Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection

Xiang Li[1,2], Wenhai Wang[3], Xiaolin Hu[4], Jun Li[1], Jinhui Tang[1], and Jian Yang[1*]

Abstract

Localization Quality Estimation (LQE) is crucial and popular in the recent advancement of dense object detectors since it can provide accurate ranking scores that benefit the Non-Maximum Suppression processing and improve detection performance. As a common practice, most existing methods predict LQE scores through vanilla convolutional features shared with object classification or bounding box regression. In this paper, we explore a completely novel and different perspective to perform LQE – based on the learned distributions of the four parameters of the bounding box. The bounding box distributions are inspired and introduced as "General Distribution" in GFLV1, which describes the uncertainty of the predicted bounding boxes well. Such a property makes the distribution statistics of a bounding box highly correlated to its real localization quality. Specifically, a bounding box distribution with a sharp peak usually corresponds to high localization quality, and vice versa. By leveraging the close correlation between distribution statistics and the real localization quality, we develop a considerably lightweight Distribution-Guided Quality Predictor (DGQP) for reliable LQE based on GFLV1, thus producing GFLV2. To our best knowledge, it is the first attempt in object detection to use a highly relevant, statistical representation to facilitate LQE. Extensive experiments demonstrate the effectiveness of our method. Notably, GFLV2 (ResNet-101) achieves 46.2 AP at 14.6 FPS, surpassing the previous state-of-the-art ATSS baseline (43.6 AP at 14.6 FPS) by absolute 2.6 AP on COCO `test-dev`*, without sacrificing the efficiency both in training and inference. Codes are available at https://github.com/implus/GFocalV2.*

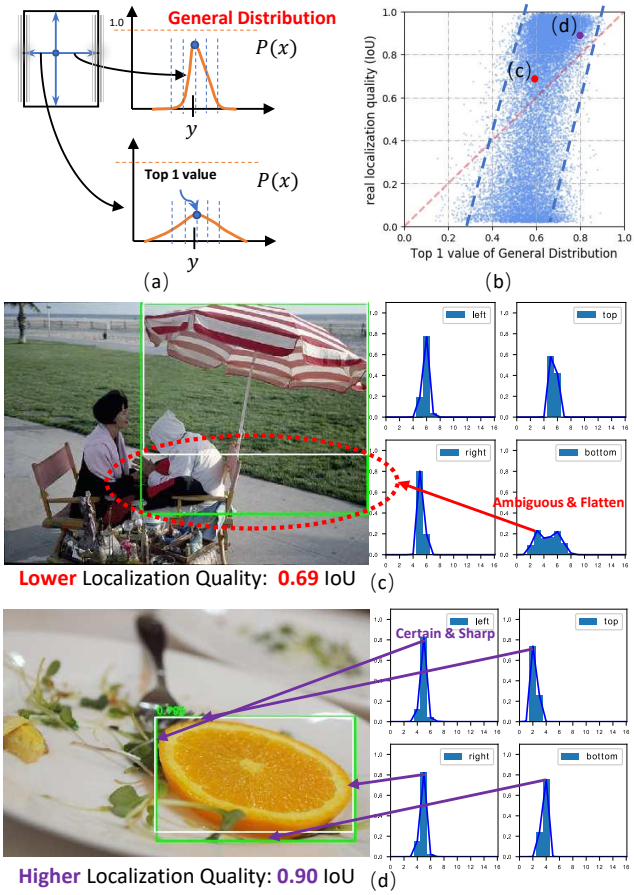

Figure 1: Motivation of utilizing the highly relevant statistics of learned bounding box distributions to guide the better generation of its estimated localization quality. (a): The illustration of General Distribution in GFLV1 [18] to represent bounding boxes, which models the probability distribution of the predicted edges. (b): The scatter diagram of the relation between Top-1 (mean of four sides) value of General Distribution of predicted boxes and their real localization quality (IoU between the prediction and ground-truth), calculated over all validation images on COCO [22] dataset, based on GFLV1 model. (c) and (d): Two specific examples from (b), where the sharp distribution corresponds to higher quality, whilst the flat one stands for lower quality usually. Green: predicted bounding boxes; White: ground-truth bounding boxes.

*Corresponding author. Xiang Li, Jun Li and Jian Yang are from PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology.

[1]Nanjing University of Science and Technology [2]Momenta [3]Nanjing University [4]Tsinghua University {xiang.li.implus, jinhuitang, csjyang}@njust.edu.cn, wangwenhai362@163.com, xlhu@mail.tsinghua.edu.cn, junl.mldl@gmail.com

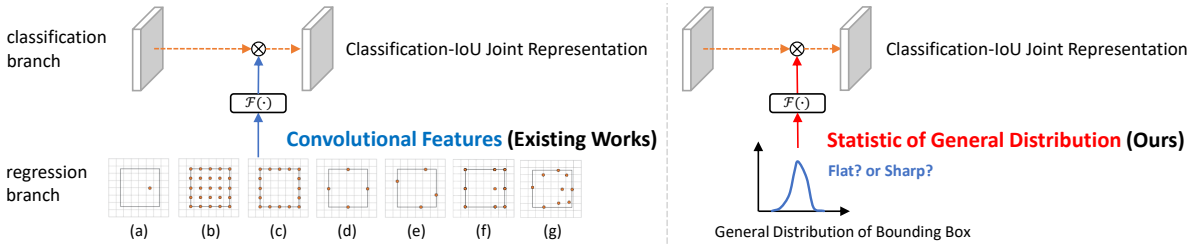

Figure 2: Comparisons of input features for predicting localization quality between existing works (left) and ours (right). Existing works focus on different *spatial locations* of convolutional features, including (a): *point* [28, 29, 30, 33, 40, 36, 14, 18], (b): *region* [13], (c): *border* [27] *dense* points, (d): *border* [27] *middle* points, (e): *border* [27] *extreme* points, (f): *regular* sampling points [39], and (g): *deformable* sampling points [4, 5]. In contrast, we use the statistic of learned box distribution to produce reliable localization quality.

