Bounding Box Regression with Uncertainty for Accurate Object Detection

Yihui He[1] Chenchen Zhu[1] Jianren Wang[1] Marios Savvides[1] Xiangyu Zhang[2]
[1]Carnegie Mellon University [2]Megvii Inc. (Face++)
{he2,chenchez,jianrenw,marioss}@andrew.cmu.edu zhangxiangyu@megvii.com

(a) (b) (c) (d)

Figure 1: In object detection datasets, the ground-truth bounding boxes have inherent ambiguities in some cases. The bounding box regressor is expected to get smaller loss from ambiguous bounding boxes with our KL Loss. (a)(c) The ambiguities introduced by inaccurate labeling. (b) The ambiguities introduced by occlusion. (d) The object boundary itself is ambiguous. It is unclear where the left boundary of the train is because the tree partially occludes it. (*better viewed in color*)

Abstract

*Large-scale object detection datasets (*e.g.*, MS-COCO) try to define the ground truth bounding boxes as clear as possible. However, we observe that ambiguities are still introduced when labeling the bounding boxes. In this paper, we propose a novel bounding box regression loss for learning bounding box transformation and localization variance together. Our loss greatly improves the localization accuracies of various architectures with nearly no additional computation. The learned localization variance allows us to merge neighboring bounding boxes during non-maximum suppression (NMS), which further improves the localization performance. On MS-COCO, we boost the Average Precision (AP) of VGG-16 Faster R-CNN from 23.6% to* ***29.1%****. More importantly, for ResNet-50-FPN Mask R-CNN, our method improves the AP and* AP^{90} *by* ***1.8%*** *and* ***6.2%*** *respectively, which significantly outperforms previous state-of-the-art bounding box refinement methods. Our code and models are available at github.com/yihui-he/KL-Loss*

1. Introduction

Large scale object detection datasets like ImageNet [6], MS-COCO [35] and CrowdHuman [47] try to define the ground truth bounding boxes as clear as possible.

However, we observe that the ground-truth bounding boxes are inherently ambiguous in some cases. The ambiguities makes it hard to label and hard to learn the bounding box regression function. Some inaccurately labeled bounding boxes from MS-COCO are shown in Figure 1 (a)(c). When the object is partially occluded, the bounding box boundaries are even more unclear, shown in Figure 1 (d) from YouTube-BoundingBoxes [40].

Object detection is a multi-task learning problem consisting of object localization and object classification. Current state-of-the-art object detectors (*e.g.*, Faster R-CNN [42], Cascade R-CNN [2] and Mask R-CNN [17]) rely on bounding box regression to localize objects.

However, the traditional bounding box regression loss (*i.e.*, the smooth L1 loss [13]) does not take such the ambiguities of the ground truth bounding boxes into account. Besides, bounding box regression is assumed to be accurate when the classification score is high, which is not always the case, illustrated in Figure 2.

To address these problems, we propose a novel bounding box regression loss, namely KL Loss, for learning bounding box regression and localization uncertainty at the same time. Specifically, to capture the uncertainties of bounding box prediction, we first model the bounding box pre-

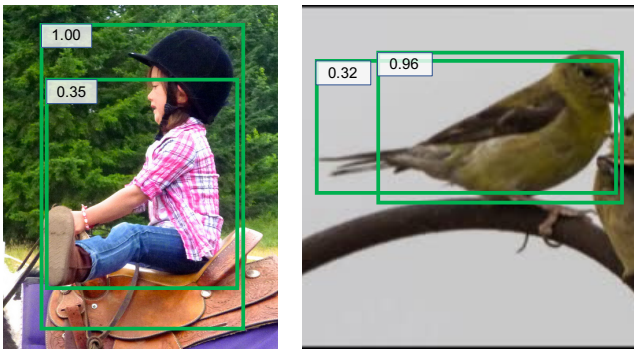

Figure 2: Illustration of failure cases of VGG-16 Faster R-CNN on MS-COCO. (a) both candidate boxes are inaccurate in a certain coordinate. (b) the left boundary of the bounding box which has the higher classification score is inaccurate. (*better viewed in color*)

diction and ground-truth bounding box as Gaussian distribution and Dirac delta function respectively. Then the new bounding box regression loss is defined as the KL divergence of the predicted distribution and ground-truth distribution. Learning with KL Loss has three benefits: (1) The ambiguities in a dataset can be successfully captured. The bounding box regressor gets smaller loss from ambiguous bounding boxes. (2) The learned variance is useful during post-processing. We propose var voting (variance voting) to vote the location of a candidate box using its neighbors' locations weighted by the predicted variances during non-maximum suppression (NMS). (3) The learned probability distribution is interpretable. Since it reflects the level of uncertainty of the bounding box prediction, it can potentially be helpful in down-stream applications like self-driving cars and robotics [7, 16, 21].

