Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report

Ross Girshick[1]   Jeff Donahue[1,2]   Trevor Darrell[1,2]   Jitendra Malik[1]
[1]UC Berkeley and [2]ICSI
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Can a large convolutional neural network trained for whole-image classification on ImageNet be coaxed into detecting objects in PASCAL? We show that the answer is yes, and that the resulting system is simple, scalable, and boosts mean average precision, relative to the venerable deformable part model, by more than 40% (achieving a final mAP of 48% on VOC 2007). Our framework combines powerful computer vision techniques for generating bottom-up region proposals with recent advances in learning high-capacity convolutional neural networks. We call the resulting system R-CNN: Regions with CNN features. The same framework is also competitive with state-of-the-art semantic segmentation methods, demonstrating its flexibility. Beyond these results, we execute a battery of experiments that provide insight into what the network learns to represent, revealing a rich hierarchy of discriminative and often semantically meaningful features.

1. Introduction

Image features are the engine of recognition. Better features immediately propel a wide array of computer vision techniques forward. The last feature revolution was, arguably, established through the introduction of SIFT [30] and then HOG [7]. Nearly all modern object detection and semantic segmentation systems (*e.g.*, [5, 17]) are built on top of one, or both, of these low-level features, serving as a testament to their effectiveness.

Yet, the hypothesis that SIFT and HOG are now bottlenecks throttling recognition performance has emerged over the last few years. This hypothesis is grounded, for example, in the wide range of papers that attempt to boost detection accuracy with work along four axes: (1) rich structured models [20, 42]; (2) multiple feature learning [38, 41]; (3) learned histogram-based features [11, 29, 32]; or (4) unsupervised feature learning [34].

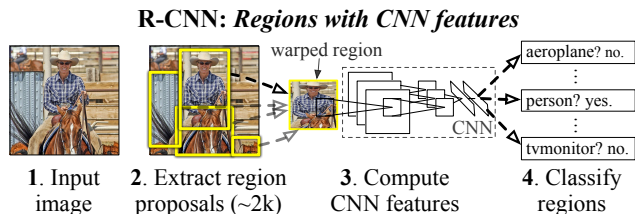The PASCAL Visual Object Classes (VOC) Challenge serves as the main benchmark for assessing object detec-

R-CNN: *Regions with CNN features*

Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. This system achieves a mean average precision (mAP) of **43.5% on PASCAL VOC 2010**. For comparison, [36] reports a mAP of 35.1% using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. Deformable part models [19] perform at 29.6%.

tor performance [15]. The 2010 and 2011 challenges were won by combining multiple types of features and making extensive use of context from ensembles of object detectors and scene classifiers. Using multiple features improved mean average precision (mAP) by at most 10% (relative), with diminishing returns for each additional feature. In the final year of the challenge (2012) systems performed no better than in the previous year. This plateau suggests current methods may be limited by the available features. Here, we take a supervised feature learning approach. Figure 1 overviews our method and highlights some of our results.

At the same time, researchers working on a broad array of "deep learning" methods were making steady progress on improving whole-image classification. (See Bengio *et al.* [3] for an excellent survey.) However, until recently these results were isolated to datasets such as CIFAR [25] and MNIST [28], slowing their adoption by computer vision researchers for use on other tasks and image domains.

Then, Krizhevsky *et al.* [26] rekindled broader interest in convolutional neural networks (CNNs) [27, 28] by showing substantially lower error rates on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9, 10]. The significance of their result was vigorously debated during

the ILSVRC workshop at ECCV 2012. The central issue of debate can be distilled as: **To what extent do the CNN *classification* results on ImageNet generalize to object *detection* results on the PASCAL VOC Challenge?** In this paper, we study object detection using features computed by a large convolutional neural network, answering this important scientific question.[1]

CNNs have been used as sliding window detectors for at least two decades, typically on constrained object categories, such as faces [33, 37] and, recently, pedestrians [34]. We also considered adopting a sliding window approach. However, the high computational cost of a CNN that includes large, densely connected (non-convolutional) layers, such as the one that we investigate, makes this avenue unattractive. Additionally, to detect objects with widely varying aspect ratios, one typically slides a *set* of detectors, with different shapes tuned to different appearance modes of the target object, further increasing computational costs.

Instead, we operate within the "recognition using regions" paradigm, as argued for by Gu *et al*. in [21]. At run-time, our method generates around 2000 category-independent region proposals for the input image, extracts a feature vector from each proposal using our CNN, and then classifies each region with category-specific linear SVMs.

Our system is surprisingly efficient. By working with region proposals, the CNN processes two orders of magnitude fewer image windows as compared to the sliding window approach. We use a simple technique to compute a fixed-length feature vector for each proposal, regardless of the region's shape, allowing features to be shared across all categories and appearance modes. Furthermore, our features are two orders of magnitude lower dimensional than those typically used in similar pipelines (*cf*. [36]). The only class-specific computations are a reasonably small matrix-matrix product and greedy non-maximum suppression.

Following this methodology, we show that a CNN trained for image classification on ImageNet outperforms existing detection methods on the PASCAL VOC Challenge by a large margin. On VOC 2007, for example, an ablated version of our system achieves a mAP of 43% compared to 34% for the highly-tuned deformable part model (DPM) [17]. Our full method boosts mAP to 48%.