1. Introduction

Dense object detector [28, 23, 42, 33, 18, 27] which directly predicts pixel-level object categories and bounding boxes over feature maps, becomes increasingly popular due to its elegant and effective framework. One of the crucial techniques underlying this framework is Localization Quality Estimation (LQE). With the help of better LQE, high-quality bounding boxes tend to score higher than low-quality ones, greatly reducing the risk of mistaken suppression in Non-Maximum Suppression (NMS) processing.

Many previous works [28, 29, 30, 33, 40, 36, 14, 18, 39, 43, 27] have explored LQE. For example, the YOLO family [28, 29, 30] first adopt *Objectness* to describe the localization quality, which is defined as the Intersection-over-Union (IoU) between the predicted and ground-truth box. After that, *IoU* is further explored and proved to be effective in IoU-Net [13], IoU-aware [36], PAA [14], GFLV1 [18] and VFNet [39]. Recently, FCOS [33] and ATSS [40] introduce *Centerness*, the distance degree to the object center, to suppress low-quality detection results. Generally, the aforementioned methods share a common characteristic that they are all based on *vanilla convolutional features*, e.g., features of points, borders or regions (see Fig. 2 (a)-(g)), to estimate the localization quality.

Different from previous works, in this paper, we explore a brand new perspective to conduct LQE – by directly utilizing *the statistics of bounding box distributions*, instead of using the *vanilla convolutional features* (see Fig. 2). Here the *bounding box distribution* is introduced as "General Distribution" in GFLV1 [18], where it learns a discrete probability distribution of each predicted edge (Fig. 1 (a)) for describing the uncertainty of bounding box regression. Interestingly, we observe that the statistic of the General Distribution has a strong correlation with its real localization quality, as illustrated in Fig. 1 (b). More specifically in Fig. 1 (c) and (d), the shape (flatness) of bounding box distribution can clearly reflect the localization quality of the predicted results: the sharper the distribution, the more accurate the predicted bounding box, and vice versa. Consequently, it can potentially be easier and very efficient to conduct better LQE by the guidance of the distribution information, as the input (distribution statistics of bounding boxes) and the output (LQE scores) are highly correlated.

Inspired by the strong correlation between the distribution statistics and LQE scores, we propose a very lightweight sub-network with only dozens of (e.g., 64) hidden units, on top of these distribution statistics to produce reliable LQE scores, significantly boosting the detection performance. Importantly, it brings negligible additional computation cost in practice and almost does not affect the training/inference speed of the basic object detectors. In this paper, we term this lightweight sub-network as Distribution-Guided Quality Predictor (DGQP), since it relies on the guidance of distribution statistics for quality predictions.

By introducing the lightweight DGQP that predicts reliable LQE scores via statistics of bounding box distributions, we develop a novel dense object detector based on the framework of GFLV1, thus termed GFLV2. To verify the effectiveness of GFLV2, we conduct extensive experiments on the challenging benchmark COCO [22]. Notably, based on ResNet-101 [11], GFLV2 achieves impressive detection performance (46.2 AP), i.e., 2.6 AP gains over the state-of-the-art ATSS baseline (43.6 AP) on COCO `test-dev`, under the same training schedule and without sacrificing the efficiency both in training and inference.

In summary, our contributions are as follows:

• To our best knowledge, our work is the first to bridge the statistics of bounding box distributions and localization quality estimation in an end-to-end dense object detection framework.

• The proposed GFLV2 is considerably lightweight and cost-free in practice. It can also be easily plugged into most dense object detectors with a consistent gain of $\sim$2 AP, and without loss of training/inference speed.

• Our GFLV2 (Res2Net-101-DCN) achieves very competitive 53.3 AP (multi-scale testing) on COCO dataset among dense object detectors.

2. Related Work

Formats of LQE: Early popular object detectors [9, 31, 1, 10] simply treat the classification confidence as the formulation of LQE score, but there is an obvious inconsistency between them, which inevitably degrades the detection performance. To alleviate this problem, AutoAssign [43] and BorderDet [27] employ additional localization features to rescore the classification confidence, but they still lack an explicit definition of LQE.

Recently, FCOS [33] and ATSS [40] introduce a novel format of LQE, termed *Centerness*, which depicts the distance degree to the center of the object. Although *Centerness* is effective, recent researches [18, 39] show that it has certain limitations and may be suboptimal for LQE. SABL [35] introduces boundary buckets for coarse localization, and utilizes the averaged bucketing confidence as a formulation of LQE.

After years of technical iterations [28, 29, 30, 13, 34, 12, 36, 14, 18, 39], *IoU* has been deeply studied and becomes increasingly popular as an excellent measurement of LQE. *IoU* is first known as the *Objectness* in YOLO [28, 29, 30], where the network is supervised to produce estimated IoUs between predicted boxes and ground-truth ones, to reduce ranking basis during NMS. Following the similar paradigm, IoU-Net [13], Fitness NMS [34], MS R-CNN [12], IoU-aware [36], PAA [14] utilize a separate branch to perform LQE in the *IoU* form. Concurrently, GFLV1 [18] and VFNet [39] demonstrate a more effective format, by merging the classification score with *IoU* to reformulate a joint representation. Due to its great success [18, 39], we build our GFLV2 based on the Classification-IoU Joint Representation [18], and develop a novel approach for reliable LQE.

Input Features for LQE: As shown in the left part of Fig. 2, previous works directly use convolutional features as input for LQE, which only differ in the way of spatial sampling. Most existing methods [28, 29, 30, 33, 40, 36, 14, 18] adopt the point features (see Fig. 2 (a)) to produce LQE scores for high efficiency. IoU-Net [13] predicts IoU based on the region features as shown in Fig. 2 (b). BorderDet [27] designs three types of border-sensitive features (see Fig. 2 (c)-(e)) to facilitate LQE. Similar with BorderDet, a star-shaped sampling manner (see Fig. 2 (f)) is designed in VFNet [39]. Alternatively, HSD [2] and RepPoints [38, 4] focus on features with learned locations (see Fig. 2 (g)) via the deformable convolution [5, 46].

