

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



Summary



- **Introduction**

- related work
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- Add localization scheme
- implementation details
- experimental results
- Qualitative results and conclude
- Samples

Introduction

- improving on two key aspects

- **object representation**

- Deeper

- **Wider**

- **multi-region CNN** : capture several different aspects of an object : pure appearance、 object parts、 context appearance and so on
- **Another CNN model Component** : Captures semantic segmentation information because of connection exists between segmentation and detection


- **object localization**

- **a more powerful localization system**

- Combine multi-region CNN model with a CNN-model for bounding box regression
- **an iterative scheme** that alternates between scoring candidate boxes and refining their coordinates.



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.



Introduction - contributions.

- ▶ a multi-region CNN recognition model yields an enriched object representation
- ▶ propose a unified neural network architecture learn semantic segmentation-aware CNN features
- ▶ significantly improve the localization capability adopt a scheme that alternates between scoring candidate boxes and refining their locations
- ▶ Our detection system achieves mAP of 78.2% and 73.9% on VOC2007



Summary



- Introduction
- **related work**
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- Add localization scheme
- implementation details
- experimental results
- Qualitative results and conclude
- Samples



related work

- Caffe <https://github.com/gidariss/mrcnn-object-detection>
- we use multiple regions designed to diversify the appearance factors captured by our representation and to improve localization, we exploit CNN-based semantic segmentation-aware features (integrated in a unified neural network architecture), and make use of a deep CNN model for bounding box regression, as well as a box-voting scheme after nonmax-suppression.
- 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 3.1, helps on boosting the detection performance.