To demonstrate the generality of KL Loss and var voting, we evaluate various CNN-based object detectors on both PASCAL VOC 2007 and MS-COCO including VGG-CNN-M-1024, VGG-16, ResNet-50-FPN, and Mask R-CNN. Our experiments suggest that our approach offers better object localization accuracy for CNN-based object detectors. For VGG-16 Faster R-CNN on MS-COCO, we improve the AP from 23.6% to **29.1%**, with only 2ms increased inference latency on the GPU (GTX 1080 Ti). Furthermore, we apply this pipeline to ResNet-50-FPN Mask R-CNN and improve the AP and AP^{90} by **1.8%** and **6.2%** respectively, which outperforms the previous state-of-the-art bounding box refinement algorithm [27].

2. Related Work

Two-stage Detectors: Although one-stage detection algorithms [36, 41, 30, 51] are efficient, state-of-the-art object detectors are based on two-stage, proposal-driven mechanism [42, 4, 5, 17, 31, 2]. Two-stage detectors generate cluttered object proposals, which result in a large number of duplicate bounding boxes. However, during the standard NMS procedure, bounding boxes with lower classification scores will be discarded even if their locations are accurate. Our var voting tries to utilize neighboring bounding boxes based on localization confidence for better localization of the selected boxes.

Object Detection Loss: To better learn object detection, different kind of losses have been proposed. UnitBox [49] introduced an Intersection over Union (IoU) loss function for bounding box prediction. Focal Loss [34] deals with the class imbalance by changing the standard cross entropy loss such that well-classified examples are assigned lower weights. [39] optimizes for the mAP via policy gradient for learning globally optimized object detector. [28] introduces uncertainties for depth estimation. The idea is further extended to the 3D object detection [10, 9]. [29] proposes to weight multi-task loss for scene understanding by considering the uncertainty of each task. With KL Loss, our model can adaptively adjust variances for the boundaries of every object during training, which can help to learn more discriminative features.

Non-Maximum Suppression: NMS has been an essential part of computer vision for many decades. It is widely used in edge detection [44], feature point detection [37] and objection detection [13, 12, 42, 45]. Recently, soft NMS and learning NMS [1, 24] are proposed to improve NMS results. Instead of eliminating all lower scored surrounding bounding boxes, soft-NMS [1] decays the detection scores of all other neighbors as a continuous function of their overlap with the higher scored bounding box. Learning NMS [24] proposed to learn a new neural network to perform NMS using only boxes and their classification scores.

Bounding Box Refinement: MR-CNN [11] is first proposed to merge boxes during iterative localization. Relation network [25] proposes to learn the relation between bounding boxes. Recently, IoU-Net [27] proposes to learn the IoU between the predicted bounding box and the ground-truth bounding box. IoU-NMS is then applied to the detection boxes, guided by the learned IoU. Different from IoU-Net, we propose to learn the localization variance from a probabilistic perspective. It enables us to learn the variances for the four coordinates of a predicted bounding box separately

instead of only IoU. Our var voting determine the new location of a selected box based on the variances of neighboring bounding boxes learned by KL Loss, which can work together with soft-NMS (Table 1 and Table 6).

3. Approach

In this section, we first introduce our bounding box parameterization. Then we propose KL Loss for training detection network with localization confidence. Finally, a new NMS approach is introduced for improving localization accuracy with our confidence estimation.

3.1. Bounding Box Parameterization

Based on a two-stage object detector Faster R-CNN or Mask R-CNN [42, 17] shown in Figure 3, we propose to regress the boundaries of a bounding box separately. Let $(x_1, y_1, x_2, y_2) \in \mathcal{R}^4$ be the bounding box representation as a 4-dimensional vector, where each dimension is the box boundary location. We adopt the parameterizations of the (x_1, y_1, x_2, y_2) coordinates instead of the (x, y, w, h) coordinates used by R-CNN [13]:

$$
\begin{aligned}
t_{x_1} &= \frac{x_1 - x_{1a}}{w_a}, t_{x_2} = \frac{x_2 - x_{2a}}{w_a} \\
t_{y_1} &= \frac{y_1 - y_{1a}}{h_a}, t_{y_2} = \frac{y_2 - y_{2a}}{h_a} \\
t^*_{x_1} &= \frac{x^*_1 - x_{1a}}{w_a}, t^*_{x_2} = \frac{x^*_2 - x_{2a}}{w_a} \\
t^*_{y_1} &= \frac{y^*_1 - y_{1a}}{h_a}, t^*_{y_2} = \frac{y^*_2 - y_{2a}}{h_a}
\end{aligned}
\tag{1}
$$

where t_{x_1}, t_{y_1}, t_{x_2}, t_{y_2} are the predicted offsets. $t^*_{x_1}$, $t^*_{y_1}$, $t^*_{x_2}$, $t^*_{y_2}$ are the ground-truth offsets. x_{1a}, x_{2a}, y_{1a}, y_{2a}, w_a, h_a are from the anchor box. x_1, y_1, x_2, y_2 are from the predicted box. In the following discussions, a bounding box coordinate is denoted as x for simplicity because we can optimize each coordinate independently.

