Learning Object Detectors from Scratch with Gated Recurrent Feature Pyramids

Zhiqiang Shen\textsuperscript{1,2}, Honghui Shi\textsuperscript{4,3}, Rogerio Feris\textsuperscript{3}, Liangliang Cao\textsuperscript{4}, Shuicheng Yan\textsuperscript{5}
Ding Liu\textsuperscript{1}, Xinchao Wang\textsuperscript{1}, Xiangyang Xue\textsuperscript{2}, Thomas S. Huang\textsuperscript{1}
\textsuperscript{1}IFP, Beckman Institute, University of Illinois at Urbana-Champaign \textsuperscript{2}Fudan University, \textsuperscript{3}IBM Research, \textsuperscript{4}HelloVera.AI, \textsuperscript{5}360 AI

Abstract

In this paper, we propose gated recurrent feature pyramid for the problem of learning object detection from scratch. Our approach is motivated by the recent work of deeply supervised object detector (DSOD) \cite{24}, but explores new network architecture that dynamically adjusts the supervision intensities of intermediate layers for various scales in object detection. The benefits of the proposed method are two-fold: First, we propose a recurrent feature-pyramid structure to squeeze rich spatial and semantic features into a single prediction layer that further reduces the number of parameters to learn (DSOD need learn 1/2, but our method need only 1/3). Thus our new model is more fit for learning from scratch, and can converge faster than DSOD (using only 50\% of iterations). Second, we introduce a novel gate-controlled prediction strategy to adaptively enhance or attenuate supervision at different scales based on the input object size. As a result, our model is more suitable for detecting small objects. To the best of our knowledge, our study is the best performed model of learning from scratch. Our method in the Pascal VOC 2012 comp3 leaderboard (which compares object detectors that are trained only with Pascal VOC data) demonstrates a significant performance jump, from previous 64\% to our 77\% (VOC 07\texttt{++}12) and 72.5\% (VOC 12). We also evaluate the performance of our method on Pascal VOC 2007, 2012 and MS COCO datasets, and find that the accuracy of our learning from scratch method can even beat a lot of the state-of-the-art detection methods which use pre-trained models from ImageNet. Code is available at: \url{https://github.com/szq0214/GRP-DSOD}.

1. Introduction

Learning object detection from scratch with high accuracy is both a challenging problem in computer vision and a promising newly proposed research direction in deep learning. While most state-of-the-art object detection systems are designed based on backbone networks that are pre-trained on the ImageNet, learning from scratch with only the target dataset has the potential benefits as the following: (i) can reduce learning bias due to the different data distribution between ImageNet and target datasets; (ii) can apply to datasets in different domains other than RGB datasets.

Recently, DSOD \cite{24} was proposed to tackle the learning from scratch problem and has demonstrated great performance on popular detection datasets such as Pascal VOC and MS COCO. One of the important design principle of DSOD, similar to many recently proposed detection frameworks such as SSD \cite{19}, DSSD \cite{6}, MS-CNN \cite{2}, FPN \cite{16}, ION \cite{1}, Focal loss \cite{17}, is using feature pyramid for multiple scale prediction during a single pass. For different pyramidal layers, a series of prediction operations are conducted to adapt the arbitrary object scales.

Despite its popular usage, a major drawback of current
feature pyramid design is their naive single-scaled feature representation in each pyramidal layer, as done in SSD [19] and FPN [16]. In these methods, the pyramidal layers are independent without any interactions. A natural way to tackle this issue is to fuse different scaled features into one pyramidal layer to enhance the representational power, as done by DSOD [24]. DSOD adopts a dense connection design to combine features from both previous and current scales into one pyramid. In this paper, we propose recurrent feature pyramid, a network that concatenates high-level semantic features and low-level spatial features in a single pyramid. Our ablation experiments show that this simple design can vastly boost the performance of object detection and reduce the model parameters.