The aforementioned methods mainly focus on extracting discriminating convolutional features with various spatial aspects for better LQE. Different from previous methods, our proposed GFLV2 is designed in an artful perspective: predicting LQE scores by its directly correlated variables—the statistics of bounding box distributions (see the right part of Fig. 2). As later demonstrated in Table 3, compared with convolutional features shown in Fig. 2 (a)-(g), the statistics of bounding box distributions achieve an impressive efficiency and a high accuracy simultaneously.

3. Method

In this section, we first briefly review the Generalized Focal Loss (i.e., GFLV1 [18]), and then derive the proposed GFLV2 based on the relevant concepts and formulations.

3.1. Generalized Focal Loss V1

Classification-IoU Joint Representation: This representation is the key component in GFLV1, which is designed to reduce the inconsistency between localization quality estimation and object classification during training and inference. Concretely, given an object with category label $c \in \{1, 2, ..., m\}$ (m indicates the total number of categories), GFLV1 utilizes the classification branch to produce the joint representation of Classification and IoU as $\mathbf{J} = [J_1, J_2, ..., J_m]$, which satisfies:

$$J_i = \begin{cases} \mathrm{IoU}(b_{pred}, b_{gt}), & \text{if } i = c; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathrm{IoU}(b_{pred}, b_{gt})$ denotes the IoU between the predict bounding box b_{pred} and the ground truth b_{gt}.

General Distribution of Bounding Box Representation: Modern detectors [31, 21, 33] usually describe the bounding box regression by Dirac delta distribution: $y = \int_{-\infty}^{+\infty} \delta(x - y)x \,\mathrm{d}x$. Unlike them, GFLV1 introduces a flexible General Distribution $P(x)$ to represent the bounding box, where each edge of the bounding box can be formulated as: $\hat{y} = \int_{-\infty}^{+\infty} P(x)x \,\mathrm{d}x = \int_{y_0}^{y_n} P(x)x \,\mathrm{d}x$, under a predefined output range of $[y_0, y_n]$. To be compatible with the convolutional networks, the continuous domain is converted into the discrete one, via discretizing the range $[y_0, y_n]$ into a list $[y_0, y_1, ..., y_i, y_{i+1}, ..., y_{n-1}, y_n]$ with even intervals Δ ($\Delta = y_{i+1} - y_i, \forall i \in \{0, 1, ..., n-1\}$). As a result, given the discrete distribution property $\sum_{i=0}^{n} P(y_i) = 1$, the estimated regression value $\hat{y}$ can be presented as:

$$\hat{y} = \sum_{i=0}^{n} P(y_i)y_i. \tag{2}$$

Compared with the Dirac delta distribution, the General Distribution $P(x)$ can faithfully reflect the prediction quality (see Fig. 1 (c)-(d)), which is the cornerstone of this work.

3.2. Generalized Focal Loss V2

Decomposed Classification-IoU Representation: Although the joint representation solves the inconsistency problem [18] between object classification and quality estimation during training and testing, there are still some limitations in using only the classification branch to predict the joint representation. In this work, we decompose

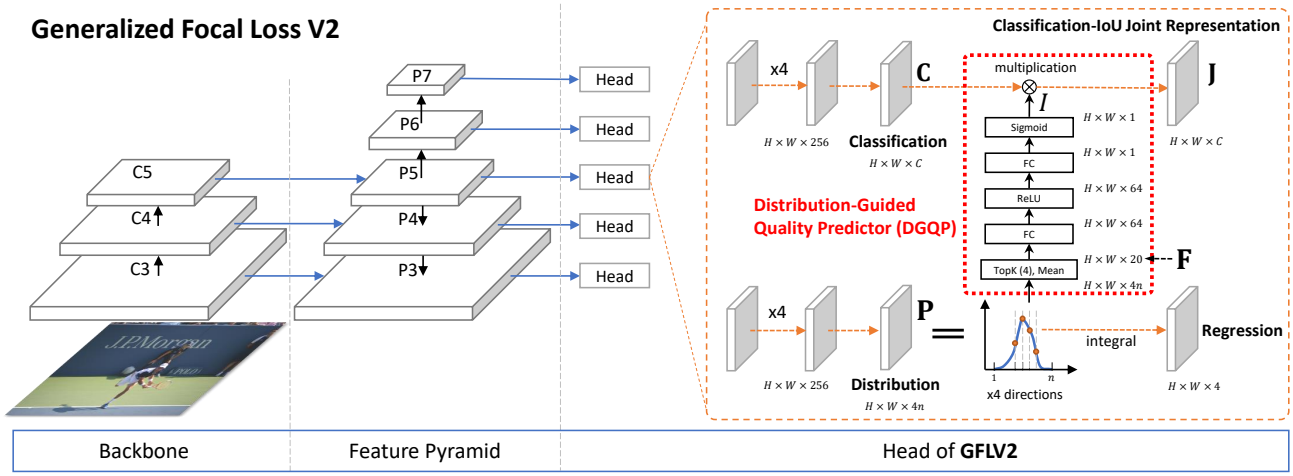

Figure 3: The illustration of the proposed Generalized Focal Loss V2 (GFLV2), where a novel and tiny Distribution-Guided Quality Predictor (DGQP) uses the statistics of learned bounding box distributions to facilitate generating reliable IoU quality estimations.

the joint representation explicitly by leveraging information from both classification ($\mathbf{C}$) and regression (I) branches:

$$\mathbf{J} = \mathbf{C} \times I, \tag{3}$$

where $\mathbf{C} = [C_1, C_2, ..., C_m], C_i \in [0, 1]$ denotes the Classification Representation of total m categories, and $I \in [0, 1]$ is a scalar that stands for the IoU Representation.

