Acquisition of Localization Confidence for Accurate Object Detection

Borui Jiang[*1,3], Ruixuan Luo[*1,3], Jiayuan Mao[*2,4],
Tete Xiao[1,3], and Yuning Jiang[4]

[1] School of Electronics Engineering and Computer Science, Peking University
[2] ITCS, Institute for Interdisciplinary Information Sciences, Tsinghua University
[3] Megvii Inc. (Face++)   [4] Toutiao AI Lab
{jbr, luoruixuan97, jasonhsiao97}@pku.edu.cn,
mjy14@mails.tsinghua.edu.cn, jiangyuning@bytedance.com

Abstract. Modern CNN-based object detectors rely on bounding box regression and non-maximum suppression to localize objects. While the probabilities for class labels naturally reflect classification confidence, localization confidence is absent. This makes properly localized bounding boxes degenerate during iterative regression or even suppressed during NMS. In the paper we propose IoU-Net learning to predict the IoU between each detected bounding box and the matched ground-truth. The network acquires this confidence of localization, which improves the NMS procedure by preserving accurately localized bounding boxes. Furthermore, an optimization-based bounding box refinement method is proposed, where the predicted IoU is formulated as the objective. Extensive experiments on the MS-COCO dataset show the effectiveness of IoU-Net, as well as its compatibility with and adaptivity to several state-of-the-art object detectors.

Keywords: object localization, bounding box regression, non-maximum suppression

1 Introduction

Object detection serves as a prerequisite for a broad set of downstream vision applications, such as instance segmentation [19, 20], human skeleton [27], face recognition [26] and high-level object-based reasoning [30]. Object detection combines both object classification and object localization. A majority of modern object detectors are based on two-stage frameworks [9, 8, 22, 16, 10], in which object detection is formulated as a multi-task learning problem: 1) distinguish foreground object proposals from background and assign them with proper class labels; 2) regress a set of coefficients which localize the object by maximizing intersection-over-union (IoU) or other metrics between detection results and the ground-truth. Finally, redundant bounding boxes (duplicated detections on the same object) are removed by a non-maximum suppression (NMS) procedure.

---

[*] indicates equal contribution.
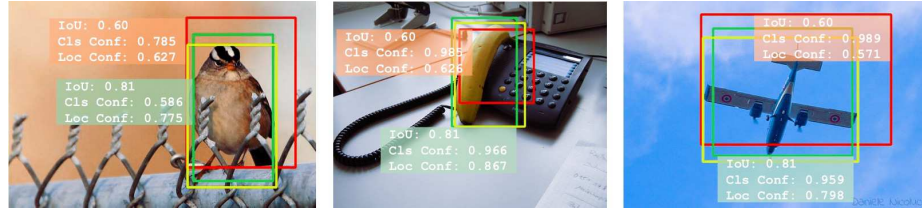
(a) Demonstrative cases of the misalignment between classification confidence and localization accuracy. The yellow bounding boxes denote the ground-truth, while the red and green bounding boxes are both detection results yielded by FPN [16]. Localization confidence is computed by the proposed IoU-Net. Using classification confidence as the ranking metric will cause accurately localized bounding boxes (in green) being incorrectly eliminated in the traditional NMS procedure. Quantitative analysis is provided in Section 2.1
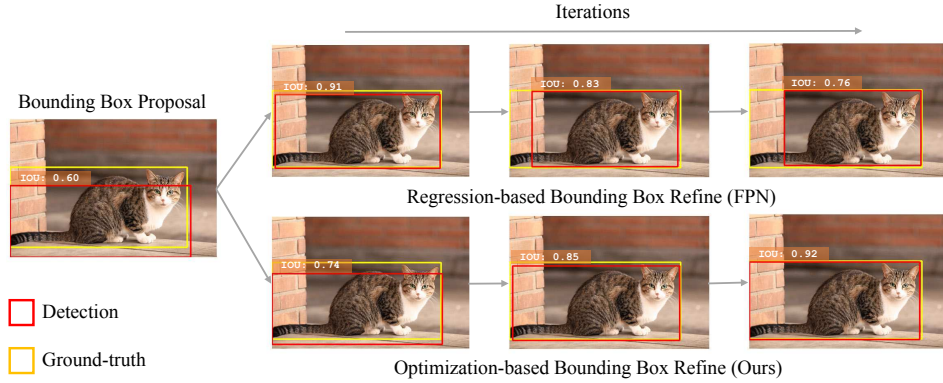
(b) Demonstrative cases of the non-monotonic localization in iterative bounding box regression. Quantitative analysis is provided in Section 2.2.

Fig. 1: Visualization on two drawbacks brought by the absence of localization confidence. Examples are selected from MS-COCO *minival* [17].

Classification and localization are solved differently in such detection pipeline. Specifically, given a proposal, while the probability for each class label naturally acts as an "classification confidence" of the proposal, the bounding box regression module finds the optimal transformation for the proposal to best fit the ground-truth. However, the "localization confidence" is absent in the loop.

