Object detection via a multi-region & semantic segmentation-aware CNN model

Spyros Gidaris
Universite Paris Est, Ecole des Ponts ParisTech
gidariss@imagine.enpc.fr

Nikos Komodakis
Universite Paris Est, Ecole des Ponts ParisTech
nikos.komodakis@enpc.fr

Abstract

We propose an object detection system that relies on a multi-region deep convolutional neural network (CNN) that also encodes semantic segmentation-aware features. The resulting CNN-based representation aims at capturing a diverse set of discriminative appearance factors and exhibits localization sensitivity that is essential for accurate object localization. We exploit the above properties of our recognition module by integrating it on an iterative localization mechanism that alternates between scoring a box proposal and refining its location with a deep CNN regression model. Thanks to the efficient use of our modules, we detect objects with very high localization accuracy. On the detection challenges of PASCAL VOC2007 and PASCAL VOC2012 we achieve mAP of 78.2% *and* 73.9% *correspondingly, surpassing any other published work by a significant margin.*

1. Introduction

One of the most studied problems of computer vision is that of object detection: given an image return all the instances of one or more type of objects in form of bounding boxes that tightly enclose them. The last two years, huge improvements have been observed on this task thanks to the recent advances of deep learning community [18, 1, 14]. Among them, most notable is the work of Sermanet et al. [24] with the Overfeat framework and the work of Girshick et al. [10] with the R-CNN framework..

Overfeat [24] uses two CNN models that apply in a sliding window fashion on multiple scales of an image. The first is used to classify if a window contains an object and the second to predict the true bounding box location of the object. Finally, the dense class and location predictions are merged with a greedy algorithm in order to produce the final set of object detections.

R-CNN [10] uses Alex Krizhevsky's Net [17] to extract features from box proposals provided by selective search [27] and then it classifies them with class specific linear SVMs. Girshick et al. [10], manage to train networks with millions of parameters by first pre-training on the auxiliary task of image classification and then fine-tuning on a small set of images annotated for the detection task. This simple pipeline surpasses by a large margin the detection performance of all the previously published systems, such as deformable parts models [8] and non-linear multi-kernel approaches [28]. This success of R-CNN comes from the fact that hand-engineered features like HOG [3] or SIFT [22] are replaced with the high level object representations produced from the last layer of a CNN model. By employing an even deeper CNN model, such as the 16-layers VGG-Net [25], they boosted the performance another 7 points [10].

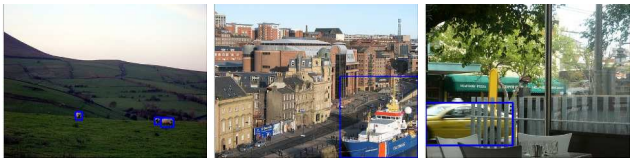


Figure 1: Left: detecting the sheep on this scene is very difficult without referring on the context, mountainish landscape. **Center:** In contrast, the context on the right image can only confuse the detection of the boat. The pure object characteristics is what a recognition model should focus on in this case. **Right:** This car instance is occluded on its right part and the recognition model should focus on the left part in order to confidently detect it.

In this paper we aim to further advance the state-of-the-art on object detection by improving on two key aspects that play a critical role in this task: object representation and object localization.

Object representation. One of the lessons learned from the above-mentioned works is that indeed features matter a lot on object detection and our work is partly motivated from this observation. However, instead of proposing only a network architecture that is deeper, here we also opt for an architecture of greater width, *i.e.*, one whose last hidden layers provide features of increased dimensionality. In doing so, our goal is to build a richer candidate box representation. This is accomplished at two levels:

(1). At a first level, we want our object representation

This work was supported by the ANR SEMAPOLIS project. Its code will become available on github.com/gidariss/mrcnn-object-detection/.

to capture several different aspects of an object such as its pure appearance characteristics, the distinct appearance of its different regions (object parts), context appearance, the joint appearance on both sides of the object boundaries, and semantics. We believe that such a rich representation will further facilitate the problem of recognising (even difficult) object instances under a variety of circumstances (like, *e.g.*, those depicted in Figure 1). In order to achieve our goal, we propose a multi-component CNN model, called *multi-region CNN* hereafter, each component of which is steered to focus on a different region of the object thus enforcing diversification of the discriminative appearance factors captured by it.

Additionally, as we will explain shortly, by properly choosing and arranging some of these regions, we aim also to help our representation in being less invariant to inaccurate localization of an object. Note that this property, which is highly desirable for detection, contradicts with the built-in invariances of CNN models, which stem from the use of max-pooling layers.

(2). At a second level, inspired by the close connection that exists between segmentation and detection, we wish to enrich the above representation so that it also captures semantic segmentation information. To that end, we extend the above CNN model such that it also learns novel CNN-based semantic segmentation-aware features. Importantly, learning these features (*i.e.*, training the extended unified CNN model) does not require having ground truth object segmentations as training data.

Object localization. Besides object representation, our work is also motivated by the observation that, due to the remarkable classification capability of the recent CNN models [17, 30, 25, 16, 13, 26], the bottleneck for good detection performance is now the accurate object localization. Indeed, Girshick et al. observed that the most common type of false positives in their R-CNN system, is the mis-localized detections [10]. They attempt to fix some of those errors by employing a post processing step of bounding box regression that is applied on the final list of detections. However, this technique only helps with small localization errors. We believe that there is much more space for improvement on this aspect. In order to prove our belief, we attempt to built a more powerful localization system that combines our multi-region CNN model with a CNN-model for bounding box regression, which are used within an iterative scheme that alternates between scoring candidate boxes and refining their coordinates.

