Rich feature hierarchies for accurate object detection and semantic segmentation

Abstract

a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30%

two key insights

one can apply high-capacity convolutional neural networks to bottom-up region proposals in order to localize and segment objects

when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost

Introduction

CNN can lead to dramatically higher object detection performance

2 problems

localizing objects with a deep network

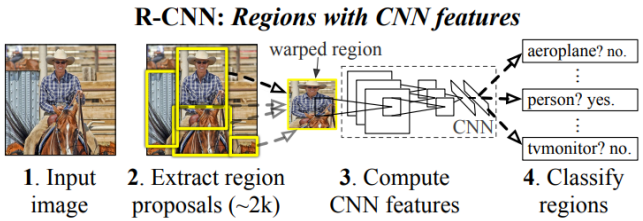
training a high-capacity model with only a small quantity of annotated detection data.

Unlike image classification, detection requires localizing (likely many) objects within an image.

recognition using regions" paradigm

2000 category-independent region proposals for the input image,

a fixed-length feature vector from each proposal using a CNN, and then classifies each region with category-specific linear SVM



A second challenge faced in detection is that labeled data is scarce and the amount currently available is insufficient for training a large CNN

supervised pre-training on a large auxiliary dataset followed by domainspecific fine-tuning on a small dataset

- effective paradigm for learning high-capacity CNNs when data is scarce

Our system is also quite efficient

a reasonably small matrix-vector product and greedy non-maximum suppression

because R-CNN operates on regions it is natural to extend it to the task of semantic segmentation

Object detection with R-CNN

three modules

1 generates category-independent region proposals

2 a large convolutional neural network

3 a set of class-specific linear SVMs

Module design

Region proposals

While R-CNN is agnostic to the particular region proposal method, we use selective search to enable a controlled comparison with prior detection work

Feature extraction

Figure 2: Warped training samples from VOC 2007 train.

dilate the tight bounding box
warp all pixels in a tight bounding box around it to the required size

Test-time detection

selective search on the test image to extract around 2000 region proposals
We warp each proposal and forward propagate it through the CNN in order to compute features
for each class, we score each extracted feature vector using the SVM trained for that class
greedy non-maximum suppression

Run-time analysis

Two properties make detection efficient.
1 all CNN parameters are shared across all categories.
2 the feature vectors computed by the CNN are low-dimensional when compared to other common approaches, such as spatial pyramids with bag-of-visual-word encodings

The result of such sharing is that the time spent computing region proposals and features (13s/image on a GPU or 53s/image on a CPU) is amortized over all classes

Training

Supervised pre-training.

pre-trained the CNN on a large auxiliary dataset using image-level annotations only

Domain-specific fine-tuning.

To adapt our CNN to the new task (detection) and the new domain (warped proposal windows)
stochastic gradient descent (SGD) training of the CNN parameters using only warped region proposals

Object category classifiers

IoU overlap threshold

We found that selecting this threshold carefully is important

Once features are extracted and training labels are applied, we optimize one linear SVM per class

Results on PASCAL VOC 2010-12

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in Section C. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