This brings about two drawbacks. (1) First, the suppression of duplicated detections is ignorant of the localization accuracy while the classification scores are typically used as the metric for ranking the proposals. In Figure 1(a), we show a set of cases where the detected bounding boxes with higher classification confidences contrarily have smaller overlaps with the corresponding ground-truth. Analog to Gresham's saying that *bad money drives out good*, the misalignment between classification confidence and localization accuracy may lead to accurately localized bounding boxes being suppressed by less accurate ones in the NMS procedure. (2) Second, the absence of localization confidence makes the widely-

adopted bounding box regression less interpretable. As an example, previous works [3] report the non-monotonicity of iterative bounding box regression. That is, bounding box regression may degenerate the localization of input bounding boxes if applied for multiple times (shown as Figure 1(b)).

In this paper we introduce IoU-Net, which predicts the IoU between detected bounding boxes and their corresponding ground-truth boxes, making the networks aware of the localization criterion analog to the classification module. This simple coefficient provides us with new solutions to the aforementioned problems:

1. IoU is a natural criterion for localization accuracy. We can replace classification confidence with the predicted IoU as the ranking keyword in NMS. This technique, namely IoU-guided NMS, help to eliminate the suppression failure caused by the misleading classification confidences.
2. We present an optimization-based bounding box refinement procedure on par with the traditional regression-based methods. During the inference, the predicted IoU is used as the optimization objective, as well as an interpretable indicator of the localization confidence. The proposed Precise RoI Pooling layer enables us to solve the IoU optimization by gradient ascent. We show that compared with the regression-based method, the optimization-based bounding box refinement empirically provides a monotonic improvement on the localization accuracy. The method is fully compatible with and can be integrated into various CNN-based detectors [16, 3, 10].

2   Delving into object localization

First of all, we explore two drawbacks in object localization: the misalignment between classification confidence and localization accuracy and the non-monotonic bounding box regression. A standard FPN [16] detector is trained on MS-COCO *trainval35k* as the baseline and tested on *minival* for the study.

2.1   Misaligned classification and localization accuracy

With the objective to remove duplicated bounding boxes, NMS has been an indispensable component in most object detectors since [4]. NMS works in an iterative manner. At each iteration, the bounding box with the maximum classification confidence is selected and its neighboring boxes are eliminated using a predefined overlapping threshold. In Soft-NMS [2] algorithm, box elimination is replaced by the decrement of confidence, leading to a higher recall. Recently, a set of learning-based algorithms have been proposed as alternatives to the parameter-free NMS and Soft-NMS. [24] calculates an overlap matrix of all bounding boxes and performs affinity propagation clustering to select exemplars of clusters as the final detection results. [11] proposes the GossipNet, a post-processing network trained for NMS based on bounding boxes and the classification confidence. [12] proposes an end-to-end network learning the relation between detected bounding boxes. However, these parameter-based methods require more computational resources which limits their real-world application.

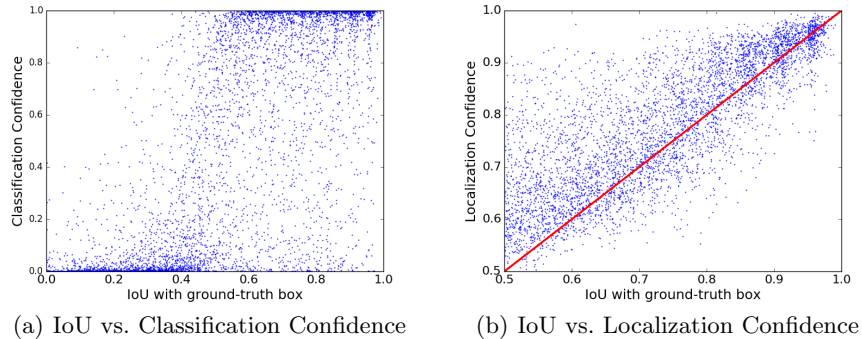(a) IoU vs. Classification Confidence   (b) IoU vs. Localization Confidence

Fig. 2: The correlation between the IoU of bounding boxes with the matched ground-truth and the classification/localization confidence. Considering detected bounding boxes having an IoU (> 0.5) with the corresponding ground-truth, the Pearson correlation coefficients are: (a) 0.217, and (b) 0.617.
(a) The classification confidence indicates the category of a bounding box, but cannot be interpreted as the localization accuracy.
(b) To resolve the issue, we propose IoU-Net to predict the localization confidence for each detected bounding box, *i.e.*, its IoU with corresponding ground-truth.

In the widely-adopted NMS approach, the classification confidence is used for ranking bounding boxes, which can be problematic. We visualize the distribution of classification confidences of all detected bounding boxes before NMS, as shown in Figure 2(a). The x-axis is the IoU between the detected box and its matched ground-truth, while the y-axis denotes its classification confidence. The Pearson correlation coefficient indicates that the localization accuracy is not well correlated with the classification confidence.

We attribute this to the objective used by most of the CNN-based object detectors in distinguishing foreground (positive) samples from background (negative) samples. A detected bounding box box_{det} is considered positive during training if its IoU with one of the ground-truth bounding box is greater than a threshold Ω_{train}. This objective can be misaligned with the localization accuracy. Figure 1(a) shows cases where bounding boxes having higher classification confidence have poorer localization.

