Localization Uncertainty Estimation for Anchor-Free Object Detection

Youngwan Lee Joong-Won Hwang Hyung-Il Kim Kimin Yun Yongjin Kwon
Electronics and Telecommunications Research Institute (ETRI)
South Korea
{yw.lee, jwhwang, hikim, kimin.yun, scocso}@etri.re.kr

Abstract

Since many safety-critical systems, such as surgical robots and autonomous driving cars, are in unstable environments with sensor noise and incomplete data, it is desirable for object detectors to take into account the confidence of localization prediction. There are three limitations of the prior uncertainty estimation methods for anchor-based object detection. 1) They model the uncertainty based on object properties having different characteristics, such as location (center point) and scale (width, height). 2) they model a box offset and ground-truth as Gaussian distribution and Dirac delta distribution, which leads to the model misspecification problem. Because the Dirac delta distribution is not exactly represented as Gaussian, i.e., for any μ and Σ. 3) Since anchor-based methods are sensitive to hyper-parameters of anchor, the localization uncertainty modeling is also sensitive to these parameters. Therefore, we propose a new localization uncertainty estimation method called Gaussian-FCOS for anchor-free object detection. Our method captures the uncertainty based on four directions of box offsets (left, right, top, bottom) that have similar properties, which enables to capture which direction is uncertain and provide a quantitative value in range [0, 1]. To this end, we design a new uncertainty loss, negative power log-likelihood loss, to measure uncertainty by weighting IoU to the likelihood loss, which alleviates the model misspecification problem. Experiments on COCO datasets demonstrate that our Gaussian-FCOS reduces false positives and finds more missing-objects by mitigating over-confidence scores with the estimated uncertainty. We hope Gaussian-FCOS serves as a crucial component for the reliability-required task.

1. Introduction

Object detection based on CNN is widely used in many automated systems such as autonomous vehicles and surgical robots [24]. In such a safety-related system, it is very important to know how reliable the estimated output is as well as good performance. Object detection is a task that combines object localization and classification, however, most of the state-of-the-art methods [1, 26, 29] provide the reliability of their algorithm as a single value (*e.g.*, confidence) for each bounding box. That is, they only use the classification score as the detection quality without the localization uncertainty. As a consequence, these methods produce mislocalized detection boxes with over-confidence [9]. For example, as shown in Fig. 2(a), the mislocalized detection with the confidence score of 50% (green box) is not removed because its classification confidence score is higher than the threshold. Therefore, in addition to the classification confidence score, the confidence of the bounding box's localization is also necessary for the detection certainty.

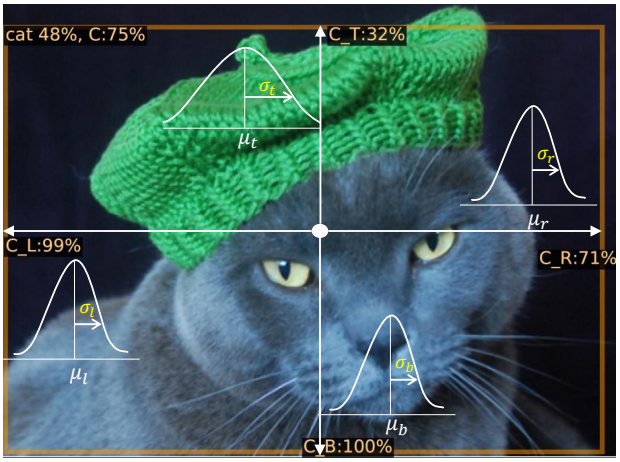
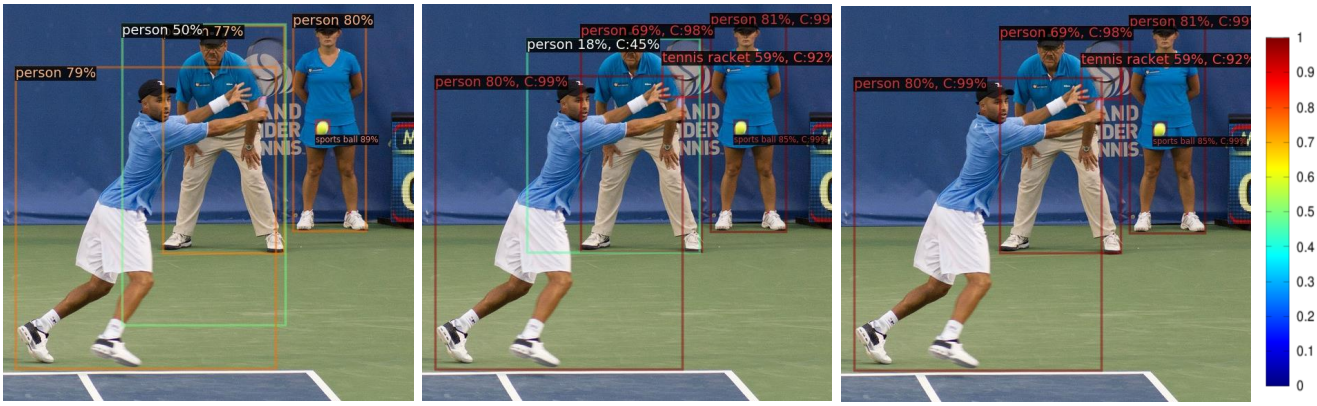


Figure 1: **Examples of 4-directions uncertainty for anchor-free object detection.** C_L, C_R, C_T, and C_B denote the estimated certainty in [0, 1] value with respect to left, right, top, and bottom. For example, Gaussian-FCOS estimates lower top-direction certainty due to its ambiguous head boundary of the cat wearing a hat. This demonstrates that our method makes it possible to quantify which direction is uncertain due to unclear or obvious objects.

