2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Dual Refinement Network for Single-Shot Object Detection

Xingyu Chen, Xiyuan Yang, Shihan Kong, Zhengxing Wu, and Junzhi Yu

Abstract— **Object detection methods fall into two categories, i.e., two-stage and single-stage detectors. The former is characterized by high detection accuracy while the latter usually has a considerable inference speed. Hence, it is imperative to fuse their merits for a better accuracy vs. speed trade-off. To this end, we propose a dual refinement network (DRN) to boost the performance of the single-stage detector. Inheriting from the advantages of two-stage approaches (i.e., two-step regression and accurate features for detection), anchor refinement and feature offset refinement are conducted in a novel anchor-offset detection, where the detection head is comprised of deformable convolutions. Moreover, to leverage contextual information for describing objects, we design a multi-deformable head, in which multiple detection paths with different receptive field sizes devote themselves to detecting objects. Extensive experiments on PASCAL VOC and ImageNet VID datasets are conducted, and we achieve a state-of-the-art detection performance in terms of both accuracy and inference speed.**

I. Introduction

Recent years have witnessed significant progress in object detection with deep convolutional neural networks (CNN). The prevalent detection networks fall into two categories, i.e., two-stage approaches [1]–[5] and single-stage detectors [6]–[12]. The two-stage process usually sees state-of-the-art detection accuracy, but it induces high time costs. As the pioneering work, YOLO [6] and SSD [8] made attempts to detect objects in real time. Thus, they got rid of region proposal and tried to localize and classify objects using a single-shot network.

It is known that high detection accuracy of two-stage approaches come with two major merits: i) Two-step regression, i.e., a region proposal process coarsely localizes objects, then a detection head precisely regresses them. ii) Accurate features for detection, i.e, through ROI polling, the features in ROI are used for classification and regression. On the contrary, ignoring the objects' location, the single-stage detector directly predicts coordinates from handcrafted anchors (reference bounding boxes), and the features for detection are spatially fixed on feature maps. Therefore, the SSD-like detector also should be endowed with the effective two-stage processes. Inheriting a part of two-stage merits (i.e., the two-step cascaded regression), Zhang *et al*. proposed a RefineDet to address class imbalance problem and elevate detection accuracy [11]. However, features for detection still did not follow the refined anchors. For augmenting the spatial sampling locations, Dai *et al*. proposed deformable convolutional networks to combat fixed geometric structures in traditional convolution operation [13]. Inspired by them, we attempt to propose a deformable architecture to seek "accurate" single-stage features for detection.

On the other hand, the receptive field of detection head in SSD-like networks is just suited to describe corresponding anchor zone. However, underlying context is usually ignored, which is crucial for object relation. For rich context, the two-stage CoupleNet utilized global features and local parts to describe a proposed region, and saw considerably improved detection accuracy [5]. Therefore, the detection head of single-stage methods is in urgent need of a contextual design.

The present study is a succeeding research and improvement on SSD [8] and RefineDet [11]. We design a novel single-stage detector with dual refinement structure, namely dual refinement network (DRN). To inherit the merits of both two-stage and single-stage detectors, our framework is a single-shot network with two-step cascaded regression. That is, refined anchors are firstly computed, which will be used for further regression [11]. Features used for detecting should also be refined to adapt anchor changes, so we predict their offsets using the refined anchors, called feature offset refinement. Composed by deformable convolutions, the detection head takes over the feature maps, refined anchors, and feature offsets for precise prediction, namely, anchor-offset detection. Different from traditional deformable convolution, the offsets used in our deformable detection head are predicted by refined anchors rather than the feature itself. In consideration of contextual information are important for describing objects, we propose a multi-deformable head with multiple detection paths to diversify detection receptive field. Our contributions are summarized as follows:

- Drawing inspiration from the merits of two-stage methods, we propose an anchor-offset detection including an anchor refinement, a feature offset refinement, and a deformable detection head to further improve the performance of the single-stage detector.
- A multi-deformable head is designed to leverage both region-level features and contextual information for describing objects.
- The DRN sees 82.0% mean average precision (mAP) vs. 55.2 frames per second (FPS) on VOC2007 test set [15] and 69.4% mAP vs. 40.5 FPS on ImageNet VID validation set [18].

