D2Det: Towards High Quality Object Detection and Instance Segmentation

Jiale Cao[1]*, Hisham Cholakkal[2]*, Rao Muhammad Anwer[2], Fahad Shahbaz Khan[2],
Yanwei Pang[1]†, Ling Shao[2]
[1]Tianjin University [2]Inception Institute of Artificial Intelligence (IIAI), UAE‡
[1]{connor,pyw}@tju.edu.cn,[2]{hisham.cholakkal,rao.anwer,fahad.khan,ling.shao}@inceptioniai.org

Abstract

We propose a novel two-stage detection method, D2Det, that collectively addresses both precise localization and accurate classification. For precise localization, we introduce a dense local regression that predicts multiple dense box offsets for an object proposal. Different from traditional regression and keypoint-based localization employed in two-stage detectors, our dense local regression is not limited to a quantized set of keypoints within a fixed region and has the ability to regress position-sensitive real number dense offsets, leading to more precise localization. The dense local regression is further improved by a binary overlap prediction strategy that reduces the influence of background region on the final box regression. For accurate classification, we introduce a discriminative RoI pooling scheme that samples from various sub-regions of a proposal and performs adaptive weighting to obtain discriminative features.

On MS COCO `test-dev`, our D2Det outperforms existing two-stage methods, with a single-model performance of 45.4 AP, using ResNet101 backbone. When using multi-scale training and inference, D2Det obtains AP of 50.1. In addition to detection, we adapt D2Det for instance segmentation, achieving a mask AP of 40.2 with a two-fold speedup, compared to the state-of-the-art. We also demonstrate the effectiveness of our D2Det on airborne sensors by performing experiments for object detection in UAV images (UAVDT dataset) and instance segmentation in satellite images (iSAID dataset). Source code is available at `https://github.com/JialeCao001/D2Det`.

1. Introduction

Recent years have witnessed formidable progress in object detection thanks to the advances in deep neural networks. Modern object detectors can be broadly divided into single-stage [35, 42, 40, 29, 27, 4] and two-stage methods [18, 43, 17, 41, 8, 21]. Two-stage detection approaches work by first generating a set of candidate proposals followed by classification and regression of these proposals. On the other hand, single-stage methods perform a direct regression and classification of default anchors into boxes by regular sampling grids on the image. Generally, two-stage methods dominate in terms of accuracy on standard benchmarks, compared to their single-stage counterparts.

High quality object detection requires both precise localization (bounding box) and accurate classification of the target object. Most existing two-stage detectors [43, 31, 15] share a similar design for the bounding box localization module. A typical design choice is a regression module, employed in most two-stage detectors, including the popular Faster R-CNN [43]. The regression module utilizes several fully connected layers to predict a single box offset of the candidate proposal. Recently, Grid R-CNN [36] extends Faster R-CNN by separating the classification and regression into two branches, as opposed to a shared network. Instead of the regression utilized in Faster R-CNN, Grid R-CNN introduces a localization scheme, based on a fully convolutional network, that searches for a set of keypoints in a fixed-sized region to identify an object boundary.

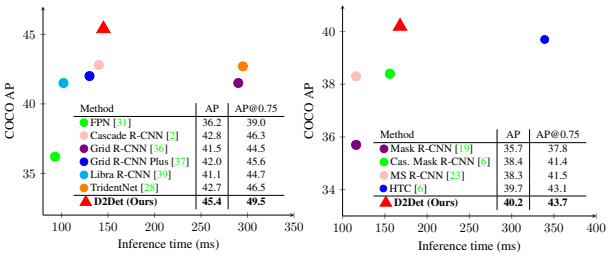
In this work, we introduce dense local regression for precise target localization. Different from the traditional regression employed in Faster R-CNN [43] that predicts a single global offset by a fully-connected network, our dense local regression predicts multiple local box offsets, termed as dense box offsets, by a fully convolutional network. Compared to the keypoint-based localization in Grid R-CNN [36], our dense local regression can more accurately localize an object due to its ability to regress any real number offset and is therefore not limited to a quantized set of keypoints within a fixed-sized region. In addition, while Grid R-CNN aims to improve localization capabilities, our method collectively addresses both precise localization and accurate classification of target object. For classification, we introduce a discriminative RoI pooling that extracts features from various sub-regions of a proposal and performs

*The first two authors contribute equally to this work.
†Corresponding author.
‡Work done at IIAI during J. Cao's research visit.

adaptive weighting to obtain discriminative features.

Contributions: We propose a two-stage object detection approach, D2Det, that targets both precise localization and accurate classification. For precise target localization, we introduce a dense local regression, where each sub-region of a candidate proposal predicts its own relative box offsets towards the four sides of ground-truth bounding box. As a result, multiple dense box offsets are obtained by a fully convolutional network, which preserves the position-sensitive characteristic for box offset prediction. To further improve our dense local regression, we introduce a binary overlap prediction that identifies each sub-region of a candidate proposal as an object region or background region, thereby reducing the influence of background region. The binary overlap prediction is trained by assuming all regions inside the ground-truth bounding box as object. For accurate classification of the target object, we introduce a discriminative RoI pooling that first samples features from various sub-regions and then performs an adaptive weighted pooling that aims to generate discriminative features.

Experiments are performed on the MS COCO [33] and UAVDT [11] datasets. Our D2Det achieves state-of-the-art performance on both datasets. On MS COCO `test-dev`, our method surpasses existing two-stage detectors, in terms of single model accuracy, with a COCO style AP of 45.4 using a ResNet101 backbone (Fig. 1(a)). Further, an absolute gain of 3.0% is obtained at AP@0.75, compared to the state-of-the-art [28], demonstrating accurate localization capabilities of our D2Det. Moreover, D2Det achieves a COCO style AP of 50.1 when using a stronger backbone with multi-scale training and inference. Additionally, we report results for instance segmentation, obtained by modifying the dense local regression branch of our two-stage detection method and utilizing instance mask annotations. Experiments are performed on two instance segmentation datasets: MS COCO and the recently introduced iSAID [51]. Our method obtains consistent improvement over existing methods on both datasets. On MS COCO `test-dev`, our approach achieves a Mask AP of 40.2 and provides a two-fold speedup over the state-of-the-art HTC [6] (Fig. 1(b)).