Although $\mathbf{J}$ is decomposed into two components, we use the final joint formulation (i.e., $\mathbf{J}$) in both the training and testing phases, so it can still avoid the inconsistency problem as mentioned in GFLV1. Specifically, we first combine $\mathbf{C}$ from the classification branch and I from the proposed Distribution-Guided Quality Predictor (DGQP) in regression branch, into the unified form $\mathbf{J}$. Then, $\mathbf{J}$ is supervised by Quality Focal loss (QFL) as proposed in [18] during training, and used directly as NMS score in inference.

Distribution-Guided Quality Predictor: DGQP is the key component of GFLV2. It delivers the statistics of the learned General Distribution $\mathbf{P}$ into a tiny sub-network (see red dotted frame in Fig. 3) to obtain the predicted IoU scalar I, which helps to generate high-quality Classification-IoU Joint Representation (Eq. (3)). Following GFLV1 [18], we adopt the relative offsets from the location to the four sides of a bounding box as the regression targets, which are represented by the General Distribution. For convenience, we mark the left, right, top and bottom sides as $\{l, r, t, b\}$, and define the discrete probabilities of the w side as $\mathbf{P}^w = [P^w(y_0), P^w(y_1), ..., P^w(y_n)]$, where $w \in \{l, r, t, b\}$.

As illustrated in Fig. 1, the flatness of the learned distribution is highly related to the quality of the final detected bounding box, and some relevant statistics can be used to reflect the flatness of the General Distribution. As a result, such statistical features have a very strong correlation with the localization quality, which will ease the training difficulty and improves the quality of estimation. Practically, we recommand to choose the Top-k values along with the mean value of each distribution vector $\mathbf{P}^w$, and concatenate them as the basic statistical feature $\mathbf{F} \in \mathbb{R}^{4(k+1)}$:

$$\mathbf{F} = \text{Concat}\big(\{\text{Topkm}(\mathbf{P}^w) \mid w \in \{l, r, t, b\}\}\big), \tag{4}$$

where $\text{Topkm}(\cdot)$ denotes the joint operation of calculating Top-k values and their mean value. $\text{Concat}(\cdot)$ means the channel concatenation. Selecting Top-k values and their mean value as the input statistics have two benefits:

- Since the sum of $\mathbf{P}^w$ is fixed (i.e., $\sum_{i=0}^{n} P^w(y_i) = 1$), Top-$k$ values along with their mean value can basically reflect the flatness of the distribution: the larger, the sharper; the smaller, the flatter;
- Top-k and mean values can make the statistical feature insensitive to its relative offsets over the distribution domain (see Fig. 4), resulting in a robust representation which is not affected by object scales.

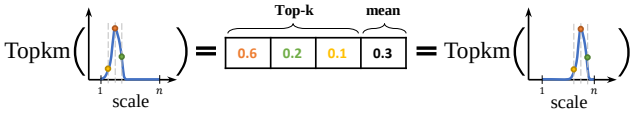

Figure 4: Topkm$(\cdot)$ feature is robust to object scales.

Given the statistical feature $\mathbf{F}$ of General Distribution as input, we design a very tiny sub-network $\mathcal{F}(\cdot)$ to predict the final IoU quality estimation. The sub-network has only two Fully-Connected (FC) layers, which are followed by ReLU [16] and Sigmoid, respectively. Consequently, the IoU scalar I can be calculated as:

$$I = \mathcal{F}(\mathbf{F}) = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{F})), \tag{5}$$

where δ and σ refer to the ReLU and Sigmoid, respectively. $\mathbf{W}_1 \in \mathbb{R}^{p \times 4(k+1)}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times p}$. k denotes the Top-k parameter and p is the channel dimension of the hidden layer ($k = 4$, $p = 64$ is a typical setting in our experiment).

Complexity: The overall architecture of GFLV2 is illustrated in Fig. 3. It is worth noting that the DGQP module is very lightweight. First, it only brings thousands of ad-

Top-k (4)	Mean	Var	Dim	AP	AP_{50}	AP_{75}
✓			16	40.8	58.5	44.2
	✓		4	40.2	58.5	43.6
		✓	4	40.3	58.3	43.7
✓	✓		20	**41.1**	**58.8**	**44.9**
✓		✓	20	40.9	58.5	44.7
✓	✓	✓	24	40.9	58.4	44.7

Table 1: Performances of different combinations of the input statistics by fixing $k = 4$ and $p = 64$. "Mean" denotes the mean value, "Var" denotes the variance number, and "Dim" is short for "Dimension" that means the total amount of the input channels.

ditional parameters, which are negligible compared to the number of parameters of the entire detection model. For example, for the model with ResNet-50 [11] and FPN [20], the extra parameters of the DGQP module only account for ∼0.003%. Second, the computational overhead of the DGQP module is also very small due to its extremely light structure. As shown in Table 5 and 8, the use of the DGQP module hardly reduces the training and inference speed of the original detector in practice.

4. Experiment

Experimental Settings: We conduct experiments on COCO benchmark [22], where `trainval35k` with 115K images is used for training and `minival` with 5K images for validation in our ablation study. Besides, we obtain the main results on `test-dev` with 20K images from the evaluation server. All results are produced under mmdetection [3] for fair comparisons, where the default hyper-parameters are always adopted. Unless otherwise stated, we apply the standard 1x learning schedule (12 epochs) without multi-scale training for the ablation study, based on ResNet-50 [11] backbone. The training/testing details follow the descriptions in previous works [18, 4].

4.1. Ablation Study

Combination of Input Statistics: In addition to the pure Top-k values, there are some statistics that may reflect more characteristics of the distributions, such as the mean and variance of these Top-k numbers. Therefore, we conduct experiments to investigate the effect of their combinations as input, by fixing $k = 4$ and $p = 64$. From Table 1, we observe that the Top-4 values with their mean number perform best. Therefore, we default to use such a combination as the standard statistical input in the following experiments.