This work was supported by the National Natural Science Foundation of China (nos. 61633004, 61633020, 61603388, 61633017, and 61725305), and by the Beijing Natural Science Foundation (no. 4161002).

X. Chen, X. Yang, S. Kong, Z. Wu, and J. Yu are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China and University of Chinese Academy of Sciences, Beijing 100049, China. J. Yu is also with the Beijing Innovation Center for Engineering Science and Advanced Technology, Peking University, Beijing 100871, China (e-mail: chenxingyu2015@ia.ac.cn, jzyangxiyuan@gmail.com, kongshihan2016@ia.ac.cn, zhengxing.wu@ia.ac.cn, junzhi.yu@ia.ac.cn).

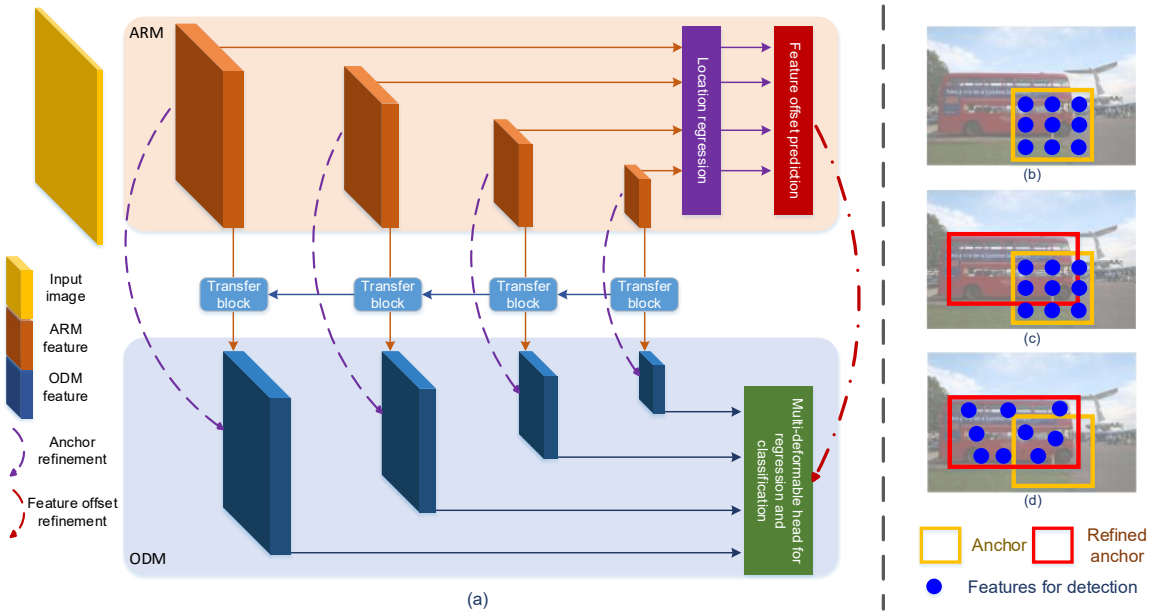

Fig. 1. Diagram of our designs. (a) The architecture of the DRN. The ARM and ODM are organized with FPN pipeline. Refined anchors are produced by coarse regression with ARM features, and they are first employed to predict feature offsets, namely, feature offset refinement. The detection head utilizes ODM feature maps, refined anchors, and feature offsets to detect objects, i.e., anchor-offset detection. Finally, a multi-deformable head is designed for rich contextual information; (b) Detection mode of SSD; (c) Detection mode of RefineDet; (d) Detection mode of DRN. Our approach aims to use features in refined anchors for detecting.