We aim to estimate the localization confidence along with the location. Formally, our network predicts a probability distribution instead of only bounding box location. Though the distribution could be more complex ones like multivariate Gaussian or a mixture of Gaussians, in this paper we assume the coordinates are independent and use single variate gaussian for simplicity:

$$
P_\Theta(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-x_e)^2}{2\sigma^2}} \tag{2}
$$

where Θ is the set of learnable parameters. x_e is the estimated bounding box location. Standard deviation σ measures uncertainty of the estimation. When $\sigma \to 0$, it means our network is extremely confident about estimated bounding box location. It is produced by a fully-connected layer on top of the fast R-CNN head (`fc7`). Figure 3 illustrates

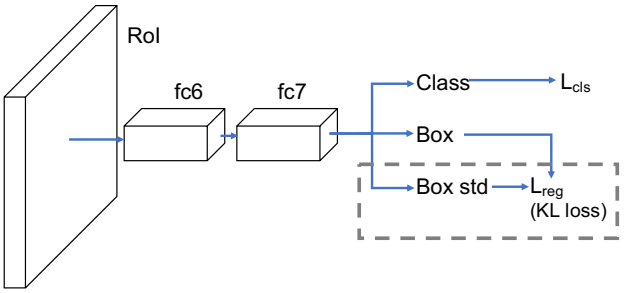

Figure 3: Our network architecture for estimating localization confidence. Different from standard fast R-CNN head of a two stage detection network, our network esitmates standard deviations along with bounding box locations, which are taken into account in our regression loss KL Loss

the fast R-CNN head of our network architecture for object detection.

The ground-truth bounding box can also be formulated as a Gaussian distribution, with $\sigma \to 0$, which is a Dirac delta function:

$$
P_D(x) = \delta(x - x_g) \tag{3}
$$

where x_g is the ground-truth bounding box location.

3.2. Bounding Box Regression with KL Loss

The goal of object localization in our context is to estimate $\hat{\Theta}$ that minimize the KL-Divergence between $P_\Theta(x)$ and $P_D(x)$ [43] over N samples:

$$
\hat{\Theta} = \arg\min_\Theta \frac{1}{N} \sum D_{KL}(P_D(x) || P_\Theta(x)) \tag{4}
$$

We use the KL-Divergence as the loss function L_{reg} for bounding box regression. The classification loss L_{cls} remains the same. For a single sample:

$$
\begin{aligned}
L_{reg} &= D_{KL}(P_D(x) || P_\Theta(x)) \\
&= \int P_D(x) \log P_D(x) \mathrm{d}x - \int P_D(x) \log P_\Theta(x) \mathrm{d}x \\
&= \frac{(x_g - x_e)^2}{2\sigma^2} + \frac{\log(\sigma^2)}{2} + \frac{\log(2\pi)}{2} - H(P_D(x))
\end{aligned}
\tag{5}
$$

Shown in Figure 4, when the location x_e is estimated inaccurately, we expect the network to be able to predict larger variance σ^2 so that L_{reg} will be lower. $\log(2\pi)/2$ and $H(P_D(x))$ do not depend on the estimated parameters Θ, hence:

$$
L_{reg} \propto \frac{(x_g - x_e)^2}{2\sigma^2} + \frac{1}{2} \log(\sigma^2) \tag{6}
$$

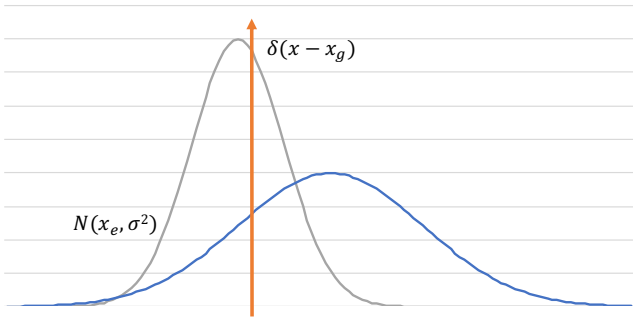

Figure 4: The Gaussian distributions in blue and gray are our estimations. The Dirac delta function in orange is the distribution of the ground-truth bounding box. When the location x_e is estimated inaccurately, we expect the network to be able to predict larger variance σ^2 so that L_{reg} will be lower (blue)

When $\sigma = 1$, KL Loss degenerates to the standard Euclidean loss:

$$L_{reg} \propto \frac{(x_g - x_e)^2}{2} \tag{7}$$

The loss is differentiable w.r.t location estimation x_e and localization standard deviation σ:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x_e} L_{reg} &= \frac{x_e - x_g}{\sigma^2} \\ \frac{\mathrm{d}}{\mathrm{d}\sigma} L_{reg} &= -\frac{(x_e - x_g)^2}{\sigma^{-3}} - \frac{1}{\sigma} \end{aligned} \tag{8}$$

However, since σ is in the denominators, the gradient sometimes can explode at the beginning of training. To avoid gradient exploding, our network predicts $\alpha = \log(\sigma^2)$ instead of σ in practice:

$$L_{reg} \propto \frac{e^{-\alpha}}{2}(x_g - x_e)^2 + \frac{1}{2}\alpha \tag{9}$$

We convert α back to σ during testing.

