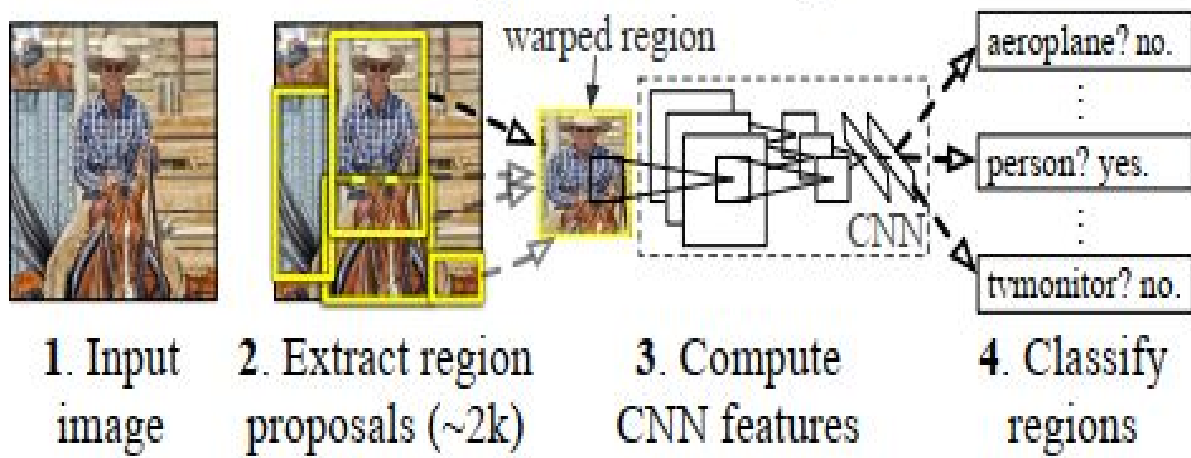


R-CNN: *Regions with CNN features*



RCNN

22.10.2014

Author

Ross Girshick

Jeff Donahue

Trevor Darrell

Jitendra Malik

Overview

CNNs saw heavy use in the 1990s, but then fell out of fashion with the rise of support vector machines in 2012. The central issue can be distilled to the following: **To what extent do the CNN classification results on ImageNet generalize to object detection results on the PASCAL VOC Challenge?** We answer this question by bridging the gap between image classification and object detection. This paper is the first to show that a CNN can lead to dramatically higher object detection performance on PASCAL VOC as compared to systems based on simpler HOG-like features. To achieve this result, we focused on two problems: localizing objects with a deep network and training a high-capacity model with only a small quantity of annotated detection data.

Unlike image classification, detection requires localizing objects within an image.

One approach frames localization as a regression problem. However, this indicates that this strategy may not fare well in practice; they report a mAP of 30.5% on VOC 2007 compared to the 58.5% achieved by RCNN method.

An alternative is to build a sliding-window detector. CNNs have been used in this way for at least two decades.

Instead, authors solve the CNN localization problem by operating within the **“recognition using regions”** paradigm, which has been successful for both object detection and semantic segmentation. At test time, this method generates around 2000 category independent region proposals for the input image, extracts a fixed-length feature vector from each proposal using a CNN, and then classifies each region with category-specific linear SVMs. Since this system combines region proposals with CNNs, they named this technique as R-CNN (Regions with CNN).

Object detection with R-CNN

This object detection system consists of three modules.

1. The first generates category-independent region proposals. These proposals define the set of candidate detections available to detectors.
2. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region.
3. The third module is a set of class specific linear SVMs.

Module design

Region proposals. A variety of recent papers offer methods for generating category independent region proposals.

Feature extraction. Authors extract a 4096-dimensional feature vector from each region proposal using the Caffe implementation of the CNN. Features are computed by forward propagating a mean-subtracted 227×227 RGB image through five convolutional layers and two fully connected layers. Regardless of the size or aspect ratio of the candidate region, authors warp all pixels in a tight bounding box around it to the required size. Prior to warping, they dilate the light bounding box so that at the warped size there are exactly p pixels of warped image context around the original box ($p = 16$). Figure shows a random sampling of warped training regions.



