LED: LOCALIZATION-QUALITY ESTIMATION EMBEDDED DETECTOR

Shiquan Zhang[1], *Xu Zhao*[1]*, *Liangji Fang*[1], *Haiping Fei*[2], *Haitao Song*[2]

[1]Department of Automation, Shanghai Jiao Tong University
[2]Industrial Internet Innovation Center (Shanghai) Co., Ltd

ABSTRACT

Classification subnetwork and box regression subnetwork are essential components in deep networks for object detection. However, we observe a contradiction that before NMS, some better localized detections do not correspond to higher classification confidences, and vice versa. This contradiction exists because classification confidences can not fully reflect the **localization-quality** (**loc-quality**) of each detection. In this work, we propose the **Localization-quality Estimation embedded Detector** abbreviated as **LED**, and a corresponding detection pipeline. In this detection pipeline, we first propose an accurate loc-quality estimation method for each detection, then combine the loc-quality with the corresponding classification confidence during inference to make each detection more reasonable and accurate. For efficiency, LED is designed as an one-stage network. Extensive experiments are conducted on Pascal VOC 2007 and KITTI car detection datasets to demonstrate the effectiveness of LED.

Index Terms— Accurate Localization-quality Estimation, Fully Convolutional Network, One-stage Detector

1. INTRODUCTION

The prevalent deep networks for object detection could be divided into two main groups: two-stage methods and one-stage methods. In two-stage methods [1, 2], the first stage aims at generating a sparse set of proposals and the second stage aims at refining the proposals. Through a sequence of advances [2, 3, 4, 5, 6, 7], the two-stage framework reaches better detection performance. The one-stage methods [8, 9, 10, 11] perform classification and box regression based on densely predefined anchors. The main advantage of one-stage approaches is high efficiency. Recently, Focal Loss [12] is proposed to demonstrate that the one-stage framework has the potential to achieve comparable performance. In aforementioned approaches, classification and box regression subnets are essential, and final detections are obtained after classification confidence based NMS.

* Corresponding author. This research has been supported by the National Key Research and Development Program of China (Grant No. 2017YFC0806501) and NSFC Program (61673269, 61273285).

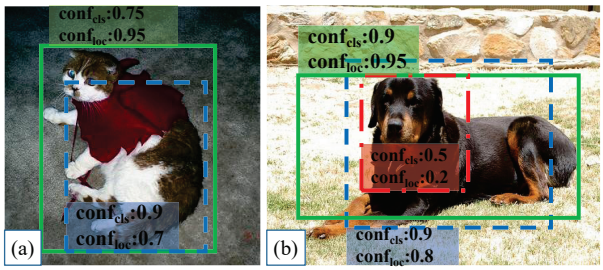

Fig. 1. Selected detections before NMS from LED. The classification confidence and the localization-quality of each detection are obtained. conf$_{cls}$ indicates the classification confidence and conf$_{loc}$ indicates the localization confidence from localization-quality estimation. (Best view in color)

In these methods, however, we find a contradiction that before NMS, some better localized detections do not correspond to higher classification confidences and vice versa. As shown in Fig. 1 (a), the better localized (solid green) detection corresponds to a lower classification confidence (0.75) while the badly localized (dashed blue) detection corresponds to a higher classification confidence (0.9). In NMS, these better localized detections may be discarded due to lower classification confidences, thus reducing the detection performance.

The contradiction arises from the independence of classification subnet and box regression subnet. Generally speaking, classification task calls for translation-invariant features, that is, the shift of an object inside the image should be indiscriminative. Meanwhile translation-sensitive features are of vital importance to box regression task, that is, the translation of an object inside the image region should produce meaningful responses for indicating how well the detected region is overlapped with the object. The two subnets are trained with different loss functions, which may lead to different optimization directions. Classification confidences can not fully reflect the loc-quality of each detection.

Previous works have been made to handle the above contradiction. YOLO [8, 9] encodes IoU (intersection of union) between a detection and a nearby object to indicate the objectiveness of the detection then combine the objectiveness score with classification score. YOLO obtains the objectiveness score by a simple regression approach with MSE loss and oth-