Structure of DGQP (i.e., k, p): We then examine the impact of different parameters of k, p in DGQP on the detection performance. Specifically, we report the effect of k and p by fixing one and varying another in Table 2. It is observed that $k = 4, p = 64$ steadily achieves the optimal accuracy among various combinations.

k	p	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0	–	40.2	58.6	43.4	23.0	44.3	53.0
1	64	40.2	58.3	44.0	23.4	44.1	52.1
2		40.9	58.5	44.6	23.3	44.8	**53.5**
3		40.9	58.5	44.6	**24.3**	**44.9**	52.3
4		**41.1**	**58.8**	**44.9**	23.5	**44.9**	53.3
8		41.0	58.6	44.5	23.5	44.5	53.4
16		40.8	58.5	44.4	23.4	44.2	53.1
4	8	40.9	58.4	44.5	23.1	44.5	52.6
	16	40.8	58.3	44.1	23.3	44.6	52.0
	32	40.9	58.7	44.3	23.1	44.6	53.2
	64	**41.1**	**58.8**	**44.9**	**23.5**	**44.9**	**53.3**
	128	40.9	58.3	44.6	23.2	44.4	52.7
	256	40.7	58.3	44.4	23.4	44.3	52.9

Table 2: Performances of various k, p in DGQP. $k = 0$ denotes the baseline version without the usage of DGQP (i.e., GFLV1).

Input Feature		AP	AP_{50}	AP_{75}	FPS
Baseline (ATSS [40] w/ QFL [18])		39.9	58.5	43.0	**19.4**
Convolutional Features	(a)	40.2	58.6	43.7	19.3
	(b)	40.5	59.0	44.0	14.0
	(c)	40.5	58.7	44.1	16.2
	(d)	40.6	59.0	44.0	18.3
	(e)	40.6	58.9	44.1	17.8
	(f)	40.7	59.0	44.1	17.9
	(g)	40.8	58.9	44.6	18.4
Distribution Statistics **(ours)**		**41.1**	58.8	44.9	**19.4**

Table 3: Comparisons among different input features by fixing the hidden layer dimension of DGQP.

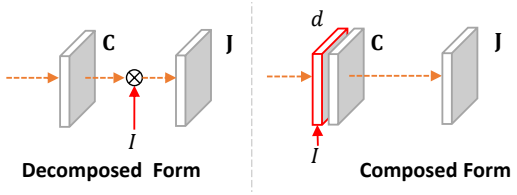

Figure 5: Different ways to utilize the distribution statistics, including Decomposed Form (left) and Composed Form (right).

Type of Input Features: To the best of our knowledge, the proposed DGQP is the first to use the statistics of learned distributions of bounding boxes for the generation of better LQE scores in the literature. Since the input (distribution statistics) and the output (LQE scores) are highly correlated, we speculate that it can be more effective or efficient than ordinary convolutional input proposed in existing methods. Therefore, we fix the hidden layer dimension of DGQP (i.e., $p = 64$) and compare our statistical input with most existing possible types of convolutional inputs, from point (a), region (b), border (c)-(e), regular points (f), and deformable points (g), respectively (Fig. 2). Table 3 shows that our distribution statistics perform best in overall AP, also fastest in inference, compared against various convolutional features.

Usage of the Decomposed Form: Next, we examine what

Type		AP	AP_{50}	AP_{75}	FPS
Baseline (GFLV1 [18])		40.2	58.6	43.4	**19.4**
Composed Form	$d = 16$	40.5	58.5	43.7	19.2
	$d = 32$	40.5	58.5	43.7	19.2
	$d = 64$	40.7	58.5	44.3	19.1
	$d = 128$	40.7	58.6	44.4	18.9
	$d = 256$	40.7	58.3	44.2	18.5
Decomposed Form **(ours)**		**41.1**	**58.8**	**44.9**	**19.4**

Table 4: Comparisons between Decomposed Form (proposed) and Composed Form (with various dimension d settings).

Method	GFLV2	AP	AP_{50}	AP_{75}	FPS
RetinaNet [21]		36.5	55.5	38.7	19.0
RetinaNet [21]	✓	**38.6** (+2.1)	**56.2**	**41.7**	19.0
FoveaNet [15]		36.4	55.8	38.8	20.0
FoveaNet [15]	✓	**38.5** (+2.1)	**56.8**	**41.6**	20.0
FCOS [33]		38.5	56.9	41.4	19.4
FCOS [33]	✓	**40.6** (+2.1)	**58.2**	**43.9**	19.4
ATSS [40]		39.2	57.4	42.2	19.4
ATSS [40]	✓	**41.1** (+1.9)	**58.8**	**44.9**	19.4

Table 5: Integrating GFLV2 into various popular dense object detectors. A consistent ~2 AP gain is observed without loss of inference speed.

is the best formulation of Classification-IoU Joint Representation in the case of using distribution statistics. There are basically two formats: the Composed Form and the proposed Decomposed Form (Sec. 3.2), as illustrated in Fig. 5. Here the "Decomposed" (the left part of Fig. 5) means that the final joint representation can be explicitly decomposed through multiplication by two components, i.e., $\mathbf{J} = \mathbf{C} \times I$ in Eq. (3). Whilst the "Composed" (the right part of Fig. 5) shows that $\mathbf{J}$ is directly obtained through FC layers where its input feature is enriched (d is the dimension of the appended feature) by the information of distribution statistics. From Table 4, our proposed Decomposed Form is always superior than the Composed Forms with various d settings in both accuracy and running speed.