Related work. We should mention that feature extraction from multiple regions has also been exploited for performing object recognition in videos by Leordeanu et al. [19]. As features they use the outputs of HOG [3]+SVM classifiers trained on each region separately and the 1000-class predictions of a CNN pre-trained on ImageNet. Instead, we fine-tune our deep networks on each region separately in order to accomplish our goal of learning deep features that will adequately capture their discriminative appearance characteristics. Furthermore, our regions exhibit more variety on their shape that, as we will see in section 2.1, helps on boosting the detection performance. Also, in the contemporaneous with us work of Zhu et al. [31], the authors extract contextual features from an additional region and utilize a MRF inference framework to exploit object segmentation proposals (obtained through parametric min-cuts).

Contributions. To summarize, our contributions are as follows: **(1)** We develop a multi-region CNN recognition model that yields an enriched object representation capable of capturing a diversity of discriminative appearance factors and of exhibiting localization sensitivity that is desired for the task of accurate object localization. **(2)** We furthermore extend the above model by proposing a unified neural network architecture that also learns semantic segmentation-aware CNN features for the task of object detection. These features are jointly learnt in a weakly supervised manner, thus requiring no additional annotation. **(3)** We show how to significantly improve the localization capability by coupling the aforementioned CNN recognition model with a CNN model for bounding box regression, adopting a scheme that alternates between scoring candidate boxes and refining their locations, as well as modifying the post-processing step of non-maximum-suppression. **(4)** Our detection system achieves mAP of 78.2% and 73.9% on VOC2007 [6] and VOC2012 [7] detection challenges respectively, thus surpassing the previous state-of-art by a quite significant margin.

The remainder of the paper is structured as follows: We describe our multi-region CNN model in §2. We show how to extend it to also learn semantic segmentation-aware CNN features in §3. Our localization scheme is described in §4 and implementation details are provided in §5. We present experimental results in §6 and conclude in §7.

2. Multi-Region CNN Model

The recognition model that we propose consists of a multi-component CNN network, each component of which is chosen so as to focus on a different region of an object. We call this a Multi-Region CNN model. We begin by describing first its overall architecture. To that end, in order to facilitate the description of our model we introduce a general CNN architecture abstraction that decomposes the computation into two different modules:

Activation maps module. This part of the network gets as input the entire image and outputs activation maps (feature maps) by forwarding it through a sequence of convolutional layers.

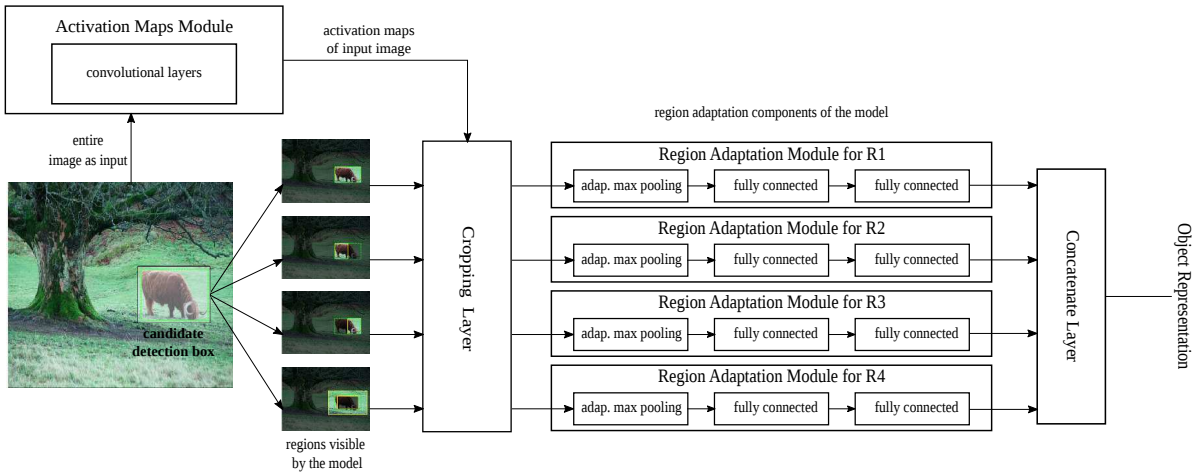


Figure 2: Multi Region CNN architecture. For clarity we present only four of the regions that participate on it. An "adaptive max pooling" layer uses spatially adaptive pooling as in [12] (but with a one-level pyramid). The above architecture can be extended to also learn semantic segmentation-aware CNN features (see section 3) by including additional 'activation-maps' and 'region-adaptation' modules that are properly adapted for this task (these are not shown here due to lack of space).

Region adaptation module. Given a region R on the image and the activation maps of the image, this module projects R on the activation maps, crops the activations that lay inside it, pools them with a spatially adaptive (max-)pooling layer [12], and then forwards them through a multi-layer network.

Under this formalism, the architecture of the Multi-Region CNN model can be seen in Figure 2. Initially, the entire image is forwarded through the activation maps module. Then, a candidate detection box B is analysed on a set of (possibly overlapping) regions $\{R_i\}_{i=1}^{k}$ each of which is assigned to a dedicated region adaptation module (note that these regions are always defined relatively to the bounding box B). As mentioned previously, each of these region adaptation modules passes the activations pooled from its assigned region through a multilayer network that produces a high level feature. Finally, the candidate box representation is obtained by concatenating the last hidden layer outputs of all the region adaptation modules.

By steering the focus on different regions of an object, our aim is: (i) to force the network to capture various complementary aspects of the objects appearance (*e.g.*, context, object parts, *etc.*), thus leading to a much richer and more robust object representation, and (ii) to also make the resulting representation more sensitive to inaccurate localization (*e.g.*, by focusing on the border regions of an object), which is also crucial for object detection.

In the next section we describe how we choose the regions $\{R_i\}_{i=1}^{k}$ to achieve the above goals, and also discuss their role in object detection.

