RFSSD : Rich Feature Single Shot Detector for Small Objects

Jiasong Guo
Shaghai Jiao Tong University
7_GUO_7@sjtu.edu.cn

Ruoyu Cheng
Shaghai Jiao Tong University
roy_account@sjtu.edu.cn

Yixiong Wang
Shaghai Jiao Tong University
wyx_ei@sjtu.edu.cn

ABSTRACT

In this paper, we propose a new method called Rich Feature Single Shot Detector (RFSSD). RFSSD is a single-stage detection method which achieves especially superior performance in small objects. Besides, RFSSD performs better than existing single-stage methods in terms of inference time. We extract low-level features with fine semantic information and add them to the original high-level features. Afterwards, we develop a more descriptive feature maps for small objects by implementing both bottom-up and top-down scheme. We perform experiments on PASCAL VOC2007 test which achieves 78.9% mAP at 22.03 FPS with only 300× 300 input. RFSSD improves the baseline by 1.5 points, particularly with a 3.6%, 2.4% gain on XS and S category.

1 Introduction

With the advent of deep convolutional networks (ConvNets [5]), the performance of object detection has been significantly improved. However, small object detection is still a challenging problem due to the relatively small area and the information in the small area picture. There are two mainly used detectors: single-stage methods [3] and two-stage [16] approaches. In general, the two-stage method is mainly better at precision, while the advantage of the single-stage method is speed.

To identify objects of all sizes, most previous detectors were based on hand-crafted features [5], using image pyramids [13]. But considering memory and inference time, those jobs are computationally expensive.

The size of the representation is critical in detection. In recent years, traditional features designed by hand have been replaced by features computed by convolutional neural networks. Recent detection systems use the top-most feature maps calculated by ConvNets on a single input scale to predict candidate bounding boxes with different scales and aspect ratios. However, the top-most feature map has a fixed receiving field that conflicts with objects of different proportions in the natural image. In particular, detection defects of small objects may impair object detection performance.

Current single-stage detectors are designed to match the accuracy of two-stage detection methods. Despite showing impressive results on large and medium objects, these detectors still perform worse than expected on small objects.

In order to solve the multi-scale problem, SSD [15] and MS-CNN [2] use pyramid-shaped feature hierarchy from bottom to top to adapt to objects of various sizes. However, the semantic information of the lowest ConvNets layer is weak, which will impair its ability to represent small objects. Latest networks try to observe and utilize pyramid features by building a top-down architecture with horizontal connections to a large extent. Compared to traditional detectors, these networks have significantly improved accuracy. However, we notice that these methods make use of the deconvolution layer of the topmost feature map, which completely loses the fine details of small objects.

As mentioned earlier, when detecting objects of different proportions (especially small objects), low-level / medium-level information and high-level semantics are required. Modern object detectors usually use top-down pyramid feature representations, where high-level information at the top or higher level is fused with high-resolution features that are semantically weaker at the bottom or previous level. Although this top-down feature pyramid representation improves performance, it only injects high-level semantics into previous layers. In addition, this pyramid representation is constructed by fusing layer by layer. In this work, we believe that fusing high-level information to the previous layer and low / medium-level information to the back layer are essential for multi-scale target detection.

2 Related Work

There are some traditional methods for detection like Haar [20] and DPM [7], which are based on sliding-window paradigm, using hand-crafted features.

In recent years, with the rapid development of ConvNets, integrating feature learning and classifier have improved the accuracy and speed to a great extent.

SPPnet [11]designs the Spatial Pyramid Pooling layer so that the input images of any sizes are feasible, which is efficient in computation. R-CNN [10] and Fast R-CNN [9] use selective search to generate bounding boxes, extracting features with CNN and classifying them by SVM. SSD, which uses multi-scale features, makes predictions by using small convolutional filters of 3×3 on six features at different depths from bottom to top. The recent methods, FPN [14] and TDM [18], adopt top-down pathway and conduct skip connections in their architectures to enhance the power of features. DSSD [8] applies deconvolution layers to the top of SSD to realize up-sampling and then achieves connection with convolutional feature maps.