Recall that in traditional NMS, when there exists duplicated detections for a single object, the bounding box with maximum classification confidence will be preserved. However, due to the misalignment, the bounding box with better localization will probably get suppressed during the NMS, leading to the poor localization of objects. Figure 3 quantitatively shows the number of positive bounding boxes after NMS. The bounding boxes are grouped by their IoU with the matched ground-truth. For multiple detections matched with the same ground-truth, only the one with the highest score is considered positive. Therefore, No-NMS could be considered as the upper-bound for the number of positive
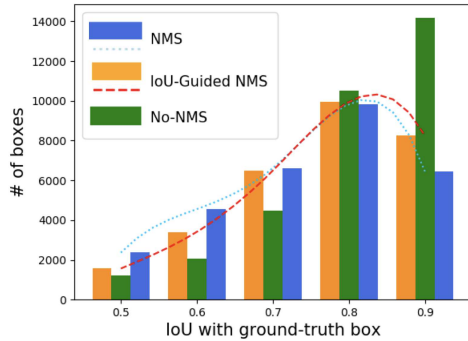
Fig. 3: The number of positive bounding boxes after the NMS, grouped by their IoU with the matched ground-truth. In traditional NMS (blue bar), a significant portion of accurately localized bounding boxes get mistakenly suppressed due to the misalignment of classification confidence and localization accuracy, while IoU-guided NMS (yellow bar) preserves more accurately localized bounding boxes.
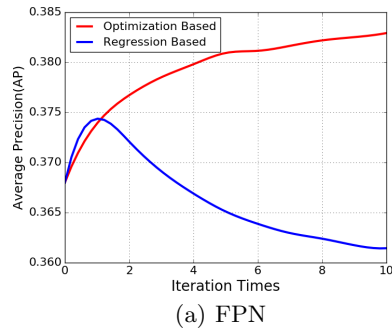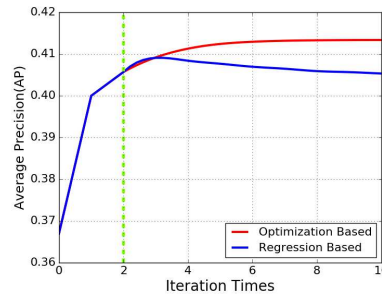
(a) FPN   (b) Cascade R-CNN

Fig. 4: Optimization-based *v.s.* Regression-based BBox refinement. **(a)** Comparison in FPN. When applying the regression iteratively, the AP of detection results firstly get improved but drops quickly in later iterations. **(b)** Camparison in Cascade R-CNN. Iteration 0, 1 and 2 represents the 1st, 2nd and 3rd regression stages in Cascade R-CNN. For iteration $i \geq 3$, we refine the bounding boxes with the regressor of the third stage. After multiple iteration, AP slightly drops, while the optimization-based method further improves the AP by 0.8%.

bounding boxes. We can see that the absence of localization confidence makes more than half of detected bounding boxes with IoU > 0.9 being suppressed in the traditional NMS procedure, which degrades the localization quality of the detection results.

2.2   Non-monotonic bounding box regression

In general, single object localization can be classified into two categories: bounding box-based methods and segment-based methods. The segment-based methods [19, 20, 13, 10] aim to generate a pixel-level segment for each instance but inevitably require additional segmentation annotation. This work focuses on the bounding box-based methods.

Single object localization is usually formulated as a bounding box regression task. The core idea is that a network directly learns to transform (*i.e.*, scale or shift) a bounding box to its designated target. In [9, 8] linear regression or fully-connected layer is applied to refine the localization of object proposals generated by external pre-processing modules (*e.g.*, Selective Search [28] or EdgeBoxes [33]). Faster R-CNN [23] proposes region proposal network (RPN) in which only predefined anchors are used to train an end-to-end object detector. [14, 32] utilize anchor-free, fully-convolutional networks to handle object scale variation. Meanwhile, Repulsion Loss is proposed in [29] to robustly detect objects with crowd occlusion. Due to its effectiveness and simplicity, bounding box regression has become an essential component in most CNN-based detectors.

A broad set of downstream applications such as tracking and recognition will benefit from accurately localized bounding boxes. This raises the demand for improving localization accuracy. In a series of object detectors [31, 7, 6, 21], refined boxes will be fed to the bounding box regressor again and go through the refinement for another time. This procedure is performed for several times, namely iterative bounding box regression. Faster R-CNN [23] first performs the bounding box regression twice to transform predefined anchors into final detected bounding boxes. [15] proposes a group recursive learning approach to iteratively refine detection results and minimize the offsets between object proposals and the ground-truth considering the global dependency among multiple proposals. G-CNN is proposed in [18] which starts with a multi-scale regular grid over the image and iteratively pushes the boxes in the grid towards the ground-truth. However, as reported in [3], applying bounding box regression more than twice brings no further improvement. [3] attribute this to the distribution mismatch in multi-step bounding box regression and address it by a resampling strategy in multi-stage bounding box regression.

We experimentally show the performance of iterative bounding box regression based on FPN and Cascade R-CNN frameworks. The Average Precision (AP) of the results after each iteration are shown as the blue curves in Figure 4(a) and Figure 4(b), respectively. The AP curves in Figure 4 show that the improvement on localization accuracy, as the number of iterations increase, is non-monotonic for iterative bounding box regression. The non-monotonicity, together with the non-interpretability, brings difficulties in applications. Besides, without localization confidence for detected bounding boxes, we can not have fine-grained control over the refinement, such as using an adaptive number of iterations for different bounding boxes.