One advantage of simpler HOG-like features is that it's easier to understand the information they carry (although [39] shows that with high-dimensional HOG features our intuition can fail us). Can we gain insight into the representation learned by our CNN? Perhaps the densely connected layers—*with more than 54 million parameters*—are the key? They are not. We "lobotomized" the CNN and

found that a surprisingly large proportion—*94%*—of its parameters can be removed with only a moderate drop in detection accuracy. Perhaps color, which HOG makes very weak use of, is the key? Removing color degrades performance only marginally. Instead, by isolating particular units in the network (as in Figure 3), we can see that the CNN learns a diverse set of rich features ranging from red-blob detectors to semantically aligned poselet-like [4] units. Equally important is understanding the failure modes of our approach. Therefore, we report results from the detection analysis tool of Hoiem *et al*. [23].

Before developing the technical details of our method, we note that it is equally applicable to the task of semantic segmentation. With a few modifications, we also achieve state-of-the-art results on the PASCAL VOC segmentation task, with an average segmentation accuracy of 47.9% on the VOC 2011 test set.

2. Object detection

Our object detection system consists of three modules. The first generates category-independent region proposals. These proposals define the set of candidate bounding boxes available to our detector. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region. The third module is a set of class-specific linear SVMs. In this section, we present our design decisions for each module, describe their test-time usage, detail how their parameters are learned, and show results on PASCAL VOC 2010-12.

2.1. Module design

Region proposals. A variety of recent papers offer methods for generating category-independent region proposals. These include objectness [1], selective search [36], category-independent object proposals [14], constrained parametric min-cuts (CPMC) [6], and a method based on merging superpixels from ultrametric contour maps [2]. Motivated by its strong prior performance on the PASCAL detection task, we use selective search.

Feature extraction. We extract a 4096-dimensional feature vector from each region proposal using our own implementation of the CNN of Krizhevsky *et al*. [26], which we built on top of the open source cuda-convnet code [24].[2] Features are computed by forward propagating a mean-subtracted 224×224 RGB image through five convolutional layers and two fully connected layers. We refer readers to [26] for more network architecture details. Ablation studies in Section 3 show how performance varies across features from each of the last three layers.

---

[1]In related work, [12] shows that the CNN of Krizhevsky *et al*., trained on ImageNet, generalizes well to a wide range of datasets and recognition tasks including scene classification, fine-grained sub-categorization, and domain adaptation.

---

[2]This is the same implementation that was used to train the CNN described in [12].

In order to compute features for a region proposal, we must first convert the image data in that region into a form that is compatible with our CNN. The network architecture requires inputs of a fixed 224×224 pixel size. Of the many possible transformations of our arbitrary-shaped regions, we opt for the simplest. Regardless of the size or aspect ratio of the candidate region, we warp all pixels in a tight bounding box around it to the required size. This leads to a fixed-length feature vector for each region. Figure 2 shows a random sampling of warped training regions. The distortion is less than one might imagine a priori.
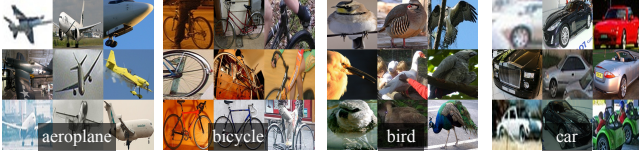

Figure 2: Warped training samples from VOC 2007 train.

2.2. Inference

We run selective search on a test image to extract around 2000 region proposals (we use selective search's "fast mode" in all experiments). We warp each proposal to the required shape, and forward propagate it through the CNN in order to read off the feature map from the desired layer. Then, for each class, we score each extracted feature map using the SVM trained for that class. Given all scored regions in an image, we apply a greedy non-maximum suppression (for each class independently) that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring unsuppressed region larger than a learned threshold (0.3 in all experiments).

Run-time analysis. Two key properties make inference very efficient. First, all CNN parameters are shared across all categories. Second, 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 features used in the UVA detection system [36], for example, are two orders of magnitude larger than ours (360k vs. 4k-dimensional).

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. The only class-specific computations are dot products between features and SVM weights and non-maximum suppression. In practice, all dot products for an image are batched into a single matrix-matrix product. The feature matrix is typically 2000×4096 and the SVM weight matrix is $4096 \times N$, where N is the number of classes.

While not immediately obvious, our method should easily 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. Moreover, this efficiency is not merely the result of using region proposals and shared features. The UVA system, due to its high-dimensional features, would be two orders of magnitude slower while requiring 134GB of memory just to store 100k linear predictors, compared to just 1.5GB for our lower-dimensional features.

It is also interesting to contrast our approach with the recent work from Dean *et al.* on scalable detection using DPMs and hashing [8]. They report a mAP of around 16% on VOC 2007 at a run-time of 5 minutes per image when introducing 10k distractor classes. With our approach, 10k detectors can run in about a minute on a CPU, and because no approximations are made mAP would remain at 48% (Section 3.2).

2.3. Training

CNN pre-training. We used a large auxiliary dataset (ILSVRC 2012) with *image-level annotations* (*i.e.*, no bounding box labels) to "pre-train" the CNN. We followed the methodology of [26] closely, aside from two small changes (implemented for simplicity). We share the same implementation and training procedure as detailed in the DeCAF tech report [12] and refer readers there for details. In brief summary, our CNN nearly matches the performance of [26], obtaining a top-1 error rate only 2.2% points higher on the ILSVRC 2012 validation set. This discrepancy is likely due to our simplifications.