2.1. Region components and their role in detection

We utilize 2 types of region shapes: rectangles and rectangular rings, where the latter type is defined in terms of an inner and outer rectangle. We describe below all of the regions that we employ, while their specifications are given in the caption of Figure 3.

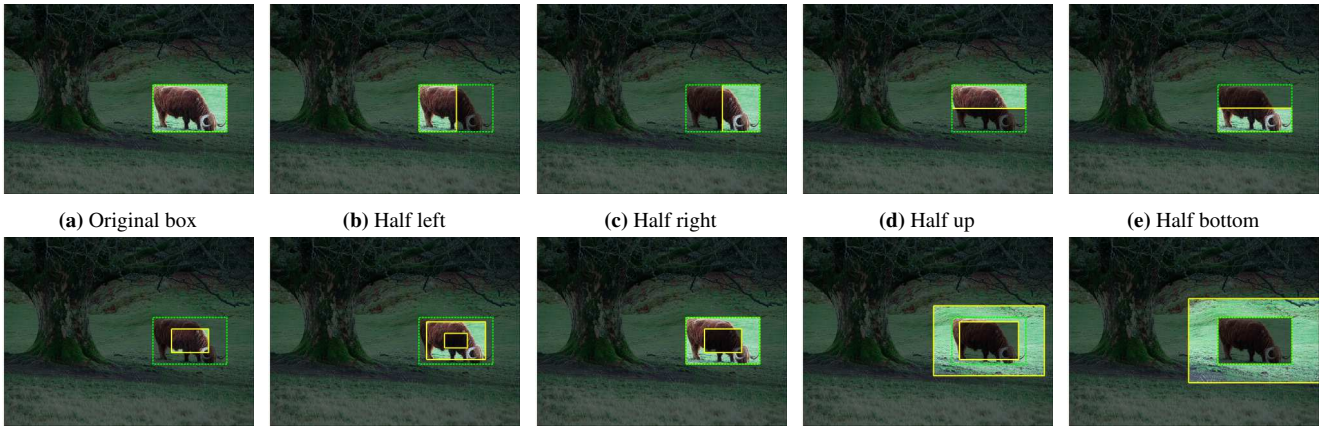
Original candidate box: this is the candidate detection box itself as being used in R-CNN [10] (Figure 3a). A network trained on this type of region is guided to capture the appearance information of the entire object. When it is used alone consists the baseline of our work.

Half boxes: those are the left/right/up/bottom half parts of a candidate box (figures 3b, 3c, 3d, and 3e). Networks trained on each of them, are guided to learn the appearance characteristics present only in each half part of an object or in each side of the objects borders, aiming to make the representation more robust with respect to occlusions.

Central Regions: there are two type of central regions in our model (figures 3f and 3g). The networks trained on them are guided to capture the pure appearance characteristics of the central part of an object that is probably less interfered by other objects next to it or by background.

Border Regions: we include two such regions, with the shape of rectangular rings (figures 3h and 3i). We expect that the dedicated on them networks will be guided to focus on the joint appearance characteristics on both sides of the object borders, also aiming to make the representation more sensitive to inaccurate localization.

Contextual Region: there is one region of this type that has rectangular ring shape (Figure 3j). Its assigned network is driven to focus on the contextual appearance that surrounds an object such as the appearance of its background or of other objects next to it.



(a) Original box **(b)** Half left **(c)** Half right **(d)** Half up **(e)** Half bottom

(f) Central Region **(g)** Central Region **(h)** Border Region **(i)** Border Region **(j)** Context. Region

Figure 3: Illustration of the regions used in the Multi-Region CNN model. With yellow solid lines are the borders of the regions and with green dashed lines are the borders of the candidate detection box. **Region a:** it is the candidate box itself as being used in R-CNN [10]. **Region b, c, d, e:** they are the left/right/up/bottom half parts of the candidate box. **Region f:** it is obtained by scaling the candidate box by a factor of 0.5. **Region g:** the inner box is obtained by scaling the candidate box by a factor of 0.3 and the outer box by a factor of 0.8. **Region h:** we obtain the inner box by scaling the candidate box by a factor of 0.5 and the outer box has the same size as the candidate box. **Region i:** the inner box is obtained by scaling the candidate box by a factor of 0.8 and the outer box by a factor of 1.5. **Region j:** the inner box is the candidate box itself and the outer box is obtained by scaling the candidate box by a factor of 1.8.

Role in detection. Concerning the general role of the regions in object detection, we briefly focus below on two of the reasons why using these regions helps:

Discriminative feature diversification. Our hypothesis is that having regions that render visible to their network-components only a limited part of the object or only its immediate surrounding forces each network-component to discriminate image boxes solely based on the visual information that appears in them thus diversifying the discriminative factors captured by our overall recognition model. For example, if the border region depicted in Figure 3i is replaced with one that includes its whole inner content, then we would expect that the network-component dedicated on it will not pay the desired attention on the visual content that is concentrated around the borders of an object. We tested such a hypothesis by conducting an experiment where we trained and tested two Multi-Region CNN models that consist of two regions each. Model A included the original box region (Figure 3a) and the border region of Figure 3i that does not contain the central part of the object. In model B, we replaced the latter region (Figure 3i), which is a rectangular ring, with a normal box of the same size. Both of them were trained on PASCAL VOC2007 [6] trainval set and tested on the test set of the same challenge. Model A achieved 64.1% mAP while Model B achieved 62.9% mAP which is 1.2 points lower and validates our assumption.

Localization-aware representation. We argue that our multi-region architecture as well as the type of regions included, address to a certain extent one of the major problems on the detection task, which is the inaccurate object localization. We believe that having multiple regions with dedicated network-components on each of them, imposes soft constraints regarding the visual content allowed in each type of region for a given candidate detection box. We experimentally justify this argument by referring to section 6.2.