(a) Object detection (b) Instance segmentation

Figure 1: Accuracy (AP) vs. speed (ms) comparison on MS COCO `test-dev`. (a) Comparison with existing two-stage detectors for object detection. (b) Comparison with state-of-the-art approaches for instance segmentation. All the methods in (a) use only box-level supervision. Further, all methods in (a) and (b) utilize same settings: input size ($\sim$ 1333 $\times$ 800, except FPN which uses $\sim$ 1000 $\times$ 600), ResNet101 with FPN (except TridentNet which introduces an alternative to FPN) and without multi-scale training or inference. The speed of all methods is reported on a Titan Xp. In addition to overall COCO AP, we report AP@0.75 for comparison at a higher overlap threshold.

2. Related Work

In recent years, two-stage detection approaches [18, 43, 17, 44, 28, 36, 5, 46] have shown continuous performance improvements in terms of detection accuracy on standard benchmarks [33, 13]. Among existing two-stage detectors, Faster R-CNN [43] is one of the most popular frameworks for object detection. In the first stage, Faster R-CNN utilizes a region proposal network (RPN) to generate class-agnostic region proposals. The second stage, also known as Fast R-CNN [17], extracts a fixed-sized region-of-interest (RoI) feature representation followed by the computation of classification scores and regressed bounding-box coordinates for each proposal. Several recent works have extended this framework by, for example, integrating pyramid representations [31, 28, 44, 5], extending to multi-stage detection [16, 2, 6, 24] and integrating a mask branch [19, 23, 34].

Most two-stage detectors represent each object in an image based on a pre-defined anchor box. Alternatively, several single-stage approaches [26, 22, 52, 50, 12] propose an anchor box free strategy that eliminates the need for anchor boxes. This typically involves using paired keypoints and keypoint estimation to detect object bounding-box. These approaches are bottom-up in that keypoints are directly generated from the entire image without defining object instances. Different from these bottom-up approaches, Grid R-CNN [36] is a top-down two-stage method which first defines instances and then generates bounding box keypoints using grid guided keypoint-based localization. This strategy searches for a set of keypoints in a fixed-sized region, obtained through an extended region mapping of RoI, to identify an object boundary. However, even an extended region mapping may fail to encompass the entire object depending on the position of the candidate proposal with respect to the ground-truth. Further, keypoint search occurs in a fixed-resolution feature space (56 $\times$ 56), which is likely to be problematic for large objects. In such a case (*e.g.*, object size $>$ 100 $\times$ 100 image pixels) the relatively smaller keypoint search space may lead to less accurate localization. Moreover, Grid R-CNN only focuses on improving the localization capabilities, while keeping the classification branch similar to original Faster R-CNN. On MS COCO, our dense local regression alone (without the proposed improvements in the classification branch) achieves a gain of

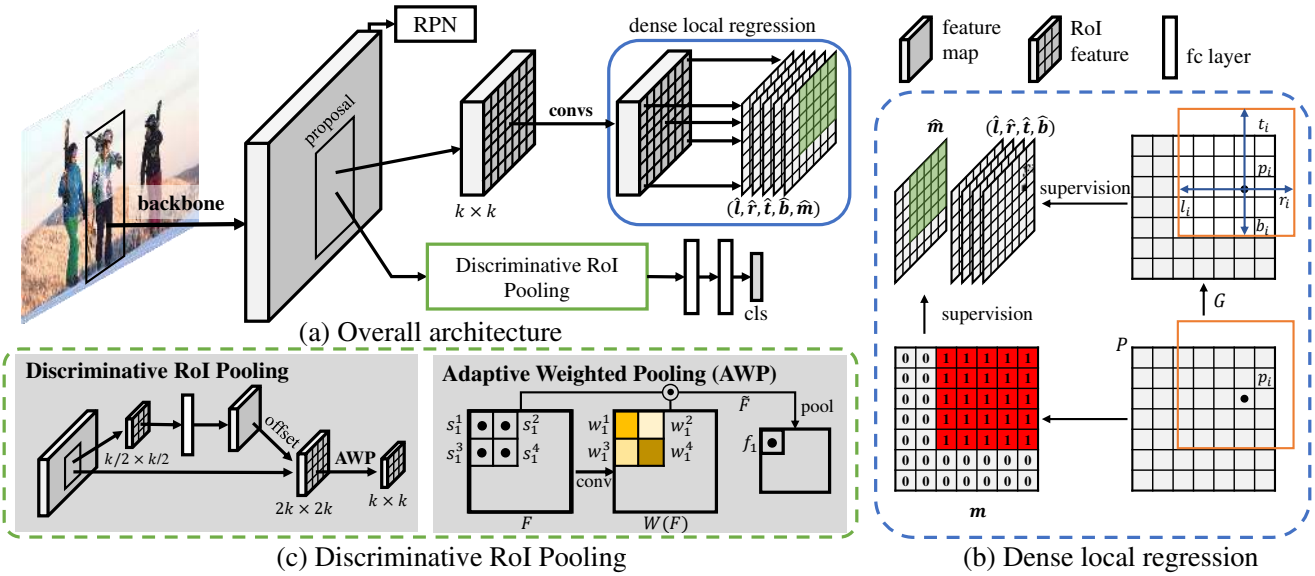


Figure 2: (a) Overall architecture of our two-stage method. The RoI feature of each candidate proposal P, generated from RPN, is passed through two different branches: dense local regression (b) and classification (c). Instead of treating RoI feature as a single global vector, our dense local regression treats them as $k \times k$ local features extracted from $k \times k$ sub-regions within RoI. These local features are used to predict multiple dense box offsets, implying each local feature $p_i \in P$ predicts its own dense box offsets $(\hat{l}_i, \hat{t}_i, \hat{r}_i, \hat{b}_i)$. To reduce the influence of background features, a binary overlap prediction $\hat{\mathbf{m}}$ (in green) is utilized that classifies each local feature as either belonging to ground-truth bounding box G (in orange) or background. To train $\hat{\mathbf{m}}$, the overlapping region $\mathbf{m}$ (in red) between G and P is assigned one ($\mathbf{m} = 1$). For classification (c), our discriminative RoI pooling first predicts the offsets of each RoI sub-region using a light-weight offset predictor, and then performs an adaptive weighting ($W(F)$) that assigns higher weights to the discriminative sampling points of an RoI.