CNN fine-tuning. To adapt our CNN to the new task (detection) and the new domain (warped PASCAL windows), we continue training the CNN parameters using only *warped region proposals* from PASCAL. During pre-training, we decreased the learning rate by a factor of 10 three times. However, little progress is made at the final rate. For fine-tuning, we start stochastic gradient descent (SGD) at a learning rate of 0.01 times that of the initial pre-training rate. This allows fine-tuning to make progress while not clobbering the initialization. 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. In each SGD iteration, we sample two training images and construct a mini-batch of size 128 by sampling 64 proposals from the approximately 2000 in each image. Because target objects are rare, we found it necessary to bias sampling so that on average 1/4 of each mini-batch is positive.

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

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 HOG [19]	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	29.6
SegDPM [18]	56.4	48.0	24.3	21.8	**31.3**	**51.3**	47.3	48.2	16.1	29.4	19.0	37.5	44.1	51.5	44.4	12.6	32.1	28.8	**48.9**	39.1	36.6
UVA [36]	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
ours (R-CNN FT fc$_7$)	**65.4**	**56.5**	**45.1**	**28.5**	24.0	50.1	**49.1**	**58.3**	**20.6**	**38.5**	**31.1**	**57.5**	**50.7**	**60.3**	**44.7**	**21.6**	**48.5**	24.9	48.0	**46.5**	**43.5**

Table 1: Detection average precision (%) on VOC 2010 test. Our method competes in the *comp4* track due to our use of outside data from ImageNet. Our system is most directly comparable to UVA (row 3) since both methods use the same selective search region proposal mechanism, but differ in features. We compare to methods before rescoring with inter-detector context and/or image classification.

is how to label a region that partially overlaps a car. We 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 on $\{0, 0.1, \ldots, 0.5\}$. We found that selecting this threshold carefully is crucial. Setting it to 0.5, as in [36], decreased mAP by 5 points. Similarly, setting it to 0 decreased mAP by 4 points. Positive examples are defined simply to be the ground-truth bounding boxes for each class.

Once features are extracted and training labels are applied, we optimize one linear SVM per class. Since the training data is too large to fit in memory, we adopt the standard hard negative mining method [17, 35]. We found that hard negative mining converges quickly and in practice mAP stops increasing after only a single pass over all images. Training is fast given precomputed feature vectors, which we store on disk. Training time for all 20 PASCAL object detection SVMs (in 5k images) takes about 1.5 hours on a single core. Computing features takes about 5ms per region on a GPU.

2.4. Results on PASCAL VOC 2010-12

Following the PASCAL "best practices" guidelines [15], we made all design decisions on the VOC 2007 dataset (see Section 3.2). For final results on the VOC 2010-12 datasets, we fine-tuned the CNN on VOC 2012 train (to avoid overfitting on the validation set). We then trained our detection SVMs on VOC 2012 trainval and submitted test results to the evaluation server only once.

Table 1 shows complete results for VOC 2010. We compare our method against three strong baselines, but exclude results from ensemble systems that use context rescoring. Such rescoring boosts all methods and is orthogonal to the focus of this paper (for clarity: we score individual windows in isolation, so no contextual information is used). The most germane comparison is to the UVA system from Uijlings *et al.* [36] (Table 1 row 3), since our systems use the same region proposal algorithm. To classify regions, their method builds a four-level spatial pyramid and populates it with densely sampled SIFT, Extended OpponentSIFT, and RGB-SIFT descriptors, each vector quantized with 4000-word codebooks. Classification is performed with a histogram intersection kernel SVM. Compared to their multi-feature, non-linear kernel SVM approach, we achieve a

large improvement in mAP, from 35.1% to 43.5% mAP, while also being much faster (Section 2.2). Our method achieves similar performance on VOC 2011/12 test with a mAP of 43.2%.

3. Visualization, ablation, and modes of error

The CNN works well in practice, but what did it learn, what aspects of its design are critical to its success, and how does it fail?

3.1. Visualizing learned features

First-layer filters can be visualized directly and are easy to understand [26]. They capture oriented edges and opponent colors. Understanding the subsequent layers is more challenging. Zeiler and Fergus present an attractive deconvolutional approach in [40]. We propose a simple (and complementary) non-parametric method that directly shows what the network learned.

The idea is to single out a particular unit (artificial "neuron") 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 response, perform non-maximum suppression (within each image), and then display the top-scoring regions. Our method lets the selected unit "speak for itself" by showing exactly which inputs it fires on. Because we avoid averaging, we have the opportunity to see multiple visual modes and gain insight into the invariances computed by the unit.

We visualize units from layer pool$_5$, which is the max-pooled output of the network's fifth and final convolutional layer. The pool$_5$ feature map is $6 \times 6 \times 256 = 9216$-dimensional. Ignoring boundary effects, each pool$_5$ unit has a receptive field of 163×163 pixels in the original 224×224 pixel input. A central pool$_5$ unit has a nearly global view, while one near the edge has a much smaller, clipped support. We selected this layer because it's the last layer for which units have a compact receptive field, making it easier to show which part of the image is responsible for the activation. Additionally, we gain some intuition into the representation learned by the next layer, fc$_6$, because it takes multiple weighted combinations of pool$_5$ activations.

