Localization Uncertainty Estimation for Anchor-Free Object Detection

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

Abstract

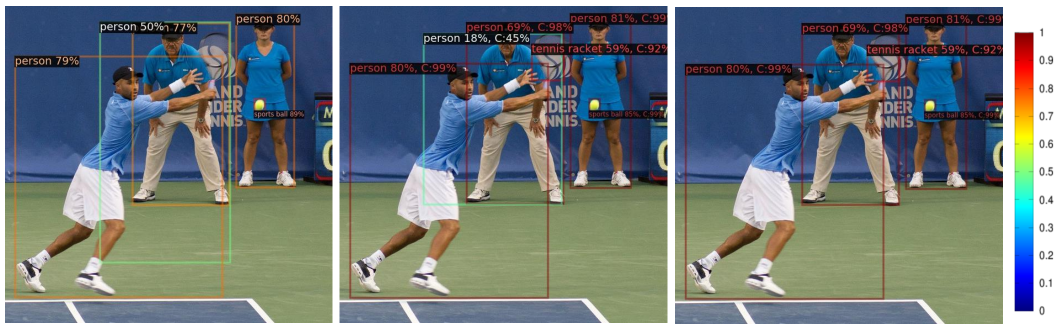
Since many safety-critical systems such as surgical robots and autonomous driving cars are in unstable environments with sensor noise or incomplete data, it is desirable for object detectors to take the confidence of the localization prediction into account. Recent attempts to estimate localization uncertainty for object detection focus only anchor-based method that captures the uncertainty of different characteristics such as location (center point) and scale (width, height). Also, anchor-based methods need to adjust sensitive anchor-box settings. Therefore, we propose a new object detector called *Gaussian-FCOS* that estimates the localization uncertainty based on an anchor-free detector that captures the uncertainty of similar property with four directions of box offsets (left, right, top, bottom) and avoids the anchor tuning. For this purpose, we design a new loss function, *uncertainty loss*, to measure how uncertain the estimated object location is by modeling the uncertainty as a Gaussian distribution. Then, the detection score is calibrated through the estimated uncertainty. Experiments on challenging COCO datasets demonstrate that the proposed new loss function not only enables the network to estimate the uncertainty but produces a synergy effect with regression loss. In addition, our Gaussian-FCOS reduces false positives with the estimated localization uncertainty and finds more missing-objects, boosting both Average Precision (AP) and Recall (AR). We hope Gaussian-FCOS serve as a baseline for the reliability-required task.

1 Introduction

Object detection based on CNNs is a key component of perception for safety-critical systems such as autonomous car and surgical robots [1]. To notify humans how trustworthy the output is, it is necessary for safety-critical applications to quantify how certain they are in the output. Also, capturing uncertainty as additional information can improve the reliability for the subsequent steps in decision-making such as sensor fusion or tracking. Object detection is a task that combines object localization and classification, and most of the existing state-of-the mart methods [2–4] show the reliability of their algorithm as a single value (i.e. confidence). In other words, they do not estimate the localization uncertainty for predicted box coordinates and only use the classification score as the detection certainty. As a consequence, the detection results of these methods contain mislocalized detections with over-confidence [5]. For example, as shown in Fig. 1(a), the mislocalization 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 coordinate based on the localization uncertainty is also needed to be measured.

Recent efforts [6, 5, 7, 8] to estimate uncertainty for object detection have been attempted. However, all of these efforts focus on the anchor-box based methods that regress center point (x,y), width, and height. That is, they model the uncertainty of location (center point) and scale (width, height)

Preprint. Under review.



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

Figure 1: **Example object detection results of FCOS and Gaussian-FCOS (proposed).** These results are based on ResNet-50 backbone. The colors of bounding boxes show 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 certainty score estimated from Gaussian-FCOS. 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 uncertainty. compared to FCOS, Gaussian-FCOS localizes the person with a tennis racket more accurately (tightly).

equally by increasing four channels in the regression output. However, each uncertainty of location and scale shows different distribution [6]. since center point, width, and height have semantically different characteristics, this approach that considers each value equally is inadequate to model the localization uncertainty. In addition, anchor-based methods have to adjust sensitive hyper-parameters of the anchor whenever the target dataset changes.

By removing such heuristic anchor-box tuning (e.g., scale, aspect ratio), anchor-free methods [9–12, 3] surpass conventional anchor-box based methods such as Faster R-CNN [13], RetinaNet [14], and their variants [2, 15]. As a representative state-of-of-the-art anchor-free method, FCOS [3] adopts the concept of centerness to filter out false positive boxes, where the centerness can be interpreted as the implicit localization uncertainty of a proposal box. However, FCOS 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., location, scale).