Another critical limitation of feature pyramid adopted in the current state-of-the-art methods, is that each pyramid has the fixed contribution to the final supervision signals. Intuitively, objects at small scales may be easier to detect on fine-resolution (lower-level) features, and thus signals from those lower-level features should be enhanced; similarly, large scale objects could be easier to detect at coarse-resolution (higher-level) feature maps, and thus signals from those higher-level feature maps should be enhanced. This (as shown in Fig.1) motivates our work to apply gating mechanism for object detection. As illustrated in Fig.4, our gate consists of three components: (i) channel-level attention; (ii) global-level attention; and (iii) identity mapping, which will be described in detail in §3.3.

We incorporate our recurrent feature pyramid and gating mechanism into DSOD, and the integrated model, called Gated Recurrent Feature Pyramid DSOD (GRP-DSOD), has achieved new state-of-the-art results on PASCAL VOC [5] and MS COCO [18] datasets, resulting in both faster convergence and higher detection accuracy, as shown in Fig.2. Such incorporation is done without bells and whistles, and could be easily extended to other detection frameworks using feature pyramids such as SSD and FPN.

Our main contributions are summarized as follows:

(1) We propose Recurrent Feature Pyramid, a novel structure design that is more fit for learning detection from scratch and further improves performance in both convergence speed and accuracy.

(2) We propose Gating Mechanism for generic object detection. To the best of our knowledge, our proposed method is one of the first successful attempts on recalibrating supervision signals for object detection.

(3) We apply Gated Recurrent Feature Pyramid on an existing detection frameworks DSOD, and the resulting GRP-DSOD method achieves new state-of-the-art for learning detection from scratch with real-time processing speed and more compact models.

Figure 2: Comparison of DSOD [24] and GRP-DSOD on VOC 2007 test set. GRP-DSOD achieves 50% faster convergence than the baseline DSOD.

2. Related Work

Object Detection. Generally, modern object detection frameworks fall into two groups. One is the two-stage detectors like R-CNN [8], Fast RCNN [7], R-FCN [3], Faster RCNN [22], Mask RCNN [10], etc. In this paper we focus on another group: one-stage detectors (also known as proposal-free detectors). OverFeat [23] is one of the first CNN-based one-stage detectors. After that, a number of recent detectors have been proposed, such as YOLO [20, 21], SSD [19, 6] and DSOD [24]. Especially, DSOD is trained from scratch without ImageNet [4] pre-trained models. The key advantage of one-stage detectors is high speed but their accuracy is moderate. Thus, the aim of this work is to further boost the performance of one-stage detectors under the setting of training from scratch.

Feature Pyramid. Adopting multiple layers for generic object detection is a common practice in a range of recently proposed approaches [24, 16, 19, 6]. For instance, DSOD [24] applies dense connections in prediction layers to combine different resolution features for detecting objects. FPN [16] develops a top-down architecture with lateral connections to build pyramidal hierarchy semantic features at all scales. DSSD [6] involves extra deconvolutional layers to capture additional large-scale context. In this paper, we propose a novel recurrent feature pyramid to not only improve accuracy but also reduce parameters.

Gating Mechanism. Gating (or attention) can be viewed as a process to adaptively adjust or allocate resource intensity towards the most informative or useful components of inputs. There are several methods for exploiting gating mechanism to improve image classification [26, 12, 27] and detection [28]. GBD-Net [28] proposes a gated bi-directional CNN for object detection that passes messages between features from different regions and uses gated functions to control message transmission. SENet [12] uses gating mecha-
3. Method

We begin by reviewing the recently proposed Deeply Supervised Object Detector (DSOD) [24]. Then we present Recurrent Feature Pyramid, a network that combines adjacent layer features for object detection. Following that, we introduce how an elaborately designed gating mechanism is used to adaptively control supervision at multiple scales in a deep network. Finally, we show how the above two structures can be applied in DSOD seamlessly to obtain Gated Recurrent-Pyramid DSOD.