3. Semantic Segmentation-Aware CNN Model

To further diversify the features encoded by our representation, we extend the Multi-Region CNN model so that it also learns semantic segmentation-aware CNN features. The motivation for this comes by the close connection between segmentation and detection and by the fact that segmentation related cues are empirically known to often help object detection [5, 11, 23]. In the context of our multi-region CNN network, the incorporation of the semantic segmentation-aware features is done by adding properly adapted versions of the two main modules of the network, *i.e.*, the 'activation-maps' and 'region-adaptation' modules:

- **Activation maps module for semantic segmentation aware features.** In order to serve the purpose of exploiting semantic segmentation aware features, for this module we adopt a Fully Convolutional Network (FCN) [21] trained to predict class specific foreground probabilities.

  Weakly supervised training. For training this FCN we use only the provided bounding box annotations for detection and not any additional segmentation annotation. To that end, we follow a weakly supervised training strategy and we create artificial foreground

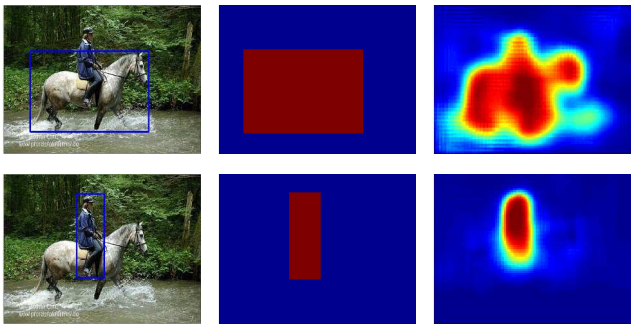


Figure 4: Illustration of the weakly supervised training of the FCN [21] used as activation maps module for the semantic segmentation aware CNN features. **Left column:** images with the ground truth bounding boxes drawn on them. The classes depicted from top to down order are horse and human. **Middle column:** the segmentation target values used during training of the FCN. They are artificially generated from the ground truth bounding box(es) on the left column. We use blue color for the background and red color for the foreground. **Right column:** the foreground probabilities estimated from our trained FCN model. These clearly verify that, despite the weakly supervised training, our extracted features carry significant semantic segmentation information.

class-specific segmentation masks by labelling the pixels that lay inside the ground truth bounding boxes as foreground and the rest as background (see left and middle column in Figure 4). As can be seen in Figure 4 right column, despite the weakly supervised way of training, the resulting activations still carry significant semantic segmentation information, enough even to delineate the boundaries of the object and separate the object from its background.

Activation maps. After the FCN has been trained on the auxiliary task of foreground segmentation, we drop the last classification layer and we use the rest of the FCN network in order to extract from images semantic segmentation aware activation maps.

- **Region adaptation module for semantic segmentation aware features.** We exploit the above activation maps by treating them as mid-level features and adding on top of them a single region adaptation module trained for our primary task of object detection. In this case, we choose to use a single region obtained by enlarging the candidate detection box by a factor of 1.5 (such a region contains semantic information also from the surrounding of a candidate detection box). The reason that we do not repeat the same regions as in the initial Multi-Region CNN architecture is for efficiency as these are already used for capturing the appearance cues of an object.

We combine the Multi-Region CNN features and the semantic segmentation aware CNN features by concatenating them. The resulting network thus jointly learns deep features of both types during training.

4. Object Localization

As already explained, our Multi-Region CNN recognition model exhibits the localization awareness property that is necessary for accurate object localization. However, by itself it is not enough. In order to make full use of it, our recognition model needs to be presented with well localized candidate boxes that in turn will be scored with high confidence from it. The solution that we adopt consists of 3 main components:

CNN region adaptation module for bounding box regression. We introduce an extra region adaptation module that, instead of being used for object recognition, is trained to predict the object bounding box. It is applied on top of the activation maps produced from the Multi-Region CNN model and, instead of a typical one-layer ridge regression model [10], consists of two hidden fully connected layers and one prediction layer that outputs 4 values (*i.e.*, a bounding box) per category. In order to allow it to predict the location of object instances that are not in the close proximity of any of the initial candidate boxes, we use as region a box obtained by enlarging the candidate box by a factor of 1.3. This combination offers a significant boost on the detection performance of our system by allowing it to make more accurate predictions and for more distant objects.

Iterative Localization. Our localization scheme starts from the selective search proposals [27] and works by iteratively scoring them and refining their coordinates. Specifically, let $\mathbf{B}_c^t = \{B_{i,c}^t\}_{i=1}^{N_{c,t}}$ denote the set of $N_{c,t}$ bounding boxes generated on iteration t for class c and image X. For each iteration $t = 1, ..., T$, the boxes from the previous iteration $\mathbf{B}_c^{t-1}$ are scored with $s_{i,c}^t = \mathcal{F}_{rec}(B_{i,c}^{t-1}|c, X)$ by our recognition model and refined into $B_{i,c}^t = \mathcal{F}_{reg}(B_{i,c}^{t-1}|c, X)$ by our CNN regression model, thus forming the set of candidate detections $\mathbf{D}_c^t = \{(s_{i,c}^t, B_{i,c}^t)\}_{i=1}^{N_{c,t}}$. For the first iteration $t = 1$, the box proposals $\mathbf{B}_c^0$ are coming from selective search [27] and are common between all the classes. Also, those with score $s_{i,c}^0$ below a threshold τ_s are rejected in order to reduce the computational burden of the subsequent iterations. This way, we obtain a sequence of candidate detection sets $\{\mathbf{D}_c^t\}_{t=1}^T$ that all-together both exhibit high recall of the objects on an image and are well localized on them.