In this paper, we propose a method, called Gaussian-FCOS, that estimates explicitly localization uncertainty for anchor-free method, FCOS. Unlike centerness in FCOS, we model the uncertainty for each of the four box offsets (left, right, top, bottom) from the center to the box to fully describe the localization uncertainty. Moreover, unlike the conventional anchor-based methods [6, 5, 7, 8] for localization uncertainty, the estimated uncertainty of the four box offsets having similar semantic characteristic make it possible to inform which direction of a box boundary is uncertain independently from the overall box uncertainty.

To do this, we model the box offset and its uncertainty of FCOS through Gaussian distribution by adding the uncertainty branch. In addition, we design a new uncertainty loss that enables the uncertainty branch to learn to estimate localization uncertainty. In particular, this new loss creates synergy with the existing box regression loss that tells the difference between the ground truth and the predicted box offset, enabling more accurate box prediction. For example, as illustrated in Fig. 1, comparing the detected box of the person with a tennis racket between (a) and (c), we can see that Gaussian-FCOS localizes objects more accurately (tightly) than FCOS.

Furthermore, we calibrate the detection score through the localization uncertainty. Specifically, we compute certainty from the estimated uncertainty and then multiply it to the classification score to get the final detection score. Fig. 1(b)-(c) shows the effectiveness of uncertainty calibration. Unlike the detection result from FCOS in Fig. 1(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 the *first* localization uncertainty estimation for anchor-free object detection that models four directions as Gaussian distribution.
- We design the new loss function, *uncertainty loss*, which makes it possible to estimate localization uncertainty and produces the synergy effect with the regression loss.
- We apply the localization uncertainty as box confidence and calibrate the detection score with the uncertainty.
- We analyze the influence of uncertainty on object localization and confirm the improvement of mislocalization and missing objects on challenging COCO dataset.

2 Related Works

2.1 Anchor-Free Object Detection

Recently, anchor-free object detectors [9, 11, 10, 12, 3] have attracted attention beyond anchor-based methods [13, 14, 2, 15] that need to tune sensitive hyper-parameters related to anchor box (e.g., scale, aspect ratio, etc). CornerNet [9] predicts an object location as a pair of keypoints (top-left and bottom-right). CenterNet [10] extends CornerNet as a triplet instead of a pair of keypoints to boost performance. ExtremeNet [11] locates four extreme points (top, bottom, left, right) and one center point to generate the object box. Zhu *et al.* [12] utilizes keypoint estimation to predict center point objects and regresses to other attributes including size, orientation, pose, and 3D location. FCOS [3] views all points inside the ground-truth box as positive samples and regresses four distances (left, right, top, bottom) from 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 [16, 17, 7]: epistemic (sampling-based) and aleatoric (sampling-free) uncertainty. Epistemic uncertainty measures the model uncertainty in the models' parameters through Bayesian neural networks [18], Monte Carlo dropout [19], and Bootstrap Ensemble [20]. 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 or problem inherent such as sensor noise and ambiguities in data. It can be estimated by explicitly modeling it as model output.

Recent works [5, 7, 8, 20] have adopted uncertainty estimation for object detection. Epistemic based methods [20, 8] use Monte Carlo dropout. Aleatoric based methods [5, 7] jointly estimate the four parameters of bounding box with Gaussian log-likelihood in anchor-based object detectors such as Faster R-CNN [13] or YOLOv3 [21]. Unlike epistemic uncertainty, since aleatoric uncertainty does not need to inference several times, it is more suitable for real-time object detection. Therefore, we also adopt aleatoric uncertainty and this is the first uncertainty estimation for anchor-free object detector (FCOS [3]).

3 Proposed Method

There are two reasons that we choose anchor-free detector, FCOS [3], for localization uncertainty compared to anchor-based methods [22, 6, 7]. 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 using anchor box, while FCOS directly regresses four boundaries (left,right,top,bottom) of a bounding box from each location, which are semantically symmetric. It can be assumed that values sharing semantic meanings have similar properties and easier to be modeled in the same method or network structure. Furthermore, it is possible 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 [3]. 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