Recently, efforts [13, 9, 16, 6] to estimate uncertainty for



(a) FCOS with conf_th:05 **(b) Gaussian-FCOS with conf_th:01** **(c) Gaussian-FCOS with conf_th:05**

Figure 2: **Example of over-confidence problem resolved by Gaussian-FCOS (proposed).** The colors of bounding boxes are chosen by its detection score (a) and certainty score (b), (c). Specifically, the higher score (up to 1.0) is, the box color is to be red otherwise blue, where 'conf_th' denotes confidence threshold for the visualization. 'C' in (b), (c) denotes the estimated certainty score. Gaussian-FCOS captures localization uncertainty for each bounding box which means how certain box location is. For example, each detected person has a higher certainty score over 90% whereas the false positive (cyan box) due to the overlap between persons gets a lower certainty score (45%). Unlike FCOS (a), Gaussian-FCOS (c) filters out the false positive by decaying the detection score with the estimated certainty score. compared to FCOS, Gaussian-FCOS localizes the person with a tennis racket more accurately (tightly).

object detection have been attempted. All of these efforts model the uncertainty of location (center point) and scale (width, height) as Gaussian distribution in the anchor-based methods by adding four channels in the regression output. However, since center point, width, and height have semantically different characteristics [13], this approach considering each value equally is inadequate for modeling localization uncertainty. For example, the estimated distributions of center point and scale shows different shapes in [13]. Besides, since anchor-based methods are sensitive to hyper-parameters of anchor, the localization uncertainty modeling on the anchor-based methods is also sensitive to these parameters.

Recently, anchor-free methods [15, 3, 32, 31, 26] that do not need the heuristic anchor-box tuning (e.g., scale, aspect ratio) surpassed conventional anchor-box based methods such as Faster R-CNN [23], RetinaNet [19], and their variants [1, 30]. As a representative anchor-free method, FCOS [26] adopts the concept of centerness to filter out false positive boxes. That is, the centerness can be interpreted as the implicit localization uncertainty of a proposal box. However, FCOS [26] heuristically measures the localization uncertainty by how well the predicted box fits the center, which does not reflect full information for localization uncertainty of box (e.g., scale).

In terms of the loss function for uncertainty modeling, the conventional methods [13, 16, 6] use the negative log-likelihood loss to regress output as Gaussian distribution. He *et al*. [9] introduce KL divergence loss for Gaussian distribution of box prediction and Dirac delta function for a ground-truth box. In the perspective of cross-entropy, however, these methods face the model misspecification problem [10] in that the Dirac delta function is not exactly represented as Gaussian distribution, *i.e.*, for any μ and Σ, $\delta(x) \neq N(x|\mu, \Sigma)$.

To deal with these limitations, in this paper, we propose a method, called *Gaussian-FCOS*, that estimates *explicitly* localization uncertainty for anchor-free method, FCOS [26]. Unlike centerness in FCOS, we model the uncertainty for each of the four box offsets (left, right, top, bottom) from the center of the box to fully describe the localization uncertainty. Moreover, unlike conventional anchor-based methods [13, 9, 16, 6] for localization uncertainty, our method estimates the uncertainty of the four box offsets having a similar semantic characteristic. It makes possible to inform which direction of a box boundary is uncertain as a quantitative value in [0,1] independently from the overall box uncertainty as shown in Fig. 1. (more examples are illustrated in Fig. 5.) To do this, we model the box offset and its uncertainty of FCOS through Gaussian distribution with the newly desgined uncertainty loss by adding the uncertainty branch.

To resolve the model misspecification [10] between Dirac delta and Gaussian distribution, we design a novel uncertainty loss, *negative power log-likelihood loss*, inspired by Power likelihood [10] (NPLL), to enable the uncertainty branch to learn to estimate localization uncertainty by weighing IoU to the log-likelihood loss. In particular,

this new loss creates a synergy with the existing box regression loss, which tells the difference between the ground truth and the predicted box offset, enabling more accurate box prediction. For example, as illustrated in Fig. 2(a) and (c), we can see that Gaussian-FCOS localizes objects more accurately (tightly) than FCOS by comparing the detected box of the person with a tennis racket. Furthermore, we calibrate the detection score through the localization uncertainty. In detail, we compute certainty from the estimated uncertainty and then multiply it to the classification score to get the final detection score. Fig. 2(b)-(c) shows the effectiveness of uncertainty calibration. Unlike the detection result from FCOS in Fig. 2(a), Gaussian-FCOS calibrates (or penalizes) the detection score with the certainty, which can filter out the mislocalized box (cyan box) between two persons.