Bounding box voting. After the last iteration T, the candidate detections $\{\mathbf{D}_c^t\}_{t=1}^T$ produced on each iteration t are merged together $\mathbf{D}_c = \cup_{t=1}^T \mathbf{D}_c^t$. Because of the multiple regression steps, the generated boxes will be highly concen-

In practice $T = 2$ iterations were enough for convergence.

We use $\tau_s = -2.1$, which was selected such that the average number of box proposals per image from all the classes together to be around 250.

trated around the actual objects of interest. We exploit this "by-product" of the iterative localization scheme by adding a step of bounding box voting. First, standard non-max suppression [10] is applied on $\mathbf{D}_c$ and produces the detections $\mathbf{Y}_c = \{(s_{i,c}, B_{i,c})\}$ using an IoU overlap threshold of 0.3. Then, the final bounding box coordinates $B_{i,c}$ are further refined by having each box $B_{j,c} \in \mathcal{N}(B_{i,c})$ (where $\mathcal{N}(B_{i,c})$ denotes the set of boxes in $\mathbf{D}_c$ that overlap with $B_{i,c}$ by more than 0.5 on IoU metric) to vote for the bounding box location using as weight its score $w_{j,c} = max(0, s_{j,c})$, or

$$B^{'}_{i,c} = \frac{\sum_{j:B_{j,c} \in \mathcal{N}(B_{i,c})} w_{j,c} \cdot B_{j,c}}{\sum_{j:B_{j,c} \in \mathcal{N}(B_{i,c})} w_{j,c}}. \quad (1)$$

The final set of object detections for class c will be $\mathbf{Y}^{'}_c = \{(s_{i,c}, B^{'}_{i,c})\}$.

5. Implementation Details

For all the CNN models involved in our proposed system, we used the publicly available 16-layers VGG model [25] pre-trained on ImageNet [4] for the task of image classification. For simplicity, we fine-tuned only the fully connected layers (fc6 and fc7) of each model while we preserved the pre-trained weights for the convolutional layers (conv1_1 to conv5_3), which are shared among all the models of our system.

Multi-Region CNN model. Its activation maps module consists of the convolutional part (layers conv1_1 to conv5_3) of the 16-layers VGG-Net that outputs 512 feature channels. The max-pooling layer right after the last convolutional layer is omitted on this module. Each region adaptation module inherits the fully connected layers of the 16-layers VGG-Net and is fine-tuned separately from the others. Regarding the regions that are rectangular rings, both the inner and outer box are projected on the activation maps and then the activations that lay inside the inner box are masked out by setting them to zero (similar to the Convolutional Feature Masking layer proposed on [2]). To train the region adaptation modules, we follow the guidelines of R-CNN [10]. As an optimization objective we use multinomial logistic loss and the minimization is performed with stochastic gradient descent (SGD). The momentum is set to 0.9, the learning rate is initially set to 0.001 and then reduced by a factor of 10 every $30k$ iterations, and the minibatch has 128 samples. The positive samples are defined as the selective search proposals [27] that overlap a ground-truth bounding box by at least 0.5. As negative samples we use the proposals that overlap with a ground-truth bounding box on the range $[0.1, 0.5)$. The labelling of the training samples is relative to the original candidate boxes and is the same across all the different regions.

https://gist.github.com/ksimonyan/

Semantic segmentation-aware CNN model. The activation maps module architecture consists of the 16-layers VGG-Net without the last classification layer and transformed to a Fully Convolutional Network [21]. To train the FCN we used logistic loss for each class independently and SGD optimization with minibatch of size 10. The momentum was set to 0.9 and the learning rate was initialized to 0.01 and decreased by a factor of 10 every 20 epochs. The architecture of the region adaptation module consists of a spatially adaptive max-pooling layer [21] that outputs feature maps of 512 channels on a 9×9 grid, and a fully connected layer with 2096 channels. In order to train it, we used the same procedure as for the region components of the Multi-Region CNN model. The weights of the layers were initialized randomly from a Gaussian distribution.

Classification SVMs. In order to train the SVMs we follow the same principles as in [10]. The ground truth bounding boxes are used as positive samples and the selective search proposals [27] that overlap with the ground truth boxes by less than 0.3, are used as negative samples. We use hard negative mining the same way as in [10, 8].

CNN region adaptation module for bounding box regression. The activation maps module used as input in this case is common with the Multi-Region CNN model. The region adaptation module for bounding box regression inherits the fully connected hidden layers of the 16-layers VGG-Net. As a loss function we use the euclidean distance between the target values and the network predictions. As training samples we use the box proposals [27] that overlap by at least 0.4 with the ground truth bounding boxes. The target values are defined the same way as in R-CNN [10]. The learning rate is initially set to 0.01 and reduced by a factor of 10 every $40k$ iterations. The momentum is set to 0.9 and the minibatch size is 128.

6. Experimental Evaluation

We evaluate our detection system on PASCAL VOC2007 [6] and on PASCAL VOC2012 [7]. We use as baseline either the *Original candidate box* region alone (Figure 3a) and/or the R-CNN framework with VGG-Net [25]. Except if otherwise stated, for all the PASCAL VOC2007 results, we trained our models on the trainval set and tested them on the test set of the same year.