For $|x_g - x_e| > 1$, we adopt a loss similar to the smooth L_1 loss defined in Fast R-CNN [12]:

$$L_{reg} = e^{-\alpha}(|x_g - x_e| - \frac{1}{2}) + \frac{1}{2}\alpha \tag{10}$$

We initialize the weights of the FC layer for α prediction with random Gaussian initialization. The standard deviation and mean are set to 0.0001 and 0 respectively, so that KL Loss will be similar to the standard smooth L1 loss at the beginning of training. (Equation 9 and Equation 10).

3.3. Variance Voting

After we obtain the variance of predicted location, it is intuitive to vote candidate bounding box location according to the learned variances of neighboring bounding boxes. Shown in Algorithm 1, we change NMS with three lines of code:

Algorithm 1 var voting

$\mathcal{B}$ is $N \times 4$ matrix of initial detection boxes. $\mathcal{S}$ contains corresponding detection scores. $\mathcal{C}$ is $N \times 4$ matrix of corresponding variances. $\mathcal{D}$ is the final set of detections. σ_t is a tunable parameter of var voting. The lines in blue and in green are soft-NMS and var voting respectively.

$\mathcal{B} = \{b_1, .., b_N\}, \mathcal{S} = \{s_1, .., s_N\}, \mathcal{C} = \{\sigma_1^2, .., \sigma_N^2\}$
$\mathcal{D} \leftarrow \{\}$
$\mathcal{T} \leftarrow \mathcal{B}$
while $\mathcal{T} \neq$ empty **do**
  $m \leftarrow$ argmax $\mathcal{S}$
  $\mathcal{T} \leftarrow \mathcal{T} - b_m$
  $\mathcal{S} \leftarrow \mathcal{S} f(IoU(b_m, T))$ ▷ soft-NMS
  $idx \leftarrow IoU(b_m, B) > 0$ ▷ var voting
  $p \leftarrow exp(-(1 - IoU(b_m, \mathcal{B}[idx]))^2/\sigma_t)$
  $b_m \leftarrow p(\mathcal{B}[idx]/\mathcal{C}[idx])/p(1/\mathcal{C}[idx])$
  $\mathcal{D} \leftarrow \mathcal{D} \bigcup b_m$
end while
return $\mathcal{D}, \mathcal{S}$

We vote the location of selected boxes within the loop of standard NMS or soft-NMS [1]. After selecting the detection with maximum score b, $\{x_1, y_1, x_2, y_2, s, \sigma_{x_1}, \sigma_{y_1}, \sigma_{x_2}, \sigma_{y_2}\}$, its new location is computed according to itself and its neighboring bounding boxes. Inspired by soft-NMS, we assign higher weights for boxes that are closer and have lower undertainties. Formally, let x be a coordinate (*e.g.*, x_1) and x_i be the coordinate of ith box. The new coordinate is computed as follow:

$$\begin{aligned} p_i &= e^{-(1-IoU(b_i,b))^2/\sigma_t} \\ x &= \frac{\sum_i p_i x_i / \sigma_{x,i}^2}{\sum_i p_i / \sigma_{x,i}^2} \\ &\text{subject to } IoU(b_i, b) > 0 \end{aligned} \tag{11}$$

σ_t is a tunable parameter of var voting. Two types of neighboring bounding boxes will get lower weights during voting: (1) Boxes with high variances. (2) Boxes that have small IoU with the selected box. Classification score is not involved in the voting, since lower scored boxes may have higher localization confidence. In Figure 5, we provide a visual illustration of var voting. With var voting, the two situations as mentioned earlier in Figure 2 that lead to detection failure can sometimes be avoided.

4. Experiments

To demonstrate our method for accurate object detection, we use two datasets: MS-COCO [35] and PASCAL VOC

