Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction

Yuting Zhang[*†], Kihyuk Sohn[†], Ruben Villegas[†], Gang Pan[*], Honglak Lee[†]

[*] Department of Computer Science,
Zhejiang University, Hangzhou, Zhejiang, China,
{zyt,gpan}@zju.edu.cn

[†] Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI, USA,
{yutingzh,kihyuks,rubville,honglak}@umich.edu

Abstract

Object detection systems based on the deep convolutional neural network (CNN) have recently made groundbreaking advances on several object detection benchmarks. While the features learned by these high-capacity neural networks are discriminative for categorization, inaccurate localization is still a major source of error for detection. Building upon high-capacity CNN architectures, we address the localization problem by 1) using a search algorithm based on Bayesian optimization that sequentially proposes candidate regions for an object bounding box, and 2) training the CNN with a structured loss that explicitly penalizes the localization inaccuracy. In experiments, we demonstrate that each of the proposed methods improves the detection performance over the baseline method on PASCAL VOC 2007 and 2012 datasets. Furthermore, two methods are complementary and significantly outperform the previous state-of-the-art when combined.

1. Introduction

Object detection is one of the long-standing and important problems in computer vision. Motivated by the recent success of deep learning [26, 21, 3, 6, 27, 4, 35] on visual object recognition tasks [25, 38, 46, 39, 42], significant improvements have been made in the object detection problem [41, 11, 20]. Most notably, Girshick et al. [18] proposed the "regions with convolutional neural network" (R-CNN) framework for object detection and demonstrated state-of-the-art performance on standard detection benchmarks (e.g., PASCAL VOC [12, 13], ILSVRC [34]) with a large margin over the previous arts, which are mostly based on deformable part model (DPM) [15].

There are two major keys to the success of the R-CNN. First, features matter [18]. In the R-CNN, the low-level image features (e.g., HOG [8]) are replaced with the CNN features, which are arguably more discriminative representations. One drawback of CNN features, however, is that they are expensive to compute. The R-CNN overcomes this issue by proposing a few hundreds or thousands candidate bounding boxes via the selective search algorithm [44] to effectively reduce the computational cost required to evaluate the detection scores at all regions of an image.

Despite the success of R-CNN, it has been pointed out through an error analysis [22] that inaccurate localization causes the most egregious errors in the R-CNN framework [18]. For example, if there is no bounding box in the close proximity of ground truth among those proposed by selective search, no matter what we have for the features or classifiers, there is no way to detect the correct bounding box of the object. Indeed, there are many applications that require accurate localization of an object bounding box, such as detecting moving objects (e.g., car, pedestrian, bicycles) for autonomous driving [17], detecting objects for robotic grasping or manipulation in robotic surgery or manufacturing [28], and many others.

In this work, we address the localization difficulty of the R-CNN detection framework with two ideas. First, we develop a fine-grained search algorithm to expand an initial set of bounding boxes by proposing new bounding boxes with scores that are likely to be higher than the initial ones. By doing so, even if the initial region proposals were poor, the algorithm can find a region that is getting closer to the ground truth after a few iterations. We build our algorithm in the Bayesian optimization framework [31, 40], where evaluation of the complex detection function is replaced with queries from a probabilistic distribution of the function values defined with a computationally efficient surrogate model. Second, we train a CNN classifier with a structured SVM objective that aims at classification and localization simultaneously. We define the structured SVM objective function with a hinge loss that balances between classification (i.e., determines whether an object exists) and localization (i.e., determines how much it overlaps with the ground truth) to be used as the last layer of the CNN.

In experiments, we evaluated our methods on PASCAL VOC 2007 and 2012 detection tasks and compared to other competing methods. We demonstrated significantly improved performance over the state-of-the-art at different lev-

els of intersection over union (IoU) criteria. In particular, our proposed method outperforms the previous arts with a large margin at higher IoU criteria (e.g., IoU = 0.7), which highlights the good localization ability of our method.

Overall, the contributions of this paper are as follows: 1) we develop a Bayesian optimization framework that can find more accurate object bounding boxes without significantly increasing the number of bounding box proposals, 2) we develop a structured SVM framework to train a CNN classifier for accurate localization, 3) the aforementioned methods are complementary and can be easily adopted to various CNN models, and finally, 4) we demonstrate significant improvement in detection performance over the R-CNN on both PASCAL VOC 2007 and 2012 benchmarks.