3   IoU-Net

To quantitatively analyze the effectiveness of IoU prediction, we first present the methodology adopted for training an IoU predictor in Section 3.1. In Section 3.2 and Section 3.3, we show how to use IoU predictor for NMS and bounding box refinement, respectively. Finally in Section 3.4 we integrate the IoU predictor into existing object detectors such as FPN [16].
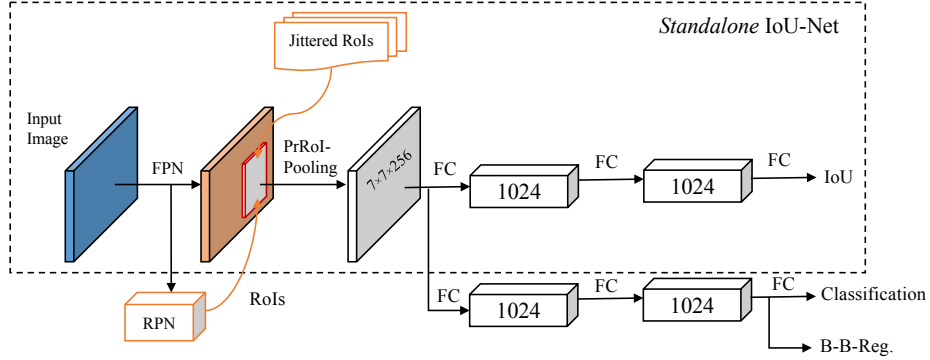
Fig. 5: Full architecture of the proposed IoU-Net described in Section 3.4. Input images are first fed into an FPN backbone. The IoU predictor takes the output features from the FPN backbone. We replace the RoI Pooling layer with a PrRoI Pooling layer described in Section 3.3. The IoU predictor shares a similar structure with the R-CNN branch. The modules marked within the dashed box form a *standalone* IoU-Net.

3.1 Learning to predict IoU

Shown in Figure 5, the IoU predictor takes visual features from the FPN and estimates the localization accuracy (IoU) for each bounding box. We generate bounding boxes and labels for training the IoU-Net by augmenting the ground-truth, instead of taking proposals from RPNs. Specifically, for all ground-truth bounding boxes in the training set, we manually transform them with a set of randomized parameters, resulting in a candidate bounding box set. We then remove from this candidate set the bounding boxes having an IoU less than $\Omega_{train} = 0.5$ with the matched ground-truth. We uniformly sample training data from this candidate set w.r.t. the IoU. This data generation process empirically brings better performance and robustness to the IoU-Net. For each bounding box, the features are extracted from the output of FPN with the proposed Precise RoI Pooling layer (see Section 3.3). The features are then fed into a two-layer feedforward network for the IoU prediction. For a better performance, we use class-aware IoU predictors.

The IoU predictor is compatible with most existing RoI-based detectors. The accuracy of a *standalone* IoU predictor can be found in Figure 2. As the training procedure is independent of specific detectors, it is robust to the change of the input distributions (*e.g.*, when cooperates with different detectors). In later sections, we will further demonstrate how this module can be jointly optimized in a full detection pipeline (*i.e.*, jointly with RPNs and R-CNN).

3.2 IoU-guided NMS

We resolve the misalignment between classification confidence and localization accuracy with a novel IoU-guided NMS procedure, where the classification confi-

---

Algorithm 1 IoU-guided NMS. Classification confidence and localization confidence are disentangled in the algorithm. We use the localization confidence (the predicted IoU) to rank all detected bounding boxes, and update the classification confidence based on a clustering-like rule.

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{S}$, $\mathcal{I}$, Ω_{nms}
    $\mathcal{B}$ is a set of detected bounding boxes.
    $\mathcal{S}$ and $\mathcal{I}$ are functions (neural networks) mapping bounding boxes to their classification confidence and IoU estimation (localization confidence) respectively.
    Ω_{nms} is the NMS threshold.
Output: $\mathcal{D}$, the set of detected bounding boxes with classification scores.
1: $\mathcal{D} \leftarrow \varnothing$
2: **while** $\mathcal{B} \neq \varnothing$ **do**
3:     $b_m \leftarrow \arg\max \mathcal{I}(b_j)$
4:     $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_m\}$
5:     $s \leftarrow \mathcal{S}(b_m)$
6:     **for** $b_j \in \mathcal{B}$ **do**
7:         **if** $\mathrm{IoU}(b_m, b_j) > \Omega_{\mathrm{nms}}$ **then**
8:             $s \leftarrow max(s, \mathcal{S}(b_j))$
9:             $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_j\}$
10:         **end if**
11:     **end for**
12:     $\mathcal{D} \leftarrow \mathcal{D} \cup \{\langle b_m, s \rangle\}$
13: **end while**
14: **return** $\mathcal{D}$

---

dence and localization confidence (an estimation of the IoU) are disentangled. In short, we use the predicted IoU instead of the classification confidence as the ranking keyword for bounding boxes. Analog to the traditional NMS, the box having the highest IoU with a ground-truth will be selected to eliminate all other boxes having an overlap greater than a given threshold Ω_{nms}. To determine the classification scores, when a box i eliminates box j, we update the classification confidence s_i of box i by $s_i = \max(s_i, s_j)$. This procedure can also be interpreted as a confidence clustering: for a group of bounding boxes matching the same ground-truth, we take the most confident prediction for the class label. A psuedo-code for this algorithm can be found in Algorithm 1.