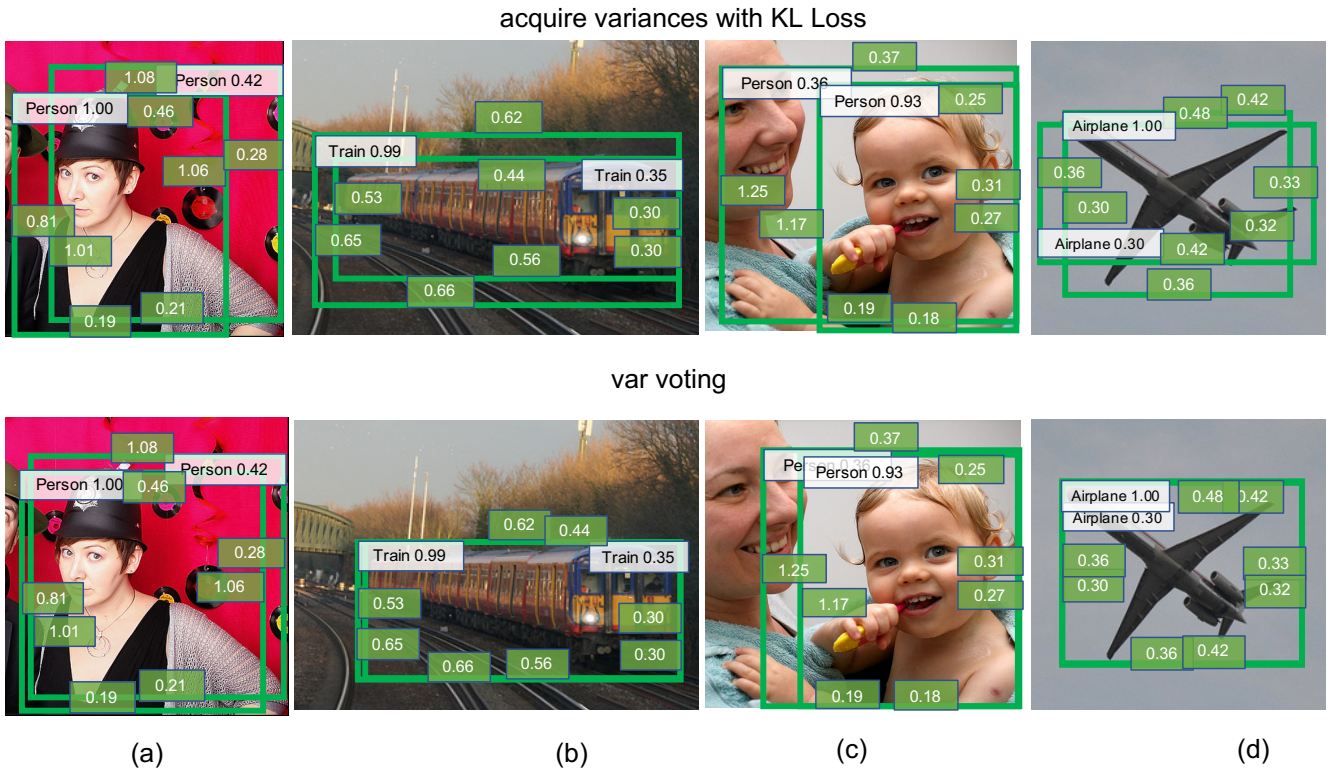

Figure 5: Results of var voting with VGG-16 Faster R-CNN on MS-COCO. The green textbox in the middle of each boundary is the corresponding standard deviation σ we predicted (Equation 2). Two failure situations corresponding to Figure 2 that can be improved by var voting: (a) When each candidate bounding box is inaccurate in some coordinates (women on the right), our var voting can incorporate their localization confidence and produce better boxes. (b) The bounding box with a higher classification score (train 0.99) actually has lower localization confidence than the bounding box with a lower classification score (train 0.35). After var voting, the box scored 0.99 moves towards the correct location. (*better viewed in color*)

2007 [8]. We use four GPUs for our experiments. The training schedule and batch size are adjusted according to the linear scaling rule [15]. For VGG-CNN-M-1024 and VGG-16 Net [48], our implementation is based on Caffe [26]. For ResNet-50 FPN [18, 33] and Mask R-CNN [17], our implementation is based on Detectron [14]. For VGG-16 [48] Faster R-CNN, following `py-faster-rcnn`[1], we train on `train2014` and test on `val2014`. For other object detection networks, we train and test on the newly defined `train2017` and `val2017` respectively. We set σ_t to 0.02. Unless specified, all hyper-parameters are set to default[2].

4.1. Ablation Study

We evaluate the contribution of each element in our detection pipeline: KL Loss, soft-NMS and var voting with VGG-16 Faster R-CNN. The detailed results are shown in Table 1.

KL Loss: Surprisingly, simply training with KL Loss greatly improves the AP by **2.8%**, which is also observed on ResNet-50 Faster R-CNN and Mask R-CNN (**1.5%** and **0.9%** improvement respectively, shown in Table 3 and Table 4). First, by learning to predict high variances for samples with high uncertainties during training, the network can learn more from useful samples. Second, the gradient for localization can be adaptively controlled by the network during training (Equation 8), which encourages the network to learn more accurate object localization. Third, KL Loss incorporates learning localization confidence which can potentially help the network to learn more discriminative features.

