Contextual Priming and Feedback for Faster R-CNN

Abhinav Shrivastava and Abhinav Gupta

Carnegie Mellon University

Abstract. The field of object detection has seen dramatic performance improvements in the last few years. Most of these gains are attributed to bottom-up, feedforward ConvNet frameworks. However, in case of humans, top-down information, context and feedback play an important role in doing object detection. This paper investigates how we can incorporate top-down information and feedback in the state-of-the-art Faster R-CNN framework. Specifically, we propose to: (a) augment Faster R-CNN with a semantic segmentation network; (b) use segmentation for top-down contextual priming; (c) use segmentation to provide top-down iterative feedback using two stage training. Our results indicate that all three contributions improve the performance on object detection, semantic segmentation and region proposal generation.

1 Introduction

The field of object detection has changed drastically over the past few years. We have moved from manually designed features [1, 2] to learned ConvNet features [3–6]; from the original sliding window approaches [2, 7] to region proposals [4, 8–11]; and from pipeline based frameworks such as Region-based CNN (R-CNN) [4] to more end-to-end learning frameworks such as Fast [10] and Faster R-CNN [11]. The performance has continued to soar higher, and things have never looked better. There seems to be a growing consensus – powerful representations learned by ConvNets are well suited for this task, and designing and learning deeper networks lead to better performance.

Most recent gains in the field have come from bottom-up, feedforward framework of ConvNets. On the other hand, in the case of human visual system, the number of feedback connections significantly outnumber the feedforward connections. In fact, many behavioral studies have shown the importance of context and top-down information for the task of object detection. This raises a few important questions – Are we on the right path as we try to develop deeper and deeper, but only feedforward networks? Is there a way we can bridge the gap between empirical results and theory, when it comes to incorporating top-down information, feedback and/or contextual reasoning in object detection?

This paper investigates how we can break the feedforward mold in current detection pipelines and incorporate context, feedback and top-down information. Current detection frameworks have two components: the first component generates region proposals and the second classifies them as an object category or background. These region proposals seem to be beneficial because (a) they reduce the search space; and (b) they reduce false positives by focusing the 'attention' in right areas. In fact, this is in line with the psychological experiments that support the idea of priming (although note that while region proposals mostly use bottom-up segmentation [8, 12], top-down context provides the priming in humans [13–15]). So, as a first attempt, we propose to use top-down information in generating region proposals. Specifically, we add segmentation as a complementary task and use it to provide top-down information to guide region proposal generation and object detection. The intuition is that semantic segmentation captures contextual relationships between objects (e.g., support, likelihood, size etc. [16]), and will essentially guide the region proposal module to focus attention in the right areas and learn detectors from them.

But contextual priming using top-down attention mechanism is only part of the story. In case of humans, the top-down information provides feedback to the whole visual pathway (as early as V1 [17, 18]). Therefore, we further explore providing top-down feedback to the entire network in order to modulate feature extraction in all layers. This is accomplished by providing the semantic segmentation output as input to different parts of the network and training another stage of our model. The hypothesis is that equipping the network with this top-down semantic feedback would guide the visual attention of feature extractors to the regions relevant for the task at hand.

To summarize, we propose to revisit the architecture of a current state-of-the-art detector (Faster R-CNN [11]) to incorporate top-down information, feedback and contextual information. Our new architecture includes:

- **Semantic Segmentation Network:** We augment Faster R-CNN with a semantic segmentation network. We believe this segmentation can be used to provide top-down feedback to Faster R-CNN (as discussed below).
- **Contextual Priming via Semantic Segmentation:** In Faster R-CNN, both region proposal and object detection modules are feedforward. We propose to use semantic segmentation to provide top-down feedback to these modules. This is analogous to contextual priming; in this case top-down semantic feedback helps propose better regions and learn better detectors.
- **Iterative Top-Down Feedback:** We also propose to use semantic segmentation to provide top-down feedback to low-level filters, so that they become better suited for the detection problem. In particular, we use segmentation as an additional input to lower layers of a second round of Faster R-CNN.

2 Related Work

Object detection was once dominated by the sliding window search paradigm [2, 7]. Soon after the resurgence of ConvNets for image classification [3, 19, 20], there were attempts at using this sliding window machinery with ConvNets [21–23]; but a key limitation was the computational complexity of brute-force search.

As a consequence, there was major paradigm shift in detection which completely bypassed the exhaustive search in favor of region-based methods and object proposals [8, 12, 24–29]. By reducing the search space, it allowed us to use sophisticated (both manually designed [9, 30, 31] and learned ConvNet [4, 11, 32–36]) features. Moreover, this also helped focus the attention of detectors to regions well supported by perceptual structures in the image. However, recently, Faster R-CNN [11] showed that even these region proposals can be generated by using ConvNet features. It removed segmentation from proposal pipeline by training a small network on top of ConvNet features that proposes a few object candidates. This raises an important question: Do ConvNet features already capture the structure that was earlier given by segmentation or does segmentation provide complementary information?

To answer this, we study the impact of using semantic segmentation in the region proposal and object detection modules of Faster R-CNN [11]. In fact, there has been a lot of interest in using segmentation in tandem with detection [30, 31, 37, 38]; e.g., Fidler et al. [30] proposed to use segmentation proposals as additional features for DPM detection hypothesis. In contrast, we propose to use semantic segmentation to guide/prime the region proposal generation itself. There is ample evidence of the importance of similar top-down contextual priming in the human visual system [15, 39], and its utility in reducing areas to focus our attention on for recognizing objects [13, 14].

This prevalence and success of region proposals is only part of the story. Another key ingredient is the powerful ConvNet features [3, 5, 6]. ConvNets are multi-layered hierarchical feature extractors, inspired by visual pathways in humans [18, 40]. But so far, our focus has been on designing deeper [5, 6] feedforward architectures, even when there is a broad agreement on the importance of feedback connections [17, 41, 42] and limitations of purely feedforward recognition [43, 44] in human visual systems. Inspired by this, we investigate how can we start incorporating top-down feedback in our current object detection architectures. There have been attempts earlier at exploiting feedback mechanisms; some well known examples are auto-context [45] and inference machines [46]. These iteratively use predictions from a previous iteration to provide contextual features to the next round of processing; however they do not trivially extend to ConvNet architectures. Closest to our goal are the contemporary works on using feedback to learn selective attention [47, 48] and using top-down iterative feedback to improve at a task at hand [49–51]. In this work, we additionally explore using top-down feedback from one task to another.