IoU-guided NMS resolves the misalignment between classification confidence and localization accuracy. Quantitative results show that our method outperforms traditional NMS and other variants such as Soft-NMS [2]. Using IoU-guided NMS as the post-processor further pushes forward the performance of several state-of-the-art object detectors.

Algorithm 2 Optimization-based bounding box refinement

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{F}$, T, λ, Ω_1, Ω_2

    $\mathcal{B}$ is a set of detected bounding boxes, in the form of (x_0, y_0, x_1, y_1).

    $\mathcal{F}$ is the feature map of the input image.

    T is number of steps. λ is the step size, and Ω_1 is an early-stop threshold and $\Omega_2 < 0$ is an localization degeneration tolerance.

    Function PrPool extracts the feature representation for a given bounding box and function IoU denotes the estimation of IoU by the IoU-Net.

Output: The set of final detection bounding boxes.

1: $\mathcal{A} \leftarrow \varnothing$
2: **for** $i = 1$ *to* T **do**
3:     **for** $b_j \in \mathcal{B}$ and $b_j \notin \mathcal{A}$ **do**
4:        $\boldsymbol{grad} \leftarrow \nabla_{b_j} \mathrm{IoU}(\mathrm{PrPool}(\mathcal{F}, b_j))$
5:        $PrevScore \leftarrow \mathrm{IoU}(\mathrm{PrPool}(\mathcal{F}, b_j))$
6:        $b_j \leftarrow b_j + \lambda * scale(\boldsymbol{grad}, b_j)$
7:        $NewScore \leftarrow \mathrm{IoU}(\mathrm{PrPool}(\mathcal{F}, b_j))$
8:        **if** $|PrevScore - NewScore| < \Omega_1$ **or** $NewScore - PrevScore < \Omega_2$ **then**
9:          $\mathcal{A} \leftarrow \mathcal{A} \cup \{b_j\}$
10:       **end if**
11:    **end for**
12: **end for**
13: **return** $\mathcal{B}$

3.3   Bounding box refinement as an optimization procedure

The problem of bounding box refinement can formulated mathematically as finding the optimal c^* *s.t.*:

$$c^* = \arg\min_c crit(\mathbf{transform}(box_{\mathrm{det}}, c), box_{\mathrm{gt}}), \qquad (1)$$

where box_{det} is the detected bounding box, box_{gt} is a (targeting) ground-truth bounding box and **transform** is a bounding box transformation function taking c as parameter and transform the given bounding box. $crit$ is a criterion measuring the distance between two bounding boxes. In the original Fast R-CNN [5] framework, $crit$ is chosen as an smooth-L1 distance of coordinates in log-scale, while in [32], $crit$ is chosen as the $-\ln(\mathrm{IoU})$ between two bounding boxes.

Regression-based algorithms directly estimate the optimal solution c^* with a feed-forward neural network. However, iterative bounding box regression methods are vulnerable to the change in the input distribution [3] and may result in non-monotonic localization improvement, as shown in Figure 4. To tackle these issues, we propose an optimization-based bounding box refinement method utilizing IoU-Net as a robust localization accuracy (IoU) estimator. Furthermore, IoU estimator can be used as an early-stop condition to implement iterative refinement with adaptive steps.

IoU-Net directly estimates $\mathrm{IoU}(box_{\mathrm{det}}, box_{\mathrm{gt}})$. While the proposed Precise RoI Pooling layer enables the computation of the gradient of IoU w.r.t. bounding

1. RoI Pooling   **2. RoI Align**   **3. PrRoI Pooling**

$(\lfloor x_1 \rfloor, \lfloor y_1 \rfloor)$

(x_1, y_1)

(x_2, y_2)

$(\lceil x_2 \rceil, \lceil y_2 \rceil)$

(x_1, y_1)

(x_2, y_2)

(a_i, b_i)

(x_1, y_1)

(x_2, y_2)

$$\frac{\sum_{i=\lfloor x_1 \rfloor}^{\lceil x_2 \rceil} \sum_{j=\lfloor y_1 \rfloor}^{\lceil y_2 \rceil} w_{i,j}}{(\lceil x_2 \rceil - \lfloor x_1 \rfloor + 1) \times (\lceil y_2 \rceil - \lfloor y_1 \rfloor + 1)} \qquad \sum_{i=1}^{N} f(a_i, b_i)/N \qquad \frac{\int_{y_1}^{y_2} \int_{x_1}^{x_2} f(x, y)\, dx dy}{(x_2 - x_1) \times (y_2 - y_1)}$$
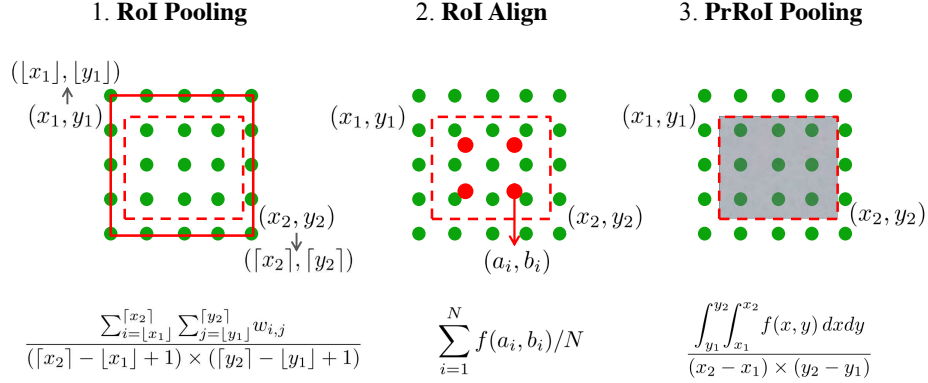
Fig. 6: Illustration of RoI Pooling, RoI Align and PrRoI Pooling.

box coordinates[§], we can directly use gradient ascent method to find the optimal solution to Equation 1. Shown in Algorithm 2, viewing the estimation of the IoU as an optimization objective, we iteratively refine the bounding box coordinates with the computed gradient and maximize the IoU between the detected bounding box and its matched ground-truth. Besides, the predicted IoU is an interpretable indicator of the localization confidence on each bounding box and helps explain the performed transformation.