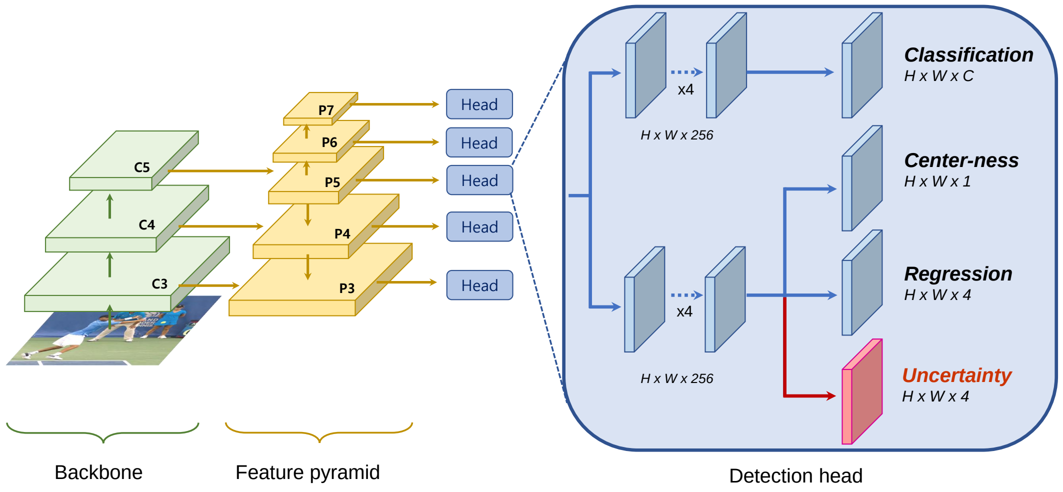


Figure 2: **Architecture of Gaussian-FCOS.** Different from FCOS [3], Gaussian-FCOS estimates localization uncertainty from *uncertainty* branch that outputs four uncertainties of box offsets (left, right, top, and bottom.)

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 [3], if the location on feature maps belongs to the ground-truth box area, it is regarded as a positive sample otherwise a negative sample. Each location (x, y) regresses the box offsets, 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,

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

where (x^g_{lt}, y^g_{lt}) and (x^g_{rb}, y^g_{rb}) denotes 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 [23] 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 idea of centerness is that the object which has its center far from the pixel location is less likely to be predicted correctly. By measuring the normalized distance from the location to the center of the object, centerness value of each location penalizes the detection score if the distance is long.

3.2 Gaussian-FCOS

Uncertainty loss. FCOS [3] predicts the class score, box offsets B=(l, r, t, b), and centerness. Although class score and centerness are between zero and one as confidence values, box coordinates are deterministic values that cannot inform how certain the box is. 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 simply determines 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 regressed the box offsets (l, r, t, b). To our network predicts probability distribution instead of only 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:

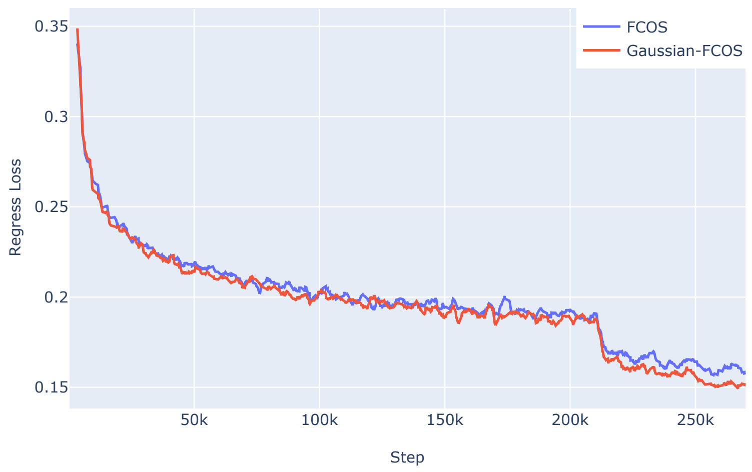


Figure 3: **Comparison of the regression Loss.** Regression loss of Gaussian-FCOS tends to be lower than that of FCOS. That is, training with the proposed uncertainty loss further reduces regression loss, and it helps to learn the better object localization.

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

$$= \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_B|^{1/2}} \exp\left\{-\frac{1}{2}\left(B^* - \boldsymbol{\mu}_B\right)^\top \boldsymbol{\Sigma}_B^{-1}\left(B^* - \boldsymbol{\mu}_B\right)\right\}, \tag{3}$$

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

In the training stage, we add *uncertainty loss* defined as negative log likelihood loss for uncertainty:

$$\mathcal{L}_{uncertainty} = -\frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \sum_{k\in\{l,r,t,b\}} IoU_{x,y} \cdot \log P_\Theta\left(B^g_{x,y,k}|\mu_k, \sigma_k^2\right) \tag{4}$$

$$= \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} IoU_{x,y} \cdot \left[\sum_{k\in\{l,r,t,b\}} \left\{\frac{(B^g_{x,y,k} - \mu_k)^2}{2\sigma_k^2} + \frac{1}{2}\log\sigma_k^2\right\} + 2\log 2\pi\right], \tag{5}$$

where N_{pos} denotes the number of positive samples, λ ($\lambda = 0.05$ in this paper) is the balance weight for $L_{uncertainty}$, and $IoU_{x,y}$ is the intersection-over-union between the predicted box and the ground-truth box. 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 [3], we use focal loss [14] for classification, binary cross-entropy loss for centerness and IoU loss [23] for regression. The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{classification} + \mathcal{L}_{centerness} + \mathcal{L}_{regression} + \mathcal{L}_{uncertainty}. \tag{6}$$