The discussion on using global top-down feedback to contextually prime object recognition is incomplete without relating it to ‘context’ in general, which has a long history in cognitive neuroscience [13–16, 52–55] and computer vision [56–63]. It is widely accepted that human visual inference of objects is heavily influenced by ‘context’, be it contextual relationships [16, 52], priming for focusing attention [13–15] or importance of scene context [39, 53–55]. These ideas have inspired lot of computer vision research (see [56, 57] for survey). However, these approaches seldom lead to strong empirical gains. Moreover, they are mostly confined to weaker visual features (e.g., [1]) and have not been explored much in ConvNet-based object detectors.

For region-based ConvNet object detectors, simple contextual features are slowly becoming popular; e.g., computing local context features by expanding the region [64–67], using other objects (e.g., people) as context [68] and using other regions [69]. In comparison, the use of context has been much more popular for semantic segmentation. E.g., CRFs are commonly used to incorporate context and post-process segmentation outputs [70–72] or to jointly reason about regions, segmentation and detection [66, 73]. More recently, RNNs have also been employed to either integrate intuitions from CRFs [72, 74, 75] in end-to-end learning systems or to capture context outside the region [36]. But empirically, at least for detection, such uses of context have mostly given feeble gains.

3 Preliminaries: Faster R-CNN

We first describe the two core modules of the Faster R-CNN [11] framework (Figure 1). The first module takes an image as input and proposes rectangular regions of interest (RoIs). The second module is the Fast R-CNN [10] (FRCN) detector that classifies these proposed regions. In this paper, both modules use the VGG16 [5] network, which has 13 convolutional (`conv`) and 2 fully connected (`fc`) layers. Both modules share all `conv` layers and branch out at `conv5_3`. Given an arbitrary sized image, the last `conv` feature map (`conv5_3`) is used as input to both the modules as described below.

Region Proposal Network (RPN). The region proposal module (Figure 1(left) in green) is a small fully convolutional network that operates on the last feature map and outputs a set of rectangular object proposals, each with a score. RPN is composed of a `conv` layer and 2 sibling `fc` layers. The `conv` layer operates on the input feature map to produce a `D`-dim. output at every spatial location; which is then fed to two `fc` layers – classification (`cls`) and box-regression (`breg`). At each spatial location, RPN considers k candidate boxes (anchors) and learns to classify them as either foreground or background based on their IOU overlap with the ground-truth boxes. For foreground boxes, `breg` layer learns to regress to the closest ground-truth box. A typical setting is $\mathtt{D} = 512$ and $k = 9$ (3 scales, 3 aspect-ratios) (see [11] for details).

Using RPN regions in FRCN. For training the Fast R-CNN (FRCN) module, a mini-batch is constructed using the regions from RPN. Each region in the mini-batch is projected onto the last `conv` feature map and a fixed-length feature vector is extracted using RoI-pooling [10, 32]. Each feature is then fed to two `fc` layers, which finally give two outputs: (1) a probability distribution over object classes and background; and (2) regressed coordinates for box re-localization. An illustration is shown in Figure 1(left) in blue.

Training Faster R-CNN. Both RPN and FRCN modules of Faster R-CNN are trained by minimizing the multi-task loss (for classification and box-regression) from [10, 11] using mini-batch SGD. To construct a mini-batch for RPN, 256 anchors are randomly sampled with 1 : 1 foreground to background ratio; and for

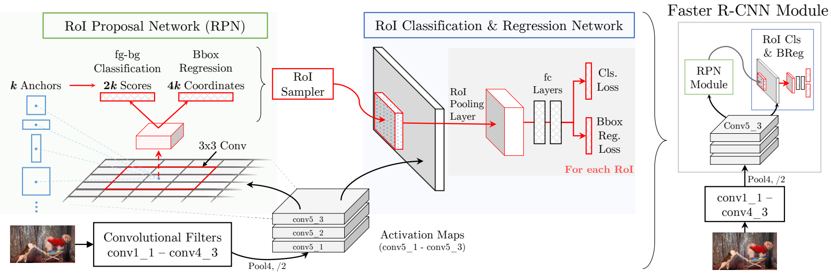


Fig. 1: Faster R-CNN. (left) Overview of Region Proposal Network (RPN) and RoI classification and box regression. (right) Shorthand diagram of Faster R-CNN.

FRCN, 128 proposals are sampled with 1 : 3 ratio. We train both modules jointly using the 'approximate joint training'. For more details, refer to [4, 10, 11, 76].

Given an image during training, a forward pass through all the `conv` layers produces `conv5_3` feature map. RPN operates on this feature to propose two sets of regions, one each for training RPN and FRCN. Independent forward-backward passes are computed for RPN and FRCN using their region sets, gradients are accumulated at `conv5_3` and back-propagated through the `conv` layers.

Why Faster R-CNN? Apart from being the current state-of-the-art object detector, Faster R-CNN is also the first framework that *learns* where to guide the 'attention' of an object detector along with the detector itself. This end-to-end learning of proposal generation and object detection provides a principled testbed for studying the proposed top-down contextual feedback mechanisms.

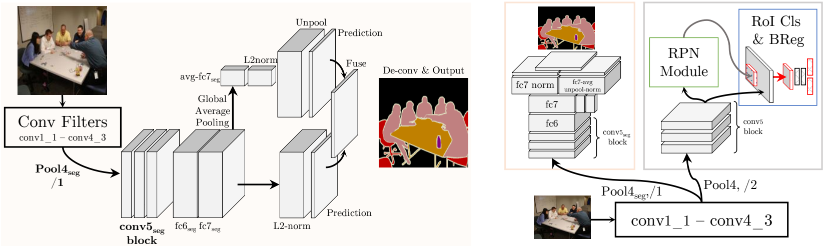
In the following sections, we first describe how we add a segmentation module to Faster R-CNN (Section 4.1) and then present how we use segmentation for top-down contextual priming (Section 4.2) and iterative feedback (Section 4.3).

4 Our Approach

We propose to use semantic segmentation as a top-down feedback signal to the RPN and FRCN modules in Faster R-CNN, and iteratively to the entire network. We argue that a raw semantic segmentation output is a compact signal that captures the desired contextual information such as relationships between objects (Section 2) along with global structures in the image, and hence is a good representation for top-down feedback.