3.1. DSOD

The Deeply Supervised Object Detector (DSOD [24]) is built on top of a backbone network that ends with densely connected layers. Compared with SSD [19], DSOD replaces the VGGNet [25] base network with the “Stem + Dense Block [13]” structure and applies dense connections into multi-scale prediction layers, as shown in Fig. 3. At each scale of prediction layers, outputs are combined with adjacent feature maps that contain different levels of spatial and semantic information. DSOD only learns half new features and reuses half. In GRP-DSOD, we follow this strategy but adopt a recurrent concatenation approach. Thus, GRP-DSOD only learns one-third new features and reuses two-thirds, and thus makes the model more compact. We will elaborate it in the following sections.

3.2. Recurrent Feature Pyramid

Our goal is to utilize multi-scale CNN features in a single prediction pass that combines detailed shape and appearance cues from lower level layers and semantics from higher levels. In this paper, we adopt the DSOD [24] framework, as a key component of our approach, to build a Recurrent Feature Pyramid Network, which can be used to train CNN-based object detectors from scratch (without the ImageNet [4] pre-trained models).

Our proposed recurrent feature pyramid is split into three parts, shown in Fig. 5. First, a down-sampling pathway takes the low level feature maps, and through a down-sampling block outputs the low-resolution feature maps that should be concatenated with the current features. Then, to incorporate higher-level semantics, an up-sampling pathway is adopted to concatenate high level features to the current layer. Finally, we repeat the concatenation operation in each scale of prediction layers in a recurrent fashion.
Down-sample and Up-sampling Pathways. The down-sampling pathway consists mainly of a max-pooling layer (kernel size = 2 × 2, stride = 2), followed by a conv-layer (kernel size = 1 × 1, stride = 1) to reduce channel dimensions, which is similar to the DSOD down-sampling block. The up-sampling pathway generates higher resolution features by upsampling spatially coarser, but semantically stronger features from the adjacent scale (we use nearest resolution in the upper layer for simplicity). We conduct a deconvolutional operation via bilinear upsampling followed by a conv-layer (kernel size = 1 × 1, stride = 1) on the spatial resolution features maps. The upsampled maps are then concatenated with features from the down-sampling pathway and the current layer. Hence, each block has very rich multi-resolution features.

Learning one-third and reusing two-thirds. Fig. 5 shows the building block that constructs our recurrent feature pyramid. With coarser-resolution and fine-resolution features, we introduce a bottleneck block with a 1 × 1 conv-layer plus a 3 × 3 conv-layer to learn new features. The number of parameters is one-third compared with DSOD.

Concatenation with a Recurrent Scheme. Each concatenation operation merges feature maps of the same spatial resolution from the down-sampling pathway and the up-sampling pathway. The process is iterated until the coarsest resolution block is generated. For 320 × 320 input images, we use six resolutions of features for predicting objects. The finest resolution is 40 × 40 and the coarsest resolution is 2 × 2. To start the iteration, we simply choose two adjacent scales of the current resolution as the inputs of down-sampling and up-sampling pathways. We also apply an extra 160 × 160 resolution as the input of the down-sampling pathway for the finest resolution (40 × 40) to improve the ability of detecting small objects.

3.3. Gate-Controlled Adaptive Recalibration

Motivation. Our goal is to ensure that the object detection network is able to adaptively select the meaningful scales for objects with different sizes, so that it can enhance the useful features in an appropriate resolution and suppress less useful ones. We propose to achieve this by adding a two-level attention mechanism and an identity mapping before each prediction layer, partially inspired by SENets [12] that won the first place in ILSVRC 2017 classification challenge. A diagram of our gate structure is shown in Fig. 4 and will be described in more detail in the following sections.