It is noted that unlike centerness, as we train the network to directly estimate 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 [3] network structure by adding the uncertainty branch as shown in Fig. 2. 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

Table 1: **Effectiveness of each component in Gaussian-FCOS.**

Method	AP	AR
FCOS	41.2	59.0
+Uncertainty Loss	41.7+0.5	59.4+0.4
+Uncertainty Calibration	42.0+0.8	60.8+1.8

Table 2: **Effectiveness of uncertainty calibration with NMS.**

Method	AP	AR
w/o Calibration	41.7	59.4
Calibration after NMS	41.5	59.4
Calibration before NMS	42.0+0.3	60.8+1.4

Table 3: **Comparison of different backbones on Gaussian-FCOS.** All models were trained using the same training protocol [25] (e.g., 3× schedule, scale-jitter) and the inference time was measured with 1 batch on the same V100 GPU machine.

Backbone	Uncertainty	AP	AR	Inference time
ResNet [26]-50		41.2	59.0	0.041
	√	**42.0**+0.8	**60.8**+1.8	0.042
ResNet [26]-101		43.1	60.6	0.054
	√	**43.7**+0.6	**61.9**+1.3	0.055
VoVNet [27]-39		43.5	61.4	0.042
	√	**44.3**+0.8	**62.8**+1.4	0.044
VoVNet [27]-57		44.4	61.6	0.048
	√	**45.2**+0.8	**63.2**+1.6	0.049

σ_k in [0, 1] . we also make the uncertainty branch share the box regression tower (4 `conv` layers) to output two kinds of statistic parameters (μ_k, σ_k) be derived from the same features.

Uncertainty calibration. With the uncertainty branch, Gaussian-FCOS can obtain the localization uncertainty σ_k and it is utilized 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 with the class score. Thus, we calibrate the detection score by multiplying the estimated box confidence to the class score to replace the class score with the calibrated detection score in NMS. Fig. 1(b) shows the false positive (cyan box) gets lower box confidence (45%) than other true positives (over 90%), which means Gaussian-FCOS estimates localization uncertainty well. Fig. 1 (c) demonstrates unlike FCOS as shown in Fig. 1(a), the false positive with the calibrated (penalized) the detection score is removed.Also ,

4 Experiments

In this section, we evaluate the effectiveness of Gaussian-FCOS on the challenging COCO [24] 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 means how many objects are correctly detected with the right classification. AR means how many objects we can detect or we don't miss. Thus, AR is a crucial metric for safety-critical applications such as autonomous cars or surgical robots. 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× schedule [25]) with a mini-batch of 16 images and initial learning rate of 0.01 which is decreased by a factor of 10 at 210K and 250K iterations, respectively. We also use the scale-jitter [25] augmentation where the shorter image side is randomly sampled from [640, 800] pixels. Unless specified, we use ResNet-50 as a backbone network with ImageNet pretrained weights in ablation study. We

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 [25] (e.g., 3× schedule, scale jitter) and same backbone (ResNet-50) implemented on `Detectron2` [29].

Method	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}	AP_S	AP_M	AP_L
Faster R-CNN [13]	40.2	**61.0**	**56.6**	49.1	36.7	13.7	24.2	43.5	52.0
RetinaNet [14]	38.7	58.0	53.6	46.4	34.7	15.7	23.3	42.3	50.3
FCOS [3]	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 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.

Method	AR	AR_{50}	AR_{60}	AR_{70}	AR_{80}	AR_{90}	AR_S	AR_M	AR_L
Faster R-CNN [13]	54.0	78.1	73.6	64.6	50.3	23.9	35.9	57.4	67.8
RetinaNet [14]	55.4	80.1	75.5	65.6	49.7	26.2	37.2	58.9	70.5
FCOS [3]	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

implement Gaussian-FCOS based on `Detectron2` [28] and use four V100 (32GB) GPUs, Pytorch1.3 and CUDA 10.0.

4.2 Ablation study

Uncertainty loss and calibration. We first validate the effectiveness of the proposed uncertainty loss and uncertainty calibration in Table 1. Fig. 3 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 the baseline (FCOS with ResNet-50). Besides, uncertainty calibration also boosts the performance, which means the detection score calibrated by localization uncertainty alleviates the over-confidence problem. 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' (second row in Table 1). We find that the AR gain of 'Calibration before NMS' is bigger than that of 'Calibration after NMS'. It means that the uncertainty calibration keeps more accurate localized objects from being filtered out by NMS. In addition, the AR of 'Calibration before NMS' is more improved that of 'Calibration after NMS'. In other words, the NMS removes the well-localized box that has low confidence, while the proposed method can keep this box by score calibration.