2. Related work

The DPM [15] and its variants [32, 16] have been the dominating methods for object detection tasks for years. These methods use image descriptors such as HOG [8], SIFT [30], and LBP [1] as features and densely sweep through the entire image to find a maximum response region. With the notable success of CNN on large scale object recognition [25], several detection methods based on CNNs have been proposed [38, 37, 41, 11, 18]. Following the traditional sliding window method for region proposal, Sermanet et al. [38] proposed to search exhaustively over an entire image using CNNs, but made it efficient by conducting a convolution on the entire image at once at multiple scales. Apart from the sliding window method, Szegedy et al. [41] used CNNs to regress the bounding boxes of objects in the image and used another CNN classifier to verify whether the predicted boxes contain objects. Girshick et al. [18] proposed the R-CNN following the "recognition using regions" paradigm [19], which also inspired several previous state-of-the-art methods [44, 45]. In this framework, a few hundreds or thousands of regions are proposed for an image via the selective search algorithm [44] and the CNN is finetuned with these region proposals. Our method is built upon the R-CNN framework using the CNN proposed in [39], but with 1) a novel method to propose extra bounding boxes in the case of poor localization, and 2) a classifier with improved localization sensitivity.

The structured SVM objective function in our work is inspired by Blaschko and Lampert [5], where they trained a kernelized structured SVM on low-level visual features (i.e., HoG [8]) to predict the object location. In contrast, we built the linear structured objective upon high-level features learned by deep CNN architectures. In addition, our negative mining step is very efficient thanks to the region-based detection framework. We present a gradient-based optimization method for training our architecture.

There have been several other related work for accurate object localization. Fidler et al. [16] incorporated the geometric consistency of bounding boxes with bottom-up segmentation as auxiliary features into the DPM. Dai and Hoiem [7] used the structured SVM with color and edge features to refine the bounding box coordinates in DPM framework. Schulter et al. [36] used the height prior of an object. These auxiliary features to aid object localization can be injected into our framework without modifications.

Localization refinement can be also taken as a CNN regression problem. Girshick et al. [18] extracted the middle layer features and linearly regressed the initially proposed regions to better locations. Sermanet et al. [38] refined bounding boxes from a grid layout to flexible locations and sizes using the higher layers of the deep CNN architecture. Erhan et al. [11] jointly conducted classification and regression in a single architecture. Our method is different in that 1) it uses the information from multiple existing regions instead of a single bounding box for predicting a new candidate region, and 2) it focuses only on maximizing the localization ability of the CNN classifier instead of doing any regression from one bounding box to another.

3. Fine-grained search for bounding box via Bayesian optimization

Let $f(x, y)$ denote a detection score of an image x at the region with the box coordinates $y = (u_1, v_1, u_2, v_2) \in \mathcal{Y}$. The object detection problem deals with finding the local maximum of $f(x, y)$ with respect to y of an unseen image x.[1] As it requires an evaluation of the score function at many possible regions, it is crucial to have an efficient algorithm to search for the candidate bounding boxes.

A sliding window method has been used as a dominant search algorithm [8, 15], which exhaustively searches over an entire image with fixed-sized windows at different scales to find a bounding box with a maximum score. However, evaluating the score function at all regions determined by the sliding window approach is prohibitively expensive when the CNN features are used as the image region descriptor. The problem becomes more severe when flexible aspect ratios are needed for handling object shape variations. Alternatively, the "recognition using regions" [19, 18] method has been proposed, which requires to evaluate significantly fewer number of regions (e.g., few hundreds or thousands) with different scales and aspect ratios, and it can use the state-of-the-art image features with high computational complexity, such as the CNN features [10]. One potential issue of object detection pipelines based on region proposal is that the correct detection will not happen when there is no region proposed in the proximity of the ground truth bounding box.[2] To resolve this issue, one can propose more bounding boxes to cover the entire image more densely, but this would significantly in-

[1] When multiple (including zero) objects exist, it involves finding the local maxima that exceed a certain threshold.

[2] We refer to selective search as a representative method for region proposal.