In the implementation, shown in Algorithm 2 Line 6, we manually scale up the gradient w.r.t. the coordinates with the size of the bounding box on that axis (*e.g.*, we scale up ∇x_1 with $width(b_j)$). This is equivalent to perform the optimization in log-scaled coordinates $(x/w, y/h, \log w, \log h)$ as in [5]. We also employ a one-step bounding box regression for an initialization of the coordinates.

Precise RoI Pooling. We introduce Precise RoI Pooling (PrRoI Pooling, for short) powering our bounding box refinement[*]. It avoids any quantization of coordinates and has a continuous gradient on bounding box coordinates. Given the feature map $\mathcal{F}$ before RoI/PrRoI Pooling (*e.g.* from Conv4 in ResNet-50), let $w_{i,j}$ be the feature at one discrete location (i, j) on the feature map. Using bilinear interpolation, the discrete feature map can be considered continuous at any continuous coordinates (x, y):

$$f(x, y) = \sum_{i,j} IC(x, y, i, j) \times w_{i,j}, \qquad (2)$$

where $IC(x, y, i, j) = max(0, 1 - |x - i|) \times max(0, 1 - |y - j|)$ is the interpolation coefficient. Then denote a bin of a RoI as $bin = \{(x_1, y_1), (x_2, y_2)\}$, where (x_1, y_1) and (x_2, y_2) are the continuous coordinates of the top-left and bottom-right

---

[§] We prefer Precise RoI-Pooling layer to RoI-Align layer [10] as Precise RoI-Pooling layer is continuously differentiable w.r.t. the coordinates while RoI-Align is not.

[*] The code is released at: https://github.com/vacancy/PreciseRoIPooling

points, respectively. We perform pooling (*e.g.*, average pooling) given *bin* and feature map $\mathcal{F}$ by computing a two-order integral:

$$\mathrm{PrPool}(bin, \mathcal{F}) = \frac{\int_{y1}^{y2}\int_{x1}^{x2} f(x,y)\, dxdy}{(x_2 - x_1) \times (y_2 - y_1)}. \tag{3}$$

For a better understanding, we visualize RoI Pooling, RoI Align [10] and our PrRoI Pooing in Figure 6: in the traditional RoI Pooling, the continuous coordinates need to be quantized first to calculate the sum of the activations in the bin; to eliminate the quantization error, in RoI Align, $N = 4$ continuous points are sampled in the bin, denoted as (a_i, b_i), and the pooling is performed over the sampled points. Contrary to RoI Align where N is pre-defined and not adaptive w.r.t. the size of the bin, the proposed PrRoI pooling directly compute the two-order integral based on the continuous feature map.

Moreover, based on the formulation in Equation 3, $\mathrm{PrPool}(Bin, \mathcal{F})$ is differentiable w.r.t. the coordinates of *bin*. For example, the partial derivative of $\mathrm{PrPool}(B, \mathcal{F})$ w.r.t. x_1 could be computed as:

$$\frac{\partial \mathrm{PrPool}(bin, \mathcal{F})}{\partial x_1} = \frac{\mathrm{PrPool}(bin, \mathcal{F})}{x_2 - x_1} - \frac{\int_{y1}^{y2} f(x_1, y)\, dy}{(x_2 - x_1) \times (y_2 - y_1)}. \tag{4}$$

The partial derivative of $\mathrm{PrPool}(bin, \mathcal{F})$ w.r.t. other coordinates can be computed in the same manner. Since we avoids any quantization, PrPool is continuously differentiable.

3.4   Joint training

The IoU predictor can be integrated into standard FPN pipelines for end-to-end training and inference. For clarity, we denote *backbone* as the CNN architecture for image feature extraction and *head* as the modules applied to individual RoIs.

Shown in Figure 5, the IoU-Net uses ResNet-FPN [16] as the backbone, which has a top-down architecture to build a feature pyramid. FPN extracts features of RoIs from different levels of the feature pyramid according to their scale. The original RoI Pooling layer is replaced by the Precise RoI Pooling layer. As for the network head, the IoU predictor works in parallel with the R-CNN branch (including classification and bounding box regression) based on the same visual feature from the backbone.

We initialize weights from pre-trained ResNet models on ImageNet [25]. All new layers are initialized with a zero-mean Gaussian with standard deviation 0.01 or 0.001. We use smooth-L1 loss for training the IoU predictor. The training data for the IoU predictor is separately generated as described in Section 3.1 within images in a training batch. IoU labels are normalized s.t. the values are distributed over $[-1, 1]$.