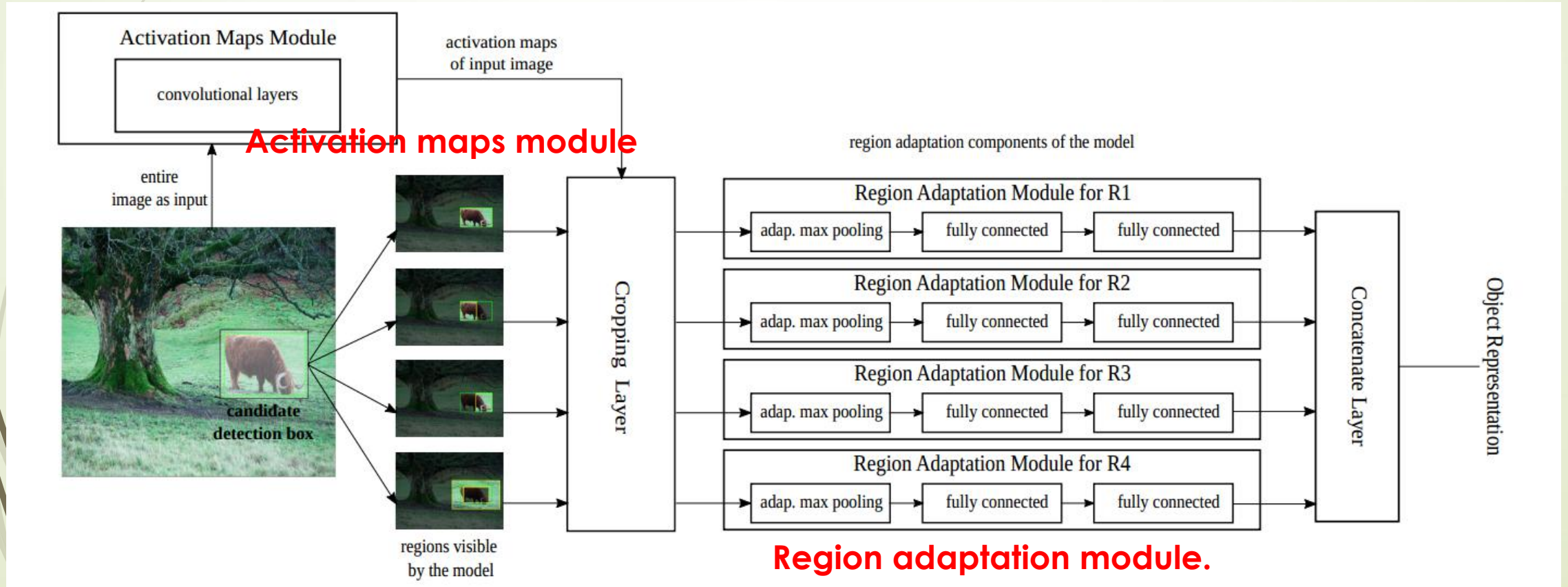
Summary



- Introduction
- related work
- **multi-region CNN model**
- Add semantic segmentation-aware CNN features
- Add localization scheme
- implementation details
- experimental results
- Qualitative results and conclude
- Samples

Multi-Region CNN Model

- our aim is:
 - (i) leading to a much richer and more robust object representation
 - (ii) to make the resulting representation more sensitive to inaccurate localization



Region components and their role on detection



Figure 3: Illustration of the regions used on 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 on 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 on detection

- Discriminative feature diversification
 - 手动强制网络可见区域，可增加物体识别所用特征的多样性。（意译）
 - 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. On 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.



Role on detection

- ▶ Localization-aware representation.
 - ▶ 增加了对物体不同位置的感知能力。（意译）
 - ▶ We observe that, by using the Multi-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. Furthermore, when our recognition model is integrated on the localization module developed for it, the reduction of false positives due to bad localization is huge. A similar observation can be deducted from figure 7 where we plot the top-ranked false positive types of the baseline and of our overall proposed system.



Summary



- Introduction
- related work
- multi-region CNN model
- **Add semantic segmentation-aware CNN features**
- Add localization scheme
- implementation details
- experimental results
- Qualitative results and conclude
- Samples

semantic segmentation-aware CNN features

- segmentation related cues are empirically known to often help object detection

Activation maps module for semantic segmentation-aware features.