3.7% on large objects, compared to Grid R-CNN.

The original Faster R-CNN employed RoIPool [17, 43] for feature pooling of candidate proposals. Recently RoIAlign [19] has replaced RoIPool in several works, including latest variants of Faster R-CNN and Grid R-CNN [36]. RoIAlign divides candidate proposals into equally sized spatial sub-regions and considers features from sub-regions inside the proposal. Four sampling points are obtained within each sub-region which are averaged by assigning equal weight to all points [19]. This can deteriorate the classification performance as discriminative regions may not appear in equally spaced sub-regions. Different from RoIAlign, deformable RoI pooling [10] obtains features that are used for both classification and regression, from various sub-regions of a candidate proposal, disregarding their distance. However, the sampling points are still averaged with equal weight, as in RoIAlign. Here, we introduce an approach that performs adaptive weighting to enhance discriminaitive features for classification.

3. Our Method

We base our method on the standard Faster R-CNN framework [43]. In our method, the proposed dense local regression (Sec. 3.1) replaces the traditional box offset regression of Faster R-CNN, while the classification is improved with a discriminative RoI pooling (Sec. 3.2). The overall architecture of our two-stage detection framework is shown in Fig. 2(a). We utilize a region proposal network (RPN) in the first stage and employ separate classification and regression branches in the second stage. The dense local regression branch (Fig. 2(b)) aims at precise localization of an object whereas the classification branch, based on discriminative RoI pooling (Fig. 2(c)), intends to improve classification of candidate proposals.

3.1. Dense Local Regression

In a two-stage detection framework, the objective of the bounding-box regression branch is to find a tight bounding-box surrounding an object. Let $P\ (x_P, y_P, w_P, h_P)$ be a candidate object proposal, and $G\ (x_G, y_G, w_G, h_G)$ be the target ground-truth box. The traditional regression in Faster R-CNN predicts a single box offset $(\Delta_x, \Delta_y, \Delta_w, \Delta_h)$, as:

$$\begin{aligned} \Delta_x &= (x_G - x_P)/w_P, \quad & \Delta_y &= (y_G - y_P)/h_P, \\ \Delta_w &= log(w_G/w_P), & \Delta_h &= log(h_G/h_P), \end{aligned} \tag{1}$$

where (x, y) indicates box centers and (w, h) represents the width and height of a given box (*i.e.*, either ground-truth bounding box G or candidate proposal P). For each

candidate proposal P, feature pooling strategies, such as RoIPool [17] or RoIAlign [19], are employed to obtain the corresponding fixed-sized ($k \times k$) RoI feature from equally spaced $k \times k$ sub-regions within the proposal. The standard Faster R-CNN treats these RoI features as a single vector, termed here as global feature representation, and predicts a single global box offset by passing them through several fully connected layers (Fig. 3(a)).

Different from the aforementioned strategy, our dense local regression approach considers the $k \times k$ dimensional RoI feature as k^2 spatially adjacent local features. One such local feature is shown as p_i in Fig. 2(b). These local RoI features are then used to predict multiple local box offsets, termed as dense box offsets, by passing through a fully convolutional network. The dense box offsets predict the distance of each local feature p_i at location (x_i, y_i) to the top-left and bottom-right corners of the ground-truth bounding box G. Let (x_l, y_t) and (x_r, y_b) represent the top-left and bottom-right corners of the ground-truth bounding box, and $\hat{l}_i$, $\hat{t}_i$, $\hat{r}_i$ and $\hat{b}_i$ represent the dense box offsets predicted by the local feature p_i in left, top, right, and bottom directions, respectively (Fig. 2(b)). The corresponding ground-truth offsets (l_i, t_i, r_i, b_i) at (index) location i, are computed,

$$\begin{aligned} l_i &= (x_i - x_l)/w_P, \quad t_i = (y_i - y_t)/h_P, \\ r_i &= (x_r - x_i)/w_P, \quad b_i = (y_b - y_i)/h_P. \end{aligned} \tag{2}$$

Here, the normalization factors w_P and h_P denote the width and height of the candidate proposal.

The number of sub-regions or local features of the candidate proposal belonging to the ground-truth bounding box depends on the overlap between the proposal and its corresponding ground-truth. Even in the case of higher overlap (majority of k^2 local features belonging to the ground-truth bounding box), several unwanted features (*e.g.*, background) are included among these k^2 local features. As a consequence, the dense box offsets predicted by these background features are less precise and are therefore desired to be ignored. With this aim, a binary overlap prediction (shown in green in Fig. 2(a) and Fig. 2(b)) is utilized in our dense local regression to classify each local feature as either belonging to ground-truth bounding box region or background. This binary overlap prediction is performed by introducing an additional output $\hat{\mathbf{m}}$, along with the dense box offsets. The local features in an overlapping region between the ground-truth bounding box G and the candidate proposal P, are assigned with a ground-truth label 1, *i.e.*,

$$m_i = \begin{cases} 1, & \text{if } p_i \in G; \quad \forall p_i \in P, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here, due to the unavailability of the ground-truth pixel-level instance mask in generic object detection, we assume all regions inside the ground-truth bounding box G as object. Note that $\hat{\mathbf{m}} = \{\hat{m}_i : i \in [1, k^2]\}$ and $\mathbf{m} = \{m_i : i \in [1, k^2]\}$. During training, the binary overlap prediction $\hat{m}_i$ at (index) location i is passed through sigmoid normalization (σ), for computing the binary cross-entropy loss with the ground-truth label m_i. During inference, our dense local regression module predicts five outputs, $(\hat{l}_i, \hat{t}_i, \hat{r}_i, \hat{b}_i, \hat{m}_i)$, at each local feature $p_i \in P$. The predicted dense box offsets at positions where $\sigma(\hat{m}_i) > 0.5$, are only used to compute the top-left and bottom-right corner points of the predicted box. Finally, the boxes predicted by multiple local features (Fig. 3(c)) are averaged to obtain a single (final) regressed bounding box (represented using top-left and bottom-right corner points).