Results on ILSVRC2013 detection

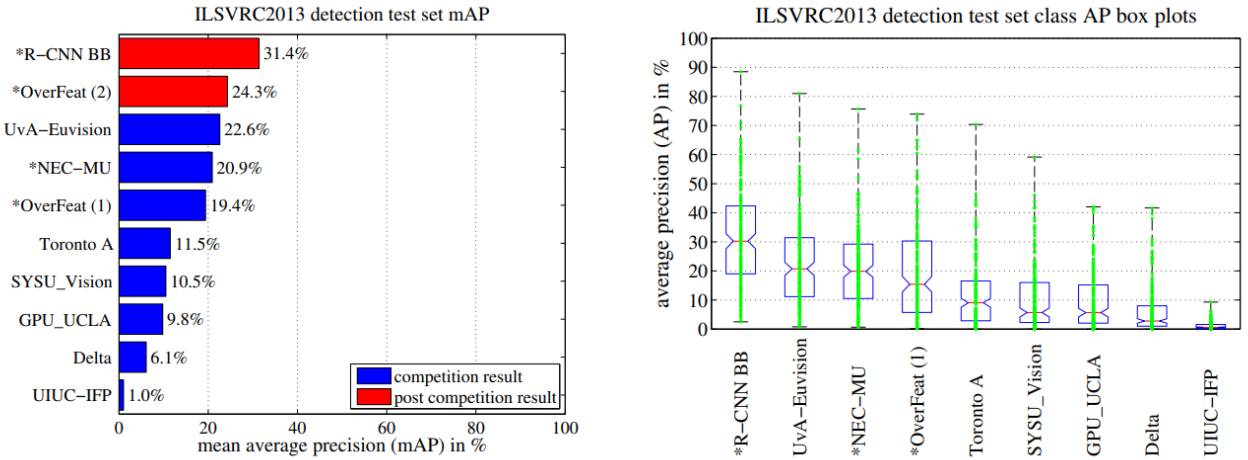


Figure 3: (Left) Mean average precision on the ILSVRC2013 detection test set. Methods preceeded by * use outside training data (images and labels from the ILSVRC classification dataset in all cases). **(Right) Box plots for the 200 average precision values per method.** A box plot for the post-competition OverFeat result is not shown because per-class APs are not yet available (per-class APs for R-CNN are in Table 8 and also included in the tech report source uploaded to arXiv.org; see `R-CNN-ILSVRC2013-APs.txt`). The red line marks the median AP, the box bottom and top are the 25th and 75th percentiles. The whiskers extend to the min and max AP of each method. Each AP is plotted as a green dot over the whiskers (best viewed digitally with zoom).

Visualization, ablation, and modes of error

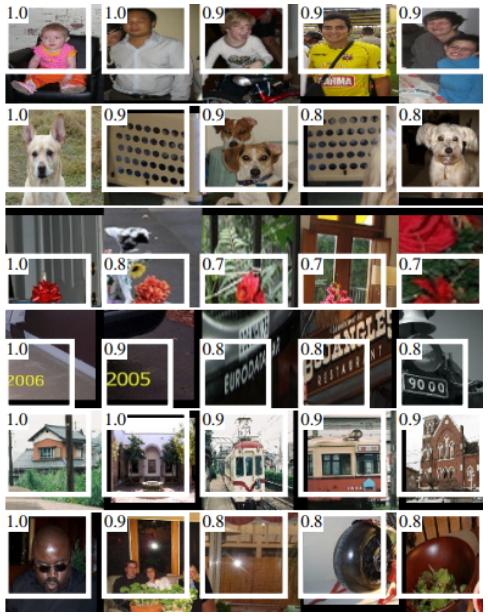
Visualizing learned features

Understanding the subsequent layers is more challenging

We propose a simple (and complementary) non-parametric method that directly shows what the network learned.

single out a particular unit (feature) in the network and use it as if it were an object detector in its own right

We avoid averaging in order to see different visual modes and gain insight into the invariances computed by the unit.



Ablation studies

Performance layer-by-layer, without fine-tuning

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN $pool_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc_6	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc_7	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7

Much of the CNN's representational power comes from its convolutional layers, rather than from the much larger densely connected layers.

Performance layer-by-layer, with fine-tuning

The boost from fine-tuning is much larger for fc6 and fc7 than for pool5

which suggests that the pool5 features learned from ImageNet are general and that most of the improvement is gained from learning domain-specific non-linear classifiers on top of them

R-CNN FT $pool_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc_6	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc_7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2

Comparison to recent feature learning methods

DPM ST

augments HOG features with histograms of “sketch token” probabilities

DPM HSC

replaces HOG with histograms of sparse codes (HSC)

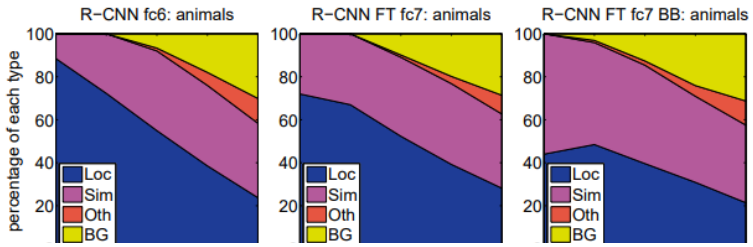
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