The main contributions are summarized as below:

- We propose a simple and effective four-directions localization uncertainty estimation for anchor-free object detection that can serve as a detection quality measure and provide which direction is uncertain as a quantitative value in [0, 1].
- We newly design the uncertainty loss function, inspired by *power likelihood*, that has IoUs as weights of negative log-likelihood loss that resolves the model misspecification problem.
- We analyze the influence of uncertainty on object localization and confirm that it improves mislocalization and reduces missing objects on challenging COCO dataset.

2. Related Works

2.1. Anchor-Free Object Detection

Recently, anchor-free object detectors [15, 32, 3, 31, 26] have attracted attention beyond anchor-based methods [23, 19, 1, 30] that need to tune sensitive hyper-parameters related to anchor box (e.g., scale, aspect ratio, etc). CornerNet [15] predicts an object location as a pair of keypoints (top-left and bottom-right). CenterNet [3] extends CornerNet as a triplet instead of a pair of key points to boost performance. ExtremeNet [32] locates four extreme points (top, bottom, left, right) and one center point to generate the object box. Zhu *et al*. [31] utilizes keypoint estimation to predict center point objects and regresses to other attributes including size, orientation, pose, and 3D location. FCOS [26] views all points inside the ground-truth box as positive samples and regresses four distances (left, right, top, bottom) from the object boundary. We propose to endow FCOS with localization uncertainty due to its simplicity and performance.

2.2. Uncertainty Estimation

Uncertainty in deep neural networks can be estimated in two types [4, 12, 16]: epistemic (sampling-based) and aleatoric (sampling-free) uncertainty. Epistemic uncertainty measures the model uncertainty in the models' parameters through Bayesian neural networks [25], Monte Carlo dropout [5], and Bootstrap Ensemble [14]. As they need to be re-evaluated several times and store several sets of weights for each network, it is hard to apply them for real-time applications. Aleatoric uncertainty is data and problem inherent such as sensor noise and ambiguities in data. It can be estimated by explicitly modeling it as model output.

Recent works [9, 16, 6, 14] have adopted uncertainty estimation for object detection. Lakshminarayanan *et al*. [14] and Harakeh *et al*. [6] use Monte Carlo dropout in Epistemic based methods. As described above, since epistemic uncertainty needs to inference several times, it is not suitable for real-time object detection. Le *et al*. [16] and Choi *et al*. [2] are aleatoric based methods and jointly estimate the uncertainties of four parameters of bounding box from SSD [21] and YOLOv3 [22]. He [9] estimates the uncertainty of bounding box by minimizing the KL-divergence loss for Gaussian distribution of predicted box and Dirac delta distribution of ground-truth box on the Faster R-CNN [23] (anchor-based method). From the cross-entropy perspective, however, Dirac delta distribution cannot be represented Gaussian distribution, which results in a misspecification problem [10]. To overcome this problem, we adopt the power likelihood concept [10] to the Gaussian log-likelihood loss. The latest concurrent work is Generalized Focal loss (GFocal) [18] that represents jointly localization quality and classification and model bounding box as arbitrary distribution. The distinct difference from GFocal [18] is that our method estimates 4-directions uncertainties as quantitative values in the range [0, 1] thus these estimated values can be used as an informative cue for decision-making.

3. Proposed Method

For the uncertainty estimation for object detector, we choose anchor-free detector, FCOS [26] based on two reasons: 1) **Simplicity**. FCOS directly regresses the target bounding boxes in a pixel-wise prediction manner without heuristic anchor tuning (aspect ratio, scales, etc). 2) **Semantic symmetry of regression**. anchor-based methods regress center point (x,y), width, and height based on each anchor box, while FCOS directly regresses four boundaries (left, right, top, bottom) of a bounding box at each location. Although the center, width, and height from anchor-based methods have different characteristics, the distances between four boundaries and each location are semantically symmetric. In terms of modeling, it is easier to model the