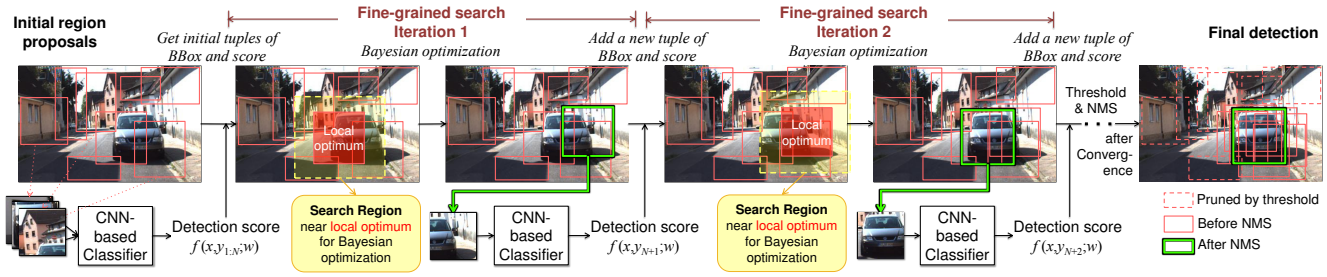

Figure 1: Pipeline of our method. 1) Initial bounding boxes are given by methods such as the selective search [44] and their detection scores are obtained from the CNN-based classifier trained with structured SVM objective. 2) The box(es) with optimal score(s) in the local regions are found by greedy NMS [18] (shown as "local optimum" boxes), and Bayesian optimization takes the neighborhood of each local optimum to propose a new box with a high chance of getting a better detection score. 3) We evaluate the detection score of the new box, take it as an observation for the next iteration of the Bayesian optimization until convergence. Note that the local optimum and the search region may change in each iteration. 4) All the bounding boxes are fed into the standard post-processing stage (e.g., thresholding and NMS, etc.).

crease the computational cost. In this section, we develop a fine-grained search (FGS) algorithm based on Bayesian optimization that sequentially proposes a new bounding box with a higher *expected* detection score than previously proposed bounding boxes without significantly increasing the number of region proposals. We first present the general Bayesian optimization framework (Section 3.1) and describe the FGS algorithm using Gaussian process as the prior for the score function (Section 3.2). We then present the local FGS algorithm that searches over multiple local regions instead of a single global region (Section 3.3), and discuss the hyperparameter learning of our FGS algorithm (Section 3.4).

3.1. General Bayesian optimization framework

Let $\{y_1, \cdots, y_N\}$ be the set of solutions (e.g., bounding boxes). In the Bayesian optimization framework, $f = f(x, y)$ is assumed to be drawn from a probabilistic model:

$$p\left(f|\mathcal{D}_N\right) \propto p\left(\mathcal{D}_N|f\right) p(f), \tag{1}$$

where $\mathcal{D}_N = \{(y_j, f_j)\}_{j=1}^N$ and $f_j = f(x, y_j)$. Here, the goal is to find a new solution y_{N+1} that maximizes the chance of improving the detection score f_{N+1}, where the chance is often defined as an acquisition function $a(y_{N+1}|\mathcal{D}_N)$. Then, the algorithm proceeds by recursively sampling a new solution y_{N+t} from $\mathcal{D}_{N+(t-1)}$, and update the set $\mathcal{D}_{N+t} = \mathcal{D}_{N+(t-1)} \cup \{y_{N+t}\}$ to draw a new sample solution $y_{N+(t+1)}$ with an updated observation.

Bayesian optimization is efficient in terms of the number of function evaluation [24], and is particularly effective when f is computationally expensive. When $a(y_{N+1}|\mathcal{D}_N)$ is much less expensive than f to evaluate, and the computation for $\arg\max_{y_{N+1}} a(y_{N+1}|\mathcal{D}_N)$ requires only a few function evaluations, we can efficiently find a solution that is getting closer to the ground truth.

3.2. Efficient region proposal via GP regression

A Gaussian process (GP) defines a prior distribution $p(f)$ over the function $f : \mathcal{Y} \to \mathbb{R}$. Due to this property, a distribution over f is fully specified by a mean function $m : \mathcal{Y} \to \mathbb{R}$ and a positive definite covariance kernel $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, i.e., $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$. Specifically, for a finite set $\{y_j\}_{j=1}^N \subset \mathcal{Y}$, the random vector $[f_j]_{1\le j\le N}$ follows a multivariate Gaussian distribution $\mathcal{N}\left([m(y_j)]_{1\le j\le N}, [k(y_i, y_j)]_{1\le i,j\le N}\right)$. A random Gaussian noise with precision β is usually added to each f_j independently in practice. Here, we used the constant mean function $m(y) = m_0$ and the squared exponential covariance kernel with automatic relevance determination (SEard) as follows: $k_{\text{SEard}}(y_i, y_j; z) =$

$$\eta \exp\left(-\frac{1}{2}\left(\Psi_z(y_i) - \Psi_z(y_j)\right)^\top \Lambda \left(\Psi_z(y_i) - \Psi_z(y_j)\right)\right),$$

where Λ is a 4×4 diagonal matrix whose diagonal entries are $\lambda_i^2, i = 1, \cdots, 4$. These form a 7-dimensional GP hyperparameter $\theta = (\beta, m_0, \eta, \lambda_1^2, \lambda_2^2, \lambda_3^2, \lambda_4^2)$ to be learned from the training data. $\Psi_z : \mathcal{Y} \to \mathbb{R}^4$ transforms the bounding box coordinates y into a new form:

$$\Psi_z(y) = \left[\frac{\bar{u}}{\exp(z)}\ ;\ \frac{\bar{v}}{\exp(z)}\ ;\ \log w\ ;\ \log h\right], \tag{2}$$

where $\bar{u} = \frac{u_1+u_2}{2}$ and $\bar{v} = \frac{v_1+v_2}{2}$ denote the center coordinates, $w = u_2 - u_1$ denotes the width, and $h = v_2 - v_1$ denotes the height of a bounding box. We introduce a latent variable z to make the covariance kernel scale-invariant.[3] We determine z in a data-driven manner by maximizing the marginal likelihood of $\mathcal{D}_N$, or

$$\hat{z} = \arg\max_z p(\{f_j\}_{j=1}^N | \{y_j\}_{j=1}^N; \theta). \tag{3}$$

[3] If the image and the bounding boxes y_i, y_j are scaled down by a certain factor, we can keep $k_{\text{SEard}}(y_i, y_j; z)$ invariant by properly setting z.