4.1 Augmenting Faster R-CNN with Segmentation

The first step is to augment Faster R-CNN framework with an additional segmentation module. This module should ideally: 1) be fast, so that we do not give up the speed advantages of [10, 11]; 2) closely follow the network used by Faster R-CNN (VGG16 in this paper), for easy integration; and 3) use minimal



(a) ParseNet Segmentation Framework

(b) Faster R-CNN with Segmentation

Fig. 2: (a) Overview of ParseNet. (b) Shorthand diagram of our multi-task setup (Faster R-CNN + Segmentation). Refer to Section 4.1 and 5.2 for details.

(preferably no) post-processing, so that we can train it jointly with Faster R-CNN. Out of several possible architectures [33, 70, 72, 77, 78], we choose the ParseNet architecture [77] because of the simplicity.

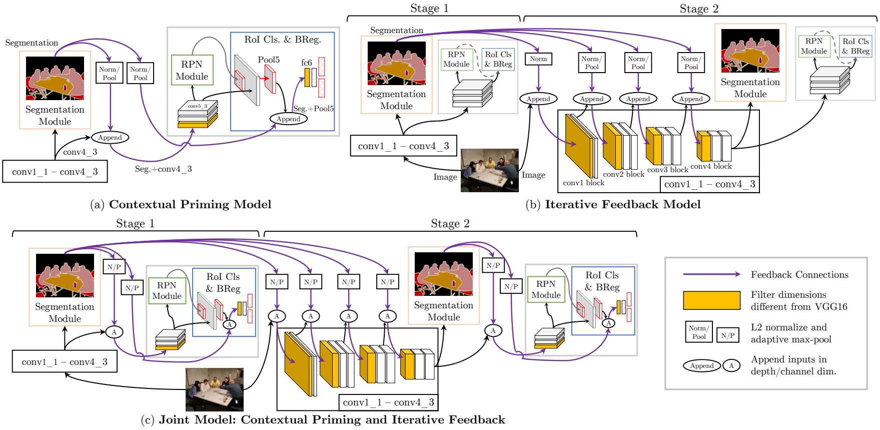
ParseNet [77] is a fully convolutional network [33] for segmentation. It is fast because it uses filter rarefication technique (a-trous algorithm) from [70]. Its architecture is similar to VGG16. Moreover, it uses no post-processing; and instead adds an average pooling layer to incorporate global context; which is shown to have similar benefits to using CRFs [70, 74].

Architecture details. An overview is shown in Figure 2(a). The key difference from standard VGG16 is that the pooling after `conv4_3` (`pool4`$_{\text{seg}}$) does no down-sampling, as opposed to the standard `pool4` which down-samples by a factor of 2. After the `conv5` block, it has two 1×1 `conv` layers with 1024 channels applied with a filter stride [70, 77]. Finally, it has a global average pooling step which given the feature map of after any layer (H×W×D) computes its spatial average (1×1×D) and 'unpools' the features. Both source and its average feature maps are normalized and used to predict per-pixel labels. These outputs are then fused and a 8× `deconv` layer is used to produce the final output.

Faster R-CNN with Segmentation – A Multi-task setup

In the joint network (Figure 2(b)), both the Faster R-CNN modules and the segmentation module share the first 10 `conv` layers (`conv1_1 - conv4_3`) and differ `pool4` onwards. For the segmentation module, we branch out `pool4`$_{\text{seg}}$ layer with stride of 1 and add the remaining ParseNet layers (`conv5_1` to `deconv`)(Figure 2). The final architecture is a multi-task setup [79], which produces both semantic segmentation and object detection outputs simultaneously.

Training details. Now that we have a joint architecture, we can train segmentation, RPN and detection modules by minimizing a multi-task loss. However, there are some key issues: 1) Faster R-CNN can operate on an arbitrary sized input image, whereas ParseNet requires a fixed 500×500 image. In this joint framework, our segmentation module is adapted to handle arbitrary sized images; 2) Faster R-CNN and ParseNet are trained using very different set of hyperparameters (e.g., learning rate schedule, batch-size etc.); and neither set of



(a) **Contextual Priming Model**

(b) **Iterative Feedback Model**

(c) **Joint Model: Contextual Priming and Iterative Feedback**

Fig. 3: Overview of the proposed models for top-down feedback. (a) **Contextual Priming via Segmentation** (Section 4.2) uses segmentation as top-down feedback signal to guide the RPN and FRCN modules of Faster R-CNN. (b) **Iterative Feedback** (Section 4.3) is a 2-unit model, where the Stage-1 provides top-down feedback for Stage-2 filters. (c) **Joint Model** (Section 4.4) uses (a) as the base unit in (b).

parameters is optimal for the other. So for joint training, we modify the hyper-parameters of segmentation module and shared layers. Details on these design decisions and analysis of their impact will be presented in Section 5.2.

This Faster R-CNN + Segmentation framework serves as the base model on top of which we add top-down contextual feedback. We will also use this multi-task model as our primary baseline (Base-MT) as it is trained using both segmentation and detection labels but does not have contextual feedback.

4.2 Contextual Priming via Segmentation

We propose to use semantic segmentation as top-down feedback to the region proposal and object detection modules of our base model. We argue that segmentation captures contextual information which will 'prime' the region proposal and object detection modules to propose better regions and learn better detectors.

In our base multi-task model, the Faster R-CNN modules operate on the `conv` feature map from the shared network. To contextually prime these modules, their input is modified to be a combination of aforementioned `conv` features and the segmentation output. Both modules can now learn to guide their operations based on the semantic segmentation of an image – it can learn to ignore background regions, find smaller objects or find large occluded objects (e.g., tables) etc. Specifically, we take the raw segmentation output and append it to the `conv4_3` feature. The `conv5` block of filters operate on this new input ('`seg+conv4_3`') and their output is input to the individual Faster R-CNN modules. Hence, a top-down feedback signal from segmentation 'primes' both Faster R-CNN modules. However, because of the RoI-pooling operation, the detection module only sees the segmentation signal local to a particular region. To provide a global context to each region, we also append segmentation to the fixed-length feature vector ('seg+pool5') before feeding it to fc6. Overview in Figure 3(a).

This entire system (three modules with connections between them) is trained jointly. After a forward pass through the shared conv layers and the segmentation module, their outputs are used as input to both Faster R-CNN modules. A forward-backward pass is performed for both RPN and FRCN. Next, the segmentation module does a backward pass using the gradients from its loss and from the other modules. Finally, gradients are accumulated at conv4_3 from all three modules and backward pass is performed for the shared conv layers.