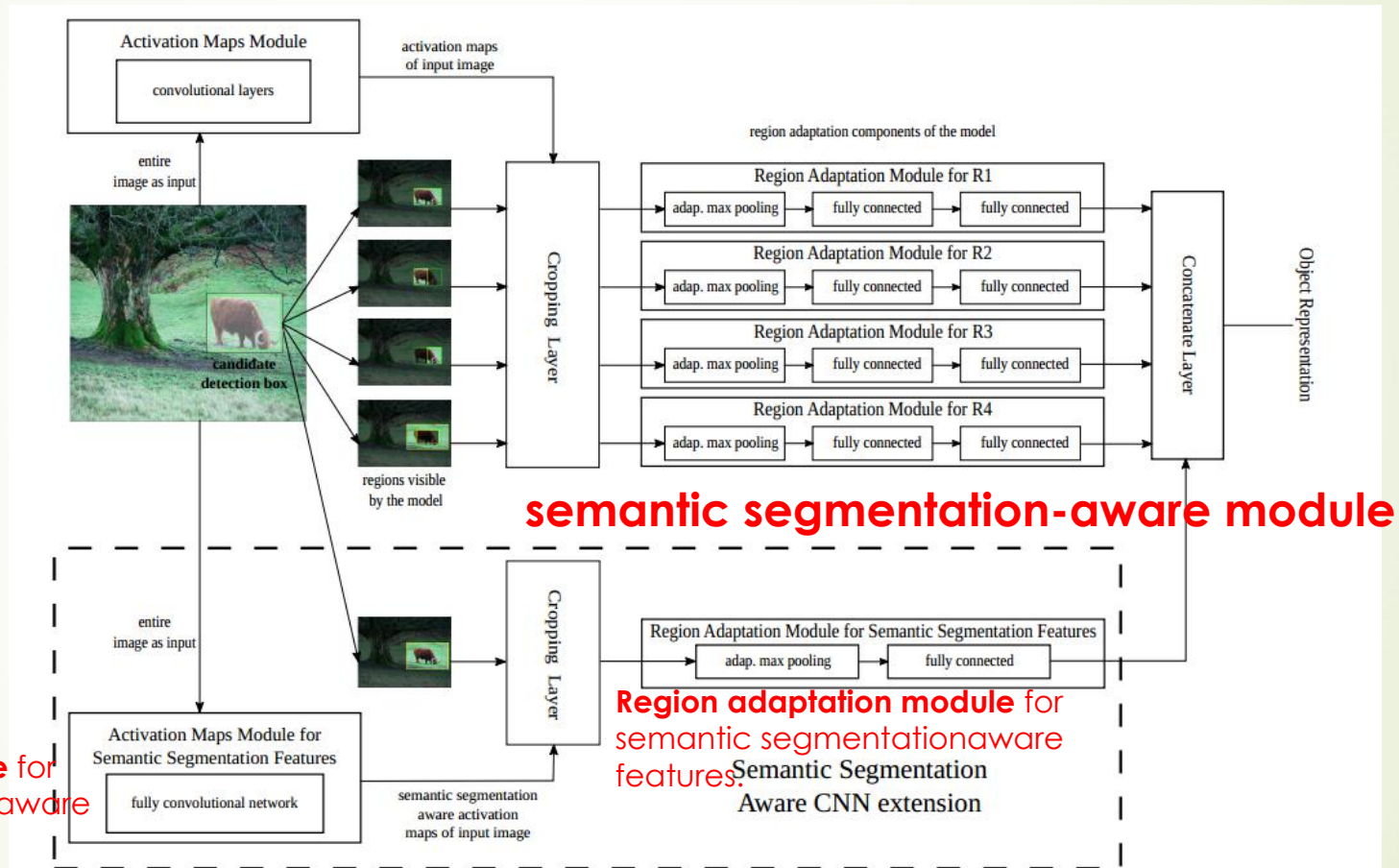


Figure 4: Multi Region CNN architecture extended with the semantic segmentation-aware CNN features.

Activation maps module for semantic segmentation-aware features

- Fully Convolutional Nets.
- Weakly Supervised Training.
- Activation Maps.

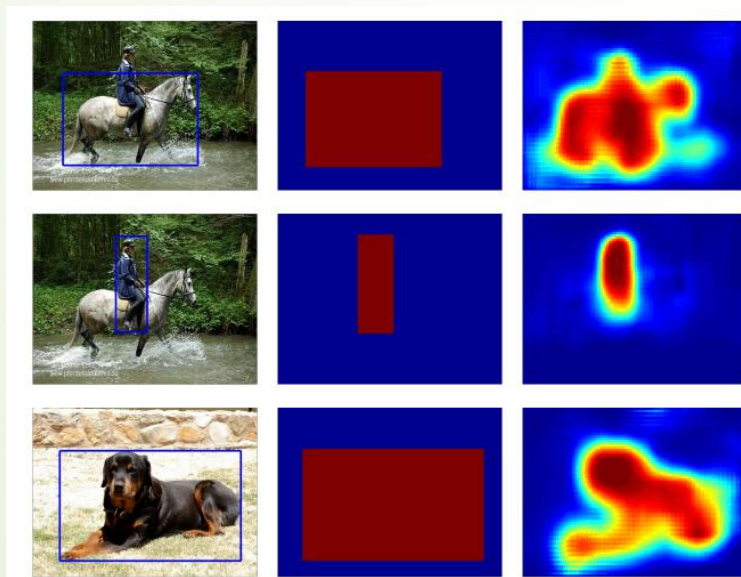
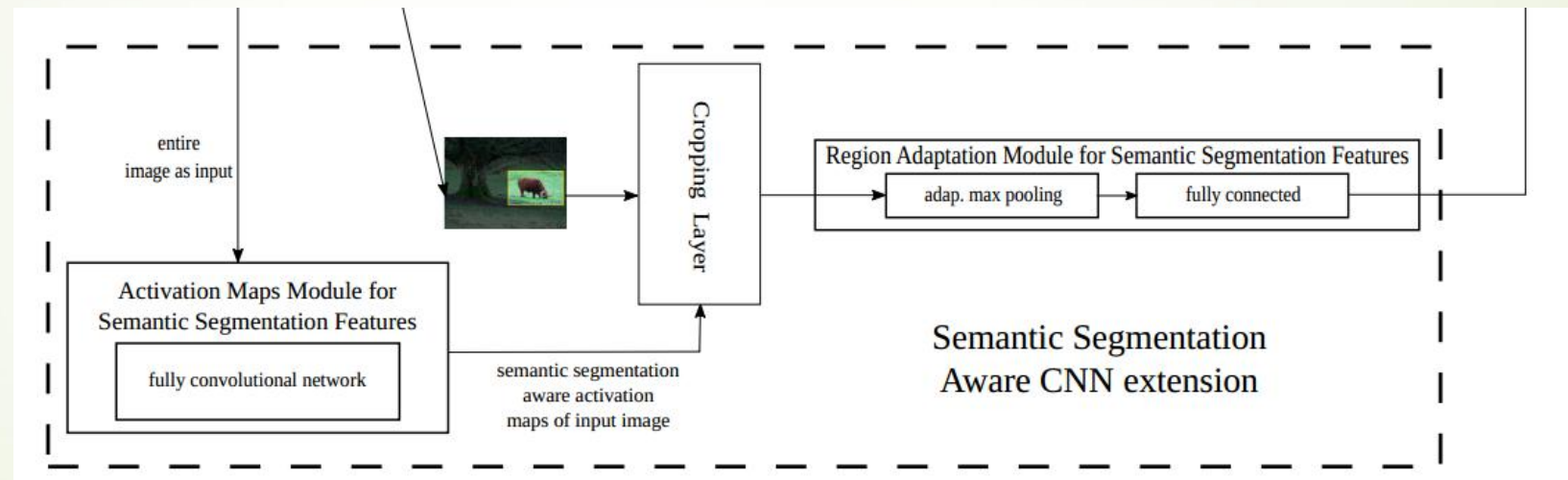