6.1. Results on PASCAL VOC2007

First, we asses the significance of each of the region adaptation modules alone on the object detection task. Results are reported in Table 1. As we expected, the best performing component is the *Original candidate box*. What is surprising is the high detection performance of individual regions like the *Border Region* on Figure 3i 54.8% or the *Contextual Region* on Figure 3j 47.2%. Despite the fact that the area visible by them includes limited or no at all

Adaptation Modules	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Original Box fig. 3a	**0.729**	**0.715**	**0.593**	**0.478**	**0.405**	**0.713**	**0.725**	**0.741**	**0.418**	**0.694**	0.591	**0.713**	0.662	**0.725**	**0.560**	**0.312**	**0.601**	0.565	0.669	**0.731**	**0.617**
Left Half Box fig. 3b	0.635	0.659	0.455	0.364	0.322	0.621	0.640	0.589	0.314	0.620	0.463	0.573	0.545	0.641	0.477	0.300	0.532	0.442	0.546	0.621	0.518
Right Half Box fig. 3c	0.626	0.605	0.470	0.331	0.314	0.607	0.616	0.641	0.278	0.487	0.513	0.548	0.564	0.585	0.459	0.262	0.469	0.465	0.573	0.620	0.502
Up Half Box fig. 3d	0.591	0.651	0.470	0.266	0.361	0.629	0.656	0.641	0.305	0.604	0.511	0.604	0.643	0.588	0.466	0.220	0.545	0.528	0.590	0.570	0.522
Bottom Half Box fig. 3e	0.607	0.631	0.406	0.397	0.233	0.594	0.626	0.559	0.285	0.417	0.404	0.520	0.490	0.649	0.387	0.233	0.457	0.344	0.566	0.617	0.471
Central Region fig. 3f	0.552	0.622	0.413	0.244	0.283	0.502	0.594	0.603	0.282	0.523	0.424	0.516	0.495	0.584	0.386	0.232	0.527	0.358	0.533	0.587	0.463
Central Region fig. 3g	0.674	0.705	0.547	0.367	0.337	0.678	0.698	0.687	0.381	0.630	0.538	0.659	0.667	0.679	0.507	0.309	0.557	0.530	0.611	0.694	0.573
Border Region fig. 3h	0.694	0.696	0.552	0.470	0.389	0.687	0.706	0.703	0.398	0.631	0.515	0.660	0.643	0.686	0.539	0.307	0.582	0.537	0.618	0.717	0.586
Border Region fig. 3i	0.651	0.649	0.504	0.407	0.333	0.670	0.704	0.624	0.323	0.625	0.533	0.594	0.656	0.627	0.517	0.223	0.533	0.515	0.604	0.663	0.548
Contextual Region fig. 3j	0.624	0.568	0.425	0.380	0.255	0.609	0.650	0.545	0.222	0.509	0.522	0.427	0.563	0.541	0.431	0.163	0.482	0.392	0.597	0.532	0.472
Semantic-aware region.	0.652	0.684	0.549	0.407	0.225	0.658	0.676	0.738	0.316	0.596	**0.635**	0.705	**0.670**	0.689	0.545	0.230	0.522	**0.598**	**0.680**	0.548	0.566

Table 1: Detection performance of individual regions on VOC2007 test set. They were trained on VOC2007 train+val set.

Approach	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN with VGG-Net	0.716	0.735	0.581	0.422	0.394	0.707	0.760	0.745	0.387	0.710	0.569	0.745	0.679	0.696	0.593	0.357	0.621	0.640	0.665	0.712	0.622
R-CNN with VGG-Net & bbox reg.	0.734	0.770	0.634	0.454	0.446	0.751	0.781	0.798	0.405	0.737	0.622	0.794	0.781	0.731	0.642	0.356	0.668	0.672	0.704	0.711	0.660
System of Yuting et al. [29]	0.725	0.788	0.67	0.452	0.510	0.738	0.787	0.783	0.467	0.738	0.615	0.771	0.764	0.739	0.665	0.392	0.697	0.594	0.668	0.729	0.665
System of Yuting et al. [29] & bbox reg.	0.741	0.832	0.670	0.508	0.516	0.762	0.814	0.772	0.481	0.789	0.656	0.773	0.784	0.751	0.701	0.414	0.696	0.608	0.702	0.737	0.685
Original Box fig. 3a	0.729	0.715	0.593	0.478	0.405	0.713	0.725	0.741	0.418	0.694	0.591	0.713	0.662	0.725	0.560	0.312	0.601	0.565	0.669	0.731	0.617
MR-CNN	0.749	0.757	0.645	0.549	0.447	0.741	0.755	0.760	0.481	0.724	0.674	0.765	0.724	0.749	0.617	0.348	0.617	0.640	0.735	0.760	0.662
MR-CNN & S-CNN	0.768	0.757	0.676	0.551	0.456	0.776	0.765	0.784	0.467	0.747	0.688	0.793	0.742	0.770	0.625	0.374	0.643	0.638	0.740	0.747	0.675
MR-CNN & S-CNN & Loc.	0.787	0.818	0.767	0.666	0.618	0.817	0.853	0.827	0.570	0.819	0.732	0.846	0.860	0.805	0.749	0.449	0.717	0.697	0.787	0.799	0.749
MR-CNN & S-CNN & Loc. & VOC07+12	**0.803**	**0.841**	**0.785**	**0.708**	**0.685**	**0.880**	**0.859**	**0.878**	**0.603**	**0.852**	**0.737**	**0.872**	**0.865**	**0.850**	**0.764**	**0.485**	**0.763**	**0.755**	**0.850**	**0.810**	**0.782**

Table 2: Detection performance of our modules on VOC2007 test set. Apart from the last entry that is trained on the superset of VOC2007 and VOC2012 train+val sets, all the other entries are trained on VOC2007 train+val set.