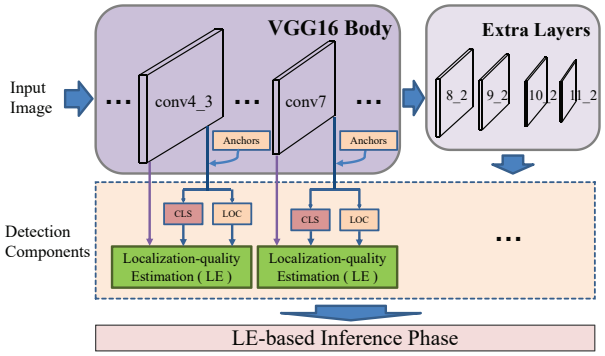

Fig. 2. Framework of LED. On each selected layer, classification subnet (CLS), box regression subnet (LOC) and Loc-quality Estimation subnet (LE) are constructed. Detection components of extra layers are omitted for clear viewing.

er approaches [10, 4] could simply encode objectiveness by regarding background as another category into classification subnet. Dai *et al.*[13] present position-sensitive RoI pooling operation to extract position-sensitive regional features, thus easing the dilemma between translation-invariance in image classification and translation-sensitiveness in box regression.

As shown in Fig.1, with the proposed Localization-quality Estimation embedded Detector (LED), we not only employ classification and box regression subnets, but also propose an explicit loc-quality estimation method for each detection. To connect classification subnet with box regression subnet meanwhile utilizing their informative features, features from classification and box regression subnets are fused as richer features for loc-quality estimation. The estimated loc-quality and the classification confidence of each detection are obtained then combined for final inference. To demonstrate the effectiveness of proposed components in LED, ablation studies are conducted on PASCAL VOC 2007. LED also achieves the state-of-the-art performance on KITTI car detection task.

2. OUR APPROACH

2.1. Framework

As shown in Fig. 2, following SSD [10], we utilize the àtrous VGG16-net [14] as our backbone and add several extra layers from conv8_1 to conv11_2. Then, LED detects multi-scale objects on selected multi-level layers respectively. On each selected layer, anchors are uniformly distributed. Meanwhile the classification subnet is built for classifying anchors while the box regression subnet is built for regressing anchors to nearby objects. The loc-quality estimation module is simultaneously constructed to obtain loc-quality of each detection, based on the richly fused features from the selected layer (*e.g.* con4_3) and the corresponding classification and box regression subnets. Last, loc-quality estimation based inference is introduced to aggregate all the detection results.

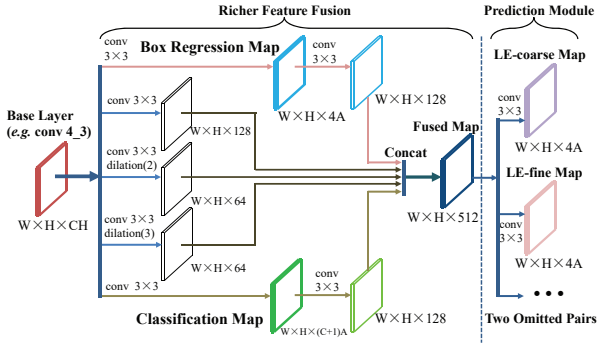

Fig. 3. Structure of the LE module. W×H×CH denotes spatial resolution and channels. A denotes the number of predefined anchors on each location while C denotes number of categories. For clear illustration, another two omitted pairs of coarse-fine maps are built, in parallel with the plotted pair.

2.2. Anchors, Classification and Box Regression

Following SSD [10], anchors are empirically set on each selected layer with multiple sizes based on the receptive field, and with multiple aspect ratios. As shown in Fig. 2, 3×3 convolutional layers are built independently on each selected layer, either for classification or for box regression. For instance, on a feature map with size $w \times h$, at each of the $w \times h$ locations, LED predicts $4A$ offsets for A anchors and $(C{+}1)A$ classification scores for $(C{+}1)$ categories (background is encoded as another category). The standard box parameterization is employed from [10]. Softmax loss is employed as the classification loss (L_{cls}) and smooth-L1 loss is employed as the box regression loss (L_{reg}).

2.3. Loc-quality Estimation (LE) Module

Model. We model the loc-quality of a detection by several spatial cues. Let S_{det}, S_{gt} and S_I denote the area of a detection, a ground truth box and the intersection of the detection and the ground truth, respectively. Naturally, the spatial cue $IoU = \frac{S_I}{S_{det}+S_{gt}-S_I}$ (intersection of union) reflects how well the detection overlaps the ground truth. Thus, IoU is referred as $overall\text{-}quality$ for localization. Furthermore, for robust and accurate estimation, we additionally define $IoD = \frac{S_I}{S_{det}}$ and $IoG = \frac{S_I}{S_{gt}}$. IoD reflects the probability that the detection contains an object, thus named $objectiveness\text{-}quality$ of the detection. Meanwhile IoG reflects the spatial ratio of an nearby object lying in the detection box, thus named $completeness\text{-}quality$ of the detection. The three qualities are adopted as the loc-quality of each detection and encoded into LED. For clear expression, we denote set V=$\{IoD, IoG, IoU\}$. V is class-agnostic because the class-specific prediction fails to reach improved performance due to limited training data in our experiments.