II. Related Work

A. Single-Stage Detector

Initially, YOLO significantly improved detection speed, but it used to miss small objects [6]. To address this problem, SSD used shallow layers with low-level features for detecting tiny objects [8]. Moreover, it employed convolution layers as the detection head and saw a favorable trade-off between accuracy and speed. Afterwards, many improved versions of SSD have emerged. For example, Fu *et al.* added deconvolution module behind convolution layers to include more high-level expression in small object detection [9]; Lin *et al.* developed a RetinaNet with feature pyramid networks [14] for propagating information in a top-down manner to enlarge shallow layers' receptive filed [10]. Zhang *et al.* proposed a RefineDet with strategies of two-step cascaded regression and negative anchor filtering to deal with class imbalance problem [11]. RefineDet inherited a part of merits of two-stage detectors, and it performed well in terms of accuracy vs. speed trade-off, i.e., 80.0% mAP vs. 40.3 FPS on VOC2007 test set [15]. In short, current single-stage detectors have advantageous inference speed and modest detection accuracy.

B. Tow-Stage Detector

Represented by RCNN family, two-stage approaches [1]–[5] are usually composed of region proposal part and detection network. The former (e.g., RPN [3]) generates sparse object proposals, while the detection module takes over ROI features for precise regression and classification. To date, the two-stage approaches still dominate the detection accuracy on generic benchmarks. For example, Zhu *et al.* developed a CoupleNet to fuse global information with local parts for detection, which achieved 82.7% mAP on VOC2007 test set [5]. However, the CoupleNet merely ran at 8.2 FPS. Simply put, the two-stage detectors perform more accurately, but they usually suffer from high time costs. In our opinion, they have two major advantageous strategies. On one hand, the process of region proposal preliminarily proposes sparse candidate objects. Conversely, the single-stage detector directly localizes objects from handcrafted anchors. On the other hand, the detection head in two-stage approaches leverages the ROI features for detecting objects. On the contrary, the features for detection in the single-stage pipeline are spatially fixed on feature maps, ignoring the objects' location. Therefore, there is an imperative need of endowing the single-stage detector with the aforementioned merits for a better accuracy vs. speed trade-off.

III. Network Architecture

As shown in Fig. 1(a), our proposed architecture is a single-shot network with a forward backbone (i.e., VGG-16 [17]) for feature extraction, where *fc6,fc7* in original VGG-16 are converted to convolutional layers, namely, *Conv6,Conv7*. The network generates a fixed number of bounding boxes and corresponding classification scores, followed by the non-maximum suppression (NMS) for duplicate removal. Similar to RefineDet [11], we also employ an anchor refinement module (ARM) and an object detection module (ODM)

for two-step regression. The ARM takes over pyramidal feature hierarchy as the input, and regresses coordinates as refined anchors. In addition, we predict feature offsets using refined anchors for ODM. Subsequently, the ODM fuses low-level features with high-level features for better semantic information. Ultimately, a creative detection head is designed with deformable convolution for final classification and regression, whose inputs are ODM features, refined anchors, and feature offsets, namely, anchor-offset detection. Furthermore, we develop a multi-deformable head to leverage contextual information for detection.

A. Anchor-Offset Detection

In general, detection in traditional SSD-like manner is based on handcrafted anchors which are rigid and usually inaccurate. As shown in Fig. 1(b), predefined anchors and fixed features could not be suited to regressing and classifying objects (e.g., the bus). Through preliminary localization, refined anchors are in favor of more precise detection (see Fig. 1(c)). However, the RefineDet still uses inaccurate features (shown with blue dots in Fig. 1(c)) for detection. To overcome these difficulties, we design an anchor-offset detection including an anchor refinement, a feature offset refinement, and a deformable detection head, whose motivation is shown in Fig. 1(d).

1) Anchor Refinement: This process is analogous in essence to RefineDet [11], i.e., using ARM to generate refined anchors that provide better initialization for the second-step regression. Similar to SSD, we firstly place regularly tiled anchors a_o on each feature map cell. Each feature layer is associated with one specific scale of anchors. In detail, we adopt the anchor size of $[32, 64, 128, 256]$ for 4-scale feature maps from low-level to high-level and tile 3 anchors at each feature map cell with aspect ratios of $[1.0, 2.0, 0.5]$. The ARM generates the same number of refined anchors a_r using ARM features f_{arm} with convolution operation,

$$a_r = (W_{ar} * f_{arm} + b_{ar}) \oplus a_o, \tag{1}$$

where $*$ denotes convolution (W, b are weights and bias); $\oplus$ represents anchor decoding operation [8].