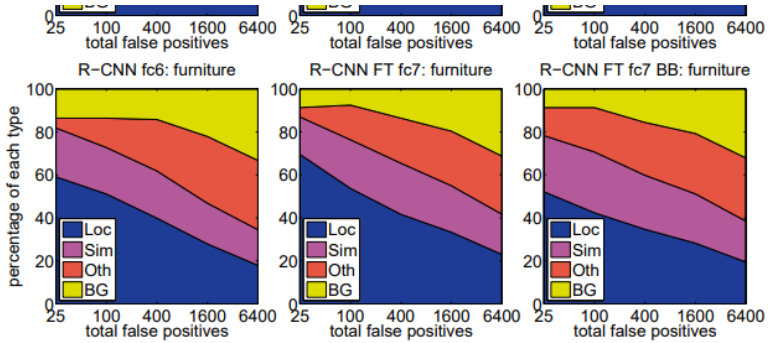
Network architectures

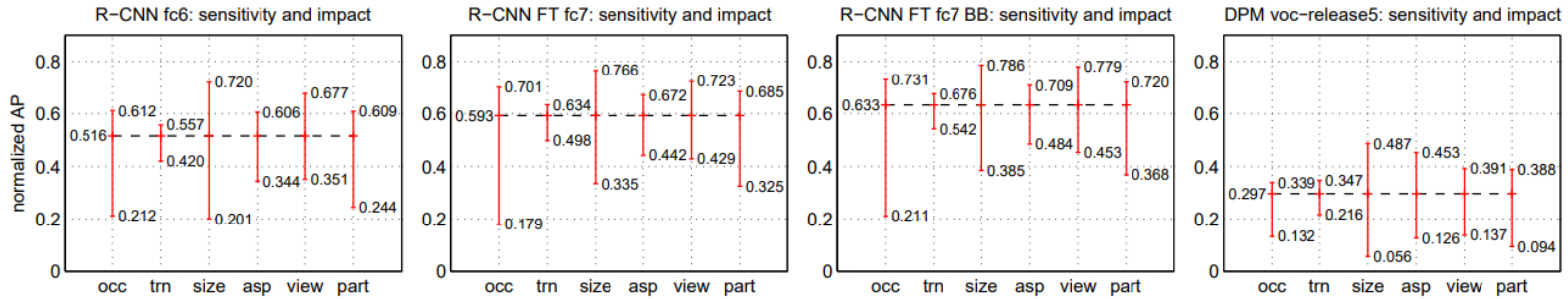
VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	**35.7**	62.1	64.0	66.5	**71.2**	62.2
R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

However there is a considerable drawback in terms of compute time, with the forward pass of O-Net taking roughly 7 times longer than T-Net.

Detection error analysis







Bounding-box regression

this simple approach fixes a large number of mislocalized detections

Qualitative results

The ILSVRC2013 detection dataset

Dataset overview

rely heavily on the val set and use some of the train images as an auxiliary source of positive examples

To use val for both training and validation, we split it into roughly equally sized “val1” and “val2” sets

approximately class-balanced partition

Region proposals

selective search is not scale invariant and so the number of regions produced depends on the image resolution

Training data

Training data is required for three procedures in R-CNN

(1) CNN fine-tuning, (2) detector SVM training, and (3) bounding-box regressor training

Validation and evaluation

Ablation study

test set	val_2	val_2	val_2	val_2	val_2	val_2	test	test
SVM training set	val_1	$\text{val}_1+\text{train}_{.5\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}+\text{train}_{1\text{k}}$	$\text{val}+\text{train}_{1\text{k}}$
CNN fine-tuning set	n/a	n/a	n/a	val_1	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$	$\text{val}_1+\text{train}_{1\text{k}}$
bbox reg set	n/a	n/a	n/a	n/a	n/a	val_1	n/a	val
CNN feature layer	fc_6	fc_6	fc_6	fc_7	fc_7	fc_7	fc_7	fc_7
mAP	20.9	24.1	24.1	26.5	29.7	**31.0**	30.2	**31.4**
median AP	17.7	21.0	21.4	24.8	29.2	**29.6**	29.0	**30.3**

Relationship to OverFeat

Semantic segmentation

CNN features for segmentation.

The first strategy (full) ignores the region's shape and computes CNN features directly on the warped window, exactly as we did for detection

second strategy (fg) computes CNN features only on a region's foreground mask - replace the background with the mean input so that background regions are zero after mean subtraction

The third strategy (full+fg) simply concatenates the full and fg features; our experiments validate their complementarity

Results on VOC 2011

	full R-CNN		*fg* R-CNN		*full+fg* R-CNN	
O_2P [4]	fc_6	fc_7	fc_6	fc_7	fc_6	fc_7
46.4	43.0	42.5	43.7	42.1	**47.9**	45.8

VOC 2011 test	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
R&P [2]	83.4	46.8	18.9	36.6	31.2	42.7	57.3	47.4	44.1	8.1	39.4	**36.1**	36.3	49.5	48.3	50.7	26.3	47.2	22.1	42.0	43.2	40.8
O_2P [4]	**85.4**	**69.7**	22.3	45.2	**44.4**	46.9	66.7	57.8	56.2	**13.5**	**46.1**	32.3	41.2	**59.1**	55.3	51.0	**36.2**	50.4	**27.8**	46.9	**44.6**	47.6
ours (*full+fg* R-CNN fc_6)	84.2	66.9	**23.7**	**58.3**	37.4	**55.4**	**73.3**	**58.7**	**56.5**	9.7	45.5	29.5	**49.3**	40.1	**57.8**	**53.9**	33.8	**60.7**	22.7	**47.1**	41.3	**47.9**

Conclusion

In recent years, object detection performance had stagnated.

presents a simple and scalable object detection algorithm that gives a 30% relative improvement over the best previous results on PASCAL VOC 2012.

two insights.

The first is to apply high-capacity convolutional neural networks to bottom-up region proposals in order to localize and segment objects.

The second is a paradigm for training large CNNs when labeled training data is scarce

We show that it is highly effective to pre-train the network— with supervision—for a auxiliary task with abundant data (image classification) and then to fine-tune the network for the target task where data is scarce (detection).