Figure 3 displays the top 16 activations for six units from a CNN that we fine-tuned on VOC 2007 trainval. The first
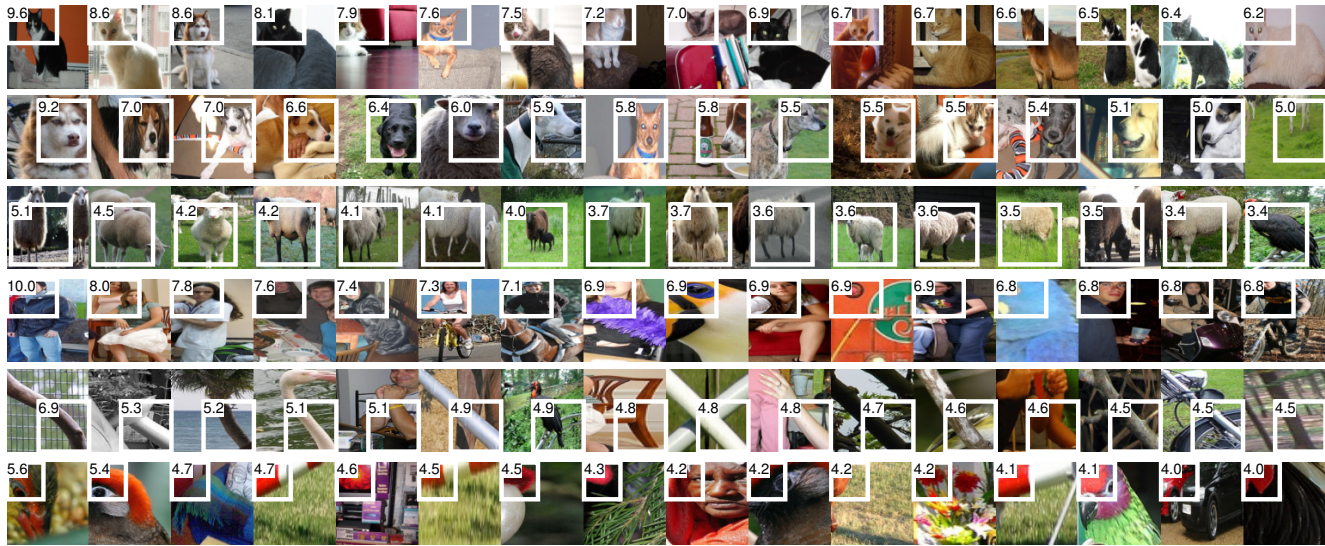
Figure 3: Top activations for six pool$_5$ units. Receptive fields and activation values are drawn in white. From top to bottom: (1) positive and (2) negative weight for cats; positive weight for (3) sheep and (4) person; selectivity for (5) diagonal bars and (6) red blobs.

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$_5$	49.3	58.0	29.7	22.2	20.6	47.7	56.8	43.6	16.0	39.7	37.7	39.6	49.6	55.6	37.5	20.6	40.5	37.4	47.8	51.3	40.1
R-CNN fc$_6$	56.1	58.8	34.4	29.6	22.6	50.4	58.0	52.5	18.3	40.1	41.3	46.8	49.5	53.5	39.7	23.0	46.4	36.4	50.8	59.0	43.4
R-CNN fc$_7$	53.1	58.9	35.4	29.6	22.3	50.0	57.7	52.4	19.1	43.5	40.8	43.6	47.6	54.0	39.1	23.0	42.3	33.6	51.4	55.2	42.6
R-CNN FT pool$_5$	55.6	57.5	31.5	23.1	23.2	46.3	59.0	49.2	16.5	43.1	37.8	39.7	51.5	55.4	40.4	23.9	46.3	37.9	49.7	54.1	42.1
R-CNN FT fc$_6$	61.8	62.0	38.8	35.7	29.4	52.5	**61.9**	53.9	22.6	49.7	40.5	**48.8**	49.9	**57.3**	44.5	**28.5**	50.4	**40.2**	54.3	61.2	47.2
R-CNN FT fc$_7$	**60.3**	**62.5**	**41.4**	**37.9**	29.0	**52.6**	61.6	**56.3**	**24.9**	**52.3**	**41.9**	48.1	54.3	57.0	**45.0**	26.9	**51.8**	38.1	**56.6**	**62.2**	**48.0**
DPM HOG [19]	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 [29]	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 [32]	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

Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show results for our CNN pre-trained on ILSVRC 2012. Rows 4-6 show results for our CNN pre-trained on ILSVRC 2012 and then fine-tuned ("FT") on VOC 2007 trainval. Rows 7-9 present DPM methods as a strong baseline comparison. The first uses only HOG, while the next two use feature learning to augment or replace it.

two units were selected because they correspond to large positive and negative weights in a cat SVM (trained on pool$_5$). The first is selective for cat faces, while the second is selective for other animal faces (primarily dogs). We also visualize units for sheep and person. The last two rows illustrate more generic units; one that fires on diagonal bars of a certain width and another that fires on red blobs. These visualizations demonstrate the ==richness of pool$_5$ features and hint at the diversity that lies therein==, with units ranging from particular animal faces to more generic shapes and textures. The subsequent fully connected layer has the ability to ==model a large set of compositions of these rich features==. Additional visualizations are included in the appendix, Figure 6.

3.2. Ablation studies

Performance layer-by-layer, without fine-tuning. To understand which layers are critical for detection performance, we analyzed results on the VOC 2007 dataset for each of the

CNN's last three layers. Layer pool$_5$ was briefly described in Section 3.1. The final two layers are summarized below.

Layer fc$_6$ is fully connected to pool$_5$. To compute features, it multiplies a 4096×9216 weight matrix by the pool$_5$ feature map (reshaped as a 9216-dimensional vector) and then adds a vector of biases. This intermediate vector is component-wise half-wave rectified (*i.e.*, $x \leftarrow \max(0, x)$).