Compatibility for Dense Detectors: Since GFLV2 is very lightweight and can be adapted to various types of dense detectors, we employ it to a series of recent popular detection methods. For those detectors that do not support the distributed representation of bounding boxes, we make the minimal and necessary modifications to enable it to generate distributions for each edge of a bounding box. Based on the results in Table 5, GFLV2 can consistently improve ~2 AP in popular dense detectors, without loss of inference speed.

4.2. Comparisons with State-of-the-arts

In this section, we compare GFLV2 with state-of-the-art approaches on COCO `test-dev` in Table 7. Following previous works [21, 33], the multi-scale ([480, 960]) training strategy and 2x learning schedule (24 epochs) are adopted during training. For a fair comparison, the results of single-model single-scale testing for all methods are reported, including their corresponding inference speeds (FPS). We also report additional multi-scale testing results for GFLV2. The visualizaion of the accuracy-speed trade-off is demonstrated in Fig. 6, and we observe that GFLV2 pushes the envelope of accuracy-speed boundary to a new level. Our best result with a single Res2Net-101-DCN model achieves considerably competitive 53.3 AP.

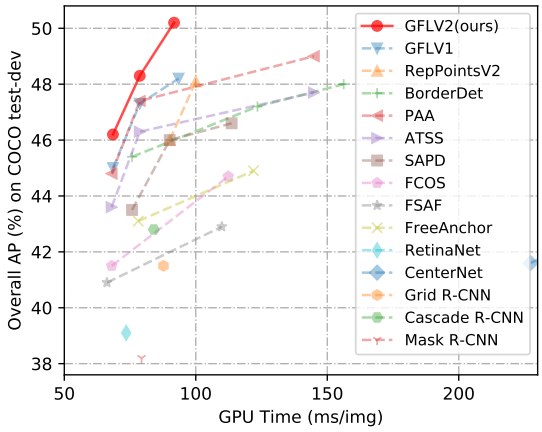

Figure 6: Single-model single-scale speed (ms) vs. accuracy (AP) on COCO `test-dev` among state-of-the-art approaches. GFLV2 achieves better speed-accuracy trade-off than its competitive counterparts.

Method	AP	FPS	PCC ↑
FCOS* [33]	39.1	19.4	0.624
ATSS* [40]	39.9	19.4	0.631
GFLV1 [18]	40.2	19.4	0.634
GFLV2 **(ours)**	**41.1** (+0.9)	19.4	**0.660** (+0.26)

Table 6: Pearson Correlation Coefficients (PCC) for representative dense object detectors. * denotes the application of Classification-IoU Joint Representation, instead of additional *Centerness* branch.

4.3. Analysis

Although the proposed DGQP module has been shown to improve the performance of dense object detectors, we would also like to understand how its mechanism operates.

DGQP Improves LQE: To assess whether DGQP is able to benefit the estimation of localization quality, we first obtain the predicted IoUs (given by four representative models with IoU as the quality estimation targets) and their corresponding real IoUs over all the positive samples on COCO `minival`. Then we calculate their Pearson Correlation Coefficient (PCC) in Table 6. It demonstrates that DGQP in GFLV2 indeed improves the linear correlation between the estimated IoUs and the ground-truth ones by a considerable margin (+0.26) against GFLV1, which eventually leads to an absolute 0.9 AP gain.