Richer Features. The classification subnet and box regres-

sion subnet are informative for loc-quality estimation, so the features from classification subnet and box regression subnet should be exploited. In addition, context information also benefits loc-quality estimation. Hence, on each selected prediction layer (*e.g.* conv4_3), features built by 3×3 filters on the classification and box regression feature maps are concatenated with the base and context feature maps, as a richly fused feature map. The feature fusion is shown in Fig. 3. Note that by the feature fusion, classification subnet and box regression subnet are connected, and both of them receive gradients from LE module. For efficiency, dilated convolution [15, 16] is adopted to encode context information.

Prediction Module. Direct regression for V can not obtain precise loc-quality of each detection in our experiments. Thus, a **coarse-to-fine** (**C2F**) prediction module is introduced. As shown in Fig. 3, on top of each fused feature map, three pairs of coarse-fine feature maps are parallel built for the three elements in V (the LE-coarse map is built for the coarse procedure while the LE-fine map is constructed for the fine procedure, one pair for one element in V).

In the coarse procedure, prediction is regarded as a classification problem. The value range 0-1 is discretized into four ranges $\{$0-0.1, 0.1-0.4, 0.4-0.7, 0.7-1.0$\}$, referred as the background value range, the low value range, the middle value range and the high value range respectively.

In the fine procedure, four independent regressors correspond to the four value ranges respectively. The regressors regress continuous values relative to "anchors" in corresponding value ranges. The "anchors" are set to $\{$0.05, 0.25, 0.55, 0.85$\}$, as the median of each value range. V is obtained by

$$v = \sum_{i=1}^{4}(prob_i \cdot val_i), \forall v \; in \; V \tag{1}$$

where v denotes IoU, IoD, or IoG. $prob_i$ denotes the probability of the i-th value range and val_i denotes the finely regressed value of the i-th value range.

LE Loss. For C2F prediction module, LE loss L_{LE} is composed of six weighted losses from two types (the coarse procedure loss L_{coarse} and the fine procedure loss L_{fine}). Each element in V donates a L_{coarse} and a L_{fine}. We adopt Softmax loss as L_{coarse} and propose the Sharp-L2 loss as L_{fine}. Compared to L2 loss, the Sharp-L2 loss up-weights the losses assigned to badly-regressed examples, thus leading to a finer regression procedure. The Sharp-L2 loss is defined as

$$Sharp{-}L2(x) = \begin{cases} \frac{1}{2} \cdot x^2 & ,|x| < 1 \\ \frac{1}{3} \cdot |x|^3 + \frac{1}{6} & ,|x| \geq 1 \end{cases} \tag{2}$$

Each loss term in L_{LE} is normalized with the number of corresponding input samples. Under this normalization, weights of the six losses are empirically set to 1.

2.4. Training

To embed LE module into our one-stage framework, we introduce a three-step mechanism to optimize LED: (1) In the first step, we train our detector without LE module, only using L_{cls} and L_{reg}. Training objective is defined as $L_1 = L_{cls} + \alpha \cdot L_{reg}$. This stage is identical to SSD [10]. (2) In the second step, we freeze all the weights and bias except LE module and only L_{LE} is employed, hence $L_2 = L_{LE}$. (3) Finally, we unfreeze all the weights and bias, then introduce a fully end-to-end training step. We employ L_{cls}, L_{reg} and L_{LE}, hence $L_3 = L_{cls} + \alpha \cdot L_{reg} + \beta \cdot L_{LE}$. Each loss term is normalized by the number of input samples. α and β are empirically set to 1 and $1/3$, respectively.

Some training strategies are utilized. (1) We match anchors with ground truth boxes to obtain positive samples. (2) Hard negative mining [17] is employed to balance negative and positive samples for classification and box regression. In addition, another independently modified hard example mining procedure is introduced for LE module, based on the L_{LE}, and we ensure that the ratio among samples from the four value ranges is around 3:1:1:1. (3) We employ data augmentation methods such as expanding, cropping and color distortion to improve the generalization performance of LED.