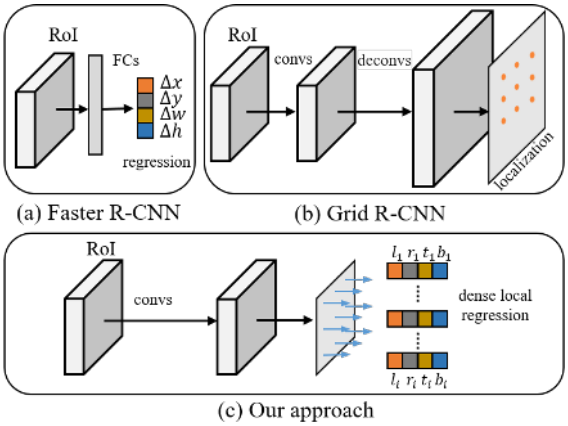


Figure 3: Comparison of our dense local regression (c) with traditional regression in Faster R-CNN (a) and keypoint-based localization in Grid R-CNN (b). Traditional regression in Faster R-CNN predicts a single global offset for a given proposal using a fully connected network. Grid R-CNN predicts bounding box keypoints using a probability heatmap. Instead, our approach yields multiple position-sensitive local offsets, termed as dense box offsets, using a fully convolutional network. Our approach can regress any real number offset and is therefore not limited to a quantized set of keypoints within a fixed region.

As discussed earlier, the traditional regression in Faster R-CNN predicts a single global offset for a given candidate proposal using a fully connected network (Fig. 3(a)). Different from the traditional regression, our dense local regression yields multiple position-sensitive box offsets using a fully convolutional network (Fig. 3(c)). Further, our binary overlap predictor reduces the influence of background regions on the final box regression. Similar to our approach, Grid R-CNN employs a fully convolutional network. However, in contrast to the keypoint-based localization strategy used in Grid R-CNN (Fig. 3(b)), our dense local regression can more accurately localize an object due to its ability to regress any real number offset and it is not limited to a quantized set of keypoints within a fixed region-of-interest. Further, our approach does not require deconvolution operations to increase spatial resolution for box localization,

thereby avoiding the additional computational overhead.

3.2. Discriminative RoI Pooling

Here, we describe the discriminative RoI pooling (Fig. 2(c)) in our classification branch. Different from the regression, the classification needs highly discriminative features. The discriminative RoI pooling is inspired by deformable RoI pooling [10] and improves it for classification in two ways. First, we use a light-weight offset prediction that requires about one-fourth of the parameters, as compared to the standard offset prediction in deformable RoI pooling. The standard offset prediction employs a RoIAlign operation to obtain features from $k \times k$ sub-regions and passes these features through three fully connected layers. Instead, the light-weight offset prediction only requires a $\frac{k}{2} \times \frac{k}{2}$ sized RoIAlign followed by the fully connected layers (light-weight due to smaller input vector).

After offset prediction, the standard deformable RoI pooling employs a RoIAlign, where all four sampling points obtained within each sub-region are averaged by assigning them equal weights. In contrast, the proposed weighted pooling aims to adaptively assign higher weights to discriminative sampling points and is motivated by [14]. Here, RoIAlign features in original sampling points, *i.e.* $F \in R^{2k \times 2k}$, are used to predict its corresponding weights $W(F) \in R^{2k \times 2k}$, which indicates the discriminative ability of the sampling points inside all $k \times k$ spatial sub-regions. Fig. 2(c) shows some sampling points (s_1^1, s_1^2, s_1^3, s_1^4) and their corresponding adaptive weights (w_1^1, w_1^2, w_1^3, w_1^4). Weighted RoI feature $\tilde{F}$ of a candidate proposal is obtained by,

$$\tilde{F} = W(F) \odot F, \tag{4}$$

where $\odot$ is the Hadamard product. Note that instead of using a fixed weight, the weight $W(F)$ is computed from F using the convolution operations. Consequently, we employ an average pooling operation with stride two on $\tilde{F}$, and obtain the discriminative RoI feature with size of $k \times k$. The discriminative RoI pooled feature of a candidate proposal is treated as a single global vector, as in the standard Faster R-CNN, followed by two fully-connected layers to obtain the classification score of the candidate proposal.

Note that the predicted offsets samples sub-regions within the candidate proposal as well its surroundings in discriminative RoI pooling. As a result, the extracted features are likely to contain discriminative information relevant to both the object and its context, which is expected to further improve the classification performance.

3.3. Instance Segmentation

The proposed method can be easily extended to instance segmentation by modifying our dense local regression branch. Instead of assuming all regions inside the ground-truth bounding box G belong to the object (Sec. 3.1), the ground-truth mask, available in instance segmentation, is used to label local features $p_i \in P$ in Eq. 3. As a result, the mask-based ground-truth binary overlap $\mathbf{m}$ is used to train the binary overlap prediction $\hat{\mathbf{m}}$ and the offset prediction in our dense regression branch (Fig. 2(b)). During inference, the binary overlap prediction $\hat{\mathbf{m}}$ provides the instance mask prediction. Further, we utilize two deconvolutional layers that increase the output spatial resolution by four times (*i.e.*, from 7×7 to 28×28) and two fully-connected layers for efficient mask scoring. Our method provides an efficient instance segmentation framework with competitive performance (see Sec. 5).