Architecture details. Given an $(\mathtt{H_I} \times \mathtt{W_I} \times 3)$ input, the conv4_3 produces a $(\mathtt{H_c} \times \mathtt{W_c} \times 512)$ feature map, where $(\mathtt{H_c}, \mathtt{W_c}) \approx (\mathtt{H_I}/8, \mathtt{W_I}/8)$. Using this feature map, the segmentation module produces a $(\mathtt{H_I} \times \mathtt{W_I} \times (\mathtt{K}+1))$ output, which is a pixel-wise probability distribution over $\mathtt{K}+1$ classes. We ignore the background class and only use $(\mathtt{H_I} \times \mathtt{W_I} \times \mathtt{K})$ output, which we refer to as $\mathtt{S}$. Now, $\mathtt{S}$ needs to be combined with conv4_3 feature for the Faster R-CNN modules and each region's $(7 \times 7 \times \mathtt{K})$-dim. pool5 feature map for FRCN, but there are 2 issues: 1) spatial dimensions of $\mathtt{S}$ does not match either, and 2) feature values from different layers are at drastically different scales [77]. To deal with the spatial dimension mis-match, we utilize the RoI/spatial-pooling layer from [10, 32]: We maxpool $\mathtt{S}$ using an adaptive grid to produce two outputs $\mathtt{S_c}$ and $\mathtt{S_p}$, which have the same spatial dimensions as conv4_3 and pool5 respectively. Now, we normalize and scale $\mathtt{S_c}$ to $\mathtt{S_{cN}}$ and $\mathtt{S_p}$ to $\mathtt{S_{pN}}$, such that their L2-norm [77] is of the same scale as the per-channel L2-norm of their corresponding features (conv4_3 and pool5 respectively). Now, we append $\mathtt{S_{cN}}$ to conv4_3 and the resulting $(\mathtt{H_c} \times \mathtt{W_c} \times (512+\mathtt{K}))$ feature is the input for Faster R-CNN. Finally, we append $\mathtt{S_{pN}}$ with each region's pool5 and the resulting $(7 \times 7 \times (512+\mathtt{K}))$ feature is the input for fc6 of FRCN. This network architecture is trained from a VGG16 initialized base model; and the additional $\mathtt{K}$ channels in conv5_3 and fc6 are initialized randomly using [5, 80]. Refer to Figure 3(a) for an overview.

4.3 Iterative Feedback via Segmentation

The architecture proposed in the previous section provides top-down semantic feedback and modulates only the Faster R-CNN module. We also propose to provide top-down information to the whole network, especially the shared conv layers, to modulate low-level filters. The hypothesis is that this feedback will help the earlier conv layers to focus on areas likely to have objects. We again build from the Base-MT model (Section 4.1).

This top-down feedback is iterative in nature and will pass from one instantiation of our base model to another. To provide this top-down feedback, we take the raw segmentation output of our base model (Stage-1) and append it to the input of the conv layer to be modulated in the second model instance (Stage-2) (see Figure 3(b)). E.g., to modulate the first conv layer of Stage-2, we append the Stage-1 segmentation signal to the input image, and use this combination as the new input to conv1_1. This feedback mechanism is trained stage-wise:

the Stage-1 model (Base-MT) is trained first; and then it is frozen and only the Stage-2 model is trained. This iterative feedback is similar to [49, 50]; the key difference being that they only focus on iteratively improving the same task, whereas in this work, we also use feedback from one task to improve another.
Architecture details. Given the pixel-wise probability output of the Stage-1 segmentation module, the background class is ignored and the remaining output ($\mathtt{S}$) is used as the semantic feedback signal. Again, $\mathtt{S}$ needs to be resized, rescaled and/or normalized to match the spatial dimensions and the feature values scale of the input to various `conv` layers. To append with the input image, $\mathtt{S}$ is re-scaled and centered element-wise to lie in $[-127, 128]$. This results in a new $(\mathtt{H_I} \times \mathtt{W_I} \times (3 + \mathtt{K}))$ input for `conv1_1`. To modulate `conv2_1`, `conv3_1` and `conv4_1`, we `maxpool` and L2-normalize $\mathtt{S}$ to match the spatial dimensions and the feature value scales of `pool1`, `pool2` and `pool3` features respectively (similar to Section 4.2). The filters corresponding to additional $\mathtt{K}$ channels in `conv1_1`, `conv2_1`, `conv3_1` and `conv4_1` are initialized using [80].

4.4 Joint Model

So far, given our multi-task base model, we have proposed a top-down feedback for contextual priming of region proposal and object detection modules and an iterative top-down feedback mechanism to the entire architecture. Next, we put these two pieces together in a single joint framework. Our final model is a 2-unit model: each individual unit being the contextual priming model (from Section 4.2), and both units being connected for iterative top-down feedback (Section 4.3). We train this 2-unit model stage-wise (Section 4.3). Architecture details of the joint model follow from Section 4.2 and 4.3 (see Figure 3(c)).

Through extensive evaluation, presented in the following sections, we show that: 1) individually, both contextual priming and iterative feedback models are effective and improve performance; and 2) the joint model is better than both individual models, indicating their complementary nature. We would like to highlight that our method is fairly general – both segmentation and detection modules can easily utilize newer network architectures (e.g., [6, 78]).

5 Experiments

We conduct experiments to better understand the impact of contextual priming and iterative feedback; and provide ablation analysis of various design decisions. Our implementation uses the Caffe [81] library.

5.1 Experimental setup

For ablation studies, we use the multi-task setup from Section 4.1 as our baseline (Base-MT). We also compare our method to Faster R-CNN [11] and ParseNet [77] frameworks. For quantitative evaluation, we use the standard mean average precision (mAP) [82] metric for object detection and mean intersection-over-union metric (mIOU) [10, 82] for segmentation.

Table 1: Ablation analysis of modifying ParseNet training methodology (Section 5.2).