Test-time detection

At test time, authors run selective search on the test image to extract around 2000 region proposals. they warp each proposal and forward propagate it through CNN in order to compute features. Then, for each class, score each extracted feature vector using the SVM trained for that class. Given all scored regions in an image, apply a greedy non-maximum suppression that rejects a region if it has an intersection-over union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

Run-time analysis. Two properties make detection efficient.

1. First, all CNN parameters are shared across all categories.
2. Second, the feature vectors computed by the CNN are low-dimensional when compared to other common approaches.

Advantage.

1. The result of such sharing is that the time spent computing region proposals and features is amortized over all classes. The only class-specific computations are dot products between features and SVM weights and non-maximum suppression.
2. This analysis shows that R-CNN can scale to thousands of object classes without resorting to approximate techniques, such as hashing. Even if there were 100k

classes, the resulting matrix multiplication takes only 10 seconds on a modern multi-core CPU.

Training

Supervised pre-training. Authors discriminatively pre-trained the CNN on a large auxiliary dataset (ILSVRC2012 classification) using image-level annotations only. Pre-training was performed using the open source Caffe CNN library. This discrepancy is due to simplifications in the training process.

Domain-specific fine-tuning. Authors continue stochastic gradient descent (SGD) training of the CNN parameters using only warped region proposals. Aside from replacing the CNN's ImageNetspecific 1000-way classification layer with a randomly initialized $(N + 1)$ -way classification layer (where N is the number of object classes, plus 1 for background), the CNN architecture is unchanged. For VOC, $N = 20$ and for ILSVRC2013, $N = 200$. We treat all region proposals with ≥ 0.5 IoU overlap with a ground-truth box as positives for that box's class and the rest as negatives. We start SGD at a learning rate of 0.001, which allows fine-tuning to make progress while not clobbering the initialization. In each SGD iteration, we uniformly sample 32 positive windows and 96 background windows to construct a mini-batch of size 128.

Object category classifiers. Consider training a binary classifier to detect cars. It's clear that an image region tightly enclosing a car should be a positive example. Similarly, it's clear that a background region, which has nothing to do with cars, should be a negative example.

Less clear is how to label a region that partially overlaps a car. Authors resolve this issue with an IoU overlap threshold, below which regions are defined as negatives. The overlap threshold, 0.3, was selected by a grid search over (0, 0.1, ..., 0.5) on a validation set.

Since the training data is too large to fit in memory, they adopt the standard hard negative mining method. Hard negative mining converges quickly and in practice mAP stops increasing after only a single pass over all images.