4. Experiments

4.1. Datasets and Implementation Details

Datasets: We conduct extensive experiments on two object detection benchmarks: MS COCO [33] and UAVDT [11]. MS COCO dataset contains 80 categories and consists of three subsets: `trainval`, `minival`, and `test-dev`. We perform training on the `trainval` set and report the results on the `test-dev` set for state-of-the-art comparison. We follow the standard protocol where the overall performance, in terms of average precision (AP), is measured by averaging over multiple intersection-over-union (IoU) thresholds, ranging from 0.5 to 0.95 with an interval of 0.05. The detection track in UAVDT dataset [11] contains three categories: car, truck and bus. Following the conventions in [11, 48], the three categories are combined into a single *vehicle* class, due to the highly imbalanced class distribution. We follow the same evaluation criteria in UAVDT [11] and report the results using PASCAL VOC style AP with the IoU threshold set to 0.7.

Implementation Details: The input image is resized during training and testing such that the shorter edge is 800 pixels. We adopt ResNet models (ResNet50 and ResNet101) [20] with FPN [31] as the backbone. In our work, RPN [43] is used to generate candidate object proposals similar to [31, 36]. All RoIs with a ground-truth overlap greater than 0.5 are considered as positive samples. From each image, we sample 512 RoIs by keeping a 1:3 positive to negative ratio and these sampled RoIs are used to train the classification branch. The dense local regression branch is trained only using positive RoIs. Like [37], we use eight convolutions of size 3×3 in dense local regression and a pooling size of 7×7 (where $k = 7$) for both classification and regression. Our method is trained on 8 GPUs (2 images per GPU) and adopts the SGD for training optimization, where the weight decay is 0.0001 and the momentum is 0.9. We adopt a $2\times$ training scheme for all MS COCO experiments. In our experiments, no data augmentation except the traditional horizontal flipping is utilized. During inference, we first classify proposals from RPN, following

Methods	Backbone	Input Size	AP	AP@0.5	AP@0.75	AP_s	AP_m	AP_l
Single-Stage Methods:								
RetinaNet w FPN [32]	ResNet101	$\sim 1333 \times 800$	39.1	59.1	42.3	21.8	42.7	50.2
ConRetinaNet w FPN [25]	ResNet101	$\sim 1333 \times 800$	40.1	59.6	43.5	23.4	44.2	53.3
EFGRNet [38]	ResNet101	512×512	39.0	58.8	42.3	17.8	43.6	54.5
CornerNet [26]	Hourglass104	511×511	40.5	56.5	43.1	19.4	42.7	53.9
FSAF w FPN [53]	ResNet101	$\sim 1333 \times 800$	40.9	61.5	44.0	24.0	44.2	51.3
RPDet w FPN [50]	ResNet101	$\sim 1333 \times 800$	41.0	62.9	44.3	23.6	44.1	51.7
FCOS w FPN [45]	ResNet101	$\sim 1333 \times 800$	41.5	60.7	45.0	24.4	44.8	51.6
HSD [3]	ResNet101	768×768	42.3	61.2	46.9	22.8	47.3	55.9
Two-Stage Methods:								
FPN [31]	ResNet101	$\sim 1000 \times 600$	36.2	59.1	39.0	18.2	39.0	48.2
Libra R-CNN w FPN [39]	ResNet101	$\sim 1333 \times 800$	41.1	62.1	44.7	23.4	43.7	52.5
Grid R-CNN w FPN [36]	ResNet101	$\sim 1333 \times 800$	41.5	60.9	44.5	23.3	44.9	53.1
Grid R-CNN Plus w FPN [37]	ResNet101	$\sim 1333 \times 800$	42.0	60.5	45.6	23.4	45.2	53.2
LIP w FPN [14]	ResNet101	$\sim 1333 \times 800$	42.0	64.3	45.8	24.7	45.2	52.3
Auto-FPN [49]	ResNet101	$\sim 1333 \times 800$	42.5	-	-	-	-	-
TridentNet [28]	ResNet101	$\sim 1333 \times 800$	42.7	63.6	46.5	23.9	46.6	56.6
Cascade R-CNN w FPN [2]	ResNet101	$\sim 1333 \times 800$	42.8	62.1	46.3	23.7	45.5	55.2
D2Det (ours) w FPN	ResNet101	$\sim 1333 \times 800$	**45.4**	**64.0**	**49.5**	**25.8**	**48.7**	**58.1**
DCN v2 [54]	ResNet101-deform v2	$\sim 1333 \times 800$	44.0	65.9	48.1	23.2	47.7	59.6
D2Det (ours)	ResNet101-deform v2	$\sim 1333 \times 800$	**47.4**	**65.9**	**51.7**	**27.2**	**50.4**	**61.3**
D2Det* (ours)	ResNet101-deform v2		50.1	69.4	54.9	32.7	52.7	62.1

Table 1: State-of-the-art object detection comparison (in terms of AP) on MS COCO `test-dev`. When using a ResNet101 backbone with FPN, our D2Det achieves the best *single-model* performance, with an overall AP of 45.4, surpassing all existing two-stage methods employing the same backbone with FPN (TridentNet and Auto-FPN do not use FPN since they introduce alternative approaches). Further, our D2Det outperforms DCN v2 [54] by a gain of 3.4%, when using the same ResNet101-deform v2 backbone. In case of multi-scale training and inference, our D2Det* achieves an overall AP of 50.1.

which we employ NMS, and select few proposals (100-125) for dense local regression, similar to [37]. On MS COCO `test-dev`, soft-NMS [1] is employed on these few proposals after dense local regression, which slightly improves detection accuracy without a significant reduction in speed.