Layer fc$_7$ is the final layer of the network. It is implemented by multiplying the features computed by fc$_6$ by a 4096×4096 weight matrix, and similarly adding a vector of biases and applying half-wave rectification.

We 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 (Table 2 rows 1-3) reveals that features from fc$_7$ offer little or no advantage over features from fc$_6$. This means that 29%, or about 16.8 million, of the CNN's parameters can be removed without degrading mAP. More surprising is that removing *both* fc$_7$ and fc$_6$ produces quite good

results even though pool$_5$ features are computed using *only 6% of the CNN's parameters*. Much of the 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 convolution layers of the CNN. This representation would enable the use of sliding window detectors, including DPM, on top of the rich pool$_5$ features.

Color. To understand how much our system benefits from color (compared to HOG-based approaches, which largely ignore it) we tested our pre-trained CNN in a grayscale world. Training SVMs on fc$_6$ features from grayscale versions of PASCAL images, and testing on grayscale images decreased mAP on VOC 2007 test from 43.4% to 40.1%.

Performance layer-by-layer, with fine-tuning. We now look at results from our CNN after having fine-tuned its parameters on VOC 2007 trainval. The improvement is striking (Table 2 rows 4-6). Fine-tuning increases mAP by 4.6 points to 48.0%. The boost from fine-tuning is much larger for fc$_6$ and fc$_7$ than for pool$_5$. This may indicate that the rich set of pool$_5$ features learned from ImageNet are largely sufficient for PASCAL and that most the improvement is gained from learning how to optimally combine them in fc$_6$.

Comparison to recent feature learning methods. Relatively few feature learning methods have been tried on PASCAL detection. We look at two recent approaches that build on deformable part models (DPM) [17]. For reference, we also include results for the standard HOG-based DPM [19].

The first DPM feature learning method, DPM ST [29], augments HOG features with histograms of "sketch token" probabilities. Intuitively, a sketch token is a tight distribution of contours passing through the center of an image patch. Sketch token probabilities are computed at each pixel by a random forest that was trained to classify 35×35 pixel patches into one of 150 sketch tokens or background.

The second method, DPM HSC [32], replaces HOG with histograms of sparse codes. To compute an HSC, sparse code activations are solved for at each pixel using a learned dictionary of 100 7×7 pixel (grayscale) atoms. The resulting activations are rectified in three ways (full and both half-waves), spatially pooled, unit ℓ_2 normalized, and then power transformed ($x \leftarrow \text{sign}(x)|x|^\alpha$).

All of our CNN methods strongly outperform the three DPM baselines (Table 2 rows 7-9), including the two that use feature learning. Compared to the latest version of DPM, which uses only HOG features, our mAP is more than 14 points higher: 48.0% *vs.* 33.7%—*a 42% relative improvement*. The combination of HOG and sketch tokens yields 2.5 mAP points over HOG alone, while HSC improves over HOG by 4 points mAP (when compared

internally to their private DPM baselines—both use non-public implementations of DPM that underperform the open source version [19]). These methods achieve mAPs of 29.1% and 34.3%, respectively.

3.3. Detection error analysis

We applied the excellent detection analysis tool from Hoiem *et al.* [23] in order to reveal our method's error modes, understand how fine-tuning changes them, and to see how our error types compare with DPM. A full summary of the analysis tool is beyond the scope of this paper and we encourage readers to consult [23] to understand some finer details (such as "normalized AP"). Since the analysis is best absorbed in the context of the associated plots, we present the discussion within the captions of Figure 4 and Figure 5.
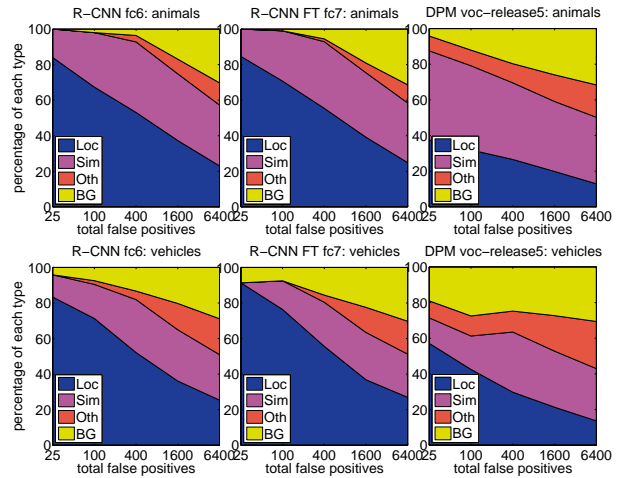

Figure 4: Distribution of top-ranked false positive (FP) types. Each plot shows the evolving distribution of FP types as more FPs are considered in order of decreasing score. Each FP is categorized into 1 of 4 types: (Loc) a detection with an IoU overlap with the correct class, but between 0.1 and 0.5 (or a duplicate) is a poor localization; (Sim) confusion with a similar category; (Oth) confusion with a dissimilar object category; (BG) a FP that fired on background. Compared with DPM, significantly more of our errors result from poor localization, rather than confusion with background and non-target objects, indicating that the CNN features are much more discriminative than HOG. The loose localization problem likely results from our use of bottom-up region proposals and the positional invariance learned from pre-training the CNN for whole-image classification. For animal categories, fine-tuning slightly reduces confusion with other animal and non-animal categories. For vehicles, fine-tuning reduces confusions with other vehicles amongst the highest scoring FPs.