Backbone network. We validate Gaussian-FCOS with various backbone networks such as ResNet [26] and VoVNet [27]. 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 more improvement of AR than that of AP, which demonstrates our certainty estimation helps to prevent objects from being missed. Also, the gap of inference time is slight (1∼2 ms), which shows that our uncertainty branch and uncertainty calibration does not cause 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 [3] but also Faster R-CNN [13] and RetinaNet [14] 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.

Table 6: **Comparison to state-of-the-art methods on COCO `test-dev2017`.** These results are tested without multi-scale testing.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
anchor-based:							
Cascade R-CNN [2]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
RetinaNet [14]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
ATSS [4]	ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
anchor-free:							
ExtremeNet [11]	Hourglass-104	40.2	55.5	43.2	20.4	43.2	53.1
CornerNet [9]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
CenterNet [10]	Hourglass-104	44.9	62.4	49.0	26.6	48.6	57.5
FCOS [3]	ResNet-101	41.5	60.7	45.0	24.4	44.8	51.6
FCOS [3]	ResNeXt-101	44.7	**64.1**	48.4	27.6	47.5	55.6
ours:							
Gaussian-FCOS	ResNet-101	43.8	61.2	48.1	25.6	46.9	54.5
Gaussian-FCOS	ResNeXt-101	45.5	63.9	49.5	28.2	48.5	56.1
Gaussian-FCOS	VoVNet-99	**46.0**	63.6	**50.6**	**28.4**	**49.0**	**56.4**

From strict metrics with 0.7 over IoU thresholds, Gaussian-FCOS shows better performance than the others. The reason is that the estimated uncertainty enables the network to distinguish 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 [3] and Gaussian-FCOS) tend to achieve better Average Recall (AR) than anchor-based methods (Faster-RCNN [13] and RetinaNet [14]) at overall IoUs and scales. This is because FCOS uses more positive samples than anchor-based methods and in turn increases 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 results in preventing well-localized objects from being missed. For small objects, as it is more likely to be missed, Gaussian-FCOS can preserve more small objects compared to FCOS.

4.4 Comparison with state-of-the-art.

We validate Gaussian-FCOS on COCO [24] `test-dev2017` dataset and compare it with state-of-the-art methods. Note that `test-dev2017` dataset has no disclosed labels and is evaluated on the test-server. Table 6 summarizes the results. Gaussian-FCOS with the same backbone ResNet-101 outperforms the all anchor-based methods. Compared to state-of-the-art anchor-free method, CenterNet [10], Gaussian-FCOS with ResNeXt-101 achieves higher performance. Moreover, our Gaussian-FCOS further boosts performance with VoVNet-99 from the baseline, achieving state-of-the-art performance.

5 Discussion

Since Gaussian-FCOS can estimate uncertainties along with 4 directions (left, right, top, bottom) of an object, we confirm whether the estimated uncertainties are effective. Figure 4 displays the estimated direction uncertainties on each object. C_{direction} denotes that the estimated certainty with respect to each direction (e.g., for left-direction, C_L). Specifically, the first row shows ambiguity examples where the cat wearing a green hat is detected with lower top-direction certainty than other direction ones due to its ambiguity. Gaussian-FCOS also estimates lower certainties on unclear or cloaked areas (e.g., the surfer's leg, the bottom and right side of the surferboard and the bird). The second row illustrates occlusion examples. The giraffe at the top-left is almost obscured by the giraffe in front of it, which results in the lower certainties (e.g., C_B:8% and C_R:41%, respectively). Also, as the right side of the man is occluded by the left side of the woman, lower certainties of those