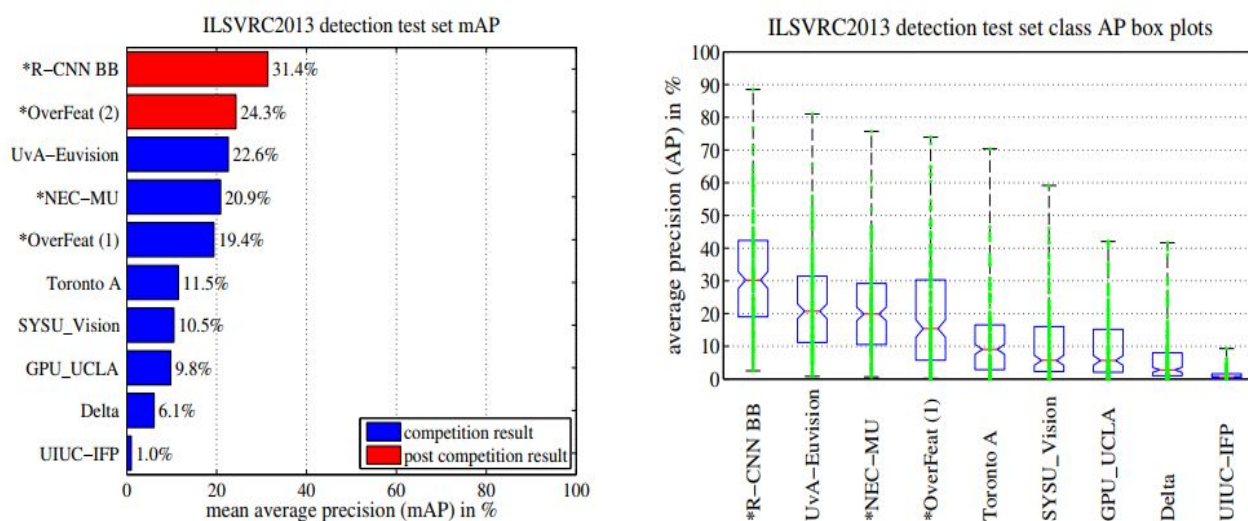
Results on PASCAL VOC 2010-12

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

RCNN achieved a large improvement in mAP, from 35.1% to 53.7% mAP, while also being much faster compared to other techniques at that time.

Results on ILSVRC2013 detection

R-CNN achieves a mAP of 31.4%, which is significantly ahead of the second-best result of 24.3% from OverFeat.




Visualization, ablation, and modes of error

Visualizing learned features

The idea is to single out a particular unit (feature) in the network and use it as if it were an object detector in its own right. That is, we compute the unit's activations on a large set of held-out region proposals (about 10 million), sort the proposals from highest to lowest activation, perform non maximum suppression, and then display the top-scoring regions. This method lets the selected unit “speak for itself” by showing exactly which inputs it fires on. 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. Authors start by looking at results from the CNN without fine-tuning on PASCAL, i.e. all CNN parameters were pre-trained on ILSVRC 2012 only. Analyzing performance layer-by-layer reveals that features from fc7 generalize worse than features from fc6. This means that 29%, or about 16.8 million, of CNN's parameters can be removed without degrading mAP. More surprising is that removing both fc7 and fc6 produces quite good results even though pool5 features are



computed using only 6% of the CNN's parameters. Much of CNN's representational power comes from its convolutional layers, rather than from the much larger densely connected layers. This finding suggests potential utility in computing a dense feature map, in the sense of HOG, of an arbitrary-sized image by using only the convolutional layers of the CNN. This representation would enable experimentation with sliding-window detectors, including DPM (Deformable Parts Model), on top of pool5 features.

Performance layer-by-layer, with fine-tuning. Now look at results from CNN after having fine-tuned its parameters on VOC 2007 trainval. The improvement is striking fine-tuning which increases mAP by 8.0 percentage points to 54.2%. 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.

Comparison to recent feature learning methods.

All R-CNN variants strongly outperform the three DPM baselines, including the two that use feature learning. Compared to the latest version of DPM, which uses only HOG features, RCNN mAP is more than 20 percentage points higher: 54.2% vs. 33.7%—a 61% relative improvement. The combination of HOG and sketch tokens yields 2.5 mAP points over HOG alone, while HSC improves over HOG by 4 mAP points. These methods achieve mAPs of 29.1% and 34.3%, respectively.

Network Architectures

Choice of architecture has a large effect on R-CNN detection performance. VGG16 network was one of the top performers in the ILSVRC 2014 classification challenge. The network has a homogeneous structure consisting of 13 layers of 3×3 convolution kernels, with five max pooling layers interspersed, and topped with three fully-connected layers. authors refer to this network as “**O-Net**” for OxfordNet and the baseline as “**T-Net**” for TorontoNet.