Method	Backbone	Epoch	MS_{train}	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Reference
multi-stage:											
Faster R-CNN w/ FPN [20]	R-101	24		14.2	36.2	59.1	39.0	18.2	39.0	48.2	CVPR17
Cascade R-CNN [1]	R-101	18		11.9	42.8	62.1	46.3	23.7	45.5	55.2	CVPR18
Grid R-CNN [24]	R-101	20		11.4	41.5	60.9	44.5	23.3	44.9	53.1	CVPR19
Libra R-CNN [26]	X-101-64x4d	12		8.5	43.0	64.0	47.0	25.3	45.6	54.6	CVPR19
RepPoints [38]	R-101	24		13.3	41.0	62.9	44.3	23.6	44.1	51.7	ICCV19
RepPoints [38]	R-101-DCN	24	✓	11.8	45.0	66.1	49.0	26.6	48.6	57.5	ICCV19
RepPointsV2 [4]	R-101	24	✓	11.1	46.0	65.3	49.5	27.4	48.9	57.3	NeurIPS20
RepPointsV2 [4]	R-101-DCN	24	✓	10.0	48.1	67.5	51.8	28.7	50.9	60.8	NeurIPS20
TridentNet [19]	R-101	24	✓	2.7*	42.7	63.6	46.5	23.9	46.6	56.6	ICCV19
TridentNet [19]	R-101-DCN	36	✓	1.3*	46.8	67.6	51.5	28.0	51.2	60.5	ICCV19
TSD [32]	R-101	20		1.1	43.2	64.0	46.9	24.0	46.3	55.8	CVPR20
BorderDet [27]	R-101	24	✓	13.2*	45.4	64.1	48.8	26.7	48.3	56.5	ECCV20
BorderDet [27]	X-101-64x4d	24	✓	8.1*	47.2	66.1	51.0	28.1	50.2	59.9	ECCV20
BorderDet [27]	X-101-64x4d-DCN	24	✓	6.4*	48.0	67.1	52.1	29.4	50.7	60.5	ECCV20
one-stage:											
CornerNet [17]	HG-104	200	✓	3.1*	40.6	56.4	43.2	19.1	42.8	54.3	ECCV18
CenterNet [7]	HG-104	190	✓	3.3*	44.9	62.4	48.1	25.6	47.4	57.4	ICCV19
CentripetalNet [6]	HG-104	210	✓	n/a	45.8	63.0	49.3	25.0	48.2	58.7	CVPR20
RetinaNet [21]	R-101	18		13.6	39.1	59.1	42.3	21.8	42.7	50.2	ICCV17
FreeAnchor [41]	R-101	24	✓	12.8	43.1	62.2	46.4	24.5	46.1	54.8	NeurIPS19
FreeAnchor [41]	X-101-32x8d	24	✓	8.2	44.9	64.3	48.5	26.8	48.3	55.9	NeurIPS19
FSAF [45]	R-101	18	✓	15.1	40.9	61.5	44.0	24.0	44.2	51.3	CVPR19
FSAF [45]	X-101-64x4d	18	✓	9.1	42.9	63.8	46.3	26.6	46.2	52.7	CVPR19
FCOS [33]	R-101	24	✓	14.7	41.5	60.7	45.0	24.4	44.8	51.6	ICCV19
FCOS [33]	X-101-64x4d	24	✓	8.9	44.7	64.1	48.4	27.6	47.5	55.6	ICCV19
SAPD [44]	R-101	24	✓	13.2	43.5	63.6	46.5	24.9	46.8	54.6	CVPR20
SAPD [44]	R-101-DCN	24	✓	11.1	46.0	65.9	49.6	26.3	49.2	59.6	CVPR20
SAPD [44]	X-101-32x4d-DCN	24	✓	8.8	46.6	66.6	50.0	27.3	49.7	60.7	CVPR20
ATSS [40]	R-101	24	✓	14.6	43.6	62.1	47.4	26.1	47.0	53.6	CVPR20
ATSS [40]	R-101-DCN	24	✓	12.7	46.3	64.7	50.4	27.7	49.8	58.4	CVPR20
ATSS [40]	X-101-32x8d-DCN	24	✓	6.9	47.7	66.6	52.1	29.3	50.8	59.7	CVPR20
PAA [14]	R-101	24	✓	14.6	44.8	63.3	48.7	26.5	48.8	56.3	ECCV20
PAA [14]	R-101-DCN	24	✓	12.7	47.4	65.7	51.6	27.9	51.3	60.6	ECCV20
PAA [14]	X-101-64x4d-DCN	24	✓	6.9	49.0	67.8	53.3	30.2	52.8	62.2	ECCV20
GFLV1 [18]	R-50	24	✓	19.4	43.1	62.0	46.8	26.0	46.7	52.3	NeurIPS20
GFLV1 [18]	R-101	24	✓	14.6	45.0	63.7	48.9	27.2	48.8	54.5	NeurIPS20
GFLV1 [18]	R-101-DCN	24	✓	12.7	47.3	66.3	51.4	28.0	51.1	59.2	NeurIPS20
GFLV1 [18]	X-101-32x4d-DCN	24	✓	10.7	48.2	67.4	52.6	29.2	51.7	60.2	NeurIPS20
GFLV2 **(ours)**	R-50	24	✓	19.4	44.3	62.3	48.5	26.8	47.7	54.1	–
GFLV2 **(ours)**	R-101	24	✓	14.6	46.2	64.3	50.5	27.8	49.9	57.0	–
GFLV2 **(ours)**	R-101-DCN	24	✓	12.7	48.3	66.5	52.8	28.8	51.9	60.7	–
GFLV2 **(ours)**	X-101-32x4d-DCN	24	✓	10.7	49.0	67.6	53.5	29.7	52.4	61.4	–
GFLV2 **(ours)**	R2-101-DCN	24	✓	10.9	50.6	69.0	55.3	31.3	54.3	63.5	–
GFLV2 **(ours)** + MS_{test}	R2-101-DCN	24	✓	–	53.3	70.9	59.2	35.7	56.1	65.6	–

Table 7: Comparisons between state-of-the-art detectors *(single-model and single-scale results except the last row)* on COCO `test-dev`. "MS_{train}" and "MS_{test}" denote multi-scale training and testing, respectively. FPS values with * are from [44] or their official repositories [27], while others are measured on the same machine with a single GeForce RTX 2080Ti GPU under the same mmdetection [3] framework, using a batch size of 1 whenever possible. "n/a" means that both trained models and timing results from original papers are not available. **R**: ResNet [11]. **X**: ResNeXt [37]. **HG**: Hourglass [25]. **DCN**: Deformable Convolutional Network [46]. **R2**: Res2Net [8].

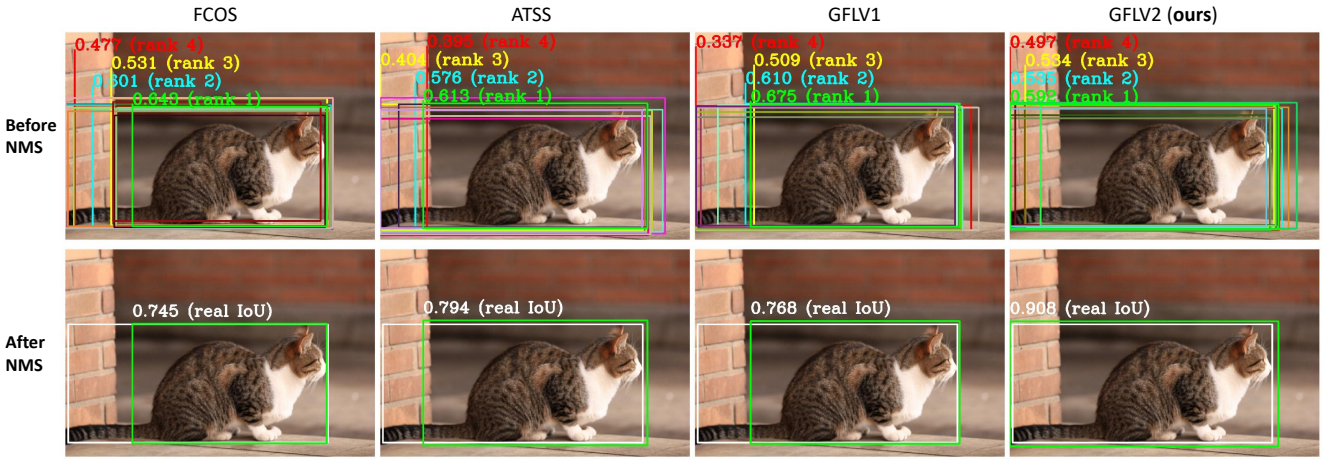

Figure 7: Visualization of predicted bounding boxes before and after NMS, along with their corresponding predicted LQE scores (only Top-4 scores are plotted for a better view). For many existing approaches [33, 40, 18], they fail to produce the highest LQE scores for the best candidates. In contrast, our GFLV2 reliably assigns larger quality scores for those real high-quality ones, thus reducing the risk of mistaken suppression in NMS processing. White: ground-truth bounding boxes; Other colors: predicted bounding boxes.