The GP regression (GPR) problem tries to find a new argument y_{N+1} given N observations $\mathcal{D}_N$ that maximizes the value of acquisition function, which, in our case, is defined with the expected improvement (EI) as: $a_{EI}(y_{N+1}|\mathcal{D}_N) =$

$$\int_{\hat{f}_N}^{\infty} (f - \hat{f}_N) \cdot p(f|y_{N+1}, \mathcal{D}_N; \theta) df \tag{4}$$

where $\hat{f}_N = \max_{1 \le j \le N} f_j$. The posterior of f_{N+1} given $(y_{N+1}, \mathcal{D}_N)$ follows Gaussian distribution: $p(f_{N+1}|y_{N+1}, \mathcal{D}_N; \theta) =$

$$\mathcal{N}\big(f_{N+1}; \mu(y_{N+1}|\mathcal{D}_N), \sigma^2(y_{N+1}|\mathcal{D}_N)\big), \tag{5}$$

with the following mean function and covariance kernels:

$$\begin{aligned}
\mu(y_{N+1}|\mathcal{D}_N) &= m_0 + \mathbf{k}_{N+1}^{\top} \mathbf{K}_N^{-1} \left(\left[f_j - m_0 \right]_{1 \le j \le N} \right), \\
\sigma^2(y_{N+1}|\mathcal{D}_N) &= k_{N+1} - \mathbf{k}_{N+1}^{\top} \mathbf{K}_N^{-1} \mathbf{k}_{N+1}, \\
k_{N+1} &= \beta^{-1} + k(y_{N+1}, y_{N+1}), \\
\mathbf{k}_{N+1} &= \left[k(y_{N+1}, y_j) \right]_{1 \le j \le N}, \\
\mathbf{K}_N &= \left[k(y_i, y_j) \right]_{1 \le i,j \le N} + \beta^{-1} \mathbf{I} \, .
\end{aligned}$$

We refer [33] for detailed derivation. By plugging (5) in (4),

$$\begin{aligned}
&a_{EI}(y_{N+1}|\mathcal{D}_N) = \sigma\left(y_{N+1}|\mathcal{D}_N\right) \times \\
&\big(\gamma(y_{N+1}) F(\gamma(y_{N+1})) + \mathcal{N}(\gamma(y_{N+1}); 0, 1)\big)
\end{aligned} \tag{6}$$

where $\gamma(y_{N+1}) = \frac{\mu(y_{N+1}|\mathcal{D}_N) - \hat{f}_N}{\sigma(y_{N+1}|\mathcal{D}_N)}$. $F(\cdot)$ is the cumulative distribution function of standard normal distribution $\mathcal{N}(\cdot)$.

3.3. Local fine-grained search

In this section, we extend the GPR-based algorithm for global maximum search to local fine-grained search (FGS).

The local FGS steps are described in Figure 1. We perform the FGS by pruning out easy negatives with low classification scores from the set of regions proposed by the selective search algorithm and sorting out a few bounding boxes with the maximum scores in local regions. Then, for each local optimum y_{best} (red boxes in Figure 1), we propose a new candidate bounding box (green boxes in Figure 1). Specifically, we initialize a set of local observations $\mathcal{D}_{\text{local}}$ for y_{best} from the set given by the selective search algorithm, whose localness is measured by an IoU between y_{best} and region proposals (yellow boxes[4] in Figure 1). $\mathcal{D}_{\text{local}}$ is used to fit a GP model, and the procedure is iterated for each local optimum at different levels of IoU until there is no more acceptable proposal. We provide a pseudocode of local FGS in Algorithm 1, where the parameters are set as: $t_{\max} = 8$, $(\rho_r)_{r=1}^{R=3} = (0.3, 0.5, 0.7)$.

In addition to the capability of handling multiple objects in a single image, better computational efficiency is another factor making local FGS preferable to global search. As a kernel method, the computational complexity of GPR increases cubically to the number of observations. By restricting the observation set to the nearby region of a local optimum, the GP fitting and proposal process can be performed efficiently. In practice, FGS introduces only $< 20\%$ computational overhead compared to the original R-CNN. Please see the supplementary materials, which are also available in our technical report [47], for more details on its practical efficiency (Sec. S4).