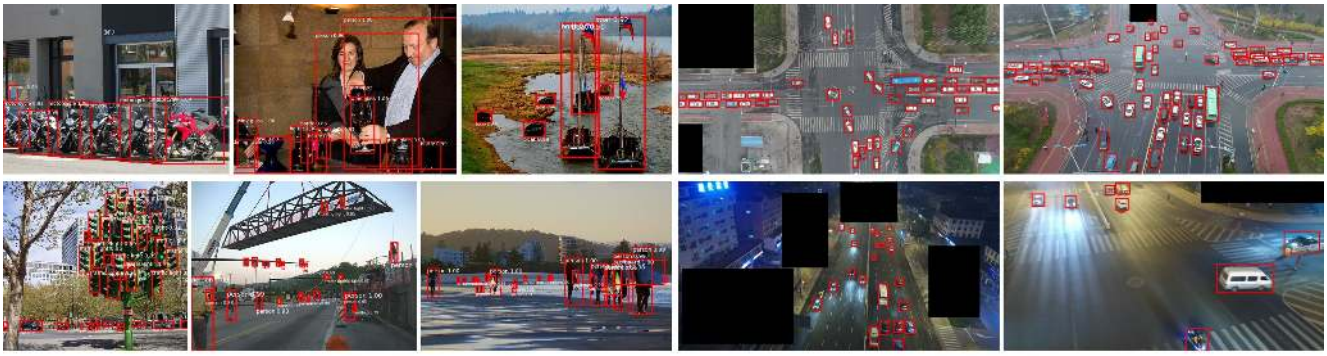
4.2. MS COCO Dataset

State-of-the-art Comparison: We first present a comparison (Tab. 1) of our detection method, D2Det, with existing detectors in literature on MS COCO `test-dev`. Note that several methods exist in the literature that exploit instance mask annotations in addition to bounding box information for object detection. For fair comparison, all detection methods in Tab. 1 only use bounding box annotations. We first discuss the results when using the popular ResNet101 backbone with FPN. Among existing two-stage detectors, Libra R-CNN [39] and Grid R-CNN [36] achieve overall AP scores of 41.1 and 41.5, respectively. Grid R-CNN Plus [37] introduces several updates to improve the performance and efficiency of Grid R-CNN and achieves 42.0 AP. TridentNet [28], which replaces FPN with a parallel multi-branch architecture having different receptive fields, achieves 42.7 AP. Cascade R-CNN [2] and LIP [14] obtain the AP scores of 42.8 and 42.0, respectively. Our D2Det significantly outperforms existing approaches by achieving an AP score of 45.4. Further, a notable absolute gain of 3.0% is obtained at strict metric (AP@0.75), compared to the state-of-the-art TridentNet [28], demonstrating the accurate localization capabilities of our detection method.

Other than ResNet101 with FPN, DCN v2 [54] utilizes a ResNet101-deform v2 backbone and reports 44.0 AP. Our D2Det achieves 47.4 AP and obtains an absolute gain of 3.4% over DCN v2, when using the same backbone. Further, our D2Det* obtains 50.1 AP in case of multi-scale training and inference.

Qualitative Analysis: To further analyze our D2Det, we utilize the error analysis protocol provided by [33]. Fig. 5 shows error plots on MS COCO `minival` for our D2Det (bottom row) and Grid R-CNN Plus [37] (top row), when using ResNet50 with FPN. As discussed earlier (Sec. 2), Grid R-CNN and its improved variant Grid R-CNN Plus utilize keypoint-based localization which is especially problematic for large objects. We therefore present error plots for both overall (left) and large objects (right). The plots in each sub-image represent a series of precision recall curves with various evaluation settings, as defined in [33].

In case of overall results (on left), Grid R-CNN Plus obtains 0.434 AP at strict AP@0.75, with AP likely increasing to 0.669 in case of perfect localization. Our D2Det detector (bottom row) achieves 0.463 AP at AP@0.75, with AP likely increasing to 0.697 in case of perfect localization. The improvement obtained by our D2Det is more promi-



(a) MS COCO (b) UAVDT

Figure 4: Qualitative results of D2Det on the COCO `test-dev` and UAVDT. In UAVDT, the black regions are ignored.

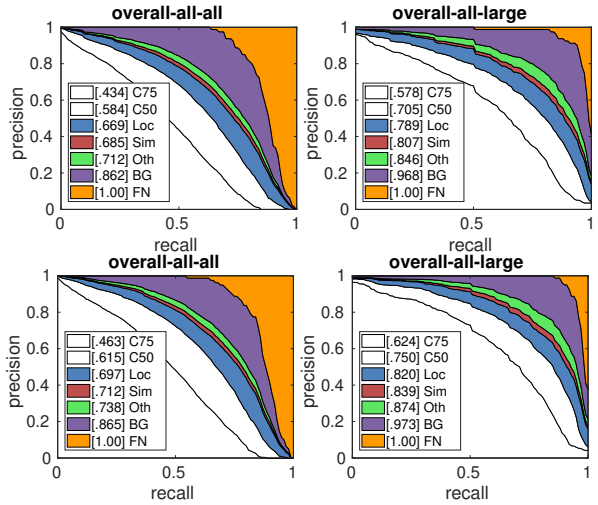


Figure 5: Error analysis plots showing a comparison of our D2Det (bottom row) with Grid R-CNN Plus (top row) across all 80 categories, on the overall (left) and the large-sized objects (right). As defined in [33], a series of precision recall curves with different evaluation settings is shown in the plots of each sub-image. We also show the area under each curve (brackets in the legends). Our D2Det achieves consistent improvements over Grid R-CNN Plus [37].

nent, when performing analysis on large-sized objects (on right). In this case, our D2Det provides a gain of 4.6% by achieving 0.624 AP at strict metric of AP@0.75, compared to 57.8 by Grid R-CNN Plus. With perfect localization, D2Det is likely to increase the AP to 0.820, compared to 0.789 by Grid R-CNN Plus. Fig. 4(a) shows detection examples with our D2Det on MS COCO `test-dev`.