4. Semantic segmentation

Region classification is a standard technique for semantic segmentation, allowing us to easily apply our CNN to the PASCAL VOC segmentation challenge. To facilitate a
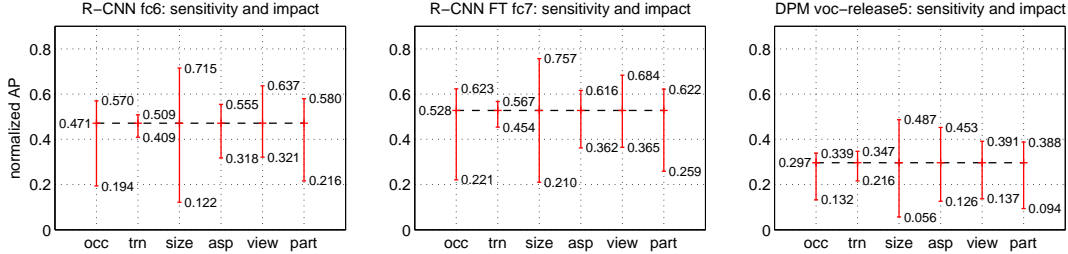
Figure 5: Sensitivity to object characteristics. Each plot shows the mean (over classes) normalized AP (see [23]) for the highest and lowest performing subsets within six different object characteristics (occlusion, truncation, bounding box area, aspect ratio, viewpoint, part visibility). We show plots for our method (R-CNN) with and without fine-tuning (FT) as well as for DPM voc-release5. Overall, fine-tuning does not reduce sensitivity (the difference between max and min), but does substantially improve both the highest and lowest performing subsets for all characteristics. This indicates that fine-tuning does more than simply improve the lowest performing subsets for aspect ratio and bounding box area, as one might conjecture based on how we distort all network inputs (by warping). Instead, fine-tuning improves robustness for all characteristics including occlusion, truncation, viewpoint, and part visibility.

direct comparison with the current leading semantic segmentation system (called O_2P for "second-order pooling") [5] we work within their open source framework. O_2P 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 (enriched variants of SIFT and LBP). We also note that Farabet *et al.* [16] recently demonstrated good results on several dense scene labeling datasets (not including PASCAL) using a CNN as a per-pixel classifier.

We follow [2, 5] and extend the PASCAL segmentation training set to include the extra annotations made available by Hariharan *et al.* [22]. Design decisions and hyperparameters were cross-validated on the VOC 2011 validation set. Final test results were evaluated only once.

CNN features for segmentation. We evaluate three strategies for computing features on CPMC regions, all of which begin by warping the rectangular window around the region to 224×224. The first strategy (*full*) ignores the region's shape and computes CNN features directly on the warped window, exactly as we 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. Therefore, 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; our experiments validate their complementarity.

	full R-CNN		fg R-CNN		full+fg R-CNN	
	fc$_6$	fc$_7$	fc$_6$	fc$_7$	fc$_6$	fc$_7$
O_2P [5] 46.4	43.0	42.5	43.7	42.1	**47.9**	45.8

Table 3: Segmentation mean accuracy (%) on VOC 2011 validation. Column 1 presents O_2P; 2-7 use our CNN pre-trained on ILSVRC 2012.

Results on VOC 2011. Table 3 shows a summary of our results on the VOC 2011 validation set compared with O_2P. (See Table 5 in the appendix for complete per-category results.) Within each feature computation strategy, layer fc$_6$ always outperformed fc$_7$ and the following discussion refers to the fc$_6$ features. The *fg* strategy slightly outperforms *full*, indicating that the masked region shape provides a stronger signal, matching intuition. However, *full+fg* achieves an average accuracy of 47.9%, our best result by a margin of 4.2% (also modestly outperforming O_2P), indicating that the context provided by the *full* features is highly informative even given the *fg* features. Notably, training the 20 SVRs on our *full+fg* features takes around an hour on a single core, compared to 10 hours for training on the O_2P features.

In Table 4 we present results on the VOC 2011 test set, comparing our best-performing method, fc$_6$ (*full+fg*), against two strong baselines. Our method achieves the highest segmentation accuracy for 11 out of 21 categories, and the highest overall segmentation accuracy of 47.9%, averaged across categories (but likely ties with the O_2P result under any reasonable margin of error). Still better performance could likely be achieved by fine-tuning.

5. Discussion

Leveraging a large auxiliary dataset is one key to our method's success. Why not give more training data to other approaches, too? One issue is that it's non-trivial to benefit from data from a different domain, and that was labeled for a different task. Training a DPM for PASCAL, for example, requires bounding box annotations for the PASCAL categories. Additionally, [42] shows that even when more data is available, DPM does not easily benefit from it. A second issue is that many methods lack large sets of shared parameters to pre-train. For instance, a bag-of-visual-words method is unlikely to benefit from training its codebook on ImageNet. These issues may be overcome in the future, but

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$_2$P [5]	**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**

Table 4: Segmentation accuracy (%) on VOC 2011 test. All methods compete in the *comp6* track due to use of outside data from [22]. We compare against two strong baselines: the "Regions and Parts" (R&P) method of [2] and the second-order pooling (O$_2$P) method of [5]. Without any fine-tuning, our CNN achieves top segmentation performance, outperforming R&P and roughly matching O$_2$P.

they are research endeavors in their own right.