Approach	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN with VGG-Net from [29]	0.402	0.433	0.234	0.144	0.133	0.482	0.445	0.364	0.171	0.340	0.279	0.363	0.268	0.282	0.212	0.103	0.337	0.366	0.316	0.489	0.308
System of Yuting et al. [29]	0.463	0.581	0.311	0.216	0.258	0.571	0.582	0.435	0.230	0.464	0.290	0.407	0.406	0.463	0.334	0.106	0.413	0.409	0.458	0.563	0.398
System of Yuting et al. [29] & bbox reg.	0.471	**0.618**	0.352	0.181	0.297	**0.660**	0.647	0.480	**0.253**	0.504	0.349	0.437	0.508	0.494	0.368	0.137	0.447	0.436	0.498	**0.605**	0.437
Original Candidate Box	0.449	0.426	0.237	0.175	0.157	0.441	0.444	0.377	0.182	0.295	0.303	0.312	0.249	0.332	0.187	0.099	0.302	0.286	0.337	0.499	0.305
MR-CNN	0.495	0.505	0.292	0.235	0.179	0.513	0.504	0.481	0.206	0.381	0.375	0.387	0.296	0.403	0.239	0.151	0.341	0.389	0.422	0.521	0.366
MR-CNN & S-CNN	0.507	0.523	0.316	0.266	0.177	0.547	0.513	0.492	0.210	0.450	0.361	0.433	0.309	0.408	0.246	0.151	0.359	0.427	0.438	0.534	0.383
MR-CNN & S-CNN & Loc.	**0.549**	0.613	**0.430**	**0.315**	**0.383**	0.646	**0.650**	**0.512**	**0.253**	**0.544**	**0.505**	**0.521**	**0.591**	**0.540**	**0.393**	**0.159**	**0.485**	**0.468**	**0.553**	0.573	**0.484**

Table 3: Detection performance of our modules on VOC2007 test set. In this case, the IoU overlap threshold for positive detections is 0.7. Each model was trained on VOC2007 train+val set.

portion of the object, they outperform previous detection systems that were based on hand crafted features. Also interesting, is the high detection performance of the semantic segmentation aware region, 56.6%.

In Table 2, we report the detection performance of our proposed modules. The Multi-Region CNN model without the semantic segmentation aware CNN features (*MR-CNN*), achieves 66.2% mAP, which is 4.2 points higher than *R-CNN with VGG-Net* (62.0%) and 4.5 points higher than the *Original candidate box* region alone (61.7%). Moreover, its detection performance slightly exceeds that of *R-CNN with VGG-Net* and bounding box regression (66.0%). Extending the Multi-Region CNN model with the semantic segmentation aware CNN features (*MR-CNN & S-CNN*), boosts the performance of our recognition model another 1.3 points and reaches the total of 67.5% mAP. Comparing to the recently published method of Yuting et al. [29], our *MR-CNN & S-CNN* model scores 1 point higher than their best performing method that includes generation of extra box proposals via Bayesian optimization and structured loss during the fine-tuning of the VGG-Net. Significant is also the improvement that we get when we couple our recognition model with the CNN model for bounding box regression under the iterative localization scheme proposed (*MR-CNN & S-CNN & Loc.*). Specifically, the detection performance is raised from 67.5% to 74.9% setting the new state-of-the-art on this test set and for this set of training data (VOC2007 train+val set). In Table 2, we also report results when our overall system is trained on the train+val sets of VOC2007 and VOC2012 (entry *MR-CNN & S-CNN & Loc. & VOC07+12*). The detection performance in this case is raised to 78.2%.

In Table 3, we report the detection performance of our system when the overlap threshold for considering a detection positive is 0.7. This metric was proposed from Yuting et al. [29] in order to reveal the localization capability of their method. From the table we observe that each of our modules exhibit very good localization capability, which was our goal when designing them, and that our overall system exceeds in that metric the approach of Yuting et al. [29].

6.2. Detection error analysis

We use the tool of Hoiem et al. [15] to analyse the detection errors of our system. In Figure 6, we plot pie charts with the percentage of detections that are false positive due to bad localization, confusion with similar category, confusion with other category, and triggered on the background or an unlabelled object. We observe that, by using the Multi-

Approach	trained on	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN [10] with VGG-Net & bbox reg.	VOC12	0.792	0.723	0.629	0.437	0.451	0.677	0.667	0.830	0.393	0.662	0.517	0.822	0.732	0.765	0.642	0.337	0.667	0.561	0.683	0.610	0.630
Network In Network [20]	VOC12	0.802	0.738	0.619	0.437	0.430	0.703	0.676	0.807	0.419	0.697	0.517	0.782	0.752	0.769	0.651	0.386	0.683	0.580	0.687	0.633	0.638
Yuting et al. [29] & bbox reg.	VOC12	0.829	0.761	0.641	0.446	0.494	0.703	0.712	0.846	0.427	0.686	0.558	0.827	0.771	0.799	0.687	0.414	0.690	0.600	0.720	0.662	0.664
MR-CNN & S-CNN & Loc. (Ours)	VOC12	0.850	0.796	0.715	0.553	0.577	0.760	0.739	0.846	0.505	0.743	0.617	0.855	0.799	0.817	0.764	0.410	0.690	0.612	0.777	0.721	0.707
MR-CNN & S-CNN & Loc. (Ours)	VOC07	0.829	0.789	0.708	0.528	0.555	0.737	0.738	0.843	0.480	0.702	0.571	0.845	0.769	0.819	0.755	0.426	0.685	0.599	0.728	0.717	0.691
MR-CNN & S-CNN & Loc. (Ours)	VOC07+12	**0.855**	**0.829**	**0.766**	**0.578**	**0.627**	**0.794**	**0.772**	**0.866**	**0.550**	**0.791**	**0.622**	**0.870**	**0.834**	**0.847**	**0.789**	**0.453**	**0.734**	**0.658**	**0.803**	**0.740**	**0.739**

Table 4: Comparative results on VOC2012 test set. Apart from the last two entries that were trained on VOC2007 train+val set and VOC2007+2012 train+val sets correspondingly, all the other entries were trained on VOC2012 train+val set.