2) Deformable Detection Head: We design a deformable detection head for final classification and localization. The standard detection head in SSD uses a regular 3×3 grid $\mathcal{R}$ to predict category probability and coordinates for a feature map cell. In the meantime, through careful anchor design, $\mathcal{R}$ can describe a specific anchor zone (see Fig. 1(b)), but it usually fails to describe the refined anchor (shown in Fig. 1(c)), which could result in inaccuracy. Thereby, allowing $\mathcal{R}$ to adapt to the anchor change, we develop a deformable detection head to capture accurate features with the feature offset δp,

$$P_{p_0} = \sum_{p \in \mathcal{R}} w(p) \cdot f_{odm}(p + \delta p). \tag{2}$$

where P is the prediction of category probability or coordinates; w is the convolution weight; p represents positions in $\mathcal{R}$ while p_0 is the center; f_{odm} denotes ODM features. The bilinear interpolation allows δp to be a fraction [13].

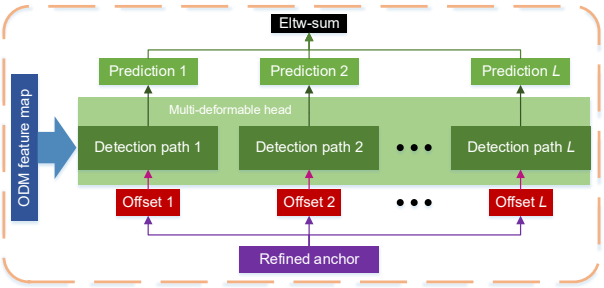

Fig. 2. Multi-deformable head. We design multiple detection paths with different receptive field sizes. Additionally, feature offset refinement is independent for each path. Finally, we fuse their results using element-wise summation.

3) Feature Offset Refinement: Originally, The offset $\Delta p = \{\delta p\}$ is computed with the feature fed into the deformable convolution, i.e.,

$$\Delta p = W_{fr} * f_{odm} + b_{fr}. \tag{3}$$

Nevertheless, there is a strong demand for describing the zone of refined anchors with the deformed grids $\mathcal{R}$ (see Fig. 1(d)). Therefore, we predict feature offsets for ODM according to refined anchors, i.e., feature offset refinement.

$$\Delta p = W_{fr} * (W_{ar} * f_{arm} + b_{ar}) + b_{fr}. \tag{4}$$

In detail, this operation is a 1×1 convolution. Since each spatial element in $(W_{ar} * f_{arm} + b_{ar})$ is related to coordinate predictions for refined anchors tiled at a specific feature map cell, we fuse its channel information for feature offset refinement.

In this way, the offsets are targeted for more effective detection, when compared to traditional deform pipeline. We call this mode anchor-offset detection, which can be formulated as

$$\begin{aligned} P_{local} &= (W_{loc} * (f_{odm}, \Delta p) + b_{loc}) \oplus a_r \\ P_{class} &= W_{conf} * (f_{odm}, \Delta p) + b_{conf}. \end{aligned} \tag{5}$$

B. Multi-Deformable Head

The CoupleNet developed local and global FCNs to detect objects [5]. The local FCN focused local features in a region proposal while the global one paid attention to the whole region-level features. In this way, more contextual information and underlying object relation are exploited for high-quality detection. Thus, we develop a spiritually similar structure for the single-stage DRN, namely, multi-deformable head. We take aim to describe the object using original, shrunken, and expansile region-level features.