Notes	Input dim.	Learning Rates (LR)			Batch-size	#iter	Normalize Loss?	mIOU (12S val)
		Base LR	Layer LR	LR Policy				
1) [77] (Original ParseNet)	500×500	10^{-8}	1	poly	8	20k	N	69.6
2) Reproducing [77]‡ (ParseNet)	500×500	10^{-8}	1	poly	8	20k	N	68.2
3) Faster R-CNN LR-policy, Norm. Loss	500×500	10^{-3}	1	step	8	20k	Y	68.5
4) Faster R-CNN batch-size, new LR	500×500	2.5×10^{-4}	1	step	2	80k	Y	67.8
5) Faster R-CNN Base-LR	500×500	10^{-3}	0.25	step	2	80k	Y	67.8
6) Faster R-CNN input dim. (ParseNet*)	[600×1000]†	10^{-3}	0.25	step	2	80k	Y	66

†min dim. is 600, max dim. capped at 1000. ‡https://github.com/weiliu89/caffe/tree/fcn

Datasets. All models in this section are trained on the PASCAL VOC12 [83] segmentation set (12S), augmented with the extra annotations (A) from [84] as is standard practice. Results are analyzed on VOC12 segmentation val set. For analysis, we chose the segmentation set, and not detection, because *all* images have *both* segmentation *and* bounding-box annotations; this helps us isolate the effects of using segmentation as top-down semantic feedback without worrying about missing segmentation labels in the standard detection split. Results on the standard splits will be presented in Section 6.

5.2 Base Model – Augmenting Faster R-CNN with Segmentation

Faster R-CNN and ParseNet both use mini-batch SGD for training, however, they follow different training methodologies. We first describe the implementation details and design decisions adopted to augment the segmentation module to Faster R-CNN and report baseline performances.

ParseNet Optimization. ParseNet is trained for 20k SGD iterations using an effective mini-batch of 8 images, an initial learning rate (LR) of 10^{-8} and polynomial LR decay policy. Compare this to Faster R-CNN, which is trained for 70k SGD iterations with a mini-batch size of 2, 10^{-3} initial LR and step LR decay policy (step at 50k). Since we are augmenting Faster R-CNN, we try to adapt ParseNet's optimization. On the 12S val set, [77] reports 69.6% (we achieved 68.2% using the released code, Table 1(1-2)). We will refer to the latter as ParseNet throughout. Similar to [33], ParseNet does not normalize the Softmax loss by number of valid pixels. But to train with Faster R-CNN in a multi-task setup, all losses need to have similar magnitude; so, we normalize the loss of ParseNet and modify the LR accordingly. Next, we change the LR decay policy from polynomial to step (step at 12.5k) to match that of Faster R-CNN. These changes result in similar performance (+0.3 points, Table 1(2-3)). We now reduce the batch size to 2 and adjust the LR appropriately (Table 1(4)). To keep the base LR of Faster R-CNN and ParseNet same, we change it to 10^{-3} and modify the LR associated with each ParseNet layer to 0.25, thus keeping the same effective LR for ParseNet (Table 1(4-5)).

Training data. ParseNet re-scales the input images and their segmentation labels to a fixed size (500×500), thus ignoring the aspect-ratio. On the other hand, Faster R-CNN maintains the aspect-ratio and re-scales the input images

Table 2: **Detection results on VOC 2012 segmentation val set**. All methods use VOC12S+A training set (Section 5.1). Legend: **S**: uses segmentation labels (Section 4.1), **P**: contextual priming (Section 4.2), **F**: iterative feedback (Section 4.3)

method	S	P	F	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Fast R-CNN [10]				71.6	88.2	79.7	83.6	62.8	42.3	84.0	69.4	87.5	41.5	73.7	57.4	84.7	77.7	85.8	75.8	35.3	73.1	67.7	85.0	76.3
Faster R-CNN [11]				75.3	92.3	80.9	86.7	65.4	49.3	87.1	78.2	89.7	42.7	79.8	61.4	87.4	82.8	89.4	82.2	46.1	78.2	64.6	86.8	75.6
Base-MT (sec. 4.1)	✓			75.6	93.0	82.5	88.1	70.2	47.2	86.5	76.5	89.3	47.7	78.3	56.4	88.0	80.2	88.9	80.7	43.6	81.5	67.9	89.4	75.2
Ours (priming, sec. 4.2)	✓	✓		77.0	91.1	82.3	85.3	70.8	47.5	90.3	75.2	90.9	46.0	82.3	65.6	88.0	83.3	91.2	81.0	49.6	81.0	69.8	92.1	76.0
Ours (feedback, sec. 4.3)	✓		✓	77.3	90.7	82.9	90.4	70.3	51.2	89.7	77.0	91.7	49.9	81.4	66.9	87.8	81.1	90.3	82.2	50.4	79.2	70.2	85.9	76.9
Ours (joint, sec. 4.4)	✓	✓	✓	**77.8**	89.8	83.8	84.0	72.1	54.2	92.0	75.5	91.2	53.6	82.1	69.8	85.7	81.7	92.4	82.5	49.9	76.2	72.5	89.3	78.4

Table 3: **Segmentation results on VOC 2012 segmentation val set**. All methods use VOC12S+A training set (Section 5.1). Legend: **S**: uses segmentation labels, **P**: contextual priming, **F**: iterative feedback

method	S	P	F	mIOU	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
ParseNet (Table 1(2))	✓			68.2	92.3	86.9	38.4	77.1	66.4	66.5	83.0	80.9	82.5	31.0	72.9	49.5	71.4	73.9	76.7	79.3	47.9	73.3	40.3	78.3	63.6
ParseNet* (Table 1(6))	✓			66.0	91.7	85.2	36.8	73.2	64.0	60.8	82.4	76.9	81.8	30.4	65.4	51.3	69.6	73.5	75.4	78.2	43.9	71.3	38.9	79.3	56.2
Base-MT (sec. 4.1)	✓			65.8	91.6	84.3	37.1	71.5	63.8	60.8	82.3	74.8	80.3	30.8	68.7	48.8	71.4	75.7	73.8	77.7	42.8	70.1	39.1	79.9	56.4
Ours (priming, sec. 4.2)	✓	✓		65.3	91.5	85.1	36.4	73.3	64.0	60.4	81.4	75.1	81.8	31.7	64.8	48.8	69.0	73.7	73.4	77.1	41.6	69.9	38.4	78.1	55.5
Ours (feedback, sec. 4.3)	✓		✓	69.5	92.8	87.3	39.4	76.9	66.7	68.1	86.9	80.6	86.4	33.4	68.1	50.9	71.8	80.1	77.3	81.3	48.6	73.3	42.0	82.8	65.5
Ours (joint, sec. 4.4)	✓	✓	✓	**69.6**	92.9	88.5	39.4	78.1	66.9	69.1	84.5	79.8	84.9	37.8	69.2	50.5	71.4	79.7	77.5	81.3	47.1	74.2	43.4	80.1	65.0

such that their shorter side is 600 pixels (and the max dim. is capped at 1000). We found that ignoring the aspect-ratio drops Faster R-CNN performance and maintaining it drops the performance of ParseNet (−1.8 points, Table 1(5-6)). Because our main task is detection, we opted to use Faster R-CNN strategy, and treat the new ParseNet (ParseNet*) as the baseline for our base model.