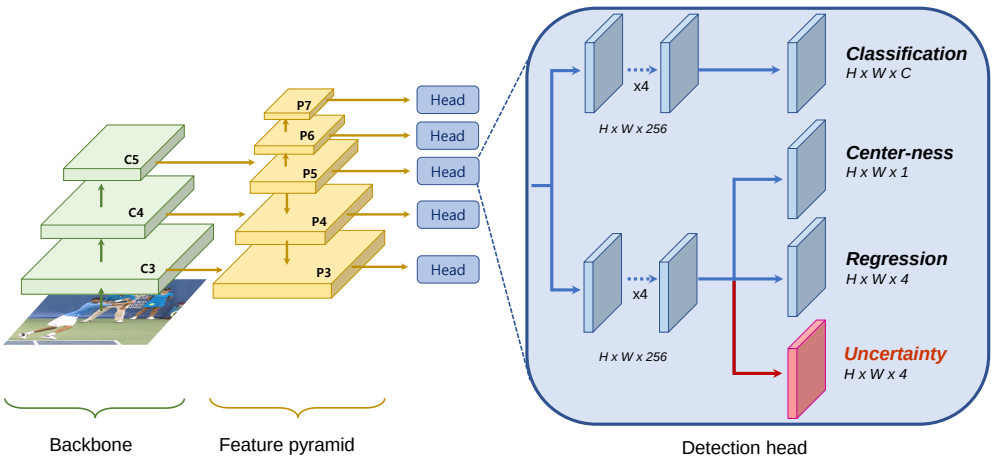


Figure 3: **Architecture of Gaussian-FCOS.** Different from FCOS [26], Gaussian-FCOS estimates localization uncertainty from *uncertainty* branch that outputs four uncertainties of box offsets (left, right, top, and bottom.)

values sharing semantic meanings that have similar properties. Furthermore, it enables to notify which direction of a box boundary is uncertain separately from the overall box uncertainty.

In this section, we first introduce the localization step of FCOS [26]. Then, we present *uncertainty loss* for modeling the uncertainty of the object coordinates in FCOS as the Gaussian parameters (i.e., the mean and variance). Next, we introduce the new uncertainty branch that estimates the localization uncertainty in addition to FCOS branches. Finally, we show how the estimated localization uncertainty is applied to provide localization confidence.

3.1. FCOS detector

In FCOS [26], if the location belongs to the ground-truth box area, it is regarded as a positive sample and as a negative sample otherwise. On each location (x, y), the box offsets are regressed as 4D vector $B_{x,y} = [l, r, t, b]^\top$ that is the distances from the location to four sides of the bounding box (i.e., left, right, top, and bottom). The regression targets $B^g_{x,y} = [l^g, r^g, t^g, b^g]^\top$ are computed as,

$$l^g = x - x^g_{lt},\ r^g = x^g_{rb} - x,\ t^g = y - y^g_{lt},\ b^g = y^g_{rb} - y, \quad (1)$$

where (x^g_{lt}, y^g_{lt}) and (x^g_{rb}, y^g_{rb}) denote the coordinates of the left-top and right-bottom corners of the ground-truth box, respectively. Then, for all locations of positive samples, the IoU loss [28] is measured between the predicted $B_{x,y}$ and the ground truth $B^g_{x,y}$ for the regression loss.

FCOS also adopts centerness to suppress low quality detected boxes in the inference stage. The main concept of centerness is to estimate not only the position of the object but also how well it fits with the center. That is, it provides confidence that the box predicted in the test step fits the center of the object well. Also, this centerness value is used as uncertainty to penalize the detection score.

3.2. Gaussian-FCOS

FCOS [26] predicts the class score, box offsets B=(l, r, t, b), and centerness. In FCOS, centerness can be regarded as implicit uncertainty because centerness is used to filter predicted boxes, but centerness alone is insufficient to measure localization uncertainty. In other words, centerness is a single value that simply shows the localization uncertainty as to how well the center of the box fits, but for accurate modeling of localization uncertainty, it is necessary to consider the four positions that make up the box. Therefore, we propose Gaussian-FCOS that estimates localization uncertainty of the box, based on the regressed box offsets (l, r, t, b). To predict the uncertainties of four box offsets, we model the box offsets through Gaussian distribution and train the network to estimate its uncertainty (standard deviation). Assuming each instance of box offsets is independent, we use multivariate Gaussian distribution of output B^* with diagonal covariance matrix $\mathbf{\Sigma}_B$ to model each box offset B:

$$P_\Theta(B^*|B) = \mathcal{N}(B^*; \boldsymbol{\mu}_B, \mathbf{\Sigma}_B), \quad (2)$$

where Θ is the learnable network parameters, and d is the dimension of B (i.e., $d = 4$). $\boldsymbol{\mu}_B = [\mu_l, \mu_r, \mu_t, \mu_b]^\top$ and $\mathbf{\Sigma}_B = \mathrm{diag}(\sigma_l^2, \sigma_r^2, \sigma_t^2, \sigma_b^2)$ denote the predicted box offset and its *uncertainty*, respectively.

Power likelihood. Prior works [13, 16, 9, 2] also model box offset and ground-truth box as Gaussian distribution and Dirac delta distribution. [13, 16, 2] adopt negative log-likelihood loss (NLL) and [9] use KL-divergence loss (KL-Loss). In cross-entropy perspective, minimizing NLL and KL-loss is equivalent as below:

$$\mathcal{L} = -\frac{1}{N}\sum P_D(x) \cdot \log P_\Theta(x), \quad (3)$$

where P_D and P_Θ are Dirac delta function and Gaussian probability density function, respectively. When the box offset is located in a ground-truth box, the P_D is 1 then Eq. 3 becomes negative log-likelihood loss. However, there is a significant problem that Dirac delta distribution does not belong to the family of Gaussian distributions, called the model misspecification problem [10]. In a number of statistical literature, to estimate parameters of interest in a robust way when the model is misspecified, the Power likelihood ($P_\Theta(\cdot)^w$), which raises the likelihood ($P_\Theta(\cdot)$) to a power (w) that controls how influential the data is, has been proposed [10]. Thus, to fill the gaps between Dirac delta distribution and Gaussian distribution, inspired by Power likelihood, we introduce a novel uncertainty loss, *negative power log-likelihood loss (NPLL)*, that exploits Intersection-over-Union (IoU) as the power since the offset that has higher IoU should be more influential. By multiplying IoU term to the log-likelihood, the new uncertainty loss is defined as :

$$\mathcal{L}_u = -\frac{\lambda}{N_{\text{pos}}} \sum_i \sum_k IoU_i \cdot \log P_\Theta \left(B^g_{i,k} | \mu_k, \sigma^2_k \right) \quad (4)$$

$$= \frac{\lambda}{N_{\text{pos}}} \sum_i IoU_i \times \left[\sum_k \left\{ \frac{(B^g_{i,k} - \mu_k)^2}{2\sigma^2_k} + \frac{1}{2} \log \sigma^2_k \right\} + 2\log 2\pi \right], \quad (5)$$

where N_{pos} denotes the number of positive samples, λ ($\lambda = 0.05$ in this paper) is the balance weight for $\mathcal{L}_u$, IoU_i is the intersection-over-union between the predicted box and the ground-truth box at location i and k is in $\{l, r, t, b\}$. The summation is calculated over all positive locations on the feature maps. From this uncertainty loss, when the predicted coordinate μ_k from the regression branch is inaccurate, the network is trained to estimate larger uncertainty σ_k. For the rest of the losses, following FCOS [26], we use focal loss [19] for classification ($\mathcal{L}_c$), binary cross-entropy loss for centerness ($\mathcal{L}_{ct}$), and IoU loss [28] for regression ($\mathcal{L}_b$). The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_{ct} + \mathcal{L}_b + \mathcal{L}_u. \quad (6)$$

It is noted that unlike centerness, our network is trained to directly estimate four localization uncertainties $(\sigma_l, \sigma_r, \sigma_t, \sigma_b)$ of each box offsets. Also, it can be estimated which direction of a box boundary is uncertain separately apart from the overall box uncertainty.

Uncertainty branch. To implement our idea, we redesign the FCOS [26] network structure by adding the uncertainty branch as shown in Fig. 3. Our network predicts a probability distribution instead of only box coordinates. The mean values μ_k of each box offsets are predicted from the regression branch in FCOS. The new uncertainty branch with sigmoid function outputs four uncertainty values σ_k in $[0, 1]$.

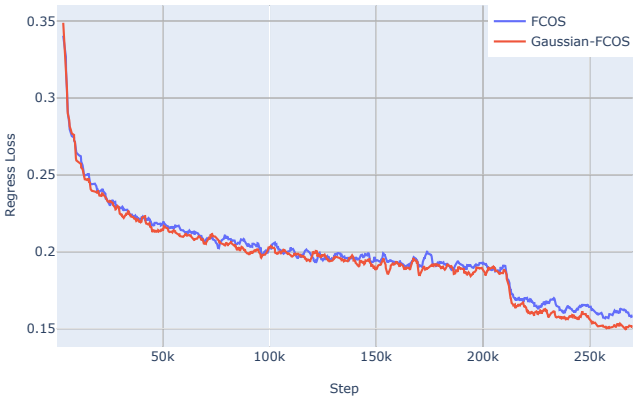


Figure 4: **Comparison of the regression Loss.** Regression loss of Gaussian-FCOS tends to be lower than that of FCOS. In other words, training with the proposed uncertainty loss further reduces the regression loss and helps the trained network estimate better object position.

The regression branch and the uncertainty branch shares same feature (4 `conv` layers) as their inputs to estimate two kinds of statistic parameters (i.e. μ_k, σ_k).

Uncertainty calibration. With the uncertainty branch, Gaussian-FCOS can obtain the localization uncertainty σ_k and utilize it to infer the box confidence. Concretely, box confidence is interpreted as the certainty that is defined as $(1 - \sigma)$, where the σ is obtained by averaging σ_k for all $k \in \{l, r, t, b\}$. In the post-processing step, Non-Maximum-Supression (NMS) is applied for removing overlapped box proposals. The class confidence score is widely adopted to decide the boxes to be removed, but this does not reflect localization uncertainty. Thus, for NMS, we calibrate the detection score by multiplying the class score by the our box confidence $(1 - \sigma)$ that considers the localization uncertainty. Fig. 2(b) shows that the false positive (cyan box) has lower box confidence (45%) than other true positives (over 90%), which means Gaussian-FCOS estimates localization uncertainty well. As shown in Fig. 2(c), Gaussian-FCOS removes the false positive through the calibrated (penalized) the detection score.