To this end, we employ multiple detection paths with different receptive field sizes. As shown in Fig. 2, each detection path is an anchor-offset detection, and their feature offset refinement is independent. Additionally, their results are fused with element-wise summation. Mathematically, the detection with multi-deformable head can be given as follows:

$$\begin{aligned} P_{local} &= \textstyle\sum_{l=1}^{L} (W_{loc_l} * (f_{odm}, \Delta p_l) + b_{loc_l}) \oplus a_r \\ P_{class} &= \textstyle\sum_{l=1}^{L} W_{conf_l} * (f_{odm}, \Delta p_l) + b_{conf_l}. \end{aligned} \tag{6}$$

IV. Training and Inference

A. Training Settings and Objective

We train the DRN in an end-to-end manner, and the pretrained VGG-16 model on ImageNet [18] is employed. The other parameters in DRN are initialized with "xavier" method [19]. L2 normalization is used to scale norms of *Conv4_3,Conv5_3* to 10 and 8, respectively. Additionally, we add batch normalization (BN) [20] in VGG-16 and extra layer for effective training. In terms of optimization, SGD optimizer with 0.9 momentum and 0.0005 weight decay is employed to train the whole network. The initial learning rate is set as 0.001, which is divided by 10 at the 130th and by 100 at the 170th epoch. The total iteration is 190 epochs. For better generalization ability, some data augmentation strategies are used to train a robust model, e.g., random clipping, flipping, expansion, photometric distortion. [8].

We design a multi-task objective to train DRN including two localization losses $\mathcal{L}_{loc-arm}, \mathcal{L}_{loc-odm}$ and a confidence loss $\mathcal{L}_{conf}$,

$$\mathcal{L} = \frac{1}{N_{arm}}\mathcal{L}_{loc-arm} + \frac{1}{N_{odm}}(\mathcal{L}_{loc-odm} + \mathcal{L}_{conf}), \quad (7)$$

where N is the number of positive boxes in ARM and ODM. The $\mathcal{L}_{loc}$ and $\mathcal{L}_{conf}$ are consistent with original SSD [8].

B. Inference

In ARM, VGG-16 and extra layers extract visual features for anchor refinement as well as feature offset refinement. Then, key features are transformed to ODM with FPN pipeline and transfer blocks. The ODM takes over refined anchors as well as feature offsets, and outputs confident object candidates (confident scores > 0.01) in the manner of anchor-offset detection and multi-deformable head. Subsequently, these candidates are processed by NMS with 0.45 jaccard overlap per class, and we retain top 200 high confident objects as the final detections.

V. Experiment

Our models are trained and evaluated on VOC2007, VOC2012, and ImageNet VID datasets, and we demonstrate a better accuracy vs. speed trade-off.

A. Runtime Performance

Our method is implemented under the PyTorch framework. The training and experiments are carried out on a workstation with an Intel 2.20 GHz Xeon(R) E5-2630 CPU, NVIDIA TITAN-X GPUs, CUDA 8.0, and cuDNN v7.

With 320×320 input image, the DRN can run at $55.2, 56.0, 40.5$ FPS on VOC2007, VOC2012 test sets and VID validaton set, respectively, which is considerably superior to two-stage methods and surpasses most single-stage detectors. Only YOLO is slightly faster than ours, but our method is fairly prominent on detection accuracy. In terms of 512×512 input, our approach achieves $32.2, 33.6$ FPS on VOC2007 and VOC2012 test sets.

TABLE I

Effectiveness of various designs. All model are trained on VOC2007 and VOC2012 trainval set, and validated on VOC2007 test set. The baseline is 79.1%.

Component		DRN320				
multi-deformable head?			✓			✓
feature offset refinement?		✓	✓		✓	✓
deformable detection head?	✓	✓	✓		✓	✓
BN for VGG&extra?				✓	✓	✓
mAP(%)	78.3	79.8	80.5	81.1	81.7	82.0

B. Ablation Studies on VOC2007

We use PASCAL VOC2007 to study proposed models in detail, which has 20 object categories. Following most methods, we train the model on the union set of VOC2007 and VOC2012 trainval set ($16,551$ images) and evaluate on VOC2007 test set ($4,592$ images). We use mAP as the criterion of detection accuracy. For the convenience of comparison, the RefineDet without negative filtering [11] is adopted as the baseline, and we obtain 79.1% mAP based on our re-produced PyTorch implementation (Note that it is 79.5% original Caffe implementation). The changes of mAP are listed in Table I.