The learned variances through our KL Loss are interpretable. Our network will output higher variances for challenging object boundaries, which can be useful in vision applications like self-driving cars and robotics. The first row

[1] github.com/rbgirshick/py-faster-rcnn
[2] github.com/facebookresearch/Detectron

KL Loss	soft-NMS	var voting	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L	AR^1	AR^{10}	AR^{100}
			23.6	44.6	22.8	6.7	25.9	36.3	23.3	33.6	34.3
	✓		24.8	45.6	24.6	7.6	27.2	37.6	23.4	39.2	42.2
✓			26.4	47.9	26.4	7.4	29.3	41.2	25.2	36.1	36.9
✓		✓	27.8	48.0	28.9	8.1	31.4	42.6	26.2	37.5	38.3
✓	✓		27.8	49.0	28.5	8.4	30.9	42.7	25.3	41.7	44.9
✓	✓	✓	**29.1**	**49.1**	**30.4**	**8.7**	**32.7**	**44.3**	**26.2**	**42.5**	**45.5**

Table 1: The contribution of each element in our detection pipeline on MS-COCO. The baseline model is VGG-16 Faster R-CNN

method	latency (ms)
baseline	99
ours	101

Table 2: Inference time comparison on MS-COCO with VGG-16 Faster R-CNN on a GTX 1080 Ti GPU, CUDA 8 [38] and CUDNN 6 [3]

fast R-CNN head	backbone	KL Loss	AP
2mlp head	FPN		37.9
		✓	$38.5^{+0.6}$
2mlp head + mask	FPN		38.6
		✓	$39.5^{+\mathbf{0.9}}$
conv5 head	RPN		36.5
		✓	$38.0^{+\mathbf{1.5}}$

Table 3: Comparison of different fast R-CNN heads. The model is ResNet-50 Faster R-CNN

of Figure 5 shows some qualitative examples of the standard deviation learned through our KL Loss.

Soft-NMS: As expected, soft-NMS performs consistently on both baseline and our network trained with KL Loss. It improves the AP by 1.2% and 1.4% on the baseline and our network respectively, shown in Table 1.

Variance Voting: Finally, with var voting, the AP is further improved to **29.1%**. We made the observation that improvement mainly comes from the more accurate localization. Notice that the AP^{50} is only improved by 0.1%. However, AP^{75}, AP^M and AP^L are improved by 1.8%, 1.8%, and 1.6% respectively, shown in Table 1. This indicates that classification confidence is not always associated with localization confidence. Therefore, learning localization confidence apart from classification confidence is important for more accurate object localization.

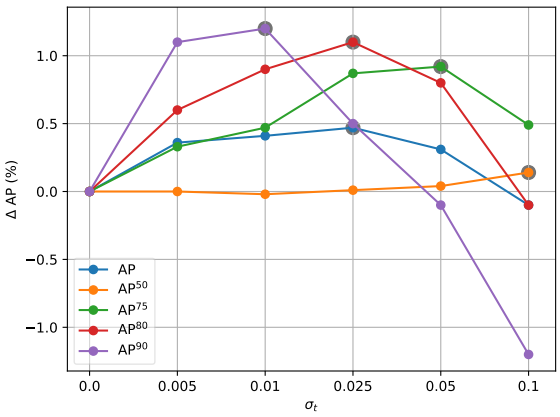

Figure 6: Varying σ_t for var voting with ResNet-50 Faster R-CNN. (*better viewed in color*)

We also found that var voting and soft-NMS can work well together. Applying var voting with the standard NMS improves the AP by 1.4%. Applying var voting after soft-NMS still can improve the AP by 1.3%. We argue that soft-NMS is good at scoring candidate bounding boxes which improve overall performance, whereas var voting is good at refining those selected bounding boxes for more accurate object localization. The second row of Figure 5 shows some qualitative examples of our var voting.

Shown in Figure 6, we test the sensitivity of the tunable parameter σ_t for var voting. When $\sigma_t = 0$, var voting is not activated. We observe that the AP^{75}, AP^{80} and AP^{90} can be significantly affected by σ_t, while AP^{50} is less sensitive to σ_t. Acceptable values of σ_t varies from around $0.005 \sim 0.05$. We use $\sigma_t = 0.02$ in all experiments.

Inference Latency: We also evaluate the inference time of our improved VGG-16 Faster R-CNN on a single GTX 1080 Ti GPU with CUDA 8 and CUDNN 6, as it is crucial for resource-limited applications [50, 20, 23, 19, 32].

	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
baseline [14]	38.6	**59.8**	55.3	47.7	34.4	11.3
MR-CNN [11]	38.9	**59.8**	55.5	48.1	$34.8^{+0.4}$	$11.9^{+0.6}$
soft-NMS [1]	39.3	59.7	**55.6**	**48.9**	$35.9^{+1.5}$	$12.0^{+0.7}$
IoU-NMS+Refine [27]	39.2	57.9	53.6	47.4	$36.5^{+2.1}$	$16.4^{+5.1}$
KL Loss	$39.5^{+0.9}$	58.9	54.4	47.6	$36.0^{+1.6}$	$15.8^{+4.5}$
KL Loss+var voting	$39.9^{+1.3}$	58.9	54.4	47.7	$36.4^{+2.0}$	$17.0^{+5.7}$
KL Loss+var voting+soft-NMS	$\mathbf{40.4}^{+1.8}$	58.7	54.6	48.5	$\mathbf{37.5}^{+3.3}$	$\mathbf{17.5}^{+6.2}$

Table 4: Comparisons of different methods for accurate object detection on MS-COCO. The baseline model is ResNet-50-FPN Mask R-CNN. We improve the baseline by $\approx 2\%$ in AP

Shown in Table 2, our approach only increases **2ms** latency on GPU. Different from IoUNet [27] which uses `2mlp` head for IoU prediction, our approach only requires a 4096×324 fully-connected layer for the localization confidence prediction.