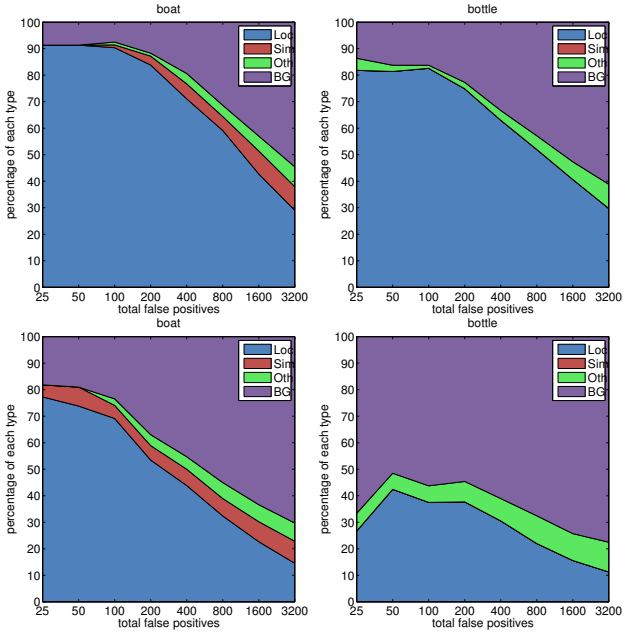


Figure 5: Top ranked false positive types. Best viewed in digital form for zooming in. **Top row:** our baseline which is the *original candidate box* only model. **Bottom row:** our overall system. Because of space limitations, we present the graphs only for the classes boat and bottle.

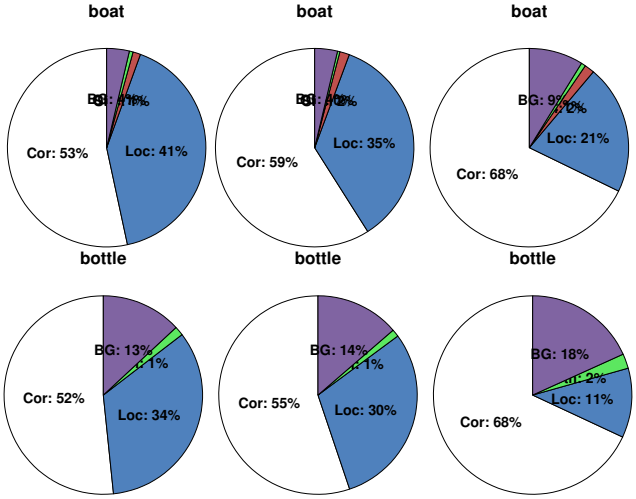


Figure 6: Fraction of top N detections (N=num of objs in category) that are correct (Cor), or false positives due to poor localization (Loc), confusion with similar objects (Sim), confusion with other VOC objects (Oth), or confusion with background or unlabelled objects (BG). **Left column:** our baseline which is the *original candidate box* only model. **Middle column:** Multi-Region CNN model without the semantic segmentation aware CNN features. **Right column:** our overall system. Because of space limitations, we present the pie charts only for the classes boat and bottle.

Region CNN model instead of the *Original Candidate Box* region alone, a considerable reduction in the percentage of false positives due to bad localization is achieved. This validates our argument that focusing on multiple regions of an object increases the localization sensitivity of our model. An even greater reduction of the false positives due to bad localization is achieved when our recognition model is integrated in the localization module developed for it. A similar observation can be made from Figure 5 where we plot the top-ranked false positive types of the baseline and of our overall proposed system.

6.3. Results on PASCAL VOC2012

In Table 4, we compare our detection system against other published work on the test set of PASCAL VOC2012 [7]. Our overall system involves the Multi-Region CNN model enriched with the semantic segmentation aware CNN features and coupled with the CNN based bounding box regression under the iterative localization scheme. We trained it on three different training sets and report results for each case: a) VOC2007 train+val set, b) VOC2012 train+val set, and c) VOC2007+2012 train+val sets. As we observe from Table 4, we achieve excellent mAP (69.1%, 70.7%, and 73.9% correspondingly) in all cases setting the new state-of-the-art on this test set. More detailed experiments are presented in section 7 of our technical report [9].

More experiments that demonstrate the localization sensitivity of our model are presented in section 7.3 of technical report [9].

7. Conclusions

We proposed a rich CNN-based representation for object detection that relies on two key factors: (i) the diversification of the discriminative appearance factors that it captures by steering its focus on different regions of the object, and (ii) the encoding of semantic segmentation-aware features. By using it in the context of a CNN-based localization refinement scheme, we showed that it achieves excellent results that surpass the state-of-the art by a significant margin.

References

[1] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 2007. 1

[2] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv preprint arXiv:1412.1283*, 2014. 6

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005. 1, 2

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009. 6

[5] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In *Computer Vision–ECCV 2014*. 2014. 4

[6] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc 2007) results (2007), 2008. 2, 4, 6

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012, 2012. 2, 6, 8

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010. 1, 6

[9] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. *arXiv preprint arXiv:1505.01749*, 2015. 8

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014. 1, 2, 3, 4, 5, 6, 8

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014*. 2014. 4

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv:1406.4729*, 2014. 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. 2

[14] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 1

[15] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *Computer Vision–ECCV 2012*. 2012. 7

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 1, 2

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989. 1

[19] M. Leordeanu, A. Radu, and R. Sukthankar. Features in concert: Discriminative feature selection meets unsupervised clustering. *arXiv preprint arXiv:1411.7714*, 2014. 2

[20] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 8

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 4, 5, 6

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004. 1

[23] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014. 4

[24] 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. 1

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2, 6

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 2

[27] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011. 1, 5, 6

[28] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2009. 1

[29] Z. Yuting, S. Kihyuk, V. Ruben, P. Gang, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. *arXiv preprint arXiv:1504.03293*, 2015. 7, 8

[30] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*. 2014. 2

[31] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. *arXiv preprint arXiv:1502.04275*, 2015. 2