Gate Definition. A gate is a series of transformation \( F_{gate} \) that transforms the input feature map \( U \) to outputs \( O (\widetilde{U} \rightarrow O) \). Let \( U = [u_1, u_2, \ldots, u_e] \) denote the set of filter maps. Suppose \( U, O, \widetilde{U} \) and \( \widetilde{V} \in \mathbb{R}^{w \times h \times c} \) are all with width \( w \), height \( h \) and \( c \) channels, where \( \widetilde{U} \) and \( \widetilde{V} \) are intermediate states. Denote \( F_{scale} \) (\( F_c \) as abbreviation, similarly hereinafter): \( U \rightarrow \widetilde{U}; F_{g, scale} (F_g): \widetilde{U} \rightarrow \widetilde{V} \) and \( F_{residual} (F_r): \widetilde{V} \rightarrow O \) and therefore a gate can be formulated as:

\[
O = F_{gate}(U) = F_r(F_g(F_c(U)))
\]  

(1)

Channel-level and Global-level Attention. The aim of channel-level attention is to model relationships between channels and the global-level attention is to adaptively enhance or attenuate different scale supervision. We apply Squeeze-and-Excitation block [12] as our channel-level attention which consists of: (i) a squeeze stage \( F_{sq} \) for global information embedding; and (ii) an excitation stage \( F_{ex} \) for channel-level recalibration. Therefore we can formulate the channel-level outputs as:

\[
\widetilde{U} = F_{ex}(F_{sq}(U))
\]  

(2)

The squeeze stage can be formulated as a global pooling operation on each channel:

\[
s_c = F_{sq}(u_c) = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} u_c(i, j)
\]  

(3)

where \( s_c \) is the \( c \)-th element of \( s \). \( s \in \mathbb{R}^c \) is a vector calculated by global-pooling filter \( u \). The excitation stage is two fully-connected layers plus a sigmoid activation:

\[
e = F_{ex}(s) = \sigma(f_c(f_{\pi}(s)))
\]  

(4)

where \( e \in \mathbb{R}^c \) is the output, \( \sigma \) is the sigmoid function. \( f_c \) and \( f_{\pi} \) are the two fully-connected layers with output dimensions of \( c \) and \( \frac{c}{16} \), respectively. Then, we can calculate \( \tilde{U} \) by:

\[
\tilde{U} = F_c(U) = e \otimes U
\]  

(5)
Before Gates
After Gates
Before Gates
After Gates
Before Gates
After Gates
Before Gates
After Gates

Figure 6: Visualization of feature maps before and after gates. In each block, left is the input image with detection results. The right-top are the feature maps before gates and the right-bottom are the maps after gates.

where \( \otimes \) denotes channel-wise multiplication. More details can be referred to the SENets [12] paper.

Our global attention takes \( s \) (the output of squeeze stage) as input, and we modify the excitation stage by generating only one element. The new excitation stage \( \bar{F}_{ex} \) (for global attention) can be formulated as:

\[
\bar{e} = \bar{F}_{ex}(s) = \sigma(f_1(f_{\pi}(s)))
\]

where \( \bar{e} \in \mathbb{R}^1 \) is the global attention. The weight of \( f_{\pi} \) is shared between \( F_{ex} \) and \( \bar{F}_{ex} \). Finally, \( \bar{V} \) is calculated by:

\[
\bar{V} = F_{\gamma}(\bar{U}) = \bar{e} \otimes \bar{U}
\]

Identity Mapping. We use an element-wise addition operation [11] to obtain the final outputs:

\[
O = U \oplus \bar{V}
\]

where \( \oplus \) denotes element-wise addition.

Fig. 6 shows several examples of feature map visualization before and after the gating operation.

3.4. Gated Recurrent Feature Pyramid for DSOD

Our proposed method is a generic solution for building recurrent feature pyramids and gates inside deep convolutional neural networks, and also very easy to apply to existing frameworks, such as SSD [19], DSOD [24], FPN [16], etc. In the following, we adopt our method in DSOD [24] for general object detection, in order to demonstrate the effectiveness and advantages of our method.