4. Experiments

In this section, we evaluate the effectiveness of Gaussian-FCOS on the challenging COCO [20] dataset which has 80 object categories. We use the COCO `train2017` set (80k images) for training and `val2017` set (5k images) for ablation studies. Final results are evaluated on `test-dev2017` in the evaluation server for the comparison with state-of-the-arts. There are two key metrics for object detection evaluation, the one is average precision (AP) and the other is average recall (AR). AP reflects how boxes are proposed properly without duplication and

Method	AP	AP_{75}	AR	AR_{75}
FCOS	41.2	44.4	59.0	63.2
$+\mathcal{L}_u$	$41.5_{+0.3}$	$45.4_{+1.0}$	$60.2_{+1.2}$	$66.6_{+3.4}$
$+\mathcal{L}_u$ w/ IoU power	$42.0_{+0.8}$	$46.2_{+1.8}$	$60.8_{+1.8}$	$67.0_{+3.8}$

Table 1: **Effectiveness of power likelihood.** $\mathcal{L}_u$ denotes negative log-likelihood loss (NLL). The proposed uncertainty loss with IoU power term (NPLL) improves the baseline.

Method	AP	AP_{75}	AR	AR_{75}
w/o Calibration	41.7	45.1	59.4	63.7
Calib. after NMS	41.5	45.2	59.4	63.7
Calib. before NMS	$42.0_{+0.3}$	$46.2_{+1.1}$	$60.8_{+1.4}$	$67.0_{+3.3}$

Table 2: **Uncertainty calibration with NMS.** Calibrating the detection score before NMS prevents well-localized boxes from being filtered out while inaccurately located boxes with over-confidence are removed in NMS step.

how proposed box are correctly classified. AR means how many objects our detector have detected without missing. Thus, AR is a crucial metric for safety-critical applications such as autonomous cars and surgical robots where missing can cause serious problem. AP and AR are averaged over IoU thresholds (.5 : .95), and the higher IoU is, the more accurate localization needs.

4.1. Implementation details

We train Gaussian-FCOS by using Stochastic Gradient Descent (SGD) for $270k$ iterations ($3\times$ schedule [7]) with a mini-batch of 16 images. An initial learning rate is 0.01, and it is decreased by a factor of 10 at $210K$ and $250K$ iterations, respectively. Unless specified, the scale-jitter [7] augmentation is applied where the shorter image side is randomly sampled from [640, 800] pixels. As a backbone network, we use ResNet-50 with ImageNet pre-trained weights in the ablation study.

4.2. Ablation study

Uncertainty loss with power likelihood. We first validate the effectiveness of the proposed uncertainty loss compared to the baseline, Negative log-Likelihood Loss (NLL), in Table 1. We can find that adopting IoU power term in the NLL improves performance AP as well as AR, suggesting that power likelihood with IoU alleviates effectively the misspecification problem. Fig. 4 also shows that the regression loss of Gaussian-FCOS tends to be lower than FCOS. It means that uncertainty loss helps the regression branch learn to reduce the error with the ground-truth location. As a result, both AP and AR are improved from FCOS [26]. Besides, uncertainty calibration also boosts the performance, which means the detection score calibrated by localization uncertainty alleviates the over-confidence problem.

Uncertainty calibration. Table 2 shows the effect according to the order of the usage of NMS and uncertainty calibration. The first row 'w/o Calibration' is the same as 'FCOS + Uncertainty Loss'. We find that the AR gain of 'Calibration before NMS' is bigger than that of 'Calibration after NMS'. It means that the uncertainty calibration prevents more well-localized objects from being filtered out by NMS. In addition, the AP of 'Calibration before NMS' is more improved that of 'Calibration after NMS'. In other words, the NMS may remove the well-localized box that has low confidence, while the proposed method can preserve this box by score calibration. This means that the score calibration before NMS makes the incorrectly located box with over-confidence removed in the NMS step.

Backbone	Uncertainty	AP	AP_{75}	AR	AR_{75}	Time
ResNet-50		41.2	44.4	59.0	63.2	0.041
	√	42.0	46.2	60.8	67.0	0.042
ResNet-101		43.1	46.7	60.6	65.4	0.054
	√	43.7	47.8	61.9	67.8	0.055
VoVNet-39		43.5	47.2	61.4	66.2	0.042
	√	44.3	48.8	62.8	69.1	0.044
VoVNet-57		44.4	47.6	61.6	66.2	0.048
	√	45.2	49.7	63.2	69.7	0.049

Table 3: **Comparison of different backbones on Gaussian-FCOS.**

Backbone network. We validate Gaussian-FCOS with various backbone networks such as ResNet [8] and VoVNet [17]. Table 3 shows that Gaussian-FCOS achieves the consistent performance gains of both AP and AR on various backbone networks. It is noted that Gaussian-FCOS obtains a large improvement of AR, which demonstrates our certainty estimation helps to prevent objects from being missed. Also, through the difference in operation time is very insignificant, (1∼2 ms), our uncertainty branch and calibration efficiently model the localization uncertainty without computational overhead.

