added to the baseline. As given in Table III, the proposed BBEAlg gains 1.3% AP from the Baseline 36.6% to 37.9% with inconsiderable complexity. Specifically, BBEAlg only increases the number of parameters and computational cost of the baseline by 0.25 M and 3.24 giga floating point operations per second (GFLOPS). Therefore, the results show the effectiveness of the BBEAlg in both accuracy and computations.
3) *Baseline+BBEAlg+Uncertainty:* This is the full implementation of the proposed BBENet, which adds an uncertainty score in the NMS step to reduce the effects of localization uncertainty on the performance. This version achieves 39.8% AP, further improving the Baseline and Baseline+BBEAlg by 3.2% and 1.9% AP, respectively. Remarkably, the performance at AP^{75} gains 3.9%, that detection has high accurate localization. This shows the detection quality joined by classification score and uncertainty score can significantly improve the overall performance. As the BBENet creates the uncertainty score based on Algorithm 1, the model complexity is the same as Baseline+BBEAlg.

3) Localization Uncertainty: The BBENet surpasses the first uncertainty method KL-Loss [20] by 0.6% AP, as shown in Table IV. As the proposed BBENet models bounding box distributions in a more flexible way, KL-Loss considers the prediction and label as fixed distributions, e.g., Gaussian and Dirac delta distribution. Alternatively, KL-Loss employs two-stage detection Faster R-CNN [23] as the baseline, while our method is a one-stage detection toward the efficient network. In comparison to localization quality, the BBENet predicting uncertainty score is superior to the IoU score [22] and centerness score [19].

4) Results on Different Detectors: We conduct the experiments on the MS-COCO validation set to evaluate the effectiveness of the bounding box encoding algorithm (BBEAlg) on different detectors. The results are given in Table V. All detectors are trained under ResNet-50, and learning schedule 1×. RetinaNet* indicates the simplified version of RetinaNet [9] when we only place one anchor box per location versus nine anchor boxes in the original RetinaNet. This simplified RetinaNet is the same as the structure of FCOS [19]. As a result, our BBEAlg achieves consistent AP improvements on all detectors (e.g., 3.6% AP, 3.2% AP, and 1.3% AP improvements), demonstrating its effectiveness and generality.

5) Comparative Visualizations: The comparative visualization between the baseline RetinaNet [9], ATSS [14], and the proposed method is illustrated in Fig. 8 with unclear boundaries and partial occlusions of objects. As expected, the proposed BBENet predicts more accurate bounding box localization than other detectors. Our bounding boxes have a finer boundary than the GT labels. In the first column, both RetinaNet and ATSS generate low localization quality (mislocalized detection) and low confidence score (achieve a score of 0.59, and 0.33—misclassified detection), while our detector produces high localization quality and confidence score. In the third column, although RetinaNet produces a highly overconfidence score (up

TABLE VI
RESULTS ON PASCAL VOC TEST SET

Method	Image size	mAP	#param	GFLOPs	FPS
FCOS [19]	1000×600	65.0	31.88	116.92	**30.8**
RetinaNet [9]	1000×600	77.3	36.50	125.88	26.8
ATSS [14]	1000×600	78.7	31.93	119.71	27.6
Faster RCNN [23]	1000×600	79.5	41.22	127.75	27.5
Our BBENet	1000×600	**83.6**	32.08	121.53	29.4

The bold values used to indicate the best performance among detectors.

TABLE VII
COMPARISON WITH DETECTORS ON CROWDHUMAN DATASET

Method	AP	AP^{75}	#param	GFLOPs	FPS
RetinaNet [9]	31.2	23.4	37.74	250.34	19.0
FCOS [19]	32.3	24.7	32.02	209.71	**22.7**
ATSS [14]	39.2	36.1	32.07	214.67	19.7
BBENet (Ours)	**42.9**	**41.7**	32.22	217.91	21.5

The bold values used to indicate the best performance among detectors.

to 0.87) but the bounding box prediction is not accurate. ATSS outputs both low localization and classification quality in this case.

VI. RESULTS ON PASCAL VOC AND CROWDHUMAN

A. Results on Pascal VOC

To verify the effectiveness of the proposed method, we present the experimental results on Pascal VOC [32] dataset. Pascal VOC consists of 20 classes for training and inference. All detectors are trained on the union *trainval* set of the Pascal VOC 2007 and 2012 and evaluated on the Pascal VOC 2007 *test* set. We reimplement all detectors with the same settings on MS-COCO dataset for fair comparisons, such as learning schedule 1× and backbone ResNet-50. We use the mean average precision (mAP) metric to evaluate the performance. Table VI shows that the proposed method surpasses the two-stage method Faster R-CNN [23] by 4.1% mAP and the strong ATSS [14] by 4.9% mAP. It demonstrates the generalization ability of our proposed BBENet.

B. Results on CrowdHuman

We conduct the experiment on a challenging dataset, e.g., CrowdHuman [33] dataset. This dataset contains a large set of uncertainty cases, addressing heavy occlusion in crowded scenes. All the hyperparameters are the same as MS-COCO dataset. Table VII tabulates the results of the proposed method, RetinaNet, FCOS, and ATSS. On the CrowdHuman dataset, we outperform the ATSS by 3.7% and the baseline RetinaNet by 11.7% AP. The performance gap between the BBENet and other detectors can be explained as follows.