Ablation Study: We perform an ablation study on the MS COCO `minival` set. Tab. 2 shows the impact of our dense local regression (Sec. 3.1) and discriminative RoI pooling (Sec. 3.2). All results are reported using the ResNet50 backbone with FPN. Note that, as opposed to a shared network, our baseline Faster R-CNN with FPN has separate fully connected branches for regression and classification. This

Baseline	DLR (Sec. 3.1)	DRP (Sec. 3.2)	AP	AP@0.5	AP@0.75
✓			38.0	59.2	41.5
✓	✓		41.5	59.6	44.8
✓		✓	39.3	61.4	42.2
✓	✓	✓	42.7	61.5	46.3

Table 2: Impact of integrating our dense local regression (DLR) and discriminative RoI pooling (DRP) into the baseline, on MS COCO `minival`. Our final method based on DLR and DRP achieves consistent improvement in performance, with an overall gain of 4.7% over the baseline.

improves the AP from 37.7 to 38.0. The integration of our dense local regression (DLR), in place of traditional regression, in the baseline leads to an AP score of 41.5, in which an AP gain of 0.7 is provided by the binary overlap predictor. Notably, our DLR provides a significant absolute gain of 3.3% at strict metric (AP@0.75), over the baseline. This large gain in detection performance at AP@0.75 shows the impact of our DLR towards achieving precise localization. Further, the integration of our discrminative RoI pooling (DRP) in the baseline leads to an overall AP score of 39.3, where our weighting scheme alone gives an AP gain of 0.4. Our final method, D2Det, provides a consistent improvement over the baseline with a significant absolute gain of 4.7% in terms of overall AP.

We also compare (Tab. 3) our dense local regression (DLR) with the keypoint-based localization utilized in the recently introduced Grid R-CNN [36] and its variant Grid R-CNN Plus [37]. For fair comparison, our DLR alone utilizes the same classification branch, as in Grid R-CNN. Further, all results are reported using the same input size, training iterations and ResNet50 backbone with FPN. Our DLR alone provides superior results compared to Grid R-CNN and its variant. Particularly, a prominent improvement in performance is obtained for large-sized objects, where our DLR alone provides an absolute gain of 2.1% over Grid R-CNN Plus. The best results in Tab. 3 are obtained by our final D2Det, highlighting the importance of both precise localization (DLR) and accurate classification (DRP) to obtain high quality object detection performance.

Method	AP	AP@0.5	AP@0.75	AP_s	AP_m	AP_l
Grid R-CNN [36]	39.6	58.3	42.4	22.6	43.8	51.5
Grid R-CNN Plus [37]	40.2	58.4	43.4	22.7	44.1	53.1
Our DLR alone	41.5	59.6	44.8	23.3	44.9	55.2
Ours Final: D2Det	42.7	61.5	46.3	24.5	46.2	56.9

Table 3: Comparisons of our dense local regression (DLR) with the grid guided keypoint-based localization utilized in Grid R-CNN and Grid R-CNN Plus. Our DLR alone provides superior results, especially for large objects, compared to Grid R-CNN and its variant.

	RetinaNet [32]	LRF-Net [47]	FPN [31]	NDFT [48]	**D2Det**
AP	33.95	37.81	49.05	52.03	**56.92**

Table 4: Object detection performance comparison on UAVDT test set. Other than LRF-Net, all methods employ ResNet101 with FPN. Our D2Det achieves superior results compared to the recently introduced NDFT detector.

4.3. UAVDT Dataset

Here, we present the results (Tab. 4) of our detector, D2Det, on UAVDT [11]. In addition to category-level annotation, all frames in UAVDT are annotated with UAV-specific nuisances: flying altitude, camera views, and weather conditions. The dataset is particularly challenging due to variations in view angle, illumination, altitude, and object scale. As in [48], we use ResNet101 with FPN. Following the authors of UAVDT [11], the results are reported using PASCAL VOC AP with IoU= 0.7. Among existing methods, the recently introduced NDFT detector [48] which explicitly learns domain-robust features by exploiting free metadata obtains 52.03 AP. Our D2Det outperforms NDFT by achieving Ap score of 56.92. Fig. 4(b) shows qualitative results on the UAVDT test set.

5. Instance Segmentation

In addition to object detection, we present the effectiveness of our D2Det, with the modifications described in Sec. 3.3, for instance segmentation task. Tab. 5 shows the state-of-the-art comparison on MS COCO `test-dev`. Among existing instance segmentation methods, the Hybrid Task Cascade (HTC) [6] which interweaves box and mask branches and employs a semantic segmentation branch to capture spatial context, obtains a mask AP of 39.7. Our method provides a two-fold speedup over HTC, while achieving a mask AP of 40.2.

We also report results (Tab. 6) on recently introduced iSAID dataset [51] for instance segmentation in satellite imagery. It contains 655,451 instances for 15 classes (roundabout, baseball diamond, large vehicle, plane, storage tank, ship, ground track field, tennis court, swimming pool, basketball court, harbor, small vehicle, bridge, helicopter, and soccer ball field). The dataset is challenging due to presence of large number of objects per image, limited appearance details, variety of small objects, large scale variations and high class imbalance. Our D2Det achieves superior results compared to existing works reported on this dataset [51]. Fig. 6 shows qualitative results on MS COCO `test-dev` (first row) and iSAID test set (second row).