4.3. Localization analysis

We investigate how Gaussian-FCOS improves the object localization (AP) and preserves objects without missing (AR). In particular, we compare Gaussian-FCOS with not only FCOS [26] but also Faster R-CNN [23] and RetinaNet [19] in that both are representative baselines in object detection field. First, we analyze Average Precision (AP) at different IoU thresholds and object scales in Table 4. At soft-metric with 0.5 and 0.6 IoU thresholds, Faster R-CNN achieves better AP than the others because Region Proposal Network (RPN) helps Faster R-CNN to remove more false positives. On the other hand, due to RPN, Faster R-CNN tends to be a lower recall rate than others as shown in Table 5. From strict metrics with 0.7 over IoU thresholds, Gaussian-FCOS shows better performance than the others.

Method	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}	AP_S	AP_M	AP_L
Faster R-CNN [23]	40.2	**61.0**	**56.6**	49.1	36.7	13.7	24.2	43.5	52.0
RetinaNet [19]	38.7	58.0	53.6	46.4	34.7	15.7	23.3	42.3	50.3
FCOS [26]	41.2	60.0	55.7	49.4	38.1	18.5	25.7	44.8	52.0
Gaussian-FCOS	**42.0**$_{+0.8}$	58.7$_{-0.3}$	55.2$_{-0.5}$	**50.3**$_{+0.9}$	**40.4**$_{+2.3}$	**20.7**$_{+2.2}$	**26.3**$_{+0.8}$	**45.8**$_{+1.0}$	**53.9**$_{+1.9}$

Table 4: **Comparison of Average Precision (AP) at different IoUs and object scales.** Note that for fair comparison, all models are trained using same training protocol [7] (e.g., 3× schedule, scale jitter) and same backbone (ResNet-50).

Method	AR	AR_{50}	AR_{60}	AR_{70}	AR_{80}	AR_{90}	AR_S	AR_M	AR_L
Faster R-CNN [23]	54.0	78.1	73.6	64.6	50.3	23.9	35.9	57.4	67.8
RetinaNet [19]	55.4	80.1	75.5	65.6	49.7	26.2	37.2	58.9	70.5
FCOS [26]	59.0	81.9	77.7	70.1	55.0	31.2	40.4	62.8	74.1
Gaussian-FCOS	**60.8**$_{+1.8}$	**82.3**$_{+0.4}$	**78.5**$_{+0.8}$	**72.2**$_{+2.1}$	**59.2**$_{+4.2}$	**32.5**$_{+1.3}$	**42.8**$_{+2.4}$	**64.9**$_{+2.1}$	**76.1**$_{+2.0}$

Table 5: **Comparison of Average Recall (AR) at different IoUs and object scales.** These results demonstrate how well Gaussian-FCOS preserves objects without missing. Due to uncertainty calibration, Gaussian-FCOS keeps well-localized objects from being filtered out.

FCOS	AP	AP_{75}	AP_S	AP_M	AP_L
+ centerness-branch [26]	38.5	41.6	**22.4**	42.4	49.1
+ IoU-branch [11, 27]	38.7	42.0	21.6	43.0	50.3
+ QFL [18]	39.0	41.9	22.0	43.1	51.0
+ ours	**39.2**	**43.2**	21.9	**43.2**	**51.0**

Table 6: **Comparison between box quality estimation methods.**

Distribution	AP	AP_{75}	AP_S	AP_M	AP_L
Dirac delta [26]	38.5	41.6	22.4	42.4	49.1
Gaussian [2, 13, 9]	38.6	41.6	21.7	42.5	50.0
General w/ DFL [18]	39.0	42.3	**22.6**	43.0	50.6
Gaussian with NPLL (ours)	**39.2**	**43.2**	21.9	**43.2**	**51.0**

Table 7: **Comparison with representations of box location.** NPLL denotes the proposed negative power log-likelihood loss.

Method	Backbone	AP	AP_{75}	AP_S	AP_M	AP_L
anchor-based:						
RetinaNet [19]	ResNet-101	39.1	42.3	21.8	42.7	50.2
ATSS [29]	ResNet-101	43.6	47.4	26.1	47.0	53.6
GFL [18]	ResNet-101	45.0	48.9	27.2	48.8	54.5
anchor-free:						
ExtremeNet [32]	Hourglass-104	40.2	43.2	20.4	43.2	53.1
CornerNet [15]	Hourglass-104	40.5	43.1	19.4	42.7	53.9
CenterNet [3]	Hourglass-104	44.9	49.0	26.6	48.6	57.5
FCOS [26]	ResNet-101	41.5	45.0	24.4	44.8	51.6
FCOS [26]	ResNeXt-101	44.7	48.4	27.6	47.5	55.6
ours:						
Gaussian-FCOS	ResNet-101	43.8	48.1	25.6	46.9	54.5
Gaussian-FCOS	ResNeXt-101	45.5	49.5	28.2	48.5	56.1
Gaussian-FCOS	VoVNet-99	**46.0**	**50.6**	**28.4**	**49.0**	**56.4**

Table 8: **Comparison to state-of-the-art methods on COCO `test-dev2017`.** These results are tested without multi-scale testing.