1) Anchor-Offset Detection: Anchor-offset detection is composed of anchor refinement, feature offset refinement, and deformable detection head, where the anchor refinement has been studied in RefineDet [11]. Thus, we analyze the latter two components in this section. At first, deformable convolution is employed as the detection head, and we obey original deform pipeline [13], i.e., the offsets are computed with ODM features (formulated by (3)). However, this strategy can not make an improvement and leads 0.8% drop in mAP (i.e., 78.3% vs. 79.1%). The shortage of this tactic is evident, i.e., the refined anchors are given by ARM while the offsets are predicted by ODM features, so they are not tightly associated. Thus, the deformed grid can hardly adapt to refined anchors without extra supervision, and it even trails original regular grid.

Therefore, the feature offset refinement is of crucial importance in DRN. In our method, the feature offset is highly correlated with refined anchors, and the network is benefited from the proposed anchor-offset detection. Finally, we find that mAP rises by 0.7% (i.e., 79.8% vs. 79.1%).

2) Multi-Deformable Head: The effect of various multi-deformable designs is shown in Table II. At first, 1×1 grid is employed to utilize shrunken region-level features, but we find it incurs negligible effectiveness. The 1×1 grid should have focused on most suitable local parts for detection, but feature offsets are computed with refined anchors in our pipeline, ignoring suitable local parts. Then, 3×3 grid with dilation is devised as one of the detection paths, but it leads to 0.4% drop in mAP. Although it expands the receptive field, the dilated 3×3 grid splits features, failing to describe objects effectively. To cover the shortage, we deem that 5×5 grid without dilation could work more effectively, and experimentally, it invites 0.7% rise in mAP (i.e., 80.5% vs.

TABLE II

EFFECTIVENESS OF VARIOUS MULTI-DEFORMABLE HEAD DESIGNS. WE USE DIFFERENT KERNEL SIZE (k) AND DILATION (d) TO VALIDATE THE EFFICACY OUR DESIGNS.

$k = 5 \times 5, d = 1$?				✓	✓
$k = 3 \times 3, d = 2$?			✓		
$k = 1 \times 1, d = 1$?		✓	✓	✓	
$k = 3 \times 3, d = 1$?	✓	✓	✓	✓	✓
mAP(%)	79.8	79.8	79.4	80.5	80.3

79.8%) because more contextual information is involved. In addition, we remove the 1×1 detection path and find this more efficient design still can reach 80.3% mAP, so this design is used for subsequent experiments. In addition, these comparisons also indicate that the improvement of multi-deformable head comes from above-analyzed reasons rather than increasing parameter size.

3) Discussion: BN is an effective tactic that solves vanishing and exploding gradient problem [20], so we try to introduce this trick for more efficient training. Experimentally, the model performs better when BN layers are added in the feature extractor (i.e., the VGG-16 and extra layer), and it sees a significant improvement in accuracy, i.e., 81.1% mAP. Subsequently, the anchor-offset detection and multi-deformable head further boost the performance. Referring to Table I, removing multi-deformable head leads to 0.3% drop in mAP, and removing anchor-offset detection invites another 0.6% mAP drop. Finally, the DRN results in the state-of-the-art detection performance with such a small input size, i.e., 82.0% mAP and 320×320 input.

C. Results on VOC2007

Referring to Table III, the DRN is compared with state-of-the-art methods. Using 320×320 input image, the DRN achieves 82.0% mAP without bells and whistles, which surpasses all methods with such small inputs. With 82.8% mAP for 512×512 input size, the DRN outperforms all compared approaches. Although the CoupleNet [5] has a similar mAP to ours, it uses ResNet-101 [16] as its backbone, and its results come with larger input size (i.e., 1000×600). Considering both mAP and FPS, we can draw a conclusion that the DRN achieves a better accuracy vs. speed trade-off.

D. Results on VOC2012

More challenging VOC2012 dataset is employed to evaluate our proposed method, and we use the union set of VOC2007 and VOC2012 trainval sets plus VOC2007 test set ($21,503$ images) for training in this experiment, and test trained model on VOC2012 test set ($10,991$ images).