Base Model Optimization. Following the changes mentioned above, our base model uses these standardized parameters: batch size of 2, 10^{-3} base LR, step decay policy (step at 50k), LR of 0.25 for segmentation and shared `conv` layers, and 80k SGD iterations. This model serves as our multi-task baseline (Base-MT).

Baselines. For comparison, re-train Fast [10] and Faster R-CNN [11] on VOC 12S+A training set. Results of the Base-MT model for detection and segmentation are reported in Table 2 and 3 respectively. Performance increases by 0.3 mAP on detection and drops by 0.1 mIOU on segmentation. This will serve as our primary baseline.

5.3 Contextual Priming

We evaluate the effects of using segmentation as top-down semantic feedback to the region proposal generation and object detection modules. We follow the same optimization hyperparameters as the Base-MT model, and report the results in Table 2 and 3. Table 2 shows that providing top-down feedback via priming to the Faster R-CNN modules improves its detection performance by **1.4** points over the Base-MT model and **1.7** points over Faster R-CNN. Results in Table 3 show that performance of segmentation drops slightly when it is used for priming.

Design Evaluation. In Table 4(a), we report the impact of providing segmentation signal to different modules. We see that just priming `conv5_1` gives a 1

Table 4: Analation analysis of Contextual Priming and Iterative Feedback on VOC 12S val set. All methods use VOC 12S+A train set for training

(a) Evaluating Priming different layers

	mAP	mIOU
Base-MT	75.6	**65.8**
Priming `conv5_1`	76.6	**65.8**
Priming `conv5_1`, each `fc6`	**77.0**	65.3

(b) Evaluating Iterative Feedback design decisions

	Stage-2 Init.	mAP	mIOU
Base-MT	-	75.6	65.8
Iterative Feedback to `conv1_1`	ImageNet	76.5	69.3
	Stage-1	76.3	69.3
Iterative Feedback to `conv{1,2,3,4}_1`	ImageNet	76.3	69.1
	Stage-1	**77.3**	**69.5**

point boost over Bast-MT and adding the segmentation signal to each individual region ('`seg+pool5`' to `fc6`) gives another 0.4 points boost. It is interesting that the segmentation performance is not affected when priming `conv5_1`, but it drops by 0.5 mIOU when we prime each region. Our hypothesis is that gradients accumulated from all regions in the mini-batch start overpowering the gradients from segmentation. To deal with this, methods like [79] can be used in the future.

5.4 Iterative Feedback

Next we study the impact of giving iterative top-down semantic feedback to the entire network. In this 2-unit setup, the first unit (Stage-1) is a trained Base-MT model and the second unit (Stage-2) is a Stage-1 initialized Base-MT model. During inference, we have the option of using the outputs from both units or just the Stage-2 unit. Given that segmentation is used as feedback, it is supposed to self-improve across units, therefore we use the Stage-2 output as our final output (similar to [49, 50]). For detection, we combine the outputs from both units; because the Stage-2 unit is modulated by segmentation, and the first unit is not, hence both might focus on different regions.

This iterative feedback improves the segmentation performance (Table 3) by **3.7** points over Base-MT (**3.5** points over ParseNet*). For detection, it improves over the Base-MT model by **1.7** points (**2** points over Faster R-CNN) (Table 2).

Design Evaluation. We study the impact of: (1) varying the degree of feedback to the Stage-2 unit, and (2) different Stage-2 initializations. In Table 4(b), we see that when initializing the Stage-2 unit with an ImageNet trained network, varying iterative feedback does not have much impact; however, when initializing with a Stage-1 model, providing more feedback leads to better performance. Specifically, iterative feedback to all shared `conv` layers improves both detection and segmentation by 1.7 mAP and 3.7 mIOU respectively, as opposed to feedback to just `conv1_1` (as in [49, 50]) which results in lower gains (Table 4(b)). Our hypothesis is that iterative feedback to a Stage-1 initialized unit allows the network to correct its mistakes and/or refine its predictions; therefore, providing more feedback leads to better performance.

Table 5: **Detection results on VOC 2007 detection test set**. All methods are trained on union of VOC07 trainval and VOC12 trainval

method	S	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Fast R-CNN [10]		70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster R-CNN [11]		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
Base-MT	✓	74.7	78.4	79.3	75.9	63.2	56.8	85.9	85.4	88.4	54.9	83.9	68.6	84.6	85.6	78.5	78.1	41.3	74.6	74.8	84.0	72.4
Ours (joint)	✓	**76.4**	79.3	80.5	76.8	72.0	58.2	85.1	86.5	89.3	60.6	82.2	69.2	87.0	87.2	81.6	78.2	44.6	77.9	76.7	82.4	71.9

Table 6: **Detection results on VOC 2012 detection test set**. All methods are trained on union of VOC07 trainval, VOC07 test and VOC12 trainval

method	S	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Fast R-CNN [10]		68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN [11]		70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Base MT	✓	71.1^	84.2	80.9	73.1	55.1	50.6	78.2	75.6	89.0	48.6	76.7	54.8	87.6	82.5	83.0	80.0	41.7	74.2	60.7	81.4	63.1
Ours (joint)	✓	**72.6**◇	84.0	81.2	75.9	60.4	51.8	81.2	77.4	90.9	50.2	77.6	58.7	88.4	83.6	82.0	80.4	41.5	75.0	64.2	82.9	65.1

^ `http://host.robots.ox.ac.uk:8080/anonymous/RUZFQC.html`, ◇ `http://host.robots.ox.ac.uk:8080/anonymous/YFSUQA.html`

5.5 Joint Model

Finally, we evaluate our joint 2-unit model, where each unit is a model with contextual priming, and both units are connected via segmentation feedback. In this setup, a trained contextual priming model is used as the Stage-1 unit as well as the initialization for the Stage-2 unit. We remove the dropout layers from Stage-2 unit. Inference follows the procedure described in Section 5.4.

As shown in Table 2, for detection, the joint model achieves **77.8**% mAP (+**2.2** points over Base-MT and +**2.5** points over Faster R-CNN), which is better than both priming only and feedback only models. This suggests that both forms of top-down feedback are complementary for object detection. The segmentation performance (Table 3) is similar to the feedback only model, which is expected since in both cases, the segmentation module receives similar feedback.