Figure 5: Illustration of the weakly supervised training of the FCN [23] used as activation maps module for the semantic segmentation aware CNN features. **Left column:** images with the ground truth bounding boxes drawn on them. The classes depicted from top to down order are horse, human, and dog. **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.

Region adaptation module for semantic segmentation-aware features

- 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).





Summary

- Introduction
- related work
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- **Add localization scheme**
- implementation details
- experimental results
- Qualitative results and conclude
- Samples



Object Localization



- CNN region adaptation module for bounding box regression.
 - applied **on top of** the activation maps produced from **the Multi-Region CNN** model
 - consists **of two hidden fully connected layers** and **one prediction** layer that outputs **4 values** (i.e., a bounding box) **per category**.
 - use as region a box obtained by **enlarging** the candidate box by a factor of **1.3**.
- Iterative Localization.
- Bounding box voting.

Iterative Localization

➤ Def: 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

➤ 第t次循环, 所有类为C的Bounding Boxes

➤ Init: \mathbf{B}_c^0 are coming from selective search are common between all the classes.

➤ For each class C :

➤ For each iteration $t = 1, \dots, T$:

$s_{i,c}^t = \mathcal{F}_{rec}(B_{i,c}^{t-1}|c, X)$ by our recognition model

$B_{i,c}^t = \mathcal{F}_{reg}(B_{i,c}^{t-1}|c, X)$ by our CNN regression

thus $\mathbf{D}_c^t = \{(s_{i,c}^t, B_{i,c}^t)\}_{i=1}^{N_{c,t}}$.

➤ Obtain $\{\mathbf{D}_c^t\}_{t=1}^T$

➤ Merge together $\hat{\mathbf{D}}_c = \cup_{t=1}^{\hat{T}} \mathbf{D}_c^t$.