DSOD is a proposal-free object detector that is trained from scratch. There are two steps to adapt Gated Recurrent Feature Pyramid for DSOD. First, we apply recurrent feature pyramid to replace the dense connection in DSOD prediction layers. Then, we add gates in each prediction layer to obtain GRP-DSOD. Other principles in DSOD are inherited in GRP-DSOD like Stem, Dense Block, etc. The full GRP-DSOD network architecture is shown here\(^1\). Implementation details and empirical results are given in the next section.

| Method                        | mAP  |
|-------------------------------|------|
| DSOD300 [24]                  | 77.7 |
| GRP-DSOD300                   | 78.5 |
| GRP-DSOD320                   | 78.7 |
| GRP-DSOD320* (using RFP only) | 79.0 |
| DSOD320* (using gates only)   | 78.6 |
| DSOD320* (using gates only)   | 78.5 |

Table 1: Ablation Experiments on PASCAL VOC 2007. “RFP” denotes our recurrent feature pyramid. * denotes we add one more aspect ratio 1.6 for default boxes at every prediction layer.

Identity Mapping. We use an element-wise addition operation [11] to obtain the final outputs:

\[
O = U \oplus \bar{V}
\]

where \( \oplus \) denotes element-wise addition.

3.4. Gated Recurrent Feature Pyramid for DSOD

Our proposed method is a generic solution for building recurrent feature pyramids and gates inside deep convolutional neural networks, and also very easy to apply to existing frameworks, such as SSD [19], DSOD [24], FPN [16], etc. In the following, we adopt our method in DSOD [24] for general object detection, in order to demonstrate the effectiveness and advantages of our method.

DSOD is a proposal-free object detector that is trained from scratch. There are two steps to adapt Gated Recurrent Feature Pyramid for DSOD. First, we apply recurrent feature pyramid to replace the dense connection in DSOD prediction layers. Then, we add gates in each prediction layer to obtain GRP-DSOD. Other principles in DSOD are inherited in GRP-DSOD like Stem, Dense Block, etc. The full GRP-DSOD network architecture is shown here\(^1\). Implementation details and empirical results are given in the next section.

4. Experiments

We conduct experiments on three widely used benchmarks: 20 category PASCAL VOC 2007, 2012 [5] and 80 category MS COCO detection datasets [18]. Following the previous practice of learning detection from scratch [19, 24], we train using the union of VOC 2007 trainval and VOC 2012 trainval (“07+12”) and test on VOC 2007 test set. For VOC 2012, we use VOC 2012 trainval and VOC 2007 trainval + test for training, and test

\[^1\]The visualization of the complete network structure is available at: http://ethereon.github.io/netscope/#/gist/4dcd890cb38a7f3c55a51dc1d67068a9.
Table 2: PASCAL VOC 2007 test detection results. SSD300S\textsuperscript{†} indicates training SSD300* from scratch with ResNet-101 or VGGNet. Note that the speed of Faster R-CNN with ResNet-101 (2.4 fps) is tested on K40, while others are tested on Titan X. For GRP-DSOD320\textsuperscript{*}, we did not include the # parameters of extra default boxes and the # parameters are 14.2M. If include, the # parameters are 16M. Table adapted from [24].

Table 3: PASCAL VOC 2012 Competition comp3 Leaderboard. GRP-DSOD320\textsuperscript{†} is trained on VOC 07+12 set and GRP-DSOD320\textsuperscript{*} is trained on VOC 12 trainval set. Note that both of the two results use single model for prediction without any experimental tricks. Result links are GRP-DSOD320\textsuperscript{†} (07+12): http://host.robots.ox.ac.uk:8080/anonymous/KJSBBP.html; GRP-DSOD320\textsuperscript{*} (12): http://host.robots.ox.ac.uk:8080/anonymous/KJSBBP.html.