6 Results

We now report results on the PASCAL VOC and MS COCO [85] datasets. We also evaluate the region proposal generation on the proxy metric of average recall.
Experimental Setup. When training on the VOC datasets with extra data (Table 5, 6 and 7), we use 100k SGD iterations (other hyperparameters follow Section 5); and for MS COCO, we use 490k SGD iterations with an initial LR of 10^{-3} and decay step size of 200k, owing to a larger epoch size.
VOC07 and VOC12 Results. Table 5 shows that on VOC07, our joint priming and feedback model improves the detection mAP by **1.7** points over Base-MT and **3.2** points over Faster R-CNN. Similarly, on VOC12 (Table 6), priming and feedback lead to **1.5** points boost over Bast-MT (**2.2** over Faster R-CNN). For segmentation on VOC12 (Table 7), we see a huge **5** point boost in mIOU over Base-MT. We would like highlight that both Base-MT and our joint model use exactly the same annotations and hyperparameters; therefore the performance boosts are because of contextual priming and iterative feedback in our model.

Table 7: **Segmentation results on VOC 2012 segmentation test set**. All methods are trained on union of VOC07 trainval, VOC07 test and VOC12 trainval

method	S	mIOU	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Base MT	✓	66.4$^\wedge$	91.3	82.0	37.7	77.6	58.8	58.8	84.0	75.6	83.1	25.1	70.9	57.8	74.0	74.6	76.4	75.0	48.8	73.7	45.6	72.3	52.0
Ours (joint)	✓	**71.4**$^\diamond$	93.0	89.3	41.4	84.1	63.8	65.2	88.1	80.9	88.6	28.4	75.4	60.6	80.3	80.9	83.1	79.7	55.4	77.9	48.2	75.8	58.8

$^\wedge$`http://host.robots.ox.ac.uk:8080/anonymous/RUZFQC.html`, $^\diamond$`http://host.robots.ox.ac.uk:8080/anonymous/YFSUQA.html`

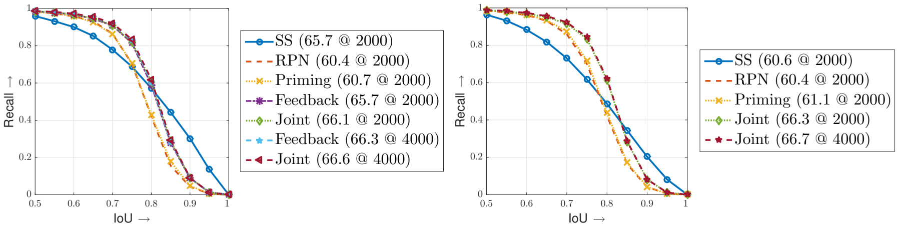


Fig. 4: Recall-to-IoU on VOC12 Segmentation val set (left) and VOC07 test set (right) (best viewed digitally).

Recall-to-IOU. Since our hypothesis is that priming and feedback lead to better proposal generation, we also evaluate the recall of region proposals by the RPN module from various models, at different IOU thresholds. In Figure 4, we show the results of using 2000 proposal per RPN module. Since feedback models have 2 units, we report their number with both 4000 and top 2000 proposals (sorted by `cls` score). As can be seen priming, feedback and joint models all lead to higher average recall (shown in legend) over the baseline RPN module.

Table 8: **Detection results on MS COCO 2015 test-dev set**. All methods use COCO trainval35k for training and results were obtained from the online 2015 test-dev server. Legend: **F**: using iterative feedback, **P**: using contextual priming, **S**: uses segmentation labels.

Method	**S**	**P**	**F**	**AP, IoU:**			**AP, Area:**			**AR, # Dets:**			**AR, Area:**		
				0.5:0.95	0.50	0.75	Small	Med.	Large	1	10	100	Small	Med.	Large
Faster R-CNN				24.5	46.0	23.7	8.2	26.4	36.9	24.0	34.8	35.5	13.4	39.2	54.3
Base-MT	✓			25.0	47.0	24.2	8.1	27.1	38.1	24.3	35.1	35.8	13.2	39.8	55.0
Ours (priming)	✓	✓		25.8	48.2	25.3	8.3	27.8	38.6	24.5	35.7	36.5	13.6	40.6	54.7
Ours (joint)	✓	✓	✓	27.5	49.2	27.8	8.9	29.5	41.5	25.5	37.4	38.3	14.6	42.5	57.4

MS COCO Results. We also perform additional analysis of contextual priming on the COCO [85] dataset. For MS COCO dataset, we only use the output from Stage-2 unit. Our priming model results in +1.2 AP points (+2.1 AP50) over Faster R-CNN and +0.8 AP points (+1.1 AP50) over Base-MT on the COCO test-dev set [85]. On further analysis, we notice that the most performance gains are for objects where context should intuitively help; e.g., +12.4 for 'parking-meter', +8.7 for 'suitcase', +8.3 for 'umbrella' etc. on AP50 wrt. to Faster R-CNN. Finally, our joint model achieves **27.5** AP points (+**3** AP points over Faster R-CNN and +**2.5** over Base-MT), further highlighting effectiveness of the proposed method.

7 Conclusion

We presented and investigated how we can incorporate top-down semantic feedback in the state-of-the-art Faster R-CNN framework. We proposed to augment a segmentation network to Faster R-CNN, which is then used to provide top-down contextual feedback to the region proposal generation and object detection modules. We also use this segmentation network to provide top-down feedback to the entire Faster R-CNN network iteratively. Our results demonstrate the effectiveness of these top-down feedback mechanisms for the tasks of region proposal generation, object detection and semantic segmentation.

Acknowledgments. We thank Ross Girshick, Ishan Misra and Sean Bell for helpful discussions. AS was supported by the Microsoft Research PhD Fellowship. This work was also partially supported by ONR MURI N000141612007. We thank NVIDIA for donating GPUs.