Input images are resized to have 800px along the short axis and a maximum of 1200px along the long axis. The classification and regression branch take 512

RoIs per image from RPNs. We use a batch size 16 for the training. The network is optimized for 160k iterations, with a learning rate of 0.01 which is decreased by a factor of 10 after 120k iterations. We also warm up the training by setting the learning rate to 0.004 for the first 10k iteration. We use a weight decay of 1e-4 and a momentum of 0.9.

During inference, we first apply bounding box regression for the initial coordinates. To speed up the inference, we first apply IoU-guided NMS on all detected bounding boxes. 100 bounding boxes with highest classification confidence are further refined using the optimization-based algorithm. We set $\lambda = 0.5$ as the step size, $\Omega_1 = 0.001$ as the early-stop threshold, $\Omega_2 = -0.01$ as the localization degeneration tolerance and $T = 5$ as the number of iterations.

4   Experiments

We perform experiments on the 80-category MS-COCO detection dataset [17]. Following [1, 16], the models are trained on the union of 80k training images and 35k validation images (*trainval35k*) and evaluated on a set of 5k validation images (*minival*). To validate the proposed methods, in both Section 4.1 and 4.2, a *standalone* IoU-Net (without R-CNN modules) is trained separately with the object detectors. IoU-guided NMS and optimization-based bounding box refinement, powered by the IoU-Net, are applied to the detection results.

4.1   IoU-guided NMS

Table 1 summarizes the performance of different NMS methods. While Soft-NMS preserve more bounding boxes (there is no real "suppression"), IoU-guided NMS improves the results by improving the localization of the detected bounding boxes. As a result, IoU-guided NMS performs significantly better than the baselines on high IoU metrics (*e.g.*, AP_{90}).

We delve deeper into the behavior of different NMS algorithms by analyzing their recalls at different IoU threshold. The raw detected bounding boxes are generated by a ResNet50-FPN without any NMS. As the requirement of localization accuracy increases, the performance gap between IoU-guided NMS and other methods goes larger. In particular, the recall at matching IoU $\Omega_{\text{test}} = 0.9$ drops to 18.7% after traditional NMS, while the IoU-NMS reaches 28.9% and the No-NMS "upper bound" is 39.7%.

4.2   Optimization-based bounding box refinement

The proposed optimization-based bounding box refinement is compatible with most of the CNN-based object detectors [16, 3, 10], as shown in Table 2. Applying the bounding box refinement after the original pipelines with the *standalone* IoU-Net further improve the performance by localizing object more accurately. The refinement further improves AP_{90} by 2.8% and the overall AP by 0.8% even for Cascade R-CNN which has a three-stage bounding box regressor.

Method	+Soft-NMS	+IoU-NMS	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
FPN			36.4	**58.0**	**53.1**	44.9	31.2	9.8
	✓		36.8	57.5	**53.1**	**45.7**	32.3	10.3
		✓	**37.3**	56.0	52.2	45.6	**33.9**	**13.3**
Cascade R-CNN			40.6	**59.3**	55.2	49.1	38.7	16.7
	✓		**40.9**	58.2	**54.7**	**49.4**	**39.9**	17.8
		✓	40.7	58.0	**54.7**	49.2	38.8	**18.9**
Mask-RCNN			37.5	**58.6**	**53.9**	46.3	33.2	10.9
	✓		37.9	58.2	**53.9**	**47.1**	34.4	11.5
		✓	**38.1**	56.4	52.7	46.7	**35.1**	**14.6**

Table 1: Comparison of IoU-guided NMS with other NMS methods. By preserving bounding boxes with accurate localization, IoU-guided NMS shows significant improvement in AP with high matching IoU threshold (*e.g.*, AP_{90}).
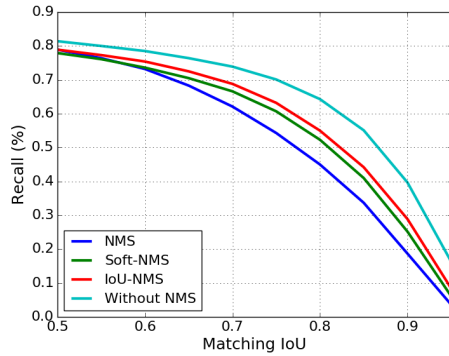

Fig. 7: Recall curves of different NMS methods at different IoU threshold for matching detected bounding boxes with the ground-truth. No-NMS (no box is suppressed) is provided as the upper bound of the recall. The proposed IoU-NMS has a higher recall and effectively narrows the gap to the upper-bound at high IoU threshold (*e.g.*, 0.9).

Method	+Refinement	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
FPN		36.4	**58.0**	**53.1**	44.9	31.2	9.8
	✓	**38.0**	57.7	**53.1**	**46.1**	**34.3**	**14.6**
Cascade R-CNN		40.6	**59.3**	55.2	49.1	38.7	16.7
	✓	**41.4**	**59.3**	**55.3**	**49.6**	**39.4**	**19.5**
Mask-RCNN		37.5	**58.6**	**53.9**	46.3	33.2	10.9
	✓	**39.2**	57.9	53.6	**47.4**	**36.5**	**16.4**

Table 2: The optimization-based bounding box refinement further improves the performance of several CNN-based object detectors.