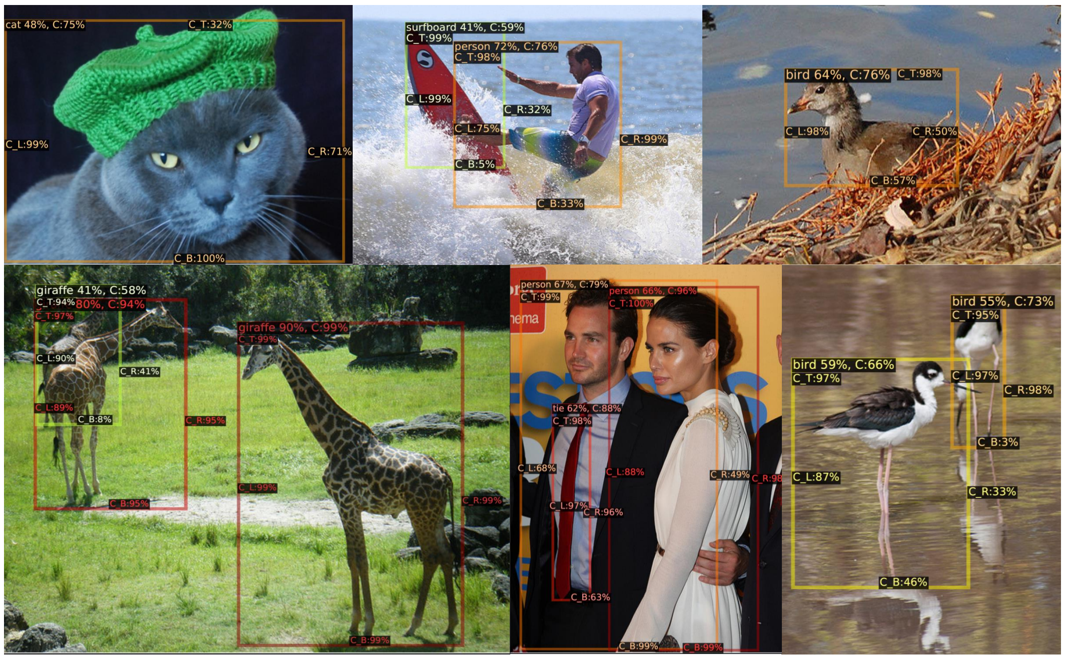


Figure 4: **Visualization of 4-direciton uncertainties for object detection.** These results are based on Gaussian-FCOS with ResNet-50. C_L, C_R, C_T, and C_B denote the estimated certainty with respect to left, right, top, and bottom. First row illustrates ambiguity examples and second row shows occlusion examples. These results demonstrate Gaussian-FCOS makes it possible to capture which direction is uncertain due to unclear or obvious objects.

directions are captured. Lastly, there are two birds located in a shallow pond. Since legs of the front bird are reflected by in the pond, Gaussian-FCOS mis-localizes the bird along bottom directions with lower certainty (46%). The behind bird is also occluded and then the bottom certainty is very low (3%). These results demonstrate our Gaussian-FCOS can provide safety-critical applications with prior information to make decision.

6 Conclusion

We have proposed Gaussian-FCOS that is the *first* localization uncertainty estimation for anchor-free object detector. The uncertainty is modeled as Gaussian distribution by adding uncertainty branch into FCOS. We also design the uncertainty loss to train the network that produces the localization uncertainty and enables accurate localization. This localization uncertainty is 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 Gaussian-FCOS achieves not only higher Average Precision (localization quality) but also Average Recall (lowering missing rate). We can expect the proposed Gaussian-FCOS can serve as a strong baseline or component in safety-critical applications.

Broader Impact

In this section, we discuss the influence of our Gaussian-FCOS. We expect it is beneficial who needs more accurate object localization ability and the localization uncertainty for better decision-making in safety-critical system. At the same time, if this method is abused in missile tracking systems, it could have negative consequences. Furthermore, we should be cautious of the result of failure of the system which could cause the mislocalization or missing the objects. Finally, we don't think our method leverages biases in the data because the COCO [24] dataset we used consists of a large number of samples and various object scales.

References

[1] D. Sarikaya, J. J. Corso, and K. 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.

[2] Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in *CVPR*, 2018, pp. 6154–6162.

[3] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in *ICCV*, 2019.

[4] 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,” 2020.

[5] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, “Bounding box regression with uncertainty for accurate object detection,” in *CVPR*, 2019.

[6] F. Kraus and K. Dietmayer, “Uncertainty estimation in one-stage object detection,” in *ITSC*, 2019.

[7] M. T. Le, F. Diehl, T. Brunner, and A. Knol, “Uncertainty estimation for deep neural object detectors in safety-critical applications,” in *ITSC*, 2018.

[8] A. Harakeh, M. Smart, and S. L. Waslander, “Bayesod: A bayesian approach for uncertainty estimation in deep object detectors,” *arXiv:1903.03838*, 2019.

[9] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *ECCV*, 2018, pp. 734–750.

[10] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint triplets for object detection,” in *ICCV*, 2019.

[11] X. Zhou, J. Zhuo, and P. Krahenbuhl, “Bottom-up object detection by grouping extreme and center points,” in *CVPR*, 2019.

[12] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” in *arXiv:1904.07850*, 2019.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *NeurIPS*, 2015.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *ICCV*, 2017.

[15] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” in *CVPR*, 2018.

[16] Y. Gal, “Uncertainty in deep learning,” *PhD Thesis*, 2016.

[17] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *NeurIPS*, 2017.