This paper proves a strong experimental claim: A large convolutional neural network is highly effective at exploiting "big visual data" to learn a rich hierarchy of features that yields previously unattainable object detection results on the gold-standard PASCAL VOC Challenge. This is no small feat. From the vantage point of the detector, ILSVRC 2012 is weakly labeled, even missing annotations for key visual concepts such as *person*. The CNN's ability to easily turn this data into top-performing detection results is truly exciting. It is significant that we achieved these results by using a combination of classical tools (bottom-up region proposals and convolutional neural networks) from computer vision *and* deep learning. Rather than opposing world views, the two are natural and inevitable partners.

A. Additional feature visualizations

Figure 6 shows additional visualizations for six pool$_5$ units. For each unit, we show the 96 region proposals that maximally activate that unit out of the full set of approximately 10 million regions in all of VOC 2007 test.

We label each unit by its (y, x, channel) position in the $6 \times 6 \times 256$ dimensional pool$_5$ feature map. Within each channel, the CNN computes exactly the same function of the input region, with the (y, x) position changing only the receptive field. From top-left to bottom-right we see units that are selective for: green plants, cat faces, human faces, squiggles, text, and stripes at a variety of orientations.

B. Per-category segmentation results

In Table 5 we show the per-category segmentation accuracy on VOC 2011 val for each of our six segmentation methods in addition to the O$_2$P method [5]. These results show which methods are strongest across each of the 20 individual PASCAL classes, plus the background class.

C. Analysis of cross-dataset redundancy

One concern when training on an auxiliary dataset is that there might be redundancy between it and the test set. Even though the tasks of object detection and whole-image classification are substantially different, making such cross-set redundancy much less worrisome, we still conducted a thorough investigation that quantifies the extent to which PASCAL test images are contained within the ILSVRC 2012

training and validation sets. Our findings may be useful to researchers who are interested in using ILSVRC 2012 as training data for the PASCAL image classification task.

We performed two checks for duplicate (and near-duplicate) images. The first test is based on exact matches of flickr image IDs, which are included in the VOC 2007 test annotations (these IDs are intentionally kept secret for subsequent PASCAL test sets). All PASCAL images, and about half of ILSVRC, were collected from flickr.com. This check turned up 31 matches out of 4952 (0.63%).

The second check uses GIST descriptor [31] matching, which was shown in [13] to have excellent performance at near-duplicate image detection in large ($>$ 1 million) image collections. Following [13], we computed GIST descriptors on warped 32×32 pixel versions of all ILSVRC 2012 trainval and PASCAL 2007 test images. Euclidean nearest-neighbor matching of GIST descriptors revealed 38 near-duplicate images (including all 31 found by flickr ID matching). The matches tend to vary slightly in JPEG compression level and resolution, and to a lesser extent cropping. These findings show that the overlap is very small, less than 1%. For VOC 2012, because flickr IDs are not available, we used the GIST matching method only. Based on GIST matches, 1.5% of VOC 2012 test images are in ILSVRC 2012 trainval. The slightly higher rate for VOC 2012 is likely due to the fact that the two datasets were collected closer together in time than VOC 2007 and ILSVRC 2012.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 2012. 2

[2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 2, 7, 8

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *TPAMI*, 2013. 1

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1, 7, 8, 10
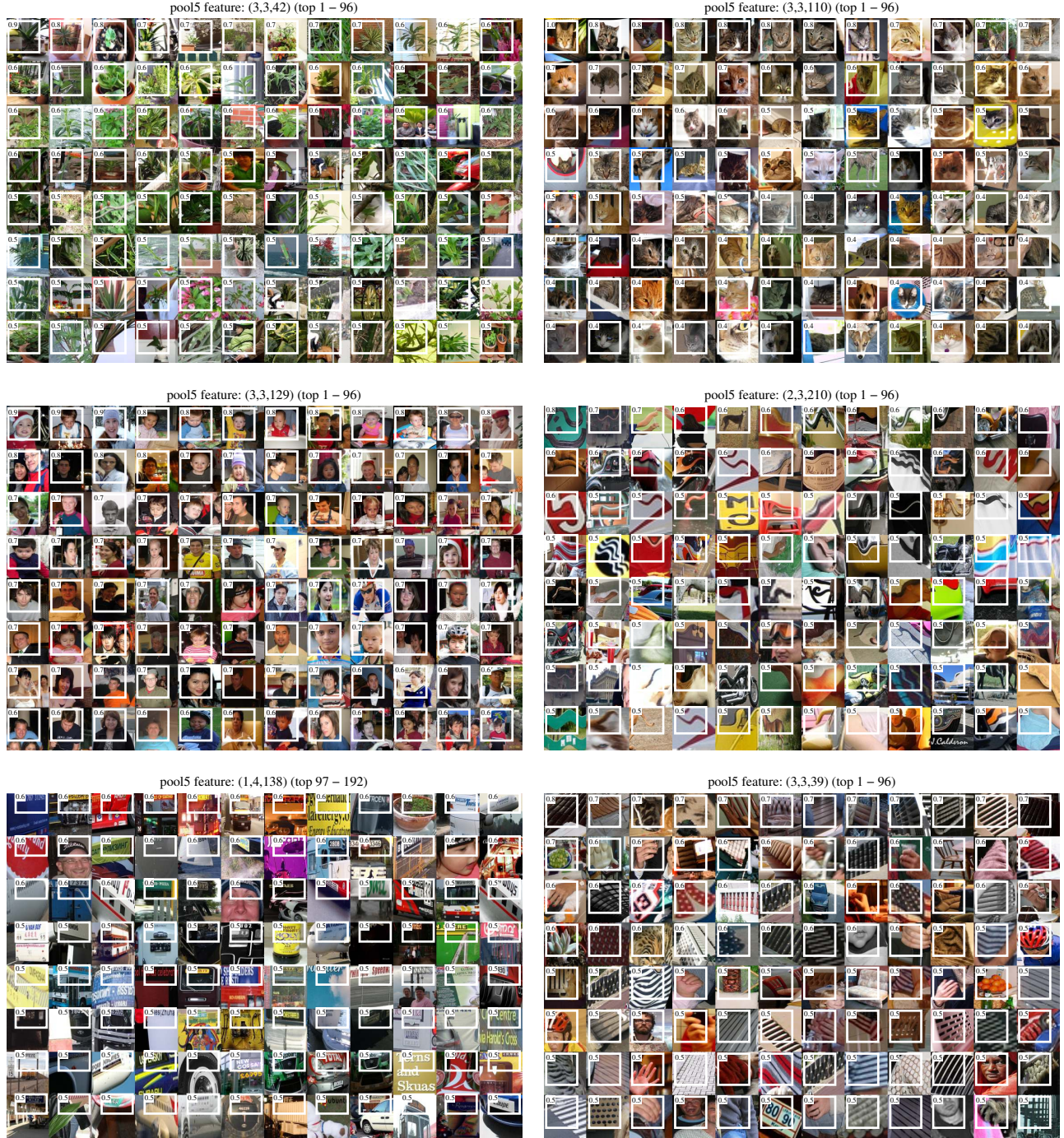
Figure 6: We show the 96 region proposals, out of the approximately 10 million regions in VOC 2007 test, that most strongly activate each of six units. Each montage is labeled by the unit's (y, x, channel) position in the $6 \times 6 \times 256$ dimensional pool_5 feature map. Each image region is drawn with an overlay of the unit's receptive field in white. The activation value (which we normalize by dividing by the max activation value over all units in a channel) is shown in the receptive field's upper-left corner. (Best viewed digitally with zoom.)