1) The conventional detectors do not identify the uncertainty of predicted bounding boxes, and it leads to mislocalized detections (FP).
2) CrowdHuman dataset contains uncertainty cases in every image.

Thus, both reasons generate many boxes with low detection quality. It largely decreases the detection performance.

Therefore, our detector performs well in general cases, while RetinaNet, FCOS, and ATSS perform well in clear cases.

VII. CONCLUSION

This article introduced the BBENet toward the accurate object detector by investigating the localization uncertainty in object detection. Specifically, the BBENet proposed the novel bounding box encoding algorithm in which bounding box predictions are learned under multiple discretized targets to be more reliable. The reconstructed bounding boxes based on the combination of four groups of the boundary are performed as statistical distribution to model the localization uncertainty of box representation. The optimal bounding box is chosen to be well aligned to target locations. Furthermore, the uncertainty scores are obtained as localization quality from the set of bounding box recombination, merged with the classification score to rank the detection during the NMS procedure. Extensive experiments on three benchmarks show that the proposed BBENet achieves accurate bounding box prediction, becoming a state-of-the-art single-shot object detector.

REFERENCES

[1] Z. Tang, E. Tian, Y. Wang, L. Wang, and T. Yang, "Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 82–89, Jan. 2020.
[2] M. D. Putro, L. Kurnianggoro, and K.-H. Jo, "High performance and efficient real-time face detector on cpu based on convolutional neural network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4449–4457, Jul. 2020.
[3] A. Masood *et al.*, "Automated decision support system for lung cancer detection and classification via enhanced RFCN with multilayer fusion RPN," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7791–7801, Dec. 2020.
[4] B. Hussain, Q. Du, A. Imran, and M. A. Imran, "Artificial intelligence-powered mobile edge computing-based anomaly detection in cellular networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 4986–4996, Aug. 2020.
[5] Y. Zhu, C. Chen, G. Yan, Y. Guo, and Y. Dong, "AR-Net: Adaptive attention and residual refinement network for copy-move forgery detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6714–6723, Oct. 2020.
[6] A. Shahbaz and K.-H. Jo, "Deep atrous spatial features based supervised foreground detection algorithm for industrial surveillance systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4818–4826, Jul. 2021.
[7] V.-T. Hoang, D.-S. Huang, and K.-H. Jo, "3-D facial landmarks detection for intelligent video systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 578–586, Jan. 2021.
[8] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999-3007.
[10] B. Li, Y. Liu, and X. Wang, "Gradient harmonized single-stage detector," in *Proc. Conf. Artif. Intell.*, 2019, pp. 8577–8584.
[11] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 840–849.
[12] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, "FreeAnchor: Learning to match anchors for visual object detection," in Proc. *Neural Inf. Process. Syst.*, 2019.
[13] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyound anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.
[14] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9756–9765.

[15] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9656–9665.
[16] J. Li *et al.*, "Pseudo-IoU: Improving label assignment in anchor-free object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2378–2387.
[17] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13034–13043.
[18] J. Liu, D. Li, R. Zheng, L. Tian, and Y. Shan, "RankDetNet: Delving into ranking constraints for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 264–273.
[19] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.
[20] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2883–2892.
[21] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 502–511.
[22] S. Wu, X. Li, and X. Wang, "IoU-aware single-stage object detector for accurate localization," *Image Vis. Comput.*, vol. 97, 2020, Art. no. 103911.
[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[24] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[26] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12592–12601.
[27] N. Wang *et al.*, "NAS-FCOS: Fast neural architecture search for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11940–11948.
[28] Q. Qian, L. Chen, H. Li, and R. Jin, "DR Loss: Improving object detection by distributional ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12161–12169.
[29] P. Sun *et al.*, "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14449–14458.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.
[32] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[33] S. Shao *et al.*, "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.

Xuan-Thuy Vo (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from the University of Science and Technology, University of Da Nang, Da Nang City, Vietnam, in 2018. He is currently working toward the M.S. and Ph.D. combined degrees with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, South Korea.

His current research interests include computer vision and deep learning focusing on object detection, object segmentation, and multiple people tracking.

Kang-Hyun Jo (Senior Member, IEEE) received the Ph.D. degree in computer-controlled machinery from Osaka University, Osaka, Japan, in 1997.

After a year of experience with ETRI as a Postdoctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea, where he is currently the Faculty Dean. His current research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence.

Dr. Jo was the Director or an AdCom Member with the Institute of Control, Robotics, and Systems, Society of Instrument and Control Engineers, and IEEE Industrial Electronics Society Technical Committee on Human Factors Chair, AdCom Member, and the Secretary until 2019. He is currently an Editorial Board Member for international journals, such as the *International Journal of Control, Automation, and Systems* and *Transactions on Computational Collective Intelligence*. He is also involved in organizing many international conferences, such as the International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, International Conference on Human System Interactions, and Annual Conference of the IEEE Industrial Electronics Society.