In our study, all experiments are trained from scratch without the ImageNet [4] pre-trained models. Our code is implemented on Caffe platform [15]. We adopt the backbone network proposed by DSOD [24] to ensure a fair comparison.

Implementation details. We adopt SGD for training our models on 8 GPUs. Following [24, 19], we use a weight decay of 0.0005 and a momentum of 0.9. All conv-layers are initialized with the “xavier” method [9].

For GRP-DSOD320, we use aspect ratios of the default boxes as: [2], [2, 3, 3], [2, 3, 3], [2], [2] for six scales of predictions, respectively. Following [6], we propose GRP-DSOD320\textsuperscript{*} with one extra aspect ratio: [1.6, 2, 3], [1.6, 2, 3], [1.6, 2, 3], [1.6, 2], [1.6, 2]. We follow the same implementation setting as in the origin DSOD paper [24].

4.1 Ablation Experiments on PASCAL VOC 2007

In this section we investigate the effectiveness of each component of our GRP-DSOD framework. We design several controlled experiments on PASCAL VOC 2007 for the ablation study, including: (i) recurrent feature pyramid; (ii) gates; and (iii) input image size. In these experiments, we train on the union of VOC 2007 trainval and 2012 trainval ("07+12"), test on the VOC 2007 test set.

Effectiveness of Recurrent Feature Pyramid. Table 1 (row 5) shows the results of our recurrent feature pyramid without the gates. The result (78.6%) is on par with GRP-DSOD320 (row 3). It indicates that our recurrent feature pyramid contributes a lot on boosting the final detection performance.
Effectiveness of Gates. Table 1 (row 6) shows the results of adding gates without the recurrent feature pyramid. The result (78.5%) is comparable to the result using recurrent feature pyramid only.

Effectiveness of large input size. As shown in Table 1 (rows 2 and 3), we observe that larger input image size (from 300 to 320) can help to improve the result. Since we only enlarge the input size marginally, the performance is slightly better (0.2% mAP improvement).

4.2. Results on PASCAL VOC 2007

We use a batch size of 128 to train our models on VOC 2007. The initial learning rate is set to 0.1, and then divided by 10 after every 20k iterations. The training finished after 200k iterations. The initial learning rate is set to 0.1, and then divided by 10 after every 20k iterations. The training finished after 200
time.

Table 2 shows our results on VOC2007 test set. Our GRP-DSOD300 achieves 78.5%, which is better than baseline method DSOD300 (77.7%). Enlarging the input image to 320 × 320 improves the result to 78.7%. Following [6], we add one more aspect ratio 1.6 at every prediction layer to SSD321 and DSSD321 with ResNet-101 backbone network. Our method is much faster than these two competitors.

Model Parameters Comparison. The number of parameters is shown in the 7-th column of Table 2. Thanks to the parameter-efficient RFP block, our model is much smaller than most competitors. For instance, our model is only 1/4 of DSOD321 and DSSD321 with ResNet-101 backbone network. Our method is much faster than these two competitors.

4.3. Results on PASCAL VOC 2012

We perform our experiments on VOC 2012 dataset and compare with two different subsets of related experimental results: one with models only trained from scratch with VOC provided data, and the other with models that are pre-trained on the larger ImageNet dataset. All other settings are the same as those used in our VOC 2007 experiments. As shown in Table 3, on PASCAL VOC 2012 Competition comp3 Leaderboard where all results are from models trained from scratch, our results outperform the second place with a large margin (13% mAP) when trained on VOC 07++12 and 8.5% mAP with VOC 12.
We then compare our results with a number of start-of-the-art object detection approaches that are proposed recently and pre-trained on the ImageNet. As shown in Table 4, our GRP-DSOD achieves 77.0% mAP, which is consistently better than baseline DSOD (76.3%) and other state-of-the-art methods using VGGNet or ResNet-101 pre-trained models like SSD321 (75.4%) and DSSD321 (76.3%).