[4] In practice, the local search region associated with $\mathcal{D}_{\text{local}}$ is not a rectangular region around local optimum since we use IoU to determine it.

Algorithm 1 Local fine-grained search (FGS)

Require: Image x, classifier f_{CNN}, a set of structured labels and classification scores $\mathcal{D}_N = \{(y_j, f_j)_{j=1}^N\}$, GP hyperparameter θ, maximum number of GP iterations $t_{\max}$, a threshold f_{prune} to prune out the bounding boxes, different levels of IoU $\rho_r, r = 1, \ldots, R$ determining the size of local regions.
Ensure: A set of structured labels and classification scores $\mathcal{D}$.

$\mathcal{D} \leftarrow \mathcal{D}_N$
for $t = 1, \cdots, t_{\max}$ **do**
  $\mathcal{D}_{\text{proposal}} = \varnothing$
  $\mathcal{D}_{\text{prune}} = \{(y, f) \in \mathcal{D} : f > f_{\text{prune}}\}$
  $\mathcal{D}_{\text{NMS}} = \text{NMS}(\mathcal{D}_{\text{prune}})$
  for each $(y_{\text{best}}, f_{\text{best}}) \in \mathcal{D}_{\text{NMS}}$ **do**
    for $r = 1, \cdots, R$ **do**
      $\mathcal{D}_{\text{local}} = \{(y, f) \in \mathcal{D} : \text{IoU}(y, y_{\text{best}}) > \rho_r\}$
      $\hat{z} = \arg\max_z p(\mathcal{D}_{\text{local}}; \theta)$ (Equation (3))
      $\hat{y} = \arg\max_y a_{EI}(y|\mathcal{D}_{\text{local}}; \theta, \hat{z})$ (Equation (6))
      $\hat{f} = f_{\text{CNN}}(x, \hat{y})$
      $\mathcal{D}_{\text{proposal}} \leftarrow \mathcal{D}_{\text{proposal}} \cup \{(\hat{y}, \hat{f})\}$
    end for
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{proposal}}$
end for

3.4. Learning GP hyperparameter

As we locally perform the FGS, the GP hyperparameter θ also needs to be trained with observations in the vicinity of ground truth objects. To this end, for an annotated object in the training set, we form a set of observations with the structured labels and corresponding classification scores of the bounding boxes that are close to the ground truth bounding box. Such an observation set is composed of the bounding boxes (given by selective search and random selection) whose IoU with the ground truth exceed a certain threshold. Finally, we fit a GP model by maximizing the joint likelihood of such observations:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i \in I_{\text{pos}}} \log p(\{(y, f) : y \in \widetilde{\mathcal{Y}}_i, f = f(x_i, y)\}; \theta),$$

where I_{pos} is the index set for positive training samples (i.e., with ground truth object annotations), and y_i is a ground

truth annotation of an image x_i.[5] We set $\widetilde{\mathcal{Y}}_i = \{y = y_i \text{ or } y \in \mathcal{Y}_i \text{ or } y \in \bar{\mathcal{Y}}_i : \text{IoU}(y, y_i) > \rho\}$, where $\mathcal{Y}_i$ consists of the bounding boxes given by selective search on x_i, $\bar{\mathcal{Y}}_i$ is a random subset of $\mathcal{Y}$, and ρ is the overlap threshold. The optimal solution $\hat{\theta}$ can be obtained via L-BFGS. Our implementation relies on the GPML toolbox [33].

4. Learning R-CNN with structured loss

This section describes a training algorithm of R-CNN for object detection using structured loss. We first revisit the object detection framework with structured output regression introduced by Blaschko and Lampert [5] in Section 4.1, and extend it to R-CNN pipeline that allows training the network with structured hinge loss in Section 4.2.

4.1. Structured output regression for detection

Let $\{x_1, x_2, \ldots, x_M\}$ be the set of training images and $\{y_1, y_2, \ldots, y_M\}$ be the set of corresponding structured labels. The structured label $y \in \mathcal{Y}$ is composed of 5 elements (l, u_1, v_1, u_2, v_2); when $l = 1$, (u_1, v_1) and (u_2, v_2) denote the top-left and bottom-right coordinates of the object, respectively, and when $l = -1$, it implies that there is no object in x_i, and there is no meaning on coordinate elements (u_1, v_1, u_2, v_2). Note that the definition of y_i is extended from Section 3 to indicate the presence of an object (l) as well as its location (u_1, v_1, u_2, v_2) when exists. When there are multiple objects in an image, we crop an image into multiple positive ($l = 1$) images, each of which contains a single object, and a negative image ($l = -1$) that doesn't contain any object.[6] Let $\phi(x, y)$ represent the feature extracted from an image x for a label y with $l = 1$. In our case, $\phi(x, y)$ denotes the top-layer representations of the CNN (excluding the classification layer) at location specified by y,[7] which are fed into the classification layer. The detection problem is to find a structured label $y \in \mathcal{Y}$ that has the highest score:

$$g(x; w) = \arg\max_{y \in \mathcal{Y}} f(x, y; w) \tag{7}$$

where

$$f(x, y; w) = w^T \widetilde{\phi}(x, y), \tag{8}$$

$$\widetilde{\phi}(x, y) = \begin{cases} \phi(x, y) & , \ l = +1 \ , \\ \mathbf{0} & , \ l = -1 \ . \end{cases} \tag{9}$$