As shown in Table III, our model obtains $79.3\%, 80.6\%$ mAP with 320×320 and 512×512 input size, respectively. The 320×320 result surpasses all the compared methods with similarly small input size and the 512×512 result is highest in Table III, which validate the effectiveness of our developed approaches again.

TABLE III

DETECTION RESULTS ON PASCAL VOC DATASET. THE FPS IS FOR VOC2007, AND THE MAP IS DEMONSTRATED IN THE FORM OF "VOC2007/VOC2012".

Method	Backbone	Input size	FPS	mAP(%)
two-stage				
Fast RCNN [2]	VGG-16	1000×600	0.5	70.0/68.4
Faster RCNN [3]	VGG-16	1000×600	7	73.2/70.4
HyperNet [21]	VGG-16	1000×600	0.9	76.3/71.4
ION [22]	VGG-16	1000×600	1.3	76.5/76.4
Faster RCNN [3]	ResNet-101	1000×600	2.4	76.4/73.8
R-FCN [4]	ResNet-101	1000×600	9	80.5/77.6
CoupleNet [5]	ResNet-101	1000×600	8.2	82.7/80.4
single-stage				
YOLO [6]	GoogleNet	448×448	45	63.4/57.9
YOLOv2 [7]	Darknet-19	544×544	40	78.6/73.4
RON384 [12]	VGG-16	384×384	15	75.4/73.0
SSD300 [8]	VGG-16	300×300	46	77.2/75.8
SSD512 [8]	VGG-16	512×512	19	79.8/78.5
SSD321 [9]	ResNet-101	321×321	11.2	77.1/75.4
SSD513 [9]	ResNet-101	513×513	6.8	80.6/79.4
DSSD321 [9]	ResNet-101	321×321	9.5	78.6/76.3
DSSD513 [9]	ResNet-101	513×513	5.5	81.5/80.0
RefineDet320 [11]	VGG-16	320×320	40.3	80.0/78.1
RefineDet512 [11]	VGG-16	512×512	24.1	81.8/80.1
DRN320	VGG-16	320×320	55.2	82.0/79.3
DRN512	VGG-16	512×512	32.2	**82.8**/**80.6**

Fig. 3. Detection examples on VOC2012 test set. We draw all detected boxes with > 0.5 confidence score. Our model works well with occlusions, truncations, inter-class interference, etc.

Intuitively, we demonstrated typical detection results in Fig. 3. It is shown that the DRN performs well in terms of challenging scenes, e.g., i) The chairs and tables are disordered; ii) The car in the second sub-figure is quite small, and another car in the last sub-figure suffers from serious occlusion or truncation; and iii) The bicycles and persons encounter motion blur. Despite these challenges, the DRN is able to localize and classify them accurately.

E. Results on ImageNet VID

We use ImageNet VID dataset to validate video detection performance of the DRN, which contains 30-class targets. Following [29], we train the DRN with VID and DET (only using the data from the 30 VID classes; $81,830$ images in total), and test it on VID validation set ($176,126$ images).

TABLE IV

COMPARISON OF THE DRN AND SEVERAL PRIOR AND CONTEMPORARY APPROACHES ON IMAGENET VID VALIDATION SET.

Method	Backbone	Real-time?	mAP(%)
temporal methods			
Closed-loop [24]	VGG-M		50.0
Seq-NMS [25]	VGG-16		52.2
LSTM-SSD [26]	MobileNet [30]	✓	54.4
TCNN [27]	DeepID+Craft		61.5
TSSD [28]	VGG-16	✓	65.4
TPN [29]	GoogLeNet [23]		68.4
D&T [31]	ResNet-101		79.8
static methods			
Faster RCNN [3]	GoogLeNet		63.0
SSD300 [8]	VGG-16	✓	63.0
RefineDet320 [11]	VGG-16	✓	66.7
DRN320	VGG-16	✓	**69.4**

For high video detection speed, we use 320×320 image as the input.