2.5. Inference

Most approaches apply NMS only by the classification confidences while LED performs NMS based on both the estimated loc-quality and the classification confidence of each detection. Forwarding an image through the network, we obtain a dense set of detections, with the classification confidence and the loc-quality set V for each detection. Based on the definitions of IoU, IoD and IoG in Section 2.3, we derive

$$IoU' = \frac{IoD \cdot IoG}{IoD + IoG - IoD \cdot IoG} \tag{3}$$

We obtain localization confidence $conf_{loc}$ from the directly predicted IoU in V and the derived IoU' in Equation 3: $conf_{loc} = \lambda \cdot IoU + (1 - \lambda) \cdot IoU'$. Integrating $conf_{cls}$ and $conf_{loc}$, overall confidence $conf$ of each detection could be simply defined as $conf = conf_{cls} \cdot conf_{loc}$ (denoted as LE-Product). By Gaussian penalty function, we define $conf$ (denoted as LE-Gaussian) as

$$conf = conf_{cls} \cdot e^{-\frac{(1-conf_{loc})^2}{\sigma}} \tag{4}$$

where λ and σ are set to 0.6 and 1 respectively, and we find that they work effectively and robustly in our experiments. Finally, NMS is applied based on the $conf$ of each detection.

3. EXPERIMENTS

Experiments are conducted on two publicly available datasets. Ablation studies are conducted on Pascal VOC 2007 dataset. We also conduct experiments on KITTI object detection (Car only) dataset to report the state-of-the-art performance of LED. All experiments are built with Caffe [19] on a single NVIDIA Titan X (Pascal) GPU.

3.1. PASCAL VOC 2007

We compare LED with Faster R-CNN [4], SSD [10] and the most recently proposed RON [18], and conduct ablation stud-

Table 1. **PASCAL VOC 2007 test results.** All methods are based on pre-trained VGG16, and trained with VOC 2007 *trainval* and VOC 2012 *trainval*. $\star$ indicates our own reproducing of SSD300, slightly higher than the original one [10].

Approach	FPS	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster R-CNN [4]	–	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
RON384 [18]	–	75.4	78.0	82.4	76.7	67.1	**56.9**	85.3	84.3	86.1	55.5	80.6	71.4	84.7	84.8	82.4	76.2	47.9	75.3	74.1	83.8	74.5
SSD300*	**94**	77.6	79.2	84.0	76.1	69.5	50.6	86.9	85.9	88.7	60.4	81.3	**76.8**	86.2	87.4	83.6	79.4	**52.9**	79.2	79.6	87.6	**77.1**
LED300	65	**78.7**	**82.7**	**86.5**	**76.9**	**71.7**	51.7	**87.1**	**88.0**	**89.9**	**60.8**	**84.0**	74.9	**88.2**	**87.9**	**85.1**	**81.3**	52.5	**79.5**	**80.8**	**87.6**	76.8

Table 2. **Ablation studies on Pascal VOC 2007.** $\surd$ denotes the setting of corresponding column is employed. Otherwise, base prediction feature map instead of richer features (RF), direct regression instead of coarse-to-fine (C2F), L2 loss instead of Sharp-L2 loss, LE-Product instead of LE-Gaussian.

Model	RF	C2F	Sharp-L2	LE-Gaussian	mAP
LED300					77.4
	$\surd$				77.9
	$\surd$	$\surd$			78.3
	$\surd$	$\surd$	$\surd$		78.5
	$\surd$	$\surd$	$\surd$	$\surd$	**78.7**
SSD300$_{240k}$					77.7

ies. Based on pre-trained VGG16 networks, all methods are trained on VOC 2007 *trainval* and VOC 2012 *trainval* then tested on VOC 2007 *test* set. We adopt the standard evaluation metric (mAP) with IoU=0.5, as described in [20].

Implementation Details: For fair comparison, LED shares the same settings for anchors, classification subnets and box regression subnets, as described in SSD [10]. Input size is set to 300×300 for LED300 and SSD300. Batch size is set to 28. LED is trained by SGD with weight decay of 0.0005 and momentum of 0.9. In the first stage, LED is trained with a learning rate of 10^{-3} for the first 80k iterations, then 10^{-4} for 20k iterations and 10^{-5} for another 20k iterations. In the second stage, learning rate is set to 10^{-4} for 20k iterations then 10^{-5} for 20k iterations. In the third stage, learning rate is set to 10^{-4} for 40k iterations then 10^{-5} for 40k iterations.