RoI Box Head: We test the effectiveness of KL Loss with different RoI box heads on a deeper backbone: ResNet-50. `res5/conv5` head consists of 9 convolutional layers which can be applied to each RoI as fast R-CNN head. `2mlp` head consists of two fully connected layers. `res5` head can learn more complex representation than the commonly used `2mlp` head. Shown in Table 3, KL Loss can improve the AP by **0.9%** with mask. KL Loss can further improve the AP by **1.5%** with `conv5` head. We hypothesize that the localization variance is much more challenging to learn than localization, therefore KL Loss can benefit more from the expressiveness of `conv5` head. Since `conv5` head is not commonly used in recent state-of-the-art detectors, we still adopt the `2mlp` head in the following experiments.

4.2. Accurate Object Detection

Table 4 summarizes the performance of different methods for accurate object detection on ResNet-50-FPN Mask R-CNN. With KL Loss, the network can learn to adjust the gradient for ambiguous bounding boxes during training. As a result, Mask R-CNN trained with KL Loss performs significantly better than the baseline for high overlap metrics like AP^{90}. Variance Voting improves the localization results by voting the location according to the localization confidences of neighboring bounding boxes. AP^{80} and AP^{90} are further improved by 0.4% and 1.2% respectively. Variance Voting is also compatible with soft-NMS. Variance Voting combined with soft-NMS improves the AP^{90} and the overall AP of the final model by **6.2%** and **1.8%** respectively. Compared with IoUNet [27]: (1) our variance and localization are learned together with KL Loss, which improves the performance. (2) KL Loss does not require a separate 2mlp head for learning localization confidence, which introduces nearly no additional computation. (3) var voting does not require iterative refinement, which is much faster.

backbone	method	mAP
VGG-CNN-M-1024	baseline	60.4
	KL Loss	62.0
	KL Loss+var voting	62.8
	KL Loss+var voting+soft-NMS	**63.6**
VGG-16	baseline	68.7
	QUBO (tabu) [46]	60.6
	QUBO (greedy) [46]	61.9
	soft-NMS [1]	70.1
	KL Loss	69.7
	KL Loss+var voting	70.2
	KL Loss+var voting+soft-NMS	**71.6**

Table 5: Comparisons of different approaches on PASCAL VOC 2007 with Faster R-CNN.

We further evaluate our approach on the feature pyramid network (ResNet-50 FPN) [33, 18], shown in Table 6.

For fast R-CNN version, training with KL Loss increases the baseline by 0.4%. After applying var voting along with soft-NMS, our model achieves **38.0%** in AP, which outperforms both IoU-NMS and soft-NMS baselines. Training end-to-end with KL Loss can help the network learn more discriminative features, which improves the baseline AP by **0.6%**. The final model achieves 39.2% in AP, which improves the baseline by 1.3%.

4.3. Experiments on PASCAL VOC 2007

Even though our approach is designed for large scale object detection, it could also generalize well on small datasets. We perform experiments with Faster R-CNN [42] on PASCAL VOC 2007, which consists of about 5k `voc_2007_trainval` images and 5k `voc_2007_test` images over 20 object categories. Backbone networks: VGG-CNN-M-1024 and VGG-16

type	method	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
fast R-CNN	baseline (1x schedule) [14]	36.4	**58.4**	39.3	**20.3**	39.8	48.1
	baseline (2x schedule) [14]	36.8	**58.4**	39.5	19.8	39.5	49.5
	IoU-NMS [27]	37.3	56.0	-	-	-	-
	soft-NMS [1]	37.4	58.2	41.0	**20.3**	40.2	50.1
	KL Loss	37.2	57.2	39.9	19.8	39.7	50.1
	KL Loss+var voting	37.5	56.5	40.1	19.4	40.2	51.6
	KL Loss+var voting+soft-NMS	**38.0**	56.4	**41.2**	19.8	**40.6**	**52.3**
Faster R-CNN	baseline (1x schedule) [14]	36.7	58.4	39.6	21.1	39.8	48.1
	IoU-Net [27]	37.0	58.3	-	-	-	-
	IoU-Net+IoU-NMS [27]	37.6	56.2	-	-	-	-
	baseline (2x schedule) [14]	37.9	59.2	41.1	21.5	41.1	49.9
	IoU-Net+IoU-NMS+Refine [27]	38.1	56.3	-	-	-	-
	soft-NMS[1]	38.6	**59.3**	42.4	21.9	**41.9**	50.7
	KL Loss	38.5	57.8	41.2	20.9	41.2	51.5
	KL Loss+var voting	38.8	57.8	41.6	21.0	41.5	52.0
	KL Loss+var voting+soft-NMS	**39.2**	57.6	**42.5**	21.2	41.8	**52.5**

Table 6: Performance comparison with FPN ResNet-50 on MS-COCO

Net [48] are tested.