As shown in Table IV, we compare the DRN with some temporal and static detection methods. The results of SSD and Faster RCNN are reported in [28], [29] while the result of RefineDet is based on our re-produced implementation. Although it ignores temporal information in video, our approach achieves 69.4% mAP and 40.5 FPS, which is superior to all compared methods except for the D&T [31]. However, the D&T result (i.e., 79.8% mAP) is induced by a two-stage detector and a tracking method, which also lead to considerable amount of computational cost. Therefore, the DRN also shows great potentials in temporal detection, and it can be employed for real-world tasks.

VI. CONCLUSION

In this paper, a novel DRN is designed for the purpose of real-time accurate object detection. Differing from existing approaches, the DRN inherits the merits of both single-stage and two-stage detectors, so it simultaneously has accurate detection performance and fast inference speed. In particular, an anchor-offset detection, including an anchor refinement, a feature offset refinement, and a deformable detection head, is proposed to migrate two effectiveness of region proposal to the single-stage detector. More specifically, to leverage both region-level features and contextual information for detection, we devise a multi-deformable head with multiple detection paths. As a result, the DRN achieves a considerably enhanced accuracy vs. speed trade-off on PASCAL VOC and ImageNet VID datasets.

In the future, we plan to further improve the temporal detection performance.

REFERENCES

[1] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Columbus, USA, Jun. 2014, pp. 580–587.

[2] R. Girshick. "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[3] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. in Neural Info. Process. Syst.*, Montreal, Canada, Dec. 2015, pp. 91–99.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Info. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 379–387.

[5] Y. Zhu *et al.*, "Couplenet: Coupling global structure with local parts for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Aug. 2017, pp. 4126–4134.

[6] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, the US, Jun. 2016, pp. 779–788.

[7] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *arXiv:1612.08242,* 2016.

[8] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, Netherlands, Oct. 2016, pp. 21–37.

[9] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.

[10] T. Y. Lin *et al.*, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[11] S. Zhang *et al.*, "Single-shot refinement neural network for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Salt Lack City, USA, Jun. 2018, pp. 4203–4212.

[12] T. Kong *et al.*, "RON: reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition,* Hawaii, USA, Jun. 2017, pp. 5936–5944.

[13] J. Dai *et al.*, "Deformable Convolutional Networks", in *Proc. Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp.764–773.

[14] T. Y. Lin *et al.*, "Feature pyramid networks for object detection," in *arXiv:1612.03144*, 2016.

[15] M. Everingham *et al.*, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[16] K. He *et al.*, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, the US, Jun. 2016, pp. 770–778

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

[18] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Inter. Conf. Artificial Intell. Statist.*, Sardinia, Italy, May 2010, pp. 249–256.

[20] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[21] T. Kong, *et al.*, "Hypernet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, the US, Jun. 2016, pp. 845–853.

[22] S. Bell *et al.*, "Insideoutside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, USA, Jun. 2016, pp. 2874–2883.

[23] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Boston, USA, Jun. 2015, pp. 1–9.

[24] L. Galteri *et al.*, "Spatio-temporal closed-loop object detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1253–1263, 2017.

[25] W. Han *et al.*, "Seq-NMS for video object detection," *arXiv:1602.08465*, 2016.

[26] M. Liu and M. Zhu, "Mobile video object detection with temporally-aware feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Salt Lack City, USA, Jun. 2018, pp. 5686–5695.

[27] K. Kang *et al.*, "T-CNN: tubelets with convolutional neural networks for object detection from videos", *IEEE Trans. Circuits Syst. Video Technol.*, DOI:10.1109/TCSVT.2017.2736553.

[28] X. Chen, J. Yu, and Z. Wu, "Temporally Identity-Aware SSD with Attentional LSTM," *IEEE Trans. Cybern.*, to be published, doi:10.1109/TCYB.2019.2894261.

[29] K. Kang *et al.*, "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Hawaii, USA, Jul, 2017, pp. 727–735.

[30] A. Howard *et al.*, "Mobilenets: efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861,* 2017.

[31] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 3038–3046.