Inspired by these researches, we propose RFSSD method which achieves especially superior performance in small objects.

3 Our approach

Our approach is based on the standard SSD framework, which takes VGG16 as a pre-trained network backbone. SSD utilizes a ConvNet's pyramidal feature hierarchy as a featurized image pyramid, intending to produce multi-scale feature maps from different layers with different levels of semantic information. The multi-scale features and single-shot architecture provide good mAP result and real-time detection for SSD, yet the performance of detecting small objects is poor due to weak semantic information on the low-level features. To optimize the performance on small objects as well as maintain the high speed of SSD structure, we propose a more complicated architecture by reusing the shallow layers in VGG backbone for small object detection with no additional layers.

Figure 1 is our overall framework for 300×300 input. As empirical result shown, shallow feature maps (conv1_3 - conv3_3) have small receptive fields, which are significant for small object detection. So we take advantage of these fine semantic information in a bi-directional network, combining them with high-level features and propagating features from the former to later layers in a cascaded fashion. In addition, we adopt a dense top-down architecture afterwards to perform a layer-by-layer fusion of CNN layers, which believes to be helpful for the detection layer.

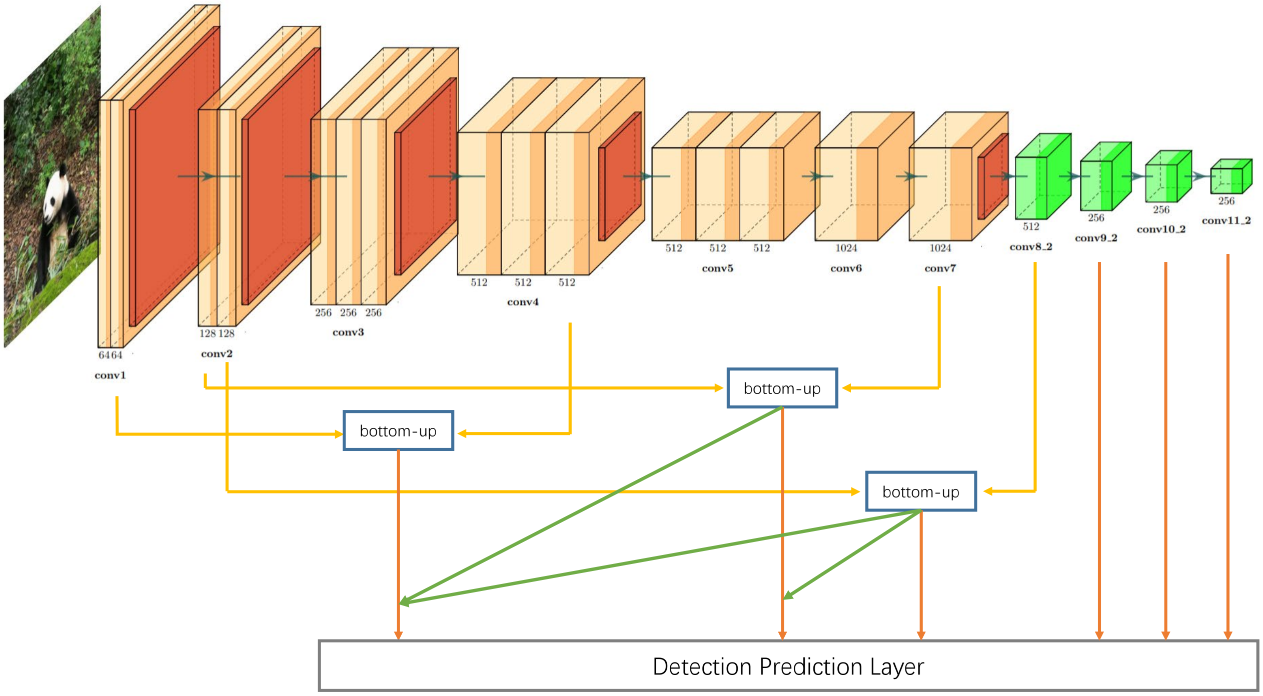


Fig. 1. Structure of RFSSD

structure of the bi-directional network

The first task for our bi-directional network is to combine shallow feature maps with high-level features. By performing element-wise multiplication, we augment low level features to make them more semantic and informative. Meanwhile, a bottom-up scheme helps to propagate the resulting features of different levels from former to later in cascaded way. To sum up, the k^{th} forward feature is calculated by:

$$f_k = \phi_k\left((m_k \otimes n_k) \oplus \phi_k^{'}(f_{k-1})\right)$$

where m_k is the k^{th} low level feature map, n_k denotes the k^{th} original prediction backbone of SSD and $\phi_k^{'}(.)$ is for down-sampling the forward feature. The $\phi_k(.)$ operation contains a ReLU block and a 3×3 conv. block. $\otimes$ and $\otimes$ represent element-wise multiplication and addition. It is noted that the first forward feature has no f_{k-1}, so we can set $\phi_k^{'}(f_{k-1})$ to all zeros.

After that, the dense top-down scheme connects all the features from later layers to the current layer. It places emphasis on high-level semantics and makes all features smoother. For the k^{th} backward feature b_k, we concatenate all higher level features:

$$b_k = \theta_k\left(\gamma_k(f_k), \gamma_{ik}\left(\sum_{k+1}^{n} \mu_k(f_i)\right)\right)$$

where $\mu_k(.)$ is for up-sampling the higher forward features for concatenation, $\gamma_k(.)$ operation is a 1×1 conv. block, and $\sum$ stands for concatenation. At the end, all backward features $b_1, b_2, ..., b_n$ are used as the prediction features, just as those in the SSD.

4 Experiments

We train our on VOC2012 and VOC2007 [6] and evaluate on VOC2007. All the experiments are carried on three NVIDIA GTX 1080 GPUs. We compare RFSSD with the state-of-the-art two stage detectors including Faster [17], ION [1], MS-CNN [2] and single-stage detectors including SSD [15], and MDSSD [21] in terms of general mAP, mAP on sizes and FPS.

4.1 PASCAL VOC2007

The trainset of VOC2007 combined with VOC2012 contains 16551 images in total. The testset contains 4952 images. Taking both the limited GPU memory and model generalization into consideration, we set the batch size to 32, the initial learning rate to 10^{-3} and decay to 10^{-4} in the 60000^{th} iteration.The momentum and weight decay are set to 0.9 and 0.0005 respectively by using SGD. We save and test our model every 5000 iterations, and we find that loss converges at 75000^{th}-iteration model.

4.2 Detection Result

Our model performs far better than others with a mAP of 78.9%. The result of RFSSD and other models is shown in Table1. We see that RFSSD is better than SSD by 1.3% mAP and 0.6% mAP compared of the state-of-the-art two stage detector MS-CNN. In particular, RFSSD performs rather good at detecting relatively small objects, such as cat (1.2% mAP to SSD) and chair (3.1% mAP to SSD), which ranks the first among all the methods. Besides, RFSSD reaches the highest mAP of bottle and overwhelms all other single-stage detectors in the experiment. It improves **8.2%** mAP compared with SSD and **3.7%** mAP for MDSSD. One thing to notice is that bottle is a bottleneck of small object detection, mAP for bottles with size XS of SSD is 0. This improvement verifies RFSSD is effective in detecting small objects and we owe it to the contribution of the bottom-up scheme and feature extraction in RFSSD, with more features remained, the predictor can thus gain more knowledge about the relatively small objects.

Method	mAp	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Two stage detectors																					
Faster [17]	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
ION [1]	75.6	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4
MS-CNN [2]	78.2	80.3	84.1	78.5	70.8	**68.5**	**88.0**	85.9	87.8	60.3	85.2	73.7	**87.2**	86.5	85.0	76.4	48.5	76.3	75.5	85.0	81.0
Single-stage detectors																					
SSD [15]	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	**77.0**	86.1	87.5	83.4	79.4	52.3	77.9	**79.5**	**87.6**	76.8
MDSSD [21]	78.6	**86.5**	**87.6**	**78.9**	70.6	55.0	86.9	**87.0**	88.1	58.5	84.8	73.4	84.8	**89.2**	**88.1**	78.0	52.3	78.6	74.5	86.8	**80.7**
RFSSD	**78.9**	81.2	85.7	77.2	**73.6**	**58.7**	85.4	86.8	**88.4**	**63.7**	**85.7**	76.6	86.4	88.0	84.1	**79.8**	**53.8**	**79.1**	80.3	87.4	76.3