4.4. Results on MS COCO

Finally we evaluate our GRP-DSOD on the MS COCO dataset [18]. The batch size is set to 128. The initial learning rate is set to 0.1 for the first 80k iterations, then divided by 10 after every 60k iterations. The total number of training iterations is 320k.

Results are summarized in Table 5. We observe that GRP-DSOD (without the use of aspect ratio 1.6) is already better than SSD321 and DSSD321 that use ResNet-101 pre-trained model from ImageNet with 2% improvement on the test-dev set (results under [0.5:0.95] IoU). Comparing with the baseline method DSOD, GRP-DSOD20 also achieves higher performance (30.0%/47.9% vs. 29.3%/47.3%) with similar input image size (320 vs. 300). Interestingly, we see that our results on Large size are slightly worse or comparable to DSOD (Avg. Precision Area: 46.3% vs. 47.0%; Avg. Recall Area: 65.0% vs. 65.0%), while results on Small and Medium sizes are much better than DSOD, SSD and DSSD. For instance, on Avg. Precision Area, our result on small size outperforms SSD321 by a very large margin (10.9% vs. 6.2%). This indicates that our proposed method is more effective to detect small objects even with very small input image size.
5. Conclusion

We have presented gated recurrent feature pyramid, a novel structure deign that is more fit for learning detection from scratch, one of the first successful attempts on recalibrating supervision signals for object detection, and improves performance in both convergence speed and accuracy. Extensive experiments on PASCAL VOC 2007, 2012 and MS COCO demonstrate the effectiveness of our proposed method on both detection performance and efficient model parameters. This work suggests that despite the high performance of training detectors from scratch, there is still room and potential to improve the performance to be on par with or even surpass the pre-trained models.

Acknowledgements

This work is supported by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network. Xiangyang Xue was supported in part by National Key R&D Program of China under Grant (No.2017YFC0803700), NSFC under Grants (No.61572138 & No.U1611461) and STCSM under Grant (No.16JC1420400).

References

[1] S. Bell, C. Lawrence Zitnick, et al. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In CVPR, 2016. 1, 7
[2] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In ECCV, 2016. 1
[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In NIPS, 2016. 2, 6, 7
[4] J. Deng, W. Dong, R. Socher, L.-J. Li, et al. ImageNet: A large-scale hierarchical image database. In CVPR, 2009. 2, 3, 6
[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010. 2, 5
[6] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 1, 2, 6, 7
[7] R. Girshick. Fast r-cnn. In ICCV, 2015. 2
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 2
[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010. 6
[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In ICCV, 2017. 2
[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 5, 7
[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507, 2017. 2, 4, 5
[13] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In CVPR, 2017. 3
[14] J. Huang, V. Rathod, C. Sun, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In CVPR, 2017. 7
[15] Y. Jia, E. Shelhamer, J. Donahue, et al. Caffe: Convolutional architecture for fast feature embedding. In ACM MM, 2014. 6
[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017. 1, 2, 5
[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In ICCV, 2017. 1
[18] T.-Y. Lin, M. Maire, S. Belongie, et al. Microsoft coco: Common objects in context. In ECCV, 2014. 2, 5, 8
[19] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multi-box detector. In ECCV, 2016. 1, 2, 3, 5, 6, 7
[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016. 2, 3
[21] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In CVPR, 2017. 2, 6, 7
[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015. 2, 6, 7
[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013. 2
[24] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue. Dsod: Learning deeply supervised object detectors from scratch. In ICCV, 2017. 1, 2, 3, 5, 6, 7
[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 3
[26] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387, 2015. 2
[27] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, A. Zisserman. The pascal visual object classes (voc) challenge. In CVPR, 2009. 2, 3
[28] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang. Gated recurrent feature pyramid networks. In ICLR, 2017. 2, 4, 5