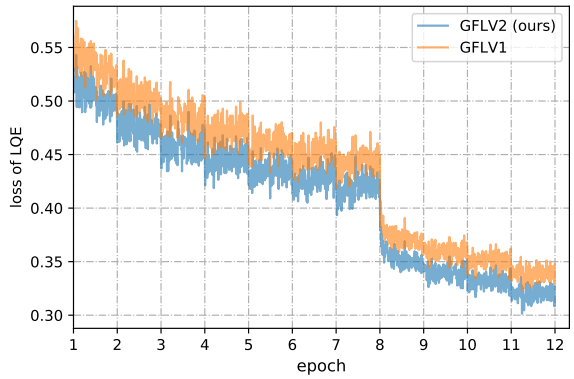

Figure 8: Comparisons of losses on LQE between GFLV1 and GFLV2. DGQP helps to ease the learning difficulty with lower losses during training.

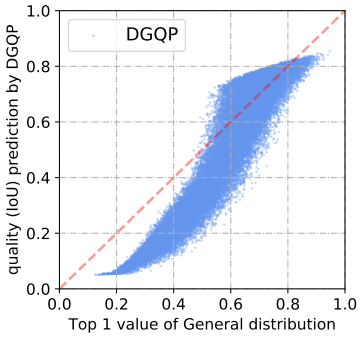

Figure 9: Visualization of the input Top-1 values (mean of four sides) from the learned distribution and the output IoU quality prediction given by DGQP.

DGQP Eases the Learning Difficulty: Fig. 8 provides the visualization of the training losses on LQE scores, where DGQP in GFLV2 successfully accelerates the training process and converges to lower losses.

Visualization on Inputs/Outputs of DGQP: To study the behavior of DGQP, we plot its inputs and corresponding outputs in Fig. 9. For a better view, we select the mean Top-1 values to represent the input statistics. It is observed that the outputs are highly correlated with inputs as expected.

Training/Inference Efficiency: We also compare the training and inference efficiency among recent state-of-the-art dense detectors in Table 8. Note that PAA [14], RepPointsV2 [4] and BorderDet [27] bring an inevitable time overhead (52%, 65%, and 22% respectively) during training, and the latter two also sacrifice inference speed by 30% and 14%, respectively. In contrast, our proposed GFLV2 can achieve top performance (~41 AP) while still maintaining the training and inference efficiency.

Qualitative Results: In Fig. 7, we qualitatively demonstrate the mechanism how GFLV2 makes use of its more reliable IoU quality estimations to maintain accurate predictions during NMS. Unfortunately for other detectors, high-quality candidates are wrongly suppressed due to their relatively lower localization confidences, which eventually leads to a performance degradation.

Method	AP	Training Hours ↓	Inference FPS ↑
ATSS* [40]	39.9	8.2	19.4
GFLV1 [18]	40.2	8.2	19.4
PAA [14]	40.4	12.5 (+52%)	19.4
RepPointsV2 [4]	41.0	14.4 (+65%)	13.5 (-30%)
BorderDet [27]	41.4	10.0 (+22%)	16.7 (-14%)
GFLV2 **(ours)**	41.1	8.2	19.4

Table 8: Comparisons of training and inference efficiency based on ResNet-50 backbone. "Training Hours" is evaluated on 8 GeForce RTX 2080Ti GPUs under standard 1x schedule (12 epochs). * denotes the application of Classification-IoU Joint Representation.

5. Conclusion

In this paper, we propose to learn reliable localization quality estimation, through the guidance of statistics of bounding box distributions. It is an entirely new and completely different perspective in the literature, which is also conceptually effective as the information of distribution is highly correlated to the real localization quality. Based on it we develop a dense object detector, namely GFLV2. Extensive experiments and analyses on COCO dataset further validate its effectiveness, compatibility and efficiency. We hope GFLV2 can serve as simple yet effective baseline for the community.

References

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

[2] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *ICCV*, 2019.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. *arXiv preprint arXiv:2007.08508*, 2020.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[6] Zhiwei Dong, Guoxuan Li, Yue Liao, Fei Wang, Pengju Ren, and Chen Qian. Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In *CVPR*, 2020.

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

[8] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *TPAMI*, 2019.

[9] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

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

[12] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019.

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

[14] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. *ECCV*, 2020.

[15] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *arXiv preprint arXiv:1904.03797*, 2019.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[17] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.

[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. In *NeurIPS*, 2020.

[19] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

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

[22] 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.

[23] 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.

[24] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *CVPR*, 2019.

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[26] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019.

[27] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun. Borderdet: Border feature for dense object detection. *ECCV*, 2020.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[29] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017.

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

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

[32] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *CVPR*, 2020.

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

[34] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, 2018.

[35] Jiaqi Wang, Wenwei Zhang, Yuhang Cao, Kai Chen, Jiangmiao Pang, Tao Gong, Jianping Shi, Chen Change Loy, and Dahua Lin. Side-aware boundary localization for more precise object detection. *ECCV*, 2020.

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

[37] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[38] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, 2019.

[39] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Varifocalnet: An iou-aware dense object detector. *arXiv preprint arXiv:2008.13367*, 2020.

[40] 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.

[41] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *NeurIPS*, 2019.

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

[43] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020.

[44] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *CVPR*, 2020.

[45] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019.

[46] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.