Table 1. Detection results on PASCAL VOC2007

We also test mAP for specific sizes of all the objects. We sort the objects within its category and add another attribute(size) into its corresponding xml file. The new attributes are divided using the following criteria: extra-small (XS: bottom 10%); small (S: next 20%); medium (M: next 40%); large (L: next 20%); extra-large (XL: next 10%). Given a specific size, we calculate the mAP within this attribute. We mainly focus on the objects whose original size category is same as the detection category and calculate its corresponding precision and recall. Finally, we derive the mAP for different sizes. The result is shown in Table2. It is easy to see that RFSSD performs really well on small object detection, it is better than SSD by 3.6% in terms of extra-small objects and 2.5% on small objects. One of our contribution here is that we improve the AP for birds from **0.1%**(SSD) to **32.9%**.

Size	Method	mAP	aero	bird	cat	cow	table	person	plant
XS	RFSSD	20.1	24.2	**32.9**	50.6	36.5	1.6	11.7	10.7
	SSD	16.5	18.7	0.1	36.9	22.5	1.9	10.9	9.7
S	RFSSD	48.9	68.7	43.3	73.8	74.6	33.6	46.6	25.7
	SSD	46.5	65.5	41.2	66.8	61.1	37.0	45.3	20.3

Table 2. Detection result on different sizes

When taking time consumption into consideration, we can assert that RFSSD is better than MDSSD and SSD. The FPS of RFSSD reaches 22 which means that RFSSD is still a real-time detector. Since there is no additional layers in RFSSD, the number of overall parameters to be learned is close to SSD, which are reflected by table 3.

Table 3. FPS on detection. The data for SSD and MDSSD are updated by us.