To use O-Net in R-CNN, authors downloaded the publicly available pre-trained network weights for the VGG ILSVRC 16 layers model from the Caffe Model Zoo. Then fine-tuned the network using the same protocol as used for T-Net. The only difference was to use smaller mini batches (24 examples) as required in order to fit within GPU memory. The results show that RCNN with O-Net substantially outperforms R-CNN with TNet, increasing mAP from 58.5% to 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.

Bounding-box regression

Based on the error analysis, Authors implemented a simple method to reduce localization errors. Inspired by the bounding-box regression employed in DPM, train a linear regression model to predict a new detection window given the pool5 features for a selective search region proposal.

The ILSVRC2013 detection dataset

Dataset overview

The ILSVRC2013 detection dataset is split into three sets: train (395,918), val (20,121), and test (40,152), where the number of images in each set is in parentheses. These images are scene-like and similar in complexity to PASCAL VOC images. The val and test splits are exhaustively annotated, meaning that in each image all instances from all 200 classes are labeled with bounding boxes. The train set, in contrast, is drawn from the ILSVRC2013 classification image distribution. These images have more variable complexity with a skew towards images of a single centered object. In any given train image, instances from the 200 classes may or may not be labeled. In addition to these image sets, each class has an extra set of negative images. The negative image sets were not used in this work.

Region proposals

Authors followed the same region proposal approach that was used for detection on PASCAL. One minor modification was required to deal with the fact that selective search is not scale invariant and so the number of regions produced depends on the image resolution. ILSVRC image sizes range from very small to a few that are several mega-pixels, and so they resize each image to a fixed width (500 pixels) before running selective search. On val, selective search resulted in an average of 2403 region proposals per image with a 91.6% recall of all ground-truth bounding boxes (at 0.5 IoU threshold). This recall is notably lower than in PASCAL, where it is approximately 98%, indicating significant room for improvement in the region proposal stage.

Training data

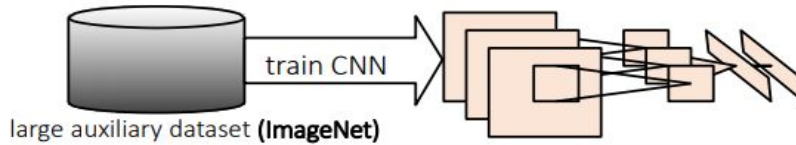
For training data, authors formed a set of images and boxes that includes all selective search and ground-truth boxes from val1 together with up to N ground-truth boxes per class from train. They'll call this dataset of images and boxes val1+trainN. Training data is required for three procedures in R-CNN:

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

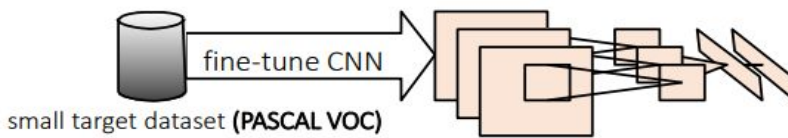
CNN fine-tuning was run for 50k SGD iteration on val1+trainN using the exact same settings as were used for PASCAL. Fine-tuning on a single NVIDIA Tesla K20 took 13 hours using Caffe. For SVM training, all ground-truth boxes from val1+trainN were used as positive

examples for their respective classes. Hard negative mining was performed on a randomly selected subset of 5000 images from val1. No negative examples were taken from the train because the annotations are not exhaustive.

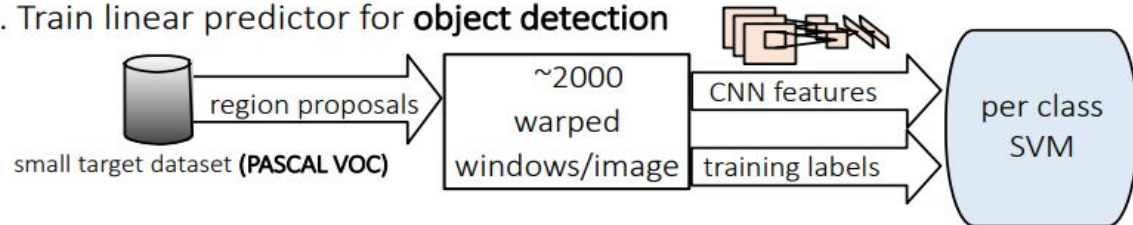
1. Pre-train CNN for **image classification**