Note that (9) includes a trick for setting the detection threshold to 0. The model parameter w is trained to minimize the structured loss $\Delta(\cdot, \cdot)$ between the predicted label $g(x_i; w)$ and the ground-truth label y_i:

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{M} \Delta(g(x_i; w), y_i) \tag{10}$$

For the detection problem, the structured loss $\Delta(y, y_i)$ is defined in terms of intersection over union (IoU) of two bounding boxes defined by y and y_i as follows:

$$\Delta(y, y_i) = \begin{cases} 1 - \text{IoU}(y, y_i) & , \text{ if } l = l_i = 1, \\ 0 & , \text{ if } l = l_i = -1, \\ 1 & , \text{ if } l \neq l_i \end{cases} \tag{11}$$

where $\text{IoU}(y, y_i) = \frac{\text{Area}(y \cap y_i)}{\text{Area}(y \cup y_i)}$. In general, the optimization problem (10) is difficult to solve due to the complicated form of structured loss. Instead, we formulate the surrogate objective in structured SVM framework [43] as follows:

$$\min_{w} \ \frac{1}{2}\|w\|^2 + \frac{C}{M} \sum_{i=1}^{M} \xi_i \quad \text{, subject to} \tag{12}$$

$$w^\top \widetilde{\phi}(x_i, y_i) \geq w^\top \widetilde{\phi}(x_i, y) + \Delta(y, y_i) - \xi_i, \forall y, \forall i \tag{13}$$

$$\xi_i \geq 0, \forall i \tag{14}$$

Using (9) and (11), the constraint (13) is written as follows:

$$w^\top \phi(x_i, y_i) \geq w^\top \phi(x_i, y) + \Delta^{\text{loc}}(y, y_i) - \xi_i, \quad \forall y \in \mathcal{Y}^{(l=1)}, \forall i \in I_{\text{pos}}, \tag{15}$$

$$w^\top \phi(x_i, y_i) \geq 1 - \xi_i, \ \forall i \in I_{\text{pos}}, \tag{16}$$

$$w^\top \phi(x_i, y) \leq -1 + \xi_i, \ \forall y \in \mathcal{Y}^{(l=1)}, \forall i \in I_{\text{neg}}, \tag{17}$$

where $\Delta^{\text{loc}}(y, y_i) = 1 - \text{IoU}(y, y_i)$, I_{pos} and I_{neg} denote the set of indices for positive and negative training examples, respectively, and $\mathcal{Y}^{(l=1)} = \{y \in \mathcal{Y} : l = 1\}$.

4.2. Gradient-based learning of R-CNN with structured SVM objective

To learn the R-CNN detector with structured loss, we propose to make several modifications to the original structured SVM formulation. First, we restrict the output space $\mathcal{Y}_i \subset \mathcal{Y}$ of ith example to regions proposed via selective search. This results in a change in notation for every $\mathcal{Y}$ in (15) and (17) of ith example to $\mathcal{Y}_i$. Second, the constraints (15, 16, 17) should be transformed into hinge loss to backpropagate the gradient to lower layers of CNN. Specifically, the objective function (12) is reformulated as follows:

$$\min_{w} \ \frac{1}{2}\|w\|^2 + \frac{1}{M}\Big(C_1 \sum_{i \in I_{\text{pos}}} h_{\text{pos},i} + C_2 \sum_{i \in I_{\text{neg}}} h_{\text{neg},i}\Big) \tag{18}$$

where $h_{\text{pos},i} = h_{\text{pos},i}(w)$, $h_{\text{neg},i} = h_{\text{neg},i}(w)$ are given as:

$$h_{\text{pos},i}(w) = \max\Big\{0, 1 - w^\top \phi(x_i, y_i), \quad \max_{y \in \mathcal{Y}_i^{(l=1)}} \big(w^\top (\phi(x_i, y) - \phi(x_i, y_i)) + \Delta^{\text{loc}}(y, y_i)\big)\Big\} \tag{19}$$

[5] We assumed one object per image. See Section 4.2 for handling multiple objects in training.

[6] We also perform the same procedure for images with a single object during the training.