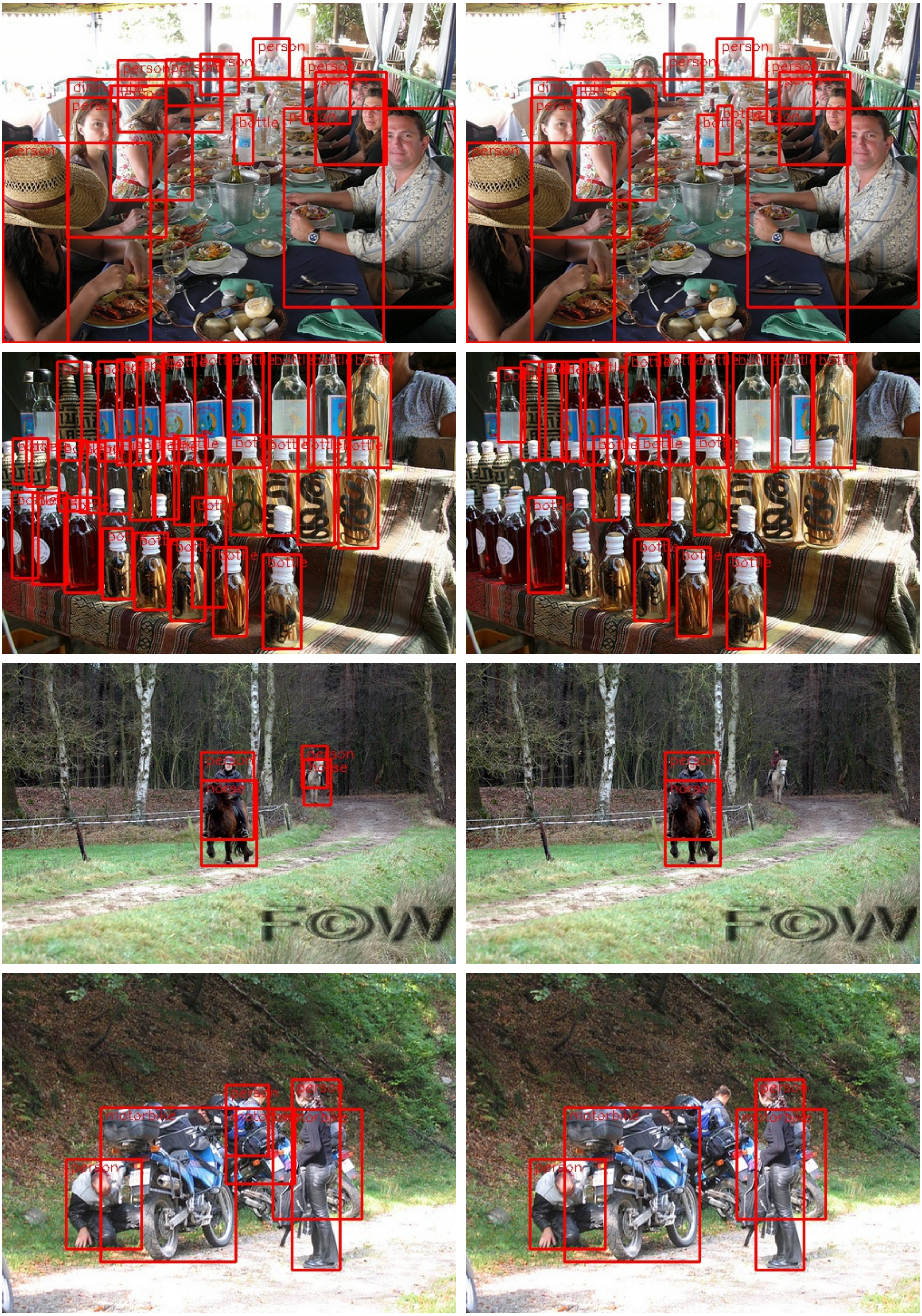
Method	backbone network	GPU	Input Size	speed(FPS)
SSD*	VGG16	1080	300 × 300	21.17
MDSSD*	VGG16	1080	300 × 300	12.66
RFSSD	VGG16	1080	300 × 300	**22.03**

The visualization of RFSSD and SSD is shown in Figure2. RFSSD model does much better in detection than SSD. And just as mentioned before, RFSSD is good at detecting bottles, the ratio of bottles detected by RFSSD is 89.5% compared with 47.4% SSD. Obviously, small objects in the background are detected by RFSSD while SSD not.

5 Conclusion and Future Work

We propose RFSSD method which performs excellent in small object detection. Not only do we extract low-level features with fine semantic information which are added to the original high-level features, but also develop a more descriptive feature maps for small objects by implementing both bottom-up and top-down scheme to low-level feature maps and make them more descriptive. Experiments conducted on benchmark datasets demonstrate the effectiveness of RFSSD. Futhermore, this time we take SSD as the baseline structure in our method , our idea can still be extended to other detectors like Faster R-CNN and so on.

Due to the limitation of time, we only carried out all experiments with VGG-16 backbone. In the future, we will try different backbones such as Resnet [19], Xception [4] and MobileNet [12]. Since that heavy-weighted networks may contribute to detection accuracy while light-weighted ones are qualified for real-time requirements, there is a trade-off in backbone selecting and we intend to find the best one to



(a) RFSSD (b) SSD

Fig. 2. Comparison of detection result between RFSSD and SSD

balance the challenge. Besides, we will modify our network input size to 512 × 512 to express more features to our model which may improve the performance in both the large and small object detections.

References

[1] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016.

[2] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *european conference on computer vision*, pages 354–370. Springer, 2016.

[3] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9705–9714, 2019.

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[8] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object

detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.

[19] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1(511-518):3, 2001.

[21] Mingliang Xu, Lisha Cui, Pei Lv, Xiaoheng Jiang, Jianwei Niu, Bing Zhou, and Meng Wang. Mdssd: Multi-scale deconvolutional single shot detector for small objects. *arXiv preprint arXiv:1805.07009*, 2018.