References

[1] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)

[2] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2010)

[3] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)

[4] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)

[5] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

[7] Viola, P., Jones, M.: Robust real-time object detection. IJCV (2001)

[8] Gu, C., Lim, J.J., Arbeláez, P., Malik, J.: Recognition using regions. In: CVPR. (2009)

[9] Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: ICCV. (2013)

[10] Girshick, R.: Fast R-CNN. In: ICCV. (2015)

[11] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497 (2015)

[12] Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. (2014)

[13] Tulving, E., Schacter, D.L.: Priming and human memory systems. Science (1990)

[14] Wig, G.S., Grafton, S.T., Demos, K.E., Kelley, W.M.: Reductions in neural activity underlie behavioral components of repetition priming. Nature neuroscience (2005)

[15] Meng, Y., Ye, X., Gonsalves, B.D.: Neural processing of recollection, familiarity and priming at encoding: Evidence from a forced-choice recognition paradigm. Brain research (2014)

[16] Biederman, I.: On the semantics of a glance at a scene. (1981)

[17] Hupe, J., James, A., Payne, B., Lomber, S., Girard, P., Bullier, J.: Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. Nature (1998)

[18] Kravitz, D.J., Saleem, K.S., Baker, C.I., Ungerleider, L.G., Mishkin, M.: The ventral visual pathway: an expanded neural framework for the processing of object quality. Trends in cognitive sciences (2013)

[19] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. IEEE (1998)

[20] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei., L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)

[21] Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. NIPS (2013)

[22] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2015)

[23] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)

[24] Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A.: Selective search for object recognition. IJCV (2013)

[25] Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR. (2010)

[26] Endres, I., Hoiem, D.: Category independent object proposals. In: ECCV. (2010)

[27] Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR. (2010)

[28] Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. TPAMI (2012)

[29] Zitnick, C.L., Dollar, P.: Edge boxes: Locating object proposals from edges. In: ECCV. (2014)

[30] Fidler, S., Mottaghi, R., Yuille, A., Urtasun, R.: Bottom-up segmentation for top-down detection. In: CVPR. (2013)

[31] Cinbis, R.G., Verbeek, J., Schmid, C.: Segmentation driven object detection with Fisher vectors. In: ICCV. (2013)

[32] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. PAMI (2015)

[33] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)

[34] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. (2015)

[35] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV. (2014)

[36] Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. arXiv preprint arXiv:1512.04143 (2015)

[37] Dong, J., Chen, Q., Yan, S., Yuille, A.: Towards unified object detection and semantic segmentation. In: 2014. (2014)

[38] Chen, X., Shrivastava, A., Gupta, A.: Enriching visual knowledge bases via object discovery and segmentation. In: CVPR. (2014)

[39] Davenport, J.L., Potter, M.C.: Scene consistency in object and background perception. Psychological Science (2004)

[40] Felleman, D.J., Van Essen, D.C.: Distributed hierarchical processing in the primate cerebral cortex. Cerebral cortex (1991)

[41] Chun, M.M., Jiang, Y.: Top-down attentional guidance based on implicit learning of visual covariation. Psychological Science (1999)

[42] Gilbert, C.D., Sigman, M.: Brain states: top-down influences in sensory processing. Neuron (2007)

[43] Lamme, V.A., Roelfsema, P.R.: The distinct modes of vision offered by feedforward and recurrent processing. Trends in neurosciences (2000)

[44] Wyatte, D., Curran, T., O'Reilly, R.: The limits of feedforward vision: Recurrent processing promotes robust object recognition when objects are degraded. Journal of Cognitive Neuroscience (2012)

[45] Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. PAMI (2010)

[46] Ross, S., Munoz, D., Hebert, M., Bagnell, J.A.: Learning message-passing inference machines for structured prediction. In: CVPR. (2011)

[47] Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: NIPS. (2014)

[48] Stollenga, M.F., Masci, J., Gomez, F., Schmidhuber, J.: Deep networks with internal selective attention through feedback connections. In: NIPS. (2014)

[49] Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. arXiv preprint arXiv:1507.06550 (2015)

[50] Li, K., Hariharan, B., Malik, J.: Iterative instance segmentation. arXiv preprint arXiv:1511.08498 (2015)

[51] Gatta, C., Romero, A., van de Veijer, J.: Unrolling loopy top-down semantic feedback in convolutional deep networks. In: CVPR Workshops. (2014)

[52] Hock, H.S., Gordon, G.P., Whitehurst, R.: Contextual relations: the influence of familiarity, physical plausibility, and belongingness. Perception & Psychophysics (1974)

[53] Oliva, A., Torralba, A.: The role of context in object recognition. Trends in cognitive sciences (2007)

[54] Palmer, t.E.: The effects of contextual scenes on the identification of objects. Memory & Cognition (1975)

[55] Hollingworth, A.: Does consistent scene context facilitate object perception? Journal of Experimental Psychology: General (1998)

[56] Galleguillos, C., Belongie, S.: Context based object categorization: A critical survey. CVIU (2010)

[57] Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR. (2009)

[58] Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: ICCV. (2003)

[59] Torralba, A.: Contextual priming for object detection. IJCV (2003)

[60] Torralba, A., Sinha, P.: Statistical context priming for object detection. In: ICCV. (2001)

[61] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV. (2007)

[62] Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: CVPR. (2012)

[63] Murphy, K., Torralba, A., Freeman, W., et al.: Using the forest to see the trees: a graphical model relating features, objects and scenes. NIPS (2003)

[64] Gidaris, S., Komodakis, N.: Object detection via a multi-region & semantic segmentation-aware cnn model. arXiv preprint arXiv:1505.01749 (2015)

[65] Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR. (2014)

[66] Zhu, Y., Urtasun, R., Salakhutdinov, R., Fidler, S.: segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In: CVPR. (2015)

[67] Mostajabi, M., Yadollahpour, P., Shakhnarovich, G.: Feedforward semantic segmentation with zoom-out features. In: CVPR. (2015)

[68] Gupta, S., Hariharan, B., Malik, J.: Exploring person context and local scene context for object detection. arXiv preprint arXiv:1511.08177 (2015)

[69] Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with RCNN. In: ICCV. (2015)

[70] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. (2015)

[71] Schwing, A.G., Urtasun, R.: Fully connected deep structured networks. arXiv preprint arXiv:1503.02351 (2015)

[72] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)

[73] Ladickỳ, L., Sturgess, P., Alahari, K., Russell, C., Torr, P.H.: What, where and how many? combining object detectors and crfs. In: ECCV. (2010)

[74] Lin, G., Shen, C., Reid, I., et al.: Efficient piecewise training of deep structured models for semantic segmentation. arXiv preprint arXiv:1504.01013 (2015)

[75] Pinheiro, P.O., Collobert, R., Dollar, P.: Learning to segment object candidates. In: NIPS. (2015)

[76] Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: CVPR. (2016)

[77] Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579 (2015)

[78] Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)

[79] Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch Networks for Multi-task Learning. In: CVPR. (2016)

[80] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. (2010)

[81] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)

[82] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)

[83] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)

[84] Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. (2011)

[85] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. (2014)