➤ standard non-max suppression applied on $\hat{\mathbf{D}}_c$ produces the detections

$$\mathbf{Y}_c = \{(s_{i,c}, B_{i,c})\}$$

Bounding box voting

➤ For each $B_{i,c}$ in $\mathbf{Y}_c = \{(s_{i,c}, B_{i,c})\}$

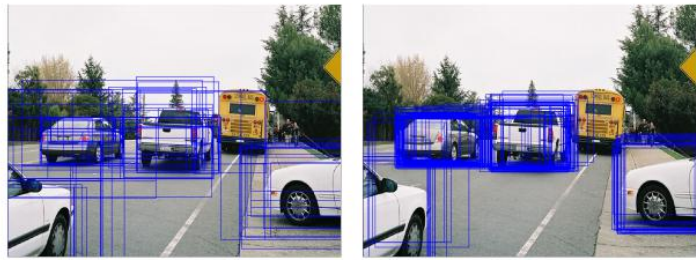
➤ Find $B_{j,c} \in \mathcal{N}(B_{i,c})$ the set of boxes in \mathbf{D}_c that overlap with $B_{i,c}$ by more than 0.5 on IoU metric

➤ Weight by S

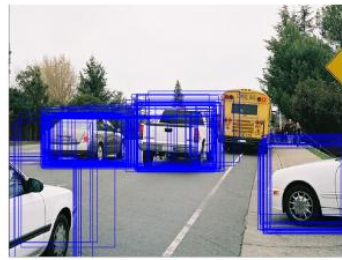
$$B'_{i,c} = \frac{\sum_{j: B_{j,c} \in \mathcal{N}(B_{i,c})} w_{j,c} \cdot B_{j,c}}{\sum_{j: B_{j,c} \in \mathcal{N}(B_{i,c})} w_{j,c}}.$$

$$w_{j,c} = \max(0, s_{j,c})$$

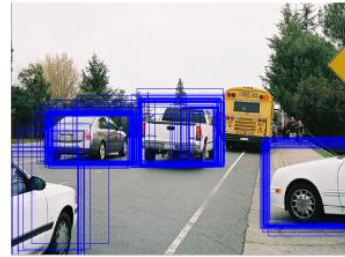
➤ Final \mathbf{Y}' The final set of object detections for class c will be $\mathbf{Y}'_c = \{(s_{i,c}, B'_{i,c})\}$.



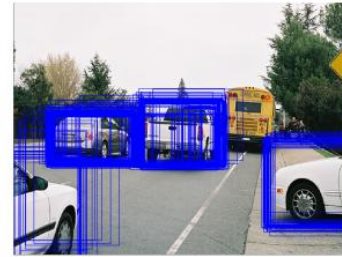
(a) Step 1



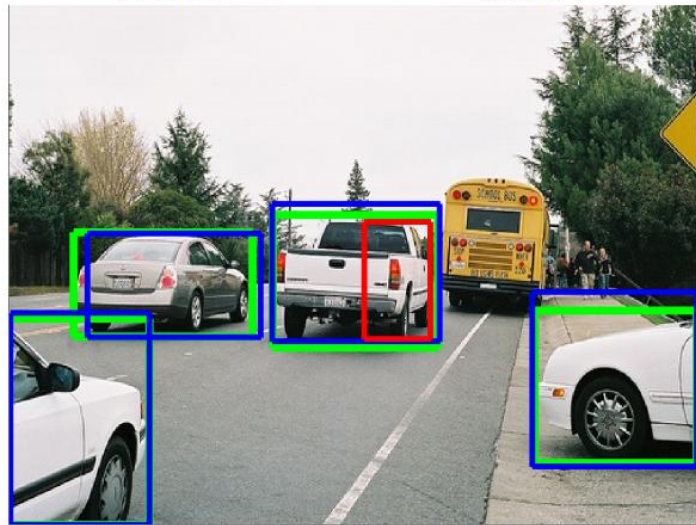
(b) Step 2



(c) Step 3



(d) Step 4



(e) Step 5

Figure 6: Illustration of the object localization scheme for instances of the class car. We describe the images from left to right and top to down order. **Step 1:** the initial box proposal of the image. For clarity we visualize only the box proposals that are not rejected after the first scoring step. **Step 2:** the new box locations obtained after performing CNN based bounding box regression on the boxes of Step 1. **Step 3:** the boxes obtained after a second step of box scoring and regressing on the boxes of Step 2. **Step 4:** the boxes of Step 2 and Step 3 merged together. **Step 5:** the detected boxes after applying non-maximum-suppression and box voting on the boxes of Step 4. On the final detections we use blue color for the true positives and red color for the false positives. Also, the ground truth bounding boxes are drawn with green color. The false positive that we see after the last step is a duplicate detection that survived from non-maximum-suppression.