[18] K. Shridhar, F. Laumann, and M. Liwicki, “A comprehensive guide to bayesian convolutional neural network with variational inference,” *arXiv:1901.02731*, 2019.

[19] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016.

[20] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *NeurIPS*, 2017.

[21] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv:1804.02767*, 2018.

[22] 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 *ICCV*, 2019.

[23] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “Unitbox: An advanced object detection network,” in *MM*, 2016.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.

[25] K. He, R. Girshick, and P. Dollar, “Rethinking imagenet pre-training,” in *ICCV*, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

[27] Y. Lee and J. Park, “Centermask: Real-time anchor-free instance segmentation,” in *CVPR*, 2020.

[28] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, “Detectron,” https://github.com/facebookresearch/detectron, 2018.

[29] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” https://github.com/facebookresearch/detectron2, 2019.

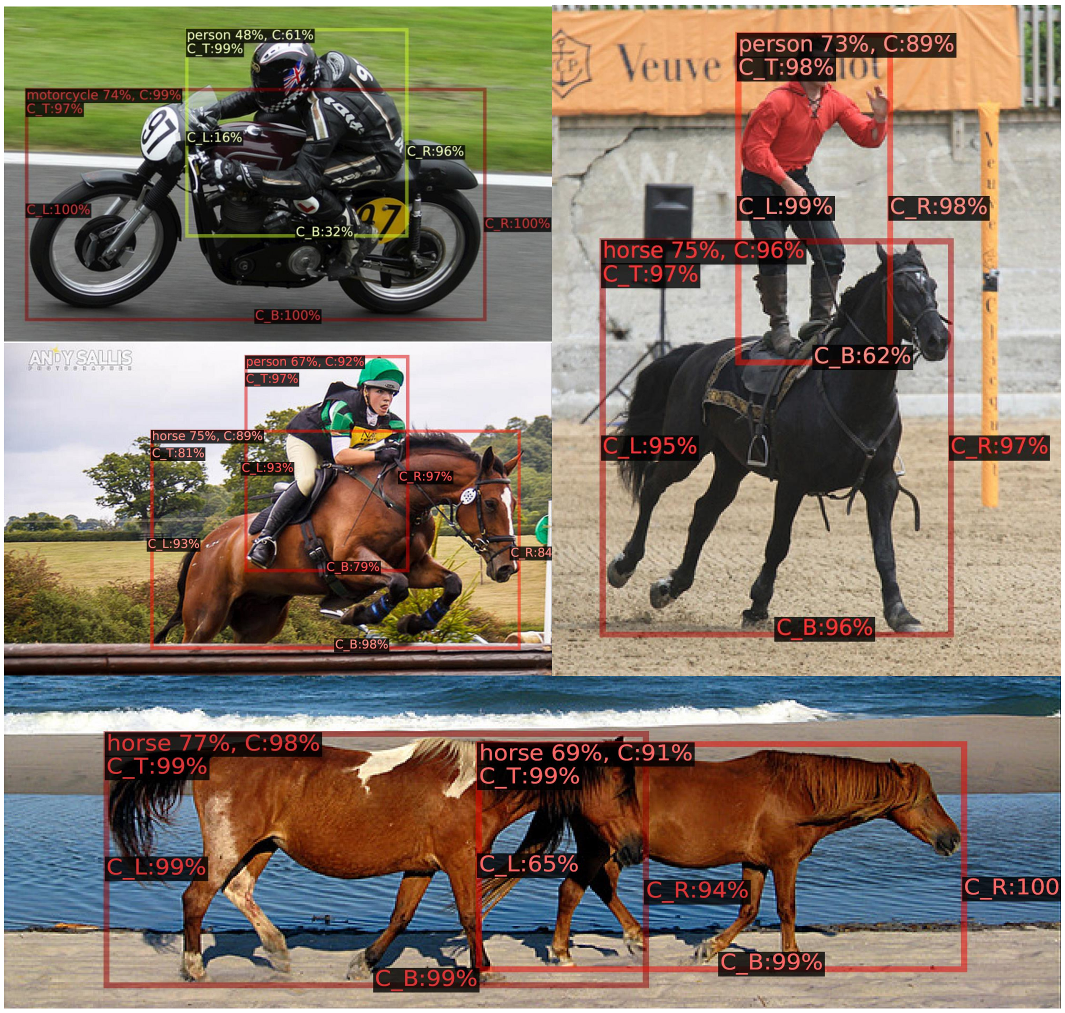
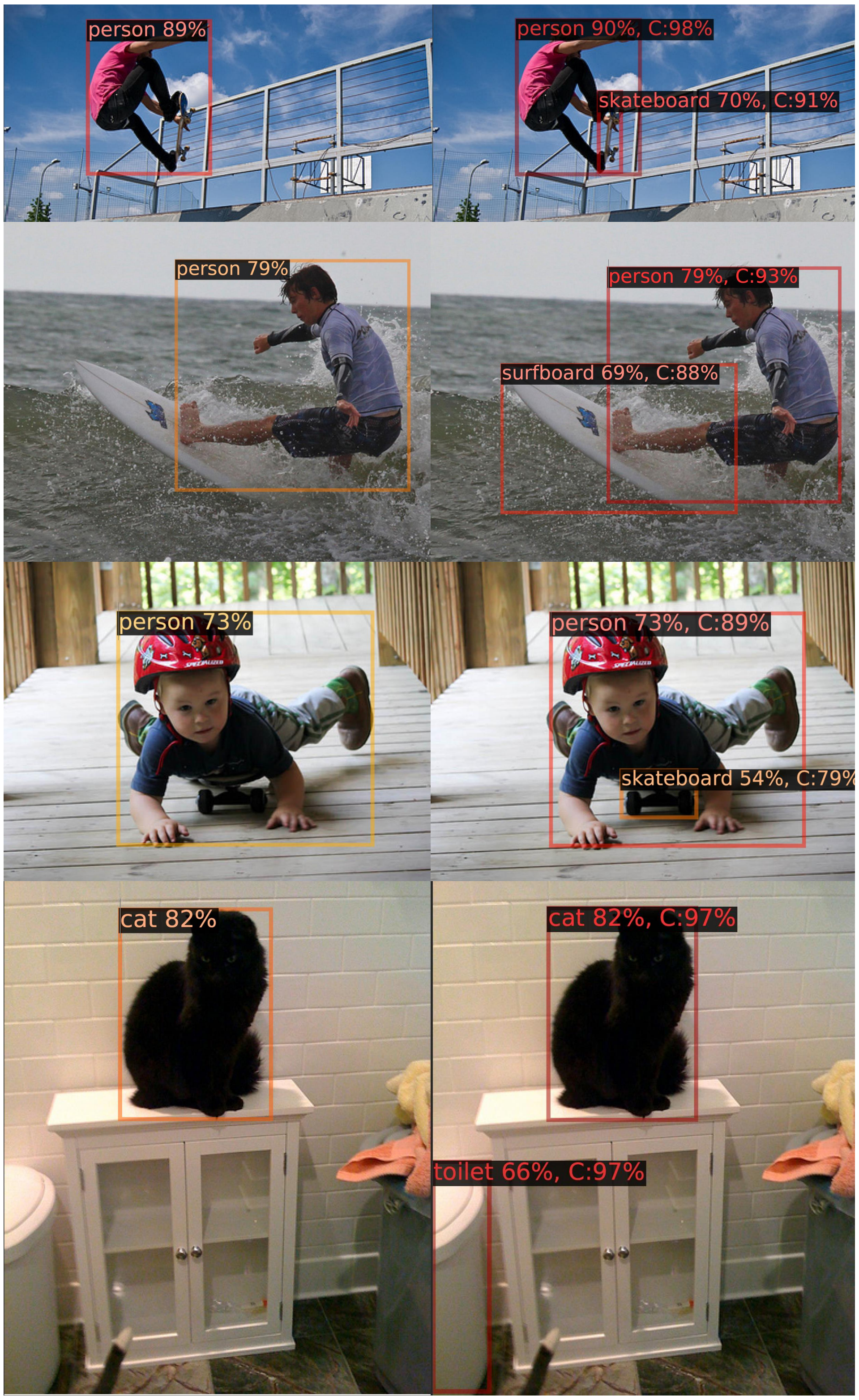


Figure 5: **More examples of 4-direciton uncertainties for object detection.** It is noted that the estimated certainties tend to be lower in boundaries between objects. For example, legs of persons on motor-bicycle or horses are likely to get lower certainties.

A Examples of estimated localization uncertainties.

We display more examples of estimated localization uncertainties. Fig. 5 illustrates how well each direction certainty is estimated. Note that the estimated certainties tend to be lower in boundaries between objects. For example, the bottom direction of persons on the motor-bicycle or horses tends to get lower certainties. Also, a front horse is localized with high and uniform certainties while the behind horse gets lower left-direction certainty because of its occluded part. Fig. 6 shows qualitative comparisons with FCOS [3]. It is noted that FCOS is more likely to miss objects that seem truncated (toilet) or unclear (skateboard or surfboard) than Gaussian-FCOS, which are consistent results as Table 5. Due to localization uncertainty, the well-localized objects are re-ordered and are preserved from being missed in NMS step.



(a) FCOS **(b) Gaussian-FCOS (ours)**

Figure 6: **Examples of missing objects.** Compared to our FCOS, our Gaussian-FCOS is less likely to miss objects that are truncated or unclear.