This is because the estimated uncertainty enables the network to detect more accurate localized objects by calibrating the detection score. In particular, Gaussian-FCOS obtains bigger AP gain on large objects. We conjecture that this is because mislocalization occurs more frequently in larger objects than in smaller objects.

We also explore the influence of Gaussian-FCOS on preventing objects from being missed (AR). Table 5 shows anchor-free methods (FCOS [26] and Gaussian-FCOS) tend to achieve better Average Recall (AR) than anchor-based methods (Faster-RCNN [23] and RetinaNet [19]) at overall IoUs and scales. This is because anchor-free methods uses more positive samples than anchor-based methods, which increases the recall rate. we can also find that Gaussian-FCOS outperforms FCOS at all metrics. We speculate that the network can re-order the calibrated scores by reflecting localization uncertainty, which keeps well-localized objects from the NMS. For small objects, which are more likely to be missed, Gaussian-FCOS can preserve more small objects compared to FCOS.

4.4. Comparison with other methods.

We also validate Gaussian-FCOS with other methods. For a fair comparison, we adopt 1x learning schedule for 12 epochs without multi-scale training. Table 6 and Table 7 show the comparison results in terms of box quality estimation [26, 11, 27, 18] and the target box representation [2, 13, 9, 18], respectively. In Table 6, our method

Figure 5: **Estimated uncertainty examples of the proposed Gaussian-FCOS.** Since there is no supervision of uncertainty, we analyze the estimated uncertainty qualitatively. Gaussian-FCOS captures lower certainties on unclear or occluded sides. For example, the left-directional certainty of the bird in the top-left image diminishes (66%) since it is occluded by a branch of a tree. Both the surfboard and the person in the top-center image have much lower bottom-directional certainties (5% and 33%) since their shapes are quite unclear due to the water. The leftmost giraffe in the bottom-left image, occluded by another girraffe. also has lower right- and bottom-directional certainties (41% and 8%).

achieves better performance than other methods. Compared to center-point [26] and IoU [11, 27, 18] that are confined to only overall quality, our 4-directions uncertainty values can reflect more degrees (l, r, t, b) of box uality. Table 7 show that our method outperforms other box representation methods including non-parametric General distribution [18]. Our Gaussian-FCOS exploits the *explicit* uncertainty loss with the proposed negative power log-likelihood loss (NPLL) that helps better understand the underlying distribution.

Lastly, we evaluate Gaussian-FCOS on COCO [20] `test-dev2017` dataset for other detection methods. Table 8 summarizes the results. Compared to the state-of-the-art methods, Gaussian-FCOS with VoVNet-99 achieves the best performance. In case of same ResNet-101 backbone, GFL shows the best performance because GFL uses the better baseline (i.e., ATSS) that our baseline (i.e., FCOS). It is expected that applying our method for the anchor-based ATSS [29] would boost performance but it is out of our research scope.

5. Conclusion

We have proposed Gaussian-FCOS that estimates 4-directions uncertainty for anchor-free object detector. To this end, We design the new uncertainty loss, negative Power log-likelihood loss (NPLL) to train the network that produces the localization uncertainty and enables accurate localization. Gaussian-FCOS captures not only the quality of the detected box but also which direction is uncertain by quantified value [0,1]. This localization uncertainty is also utilized as box confidence with which the detection score is calibrated, boosting localization quality and preventing objects from being missed. Experiments on challenging COCO dataset demonstrate that our Gaussian-FCOS improves the overall performance and especially improves the average recall by reducing the missing objects. We can expect the proposed Gaussian-FCOS can serve as a component providing an important cue for safety-critical application or decision-making system.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[2] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *ICCV*, 2019.

[3] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.

[4] Yarin Gal. Uncertainty in deep learning. *PhD Thesis*, 2016.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[6] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. *arXiv:1903.03838*, 2019.

[7] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking imagenet pre-training. In *ICCV*, 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, 2019.

[10] CC Holmes and SG Walker. Assigning a value to a power likelihood in a general bayesian model. *Biometrika*, 104(2):497–503, 2017.

[11] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[13] Florian Kraus and Klaus Dietmayer. Uncertainty estimation in one-stage object detection. In *ITSC*, 2019.

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

[15] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018.

[16] Michael Truong Le, Frederik Diehl, Thomas Brunner, and Alois Knol. Uncertainty estimation for deep neural object detectors in safety-critical applications. In *ITSC*, 2018.

[17] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020.

[18] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *NeurIPS*, 2020.

[19] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[22] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[24] Duygu Sarikaya, Jason J Corso, and Khurshid A Guru. Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. *IEEE transactions on medical imaging*, 2017.

[25] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv:1901.02731*, 2019.

[26] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[27] Shengkai Wu, Xiaoping Li, and Xinggang Wang. Iou-aware single-stage object detector for accurate localization. *Image and Vision Computing*, 2020.

[28] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *MM*, 2016.

[29] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

[30] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018.

[31] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv:1904.07850*, 2019.

[32] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019.