Summary



- Introduction
- related work
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- Add localization scheme
- **implementation details**
- experimental results
- Qualitative results and conclude
- Samples



Implementation Details



- Pretrained Vgg-16 , Only Transfer Learning on fc6、 fc7 (all)
- Multi-Region CNN model:
 - 16-layers VGG-Net that outputs 512 feature channels.
 - 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
 - fine-tuned separately from the others.
 - follow the guidelines of R-CNN

Implementation Details

- ▶ Activation maps module for semantic segmentation aware features.
 - ▶ 16-layers VGG-Net without the last classification layer
 - ▶ transformed to a FCN
 - ▶ reshaping the fc6 and fc7 full connected layers to convolutional ones with kernel size of 7×7 and 1×1 correspondingly
 - ▶ Channel fc7 = 512
 - ▶ auxiliary fc8
 - ▶ kernel size 1×1 ,
 - ▶ outputs as many channels as our classes
 - ▶ a binary (foreground vs background) logistic loss applied on each spatial cell



Implementation Details



- Region adaptation module for semantic segmentation aware features.
 - spatially adaptive max-pooling layer
 - 512 channels on a 9×9 grid
 - a fully connected layer with 2096 channels.
- Classification SVMs.
 - Same as RCNN
- CNN region adaptation module for bounding box regression.
 - Multi-Region CNN model as input
 - target values are defined the same way as in R-CNN
 - Loss : euclidean distance



Implementation Details

- Multi-Scale Implementation.
 - Multi-Region CNN model:
 - 7 scales {480, 576, 688, 874, 1200, 1600, 2100}
 - For training : region adaptation modules are applied on a random scale
 - For testing : a single scale is used such that the area of the scaled region is closest to 224×224
 - Semantic Segmentation-Aware CNN model:
 - {576, 874, 1200}, closest to 288×288
 - Bounding Box Regression CNN model:
 - {480, 576, 688, 874, 1200, 1600, 2100}. Both training and testing closest 224×224 pixels



Summary



- Introduction
- related work
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- Add localization scheme
- implementation details
- **experimental results**
- Qualitative results and conclude
- Samples

experimental results

Adaptation Modules	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
<i>Original Box fig. 3a</i>	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
<i>Left Half Box fig. 3b</i>	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
<i>Right Half Box fig. 3c</i>	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
<i>Up Half Box fig. 3d</i>	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
<i>Bottom Half Box fig. 3e</i>	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
<i>Central Region fig. 3f</i>	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
<i>Central Region fig. 3g</i>	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
<i>Border Region fig. 3h</i>	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
<i>Border Region fig. 3i</i>	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
<i>Contextual Region fig. 3j</i>	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
<i>Semantic-aware region.</i>	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
<i>R-CNN with VGG-Net</i>	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
<i>R-CNN with VGG-Net & bbox reg.</i>	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
<i>Best approach of [36]</i>	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
<i>Best approach of [36] & bbox reg.</i>	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
<i>Original Box fig. 3a</i>	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
<i>MR-CNN</i>	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
<i>MR-CNN & S-CNN</i>	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
<i>MR-CNN & S-CNN & Loc.</i>	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

Table 2: Detection performance of our modules on VOC2007 test set. Each model was trained on VOC2007 train+val set.

experimental results

Approach	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
<i>Original candidate box-Baseline</i>	0.7543	0.7325	0.6634	0.5816	0.5775	0.7109	0.7390	0.7277	0.5718	0.7112	0.6007	0.7000	0.7039	0.7194	0.6607	0.5339	0.6855	0.6461	0.6903	0.7359
<i>MR-CNN</i>	0.7938	0.7864	0.7180	0.6424	0.6222	0.7609	0.7918	0.7758	0.6186	0.7483	0.6802	0.7448	0.7562	0.7569	0.7166	0.5753	0.7268	0.7148	0.7391	0.7556

Table 4: Correlation between the IoU overlap of selective search box proposals [34] (with the closest ground truth bounding box) and the scores assigned to them.