Results and Analysis: In Table 1, LED outperforms SSD300 by 1.1% mAP. LED reaches the best performance among these approaches. As shown in Table 2, the loc-quality estimation related components in LED help to improve detection performance. Considering the three-step training phase of LED, for fair comparison, we also fine tune SSD300 model for more iterations and denote it as SSD300$_{240k}$ in Table 2. The performance of SSD300$_{240k}$ further verifies that the gains of LED come from the designs described in Section 2.

In Table 1, we find that LED does well in most cases while hurting the performance for inaccurately annotated categories like dining table. LED also fails to reach high mAP on very small objects like bottles, which may be caused by the weakly semantic features of conv4_3. Last, Inference speed will drop a little due to additionally added structures.

3.2. KITTI Car Detection

We also conduct experiments on the challenging KITTI [21] car detection task. Each ground truth is annotated with several attributes indicating difficulties (Easy, Moderate and Hard). IoU threshold is set to 0.7 for Car in evaluation. All methods are ranked based on the **moderately** difficult results.

Table 3. **KITTI car detection results on validation subset.** All methods share the same dataset splits. $\star$ indicates that the detection results and inference time are obtained from corresponding references, otherwise from our experiments. *Time* indicates mean inference time for one image. *Mod* denotes moderate difficulty and is the metric for ranking.

Approach	Time	Easy	**Mod**	Hard
3DVP [24]$\star$	40s	80.48	68.05	57.20
Faster R-CNN [4]$\star$	2s	82.91	77.83	66.25
SubCNN [22]$\star$	2s	95.77	86.64	74.07
DeepMANTA (GoogLenet) [23]$\star$	0.7s	**97.90**	91.01	**83.14**
DeepMANTA (VGG16) [23]$\star$	2s	97.45	91.47	81.79
SSD	0.07s	96.50	88.11	77.52
LED (single)	0.11s	97.31	91.32	81.23
LED (ensemble)	0.33s	97.51	**91.93**	83.11

Implementation Details: Since the annotations of the KITTI test set are not available, training images are split into train subset (3762 images) and validation subset (3799 images) as described in [22, 23]. Input size is set to 1920×576. Similar to [9], by K-means clustering parameters on train subset, we set aspect ratios of anchors as $\{1.0, 1.5, 1.8, 2.2, 2.7\}$. As KITTI is more challenging, we train LED with a learning rate of 10^{-5} for 30k iterations then 10^{-6} for another 30k iterations in all the three training steps. Our implemented SSD and LED share the same settings except the LE module.

Experimental Results: Table 3 shows that LED reaches the state-of-the-art performance with a fast inference speed, outperforming our implemented SSD by more than 3% mAP. Compared to [23] and [22], LED is less time consuming due to the efficient one-stage framework. Last, single model of LED reaches comparable performance as [23], and LED could achieve high mAP of 91.93% with the ensemble results from multi-scale and flipping testing methods [25].

4. CONCLUSION

To conclude, we propose an accurate loc-quality estimation method within an one-stage framework. The proposed loc-quality estimation module helps to obtain the accurate estimated loc-quality of each detection. Then in the LE-based inference phase, the loc-quality and the classification confidence of each detection are combined to make each detection more reasonable, thus boosting the detection performance.

5. REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[2] Ross Girshick, "Fast r-cnn," in *CVPR*, 2015, pp. 1440–1448.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*. Springer, 2014, pp. 346–361.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[5] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*. Springer, 2016, pp. 354–370.

[6] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *CVPR*, July 2017.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick, "Mask r-cnn," in *ICCV*, Oct 2017.

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

[9] Joseph Redmon and Ali Farhadi, "Yolo9000: Better, faster, stronger," in *CVPR*, July 2017.

[10] 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.

[11] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg, "DSSD : Deconvolutional single shot detector," *CoRR*, vol. abs/1701.06659, 2017.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal loss for dense object detection," in *ICCV*, Oct 2017.

[13] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016, pp. 379–387.

[14] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser, "Dilated residual networks," in *CVPR*, 2017, vol. 1.

[16] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[17] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016, pp. 761–769.

[18] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen, "Ron: Reverse connection with objectness prior networks for object detection," in *CVPR*, 2017, vol. 1, p. 2.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[20] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman, "The pascal visual object classes challenge 2007 (voc 2007) results (2007)," 2008.

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012, pp. 3354–3361.

[22] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *WACV*. IEEE, 2017, pp. 924–933.

[23] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Celine Teuliere, and Thierry Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *CVPR*, July 2017.

[24] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese, "Data-driven 3d voxel patterns for object category recognition," in *CVPR*, 2015, pp. 1903–1911.

[25] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016, pp. 2874–2883.