Shown in Table 5, we compare our approach with soft-NMS and quadratic unconstrained binary optimization (QUBO [46]). For QUBO, we test both greedy and classical tabu solver (we manually tuned the penalty term for both solvers to get better performance). We observe that it is much worse than the standard NMS, though it was reported to be better for pedestrian detection. We hypothesize that QUBO is better at pedestrian detection since there are more occluded bounding boxes [47]. For VGG-CNN-M-1024, training with var voting improves the mAP by **1.6%**. var voting further improves the mAP by **0.8%**. For VGG-16, our approach improves the mAP by **2.9%**, combined with soft-NMS. We notice that var voting could still improve performance even after soft-NMS is applied to the initial detection boxes. This observation is consistent with our experiments on MS-COCO (Table 1).

5. Conclusion

To conclude, the uncertainties in large-scale object detection datasets can hinder the performance of state-of-the-art object detectors. Classification confidence is not always strongly related to localization confidence. In this paper, a novel bounding box regression loss with uncertainty is proposed for learning more accurate object localization. By training with KL Loss, the network learns to predict localization variance for each coordinate. The resulting variances empower var voting, which can refine the selected bounding boxes via voting. Compelling results are demonstrated for VGG-16 Faster R-CNN, ResNet-50 FPN and Mask R-CNN on both MS-COCO and PASCAL VOC 2007.

Acknowledgement

This research was partially supported by National Key R&D Program of China (No. 2017YFA0700800).

We would love to express our appreciation to Prof. Kris Kitani and Dr. Jian Sun for the useful discussions during this research.

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms – improving object detection with one line of code. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5562–5570. IEEE, 2017. 2, 4, 7, 8

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *arXiv preprint arXiv:1712.00726*, 2017. 1, 2

[3] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014. 6

[4] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 2

[5] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1

[7] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider. Motion prediction of traffic ac-

tors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018. 2

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 4

[9] D. Feng, L. Rosenbaum, and K. Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273. IEEE, 2018. 2

[10] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer. Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection. *arXiv preprint arXiv:1809.05590*, 2018. 2

[11] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015. 2, 7

[12] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2, 4

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 2, 3

[14] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. github.com/facebookresearch/detectron, 2018. 5, 7, 8

[15] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5

[16] M. Gualtieri and R. Platt. Learning 6-dof grasping and pick-place using attention focus. *arXiv preprint arXiv:1806.06134*, 2018. 2

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 2, 3, 5

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5, 7

[19] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018. 7

[20] Y. He, X. Liu, H. Zhong, and Y. Ma. Addressnet: Shift-based primitives for efficient convolutional neural networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1213–1222. IEEE, 2019. 7

[21] Y. He, X. Ma, X. Luo, J. Li, M. Zhao, B. An, and X. Guan. Vehicle traffic driven camera placement for better metropolis security surveillance. *arXiv preprint arXiv:1705.08508*, 2017. 2

[22] Y. He, X. Zhang, M. Savvides, and K. Kitani. Softer-nms: Rethinking bounding box regression for accurate object detection. *arXiv preprint arXiv:1809.08545*, 2018.

[23] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1389–1397, 2017. 7

[24] J. H. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. In *CVPR*, pages 6469–6477, 2017. 2

[25] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. *arXiv preprint arXiv:1711.11575*, 8, 2017. 2

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 5

[27] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018. 2, 7, 8

[28] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017. 2

[29] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3, 2017. 2

[30] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 2

[31] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. 2

[32] Y. Liang, Z. Yang, K. Zhang, Y. He, J. Wang, and N. Zheng. Single image super-resolution via a lightweight residual convolutional neural network. *arXiv preprint arXiv:1703.08173*, 2017. 7

[33] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, page 3, 2017. 5, 7

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[35] 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 *European conference on computer vision*, pages 740–755. Springer, Cham, 2014. 1, 4

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[37] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[38] J. Nickolls, I. Buck, M. Garland, and K. Skadron. Scalable parallel programming with cuda. In *ACM SIGGRAPH 2008 classes*, page 16. ACM, 2008. 6

[39] Y. Rao, D. Lin, J. Lu, and J. Zhou. Learning globally optimized object detector via policy gradient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6190–6198, 2018. 2

[40] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 7464–7473. IEEE, 2017. 1

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 3, 7

[43] C. Robert. Machine learning, a probabilistic perspective, 2014. 3

[44] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, (5):562–569, 1971. 2

[45] R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian Conference on Computer Vision*, pages 290–306. Springer, 2014. 2

[46] S. Rujikietgumjorn and R. T. Collins. Optimized pedestrian detection for multiple and occluded people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3690–3697, 2013. 7, 8

[47] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 1, 8

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 8

[49] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 516–520. ACM, 2016. 2

[50] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7

[51] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2