Methods	Backbone	Time	Mask AP	AP@0.5	AP@0.75
MNC [9]	ResNet101	-	24.6	44.3	24.8
FCIS [30]	ResNet101	-	29.2	49.5	-
MaskLab [7]	ResNet101	-	35.4	57.4	37.4
Mask R-CNN [19]	ResNet101	116	35.7	58.0	37.8
PANet [34]	ResNet50	-	36.6	58.0	39.3
MS R-CNN [23]	ResNet101	116	38.3	58.8	41.5
Cascade Mask R-CNN [6]	ResNet101	156	38.4	60.2	41.4
HTC [6]	ResNet101	339	39.7	**61.8**	43.1
D2Det (Ours)	ResNet101	168	**40.2**	61.5	**43.7**

Table 5: State-of-the-art instance segmentation comparison (with a single model performance) in Mask AP on MS COCO `test-dev`. Other than MNC, FCIS and MaskLab, all methods employ FPN. The speed of all the methods is reported on Titan Xp. Our D2Det provides a two-fold speedup over HTC, while achieving a 40.2 mask AP.

Methods	Backbone	Mask AP	AP@0.5	AP@0.75
Mask R-CNN [19]	ResNet101	25.7	51.3	22.7
PANet [34]	ResNet101	34.2	56.6	35.8
D2Det (Ours)	ResNet101	**37.5**	**61.0**	**39.8**

Table 6: State-of-the-art instance segmentation comparison in Mask AP on iSAID test set.

Figure 6: Instance segmentation results of our D2Det on COCO `test-dev` (top row) and iSAID test (bottom row).

6. Conclusions

We propose a two-stage detection method that addresses both precise object localization and accurate classification. For precise localization, we introduce dense local regression that predicts multiple dense box offsets for a proposal. Further, a discriminative RoI pooling scheme is proposed which performs adaptive weighting to enhance discriminative features. Our D2Det achieves state-of-the-art detection results on MS COCO and UAVDT. Additionally, we present results for instance segmentation on MS COCO and iSAID, achieving promising results compared to existing methods.

Acknowledgments This work was supported by National Natural Science Foundation of China (Nos. 61906131, 61632018), Postdoctoral Program for Innovative Talents (No. BX20180214), and China Postdoctoral Science Foundation (No. 2018M641647).

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. *Proc. IEEE International Conf. Computer Vision*, 2017. 6

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018. 2, 6

[3] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. *Proc. IEEE International Conf. Computer Vision*, 2019. 6

[4] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 1

[5] Jiale Cao, Yanwei Pang, Shengjie Zhao, and Xuelong Li. High-level semantic networks for multi-scale object detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 2020. 2

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 2, 8

[7] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018. 8

[8] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Schmidt Feris, Jinjun Xiong, and Thomas S. Huang. Revisiting rcnn: on awakening the classification power of faster rcnn. *Proc. European Conf. Computer Vision*, 2018. 1

[9] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016. 8

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *Proc. IEEE International Conf. Computer Vision*, 2017. 3, 5

[11] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proc. European Conf. Computer Vision*, 2018. 2, 5, 8

[12] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 2

[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2

[14] Ziteng Gao, Limin Wang, and Gangshan Wu. Lip: Local importance-based pooling. *Proc. IEEE International Conf. Computer Vision*, 2019. 5, 6

[15] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 1

[16] Spyros Gidaris and Nikos Komodakis. Attend refine repeat: Active box proposal generation via in-out localization. *Proc. British Machine Vision Conference*, 2016. 2

[17] Ross Girshick. Fast R-CNN. *Proc. IEEE International Conf. Computer Vision*, 2015. 1, 2, 3, 4

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014. 1, 2

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *Proc. IEEE International Conf. Computer Vision*, 2017. 2, 3, 4, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proc. IEEE International Conf. Computer Vision*, 2016. 5

[21] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 1

[22] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv:1509.04874*, 2015. 2

[23] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 2, 8

[24] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. *Proc. European Conf. Computer Vision*, 2018. 2

[25] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Consistent optimization for single-shot object detection. *arXiv:1901.06563*, 2019. 6

[26] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *Proc. European Conf. Computer Vision*, 2018. 2, 6

[27] Shuai Li, Lingxiao Yang, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Dynamic anchor feature selection for single-shot object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 1

[28] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 2, 6

[29] Yazhao Li, Yanwei Pang, Jianbing Shen, Jiale Cao, and Ling Shao. Netnet: Neighbor erasing and transferring network for bettersingle shot object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2020. 1

[30] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017. 8

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid

networks for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017. 1, 2, 5, 6, 8

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Proc. IEEE International Conf. Computer Vision*, 2017. 6, 8

[33] 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. *Proc. European Conf. Computer Vision*, 2014. 2, 5, 6, 7

[34] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018. 2, 8

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Proc. European Conf. Computer Vision*, 2016. 1

[36] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 5, 6, 7, 8

[37] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN plus: Faster and better. *arXiv:1906.05688*, 2019. 2, 5, 6, 7, 8

[38] Jing Nie, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Enriched feature guided refinement network for object detection. In *The IEEE International Conference on Computer Vision*, 2019. 6

[39] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: Towards balanced learning for object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 2, 6

[40] Yanwei Pang, Tiancai Wang, R. M. Anwer, F. S. Khan, and L. Shao. Efficient featurized image pyramid network for single shot detector. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 1

[41] Junran Peng, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Pod: Practical object detection with scale-sensitive network. *Proc. IEEE International Conf. Computer Vision*, 2019. 1

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016. 1

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Proc. Advances in Neural Information Processing Systems*, 2015. 1, 2, 3, 5

[44] Bharat Singh, Mahyar Najibi, and Larry S. Davis. SNIPER: Efficient multi-scale training. *Proc. Advances in Neural Information Processing Systems*, 2018. 2

[45] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 6

[46] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 2

[47] Tiancai Wang, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 8

[48] Zhenyu Wu, Karthik Suresh, Priya Narayanan, Hongyu Xu, Heesung Kwon, and Zhangyang Wang. Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach. In *Proc. IEEE International Conf. Computer Vision*, 2019. 5, 8

[49] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. *Proc. IEEE International Conf. Computer Vision*, 2019. 6

[50] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. *Proc. IEEE International Conf. Computer Vision*, 2019. 2, 6

[51] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2, 8

[52] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 2

[53] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 6

[54] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019. 6