[7] Following [18], we crop and warp the image patch of x at location given by y to a fixed size (e.g., 224×224) to compute the CNN features.

$$h_{\text{neg},i}(w) = \max\left\{0,\ \max_{y \in \mathcal{Y}_i^{(l=1)}} (1 + w^\top \phi(x_i, y))\right\} \quad (20)$$

Note that we use different C values for positive and negative examples. In experiments, $C_1 = 2$ and $C_2 = 1$.

Structured SVM objective may cause a slow convergence in parameter estimation since it utilizes at most one instance y among a large number of instances in the (restricted) output space $\mathcal{Y}_i$, whose size varies from few hundreds to thousands. To overcome this issue, we alternately perform a gradient-based parameter estimation and hard negative data mining that effectively adapts the number of training examples to be evaluated for updating the parameters (Sec. S2 in the supplementary materials).

For model parameter estimation, we use L-BFGS to first learn parameters of the classification layer only. We found that this already resulted in a good detection performance. Then, we optionally use stochastic gradient descent to finetune the whole CNN classifiers (Sec. S1 in the supplementary materials).

5. Experimental results

We applied our proposed methods to standard visual object detection tasks on PASCAL VOC 2007 [12] and 2012 [14]. In all experiments, we consider R-CNNs [18] as baseline models. Following [18], we used the CNN models pretrained on ImageNet database [9] with $1,000$ object categories [25, 39], and finetuned the whole network using the target database by replacing the existing softmax classification layer to a new one with a different number of classes (e.g., 20 classes for VOC 2007 and 2012). We provide the learning details in the supplementary materials (Sec. S3). Our implementation is based on the Caffe toolbox [23].

Setting the R-CNN as a baseline method, we compared the detection performance of our proposed methods, such as R-CNN with FGS (R-CNN + FGS), R-CNN trained with structured SVM objective (R-CNN + StructObj), and their combination (R-CNN + StructObj + FGS). Since our goal is to localize the bounding boxes more accurately at the object regions, we also consider the IoU of 0.7 for an evaluation criterion, which only counts the detection results as correct when the overlap between the predicted bounding box and the ground truth is greater than 70%. This is more challenging than common practices (e.g., IoU ≥ 0.5), but will be a good indicator for a better localization of an object bounding box if successful.

5.1. FGS efficacy test with oracle detector

Before reporting the performance of the proposed methods in R-CNN framework, we demonstrate the efficacy of FGS algorithm using an oracle detector. We design a hypothetical oracle detector whose score function is defined as $f_{\text{ideal}}(x_i, y) = \text{IoU}(y, y_i)$, where y_i is a ground truth annotation for an image x_i. The score function is ideal in the sense that it outputs high scores for bounding boxes with high overlap with the ground truth and vice versa, overall achieving 100% mAP.

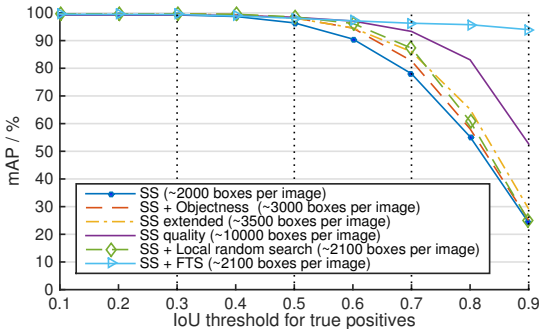

Figure 2: mAP at different IoU criteria on PASCAL VOC 2007 test set using an oracle detector. We used different numbers of bounding boxes proposed by selective search (SS) [44], Objectness [2], local random search, and our proposed FTS methods. The average numbers of bounding boxes used for evaluation are specified.

We summarize the results in Figure 2. We report the performance on the VOC 2007 test set at different levels of IoU criteria $(0.1, \cdots, 0.9)$ for the baseline selective search (SS; "fast mode" in [44]), selective search with objectness [2] (SS + Objectness), selective search with extended superpixel similarity measurements (SS extended) [44], "quality mode" of selective search (SS quality) [44], local random search,[8] and the proposed FGS method with the baseline selective search.

For low values of IoU (≤ 0.3), all methods using the oracle detectors performed almost perfectly due to the ideal score function. However, we found that the detection performance with different region proposal schemes other than our proposed FGS algorithm start to break down at high values of IoU. For example, the performance of SS, SS + Objectness, SS extended, and local random search methods, which used around $2,000 \sim 3,500$ bounding boxes per image in average, significantly dropped at IoU ≥ 0.5. SS quality method kept pace with the FGS method until IoU of 0.6, but again, the performance started to drop at IoU ≥ 0.7.