2. Fine-tune CNN for **object detection**



3. Train linear predictor for **object detection**



		VOC 2007	VOC 2010
reference	DPM v5 (Girshick et al. 2011)	33.7%	29.6%
	UVA sel. search (Uijlings et al. 2012)		35.1%
	Regionlets (Wang et al. 2013)	41.7%	39.7%
pre-trained only	R-CNN pool ₅	44.2%	
	R-CNN fc ₆	46.2%	
	R-CNN fc ₇	44.7%	
fine-tuned	R-CNN pool ₅	47.3%	
	R-CNN fc ₆	53.1%	
	R-CNN fc ₇	54.2%	50.2%%
	R-CNN fc ₇ (Bounding Box regression)	58.5%	53.7%

Validation and evaluation

All system hyperparameters (e.g., SVM C hyperparameters, padding used in region warping, NMS thresholds, bounding-box regression hyperparameters) were fixed at the same values used for PASCAL. After selecting the best choices on val2, authors submitted exactly two result files to the ILSVRC2013 evaluation server.

The first submission was without bounding-box regression and

The second submission was with bounding-box regression.

Semantic segmentation

Region classification is a standard technique for semantic segmentation, allowing us to easily apply R-CNN to the PASCAL VOC segmentation challenge. O2P (second-order pooling) uses CPMC to generate 150 region proposals per image and then predicts the quality of each region, for each class, using support vector regression (SVR). The high performance of their approach is due to the quality of the CPMC regions and the powerful second-order pooling of multiple feature types.

CNN features for segmentation. Authors evaluate three strategies for computing features on CPMC regions, all of which begin by warping the rectangular window around the region to 227×227 .

The first strategy (full) ignores the region's shape and computes CNN features directly on the warped window, exactly as did for detection. However, these features ignore the non-rectangular shape of the region. Two regions might have very similar bounding boxes while having very little overlap.

The second strategy (fg) computes CNN features only on a region's foreground mask. We 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.

Results on VOC 2011.

Within each feature computation strategy, layer fc6 always outperforms fc7 and the following discussion refers to the fc6 features. The fg strategy slightly outperforms full, indicating that the masked region shape provides a stronger signal, matching our intuition. However, full+fg achieves an average accuracy of 47.9%, our best result by a margin of 4.2%. Notably, training the 20 SVRs on our full+fg features takes an hour on a single core, compared to 10+ hours for training on O2P features.

Summary

To bypass the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use selective search to extract just 2000 regions from the image and he called them region proposals. Therefore, now, instead of trying to classify a huge number of regions, you can just work with 2000 regions. These 2000 region proposals are generated using the selective search algorithm which is written below.

Selective Search:

1. Generate initial sub-segmentation, we generate many candidate regions
2. Use greedy algorithm to recursively combine similar regions into larger ones
3. Use the generated regions to produce the final candidate region proposals

To know more about the selective search algorithm, follow this [link](#). These 2000 candidate region proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output. The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal. In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values to increase the precision of the bounding box. For example, given a region proposal, the algorithm would have predicted the presence of a person but the face of that person within that region proposal could've been cut in half. Therefore, the offset values help in adjusting the bounding box of the region proposal.

Problems with R-CNN

It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.

It cannot be implemented real time as it takes around 47 seconds for each test image.

The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

Reference

Paper (<https://arxiv.org/abs/1311.2524>)



Selective Search

(<https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf>)

R-CNN For Object Detection

(https://courses.cs.washington.edu/courses/cse590v/14au/cse590v_wk1_rcnn.pdf)