- Box proposal 越好，最后score越高。高相关性。

Approach	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
<i>Original candidate box-Baseline</i>	0.9327	0.9324	0.9089	0.8594	0.8570	0.9389	0.9455	0.9250	0.8603	0.9237	0.8806	0.9209	0.9263	0.9317	0.9151	0.8415	0.8932	0.9060	0.9241	0.9125
<i>MR-CNN</i>	0.9462	0.9479	0.9282	0.8843	0.8740	0.9498	0.9593	0.9355	0.8790	0.9338	0.9127	0.9358	0.9393	0.9440	0.9341	0.8607	0.9120	0.9314	0.9413	0.9210

Table 5: The Area-Under-Curve (AUC) measure for the well-localized box proposals against the mis-localized box proposals.

- Box proposal 越好，最后物体识别率越高。

experimental results

➡ 更多数据更好

Approach	trained on	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
<i>R-CNN [10] with VGG-Net & bbox reg.</i>	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
<i>Network In Network [21]</i>	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
<i>Best approach of [36] & bbox reg.</i>	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
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	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
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	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

Table 6: Comparative results on VOC 2012 test set.

Approach	trained on	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	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
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	VOC07	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
<i>Faster R-CNN [28]</i>	VOC07+12	0.765	0.790	0.709	0.655	0.521	0.831	0.847	0.864	0.520	0.819	0.657	0.848	0.846	0.775	0.767	0.388	0.736	0.739	0.830	0.726	0.732
<i>NoC [29]</i>	VOC07+12	0.763	0.814	0.744	0.617	0.608	0.847	0.782	0.829	0.530	0.792	0.692	0.832	0.832	0.785	0.680	0.450	0.716	0.767	0.822	0.757	0.733
<i>Fast R-CNN [9]</i>	VOC07+12	0.770	0.781	0.693	0.594	0.383	0.816	0.786	0.867	0.428	0.788	0.689	0.847	0.820	0.766	0.699	0.318	0.701	0.748	0.804	0.704	0.700

Table 7: Comparative results on VOC 2007 test set for models trained with extra data.

Approach	trained on	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	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
<i>MR-CNN & S-CNN & Loc. (Ours)</i>	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
<i>Faster R-CNN [28]</i>	VOC07+12	0.849	0.798	0.743	0.539	0.498	0.775	0.759	0.885	0.456	0.771	0.553	0.869	0.817	0.809	0.796	0.401	0.726	0.609	0.812	0.615	0.704
<i>Fast R-CNN & YOLO [27]</i>	VOC07+12	0.830	0.785	0.737	0.558	0.431	0.783	0.730	0.892	0.491	0.743	0.566	0.872	0.805	0.805	0.747	0.421	0.708	0.683	0.815	0.670	0.704
<i>Deep Ensemble COCO [11]</i>	VOC07+12, COCO [22]	0.840	0.794	0.716	0.519	0.511	0.741	0.721	0.886	0.483	0.734	0.578	0.861	0.800	0.807	0.704	0.466	0.696	0.688	0.759	0.714	0.701
<i>NoC [29]</i>	VOC07+12	0.828	0.790	0.716	0.523	0.537	0.741	0.690	0.849	0.469	0.743	0.531	0.850	0.813	0.795	0.722	0.389	0.724	0.595	0.767	0.681	0.688
<i>Fast R-CNN [9]</i>	VOC07+12	0.823	0.784	0.708	0.523	0.387	0.778	0.716	0.893	0.442	0.730	0.550	0.875	0.805	0.808	0.720	0.351	0.683	0.657	0.804	0.642	0.684

Table 8: Comparative results on VOC 2012 test set for models trained with extra data.



Summary

- Introduction
- related work
- multi-region CNN model
- Add semantic segmentation-aware CNN features
- Add localization scheme
- implementation details
- experimental results
- **Qualitative results and conclude**
- Samples



Qualitative results & Conclusions

- Failure cases.
 - 连接物比较难处理
- Missing annotations.
 - 找到了样本中漏标的东西
- Conclude:
 - we show that it achieves excellent results that surpass the state-of-the art by a significant margin.