[6] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[8] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijaya-narasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013.

VOC 2011 val	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
O$_2$P [5]	**84.0**	**69.0**	21.7	47.7	42.2	42.4	**64.7**	**65.8**	57.4	**12.9**	37.4	20.5	43.7	35.7	52.7	51.0	**35.8**	**51.0**	28.4	59.8	49.7	46.4
full R-CNN fc$_6$	81.3	56.2	23.9	42.9	40.7	38.8	59.2	56.5	53.2	11.4	34.6	16.7	48.1	37.0	51.4	46.0	31.5	44.0	24.3	53.7	51.1	43.0
full R-CNN fc$_7$	81.0	52.8	**25.1**	43.8	40.5	42.7	55.4	57.7	51.3	8.7	32.5	11.5	48.1	37.0	50.5	46.4	30.2	42.1	21.2	57.7	**56.0**	42.5
fg R-CNN fc$_6$	81.4	54.1	21.1	40.6	38.7	**53.6**	59.9	57.2	52.5	9.1	36.5	**23.6**	46.4	38.1	53.2	51.3	32.2	38.7	**29.0**	53.0	47.5	43.7
fg R-CNN fc$_7$	80.9	50.1	20.0	40.2	34.1	40.9	59.7	59.8	52.7	7.3	32.1	14.3	48.8	42.9	54.0	48.6	28.9	42.6	24.9	52.2	48.8	42.1
full+fg R-CNN fc$_6$	83.1	60.4	23.2	48.4	**47.3**	52.6	61.6	60.6	**59.1**	10.8	**45.8**	20.9	**57.7**	43.3	**57.4**	**52.9**	34.7	48.7	28.1	60.0	48.6	**47.9**
full+fg R-CNN fc$_7$	82.3	56.7	20.6	**49.9**	44.2	43.6	59.3	61.3	57.8	7.7	38.4	15.1	53.4	**43.7**	50.8	52.0	34.1	47.8	24.7	**60.1**	55.2	45.7

Table 5: Per-category segmentation accuracy (%) on the VOC 2011 validation set.

3

[9] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/. 1

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[11] M. Dikmen, D. Hoiem, and T. S. Huang. A data driven method for feature transformation. In *CVPR*, 2012. 1

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv e-prints*, arXiv:1310.1531 [cs.CV]. 2, 3

[13] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proc. of the ACM International Conference on Image and Video Retrieval*, 2009. 8

[14] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 2

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1, 4

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013. 7

[17] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 1, 2, 4, 6

[18] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 4

[19] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://www.cs.berkeley.edu/~rbg/latent-v5/. 1, 4, 5, 6

[20] R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011. 1

[21] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 2

[22] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 7, 8

[23] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*. 2012. 2, 6, 7

[24] A. Krizhevsky. cuda-convnet. https://code.google.com/p/cuda-convnet/. 2

[25] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 1

[26] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 3, 4

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

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998. 1

[29] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013. 1, 5, 6

[30] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[31] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 8

[32] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013. 1, 5, 6

[33] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *TPAMI*, 1998. 2

[34] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, 2013. 1, 2

[35] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical Report A.I. Memo No. 1521, Massachussets Institute of Technology, 1994. 4

[36] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 2, 3, 4

[37] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc on Vision, Image, and Signal Processing*, 1994. 2

[38] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1

[39] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: visualizing object detection features. *ICCV*, 2013. 2

[40] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *CVPR*, 2011. 4

[41] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured HOG-LBP for object localization. In *CVPR*, 2011. 1

[42] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012. 1, 7