4.3   Joint training

IoU-Net can be end-to-end optimized in parallel with object detection frameworks. We find that adding IoU predictor to the network helps the network to learn more discriminative features which improves the overall AP by 0.6 and 0.4 percent for ResNet50-FPN and ResNet101-FPN respectively. The IoU-guided NMS and bounding box refinement further push the performance forward. We achieve 40.6% AP with ResNet101-FPN compared to the baseline 38.5% (improved by

Backbone	Method	+IoU-NMS	+Refine	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
ResNet-50	FPN			36.4	58.0	53.1	44.9	31.2	9.8
	IoU-Net			37.0	**58.3**	**53.8**	45.7	31.9	10.7
		✓		37.6	56.2	52.4	46.0	34.1	14.0
		✓	✓	**38.1**	56.3	52.4	**46.3**	**35.1**	**15.5**
ResNet-101	FPN			38.5	**60.3**	**55.5**	47.6	33.8	11.3
	IoU-Net			38.9	60.2	**55.5**	47.8	34.6	12.0
		✓		40.0	59.0	55.1	48.6	37.0	15.5
		✓	✓	**40.6**	59.0	55.2	**49.0**	**38.0**	**17.1**

Table 3: Final experiment results on MS-COCO. IoU-Net denotes ResNet-FPN embedded with IoU predictor. We improve the FPN baseline by $\approx 2\%$ in AP.

Method	FPN	Mask-RCNN	Cascade R-CNN	IoU-Net
Speed (sec./image)	0.255	0.267	0.384	0.305

Table 4: Inference speed of multiple object detectors on a single TITAN X GPU. The models share the same backbone network ResNet50-FPN. The input resolution is 1200x800. All hyper-parameters are set to be the same.

2.1%). The inference speed is demonstrated in Table 3, showing that IoU-Net improves the detection performance with tolerable computation overhead.

We mainly attribute the inferior results on AP_{50} in Table 3 to the IoU estimation error. When the bounding boxes have a lower IoU with the ground-truth, they have a larger variance in appearance. Visualized in Figure 2(b), the IoU estimation becomes less accurate for boxes with lower IoU. This degenerates the performance of the downstream refinement and suppression. We empirically find that this problem can be partially solved by techniques such as sampling more bounding boxes with lower IoU during the training.

5   Conclusion

In this paper, a novel network architecture, namely IoU-Net, is proposed for accurate object localization. By learning to predict the IoU with matched ground-truth, IoU-Net acquires "localization confidence" for the detected bounding box. This empowers an IoU-guided NMS procedure where accurately localized bounding boxes are prevented from being suppressed. The proposed IoU-Net is intuitive and can be easily integrated into a broad set of detection models to improve their localization accuracy. Experimental results on MS-COCO demonstrate its effectiveness and potential in practical applications.

This paper points out the misalignment of classification and localization confidences in modern detection pipelines. We also formulate an novel optimization view on the problem of bounding box refinement, and the proposed solution surpasses the regression-based methods. We hope these novel viewpoints provide insights to future works on object detection, and beyond.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2874–2883 (2016)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Improving object detection with one line of code. arXiv preprint arXiv:1704.04503 (2017)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. arXiv preprint arXiv:1712.00726 (2017)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
5. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1532–1545 (2014)
6. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1134–1142 (2015)
7. Gidaris, S., Komodakis, N.: Attend refine repeat: Active box proposal generation via in-out localization. arXiv preprint arXiv:1606.04446 (2016)
8. Girshick, R.: Fast r-cnn. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
11. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. arXiv preprint (2017)
12. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. arXiv preprint arXiv:1711.11575 (2017)
13. Hu, H., Lan, S., Jiang, Y., Cao, Z., Sha, F.: Fastmask: Segment multi-scale object candidates in one shot. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 991–999 (2017)
14. Huang, L., Yang, Y., Deng, Y., Yu, Y.: Densebox: Unifying landmark localization with end to end object detection. arXiv preprint arXiv:1509.04874 (2015)
15. Li, J., Liang, X., Li, J., Wei, Y., Xu, T., Feng, J., Yan, S.: Multi-stage object detection with group recursive learning. IEEE Transactions on Multimedia (2017)
16. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
18. Najibi, M., Rastegari, M., Davis, L.S.: G-cnn: an iterative grid based object detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2369–2377 (2016)
19. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. pp. 1990–1998 (2015)

20. Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: European Conference on Computer Vision. pp. 75–91. Springer (2016)
21. Rajaram, R.N., Ohn-Bar, E., Trivedi, M.M.: Refinenet: Iterative refinement for accurate object localization. In: Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. pp. 1528–1533. IEEE (2016)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf
24. Rothe, R., Guillaumin, M., Van Gool, L.: Non-maximum suppression for object detection by passing messages between windows. In: Asian Conference on Computer Vision. pp. 290–306. Springer (2014)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
26. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1708 (2014)
27. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1653–1660 (2014)
28. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**(2), 154–171 (2013)
29. Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C.: Repulsion loss: Detecting pedestrians in a crowd. arXiv preprint arXiv:1711.07752 (2017)
30. Wu, J., Lu, E., Kohli, P., Freeman, W.T., Tenenbaum, J.B.: Learning to see physics via visual de-animation. In: Advances in Neural Information Processing Systems (2017)
31. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Craft objects from images. arXiv preprint arXiv:1604.03239 (2016)
32. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 516–520. ACM (2016)
33. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision. pp. 391–405. Springer (2014)