On the other hand, the performance of FGS dropped 5% in mAP at IoU of 0.9 by only introducing approximately 100 new bounding boxes per image. Given that SS quality requires $10,000$ region proposals per image, our proposed FGS method is much more computationally efficient ($\sim 80\%$ less bounding boxes) while localizing the bounding boxes much more accurately. This provides an insight that, if the detector is accurate, our Bayesian optimization framework would limit the number of bounding boxes to a manageable number (e.g., few thousands per image on average) to achieve almost perfect detection results.

[8] Like local FGS, local random search first determine the local search regions by NMS. However, it randomly choose a fixed number of bounding box in those regions rather than sequentially proposing new boxes based on some informed method.

Model	BBoxReg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN (AlexNet)	No	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 (VGG)	No	68.5	74.5	61.0	37.9	40.6	69.2	73.7	69.9	37.2	68.6	56.8	70.6	69.0	67.1	59.6	33.4	63.9	58.9	62.6	68.5	60.6
+ StructObj	No	68.7	73.5	62.6	40.6	41.5	69.6	73.5	71.1	39.9	69.6	58.1	70.0	67.5	69.8	59.8	35.9	63.6	59.0	62.6	67.7	61.2
+ StructObj-FT	No	69.3	75.2	62.2	39.4	42.3	70.7	74.5	74.3	40.4	71.3	59.8	72.0	69.8	69.4	60.3	35.3	64.5	**62.0**	63.7	69.8	62.3
+ FGS	No	70.6	78.4	65.7	46.2	48.8	**74.6**	77.0	74.3	42.7	70.8	60.9	75.1	75.8	70.7	66.3	37.1	66.3	57.6	66.6	71.0	64.8
+ StructObj + FGS	No	**73.4**	**80.9**	64.5	**46.7**	49.1	73.9	78.2	76.8	44.8	**75.3**	**63.0**	75.3	74.2	72.7	**68.5**	37.0	67.5	58.1	**66.9**	70.5	65.9
+ StructObj-FT + FGS	No	72.5	78.8	**67.0**	45.2	**51.0**	73.8	**78.7**	**78.3**	**46.7**	73.8	61.5	**77.1**	**76.4**	**73.9**	66.5	**39.2**	**69.7**	59.4	66.8	**72.9**	**66.5**
R-CNN (AlexNet)	Yes	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 (VGG)	Yes	70.8	77.1	**69.4**	45.8	48.4	74.0	77.0	75.0	42.2	72.5	61.5	75.6	77.7	66.6	65.3	39.1	65.8	64.2	68.6	71.5	65.4
+ StructObj	Yes	73.1	77.5	69.2	47.6	47.6	74.5	78.2	75.4	44.5	76.3	64.9	76.7	76.3	69.9	68.1	39.4	67.0	**65.6**	68.7	70.9	66.6
+ StructObj-FT	Yes	72.6	79.4	**69.4**	45.2	47.8	74.4	77.8	76.5	45.4	76.3	61.4	**80.2**	77.1	73.8	66.8	41.1	67.8	64.7	67.9	72.3	66.9
+ FGS	Yes	**74.2**	78.9	67.8	**51.6**	52.3	75.7	78.7	76.6	45.4	72.4	63.1	76.6	**79.3**	70.7	68.0	40.3	67.8	61.8	**70.2**	71.6	67.2
+ StructObj + FGS	Yes	74.1	**83.2**	67.0	50.8	51.6	**76.2**	**81.4**	77.2	48.1	**78.9**	**65.6**	77.3	78.4	**75.1**	**70.1**	41.4	69.6	60.8	**70.2**	**73.7**	**68.5**
+ StructObj-FT + FGS	Yes	71.3	80.5	69.3	49.6	**54.2**	75.4	80.7	**79.4**	**49.1**	76.0	65.2	79.4	78.4	75.0	68.4	**41.6**	**71.3**	61.2	68.2	73.3	68.4

Table 1: Test set mAP of VOC 2007 with IoU > 0.5. The entries with the best APs for each object category are bold-faced.

Model	BBoxReg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN (AlexNet)	No	32.9	40.1	19.7	18.7	11.1	39.4	40.5	26.5	14.8	29.8	24.5	26.4	23.7	31.9	18.5	**13.3**	27.6	25.8	26.6	39.5	26.6
R-CNN (VGG)	No	40.2	43.3	23.4	14.4	13.3	48.2	44.5	36.4	17.1	34.0	27.9	36.3	26.8	28.2	21.2	10.3	33.7	36.6	31.6	48.9	30.8
+ StructObj	No	42.5	44.4	24.5	17.8	15.3	46.8	46.4	37.9	17.6	33.4	26.6	36.8	24.3	31.5	21.3	10.4	30.0	36.1	30.6	46.3	31.0
+ StructObj-FT	No	44.1	47.1	23.4	16.6	16.4	50.1	48.7	39.7	18.4	39.4	28.6	38.6	27.5	32.4	23.6	11.1	33.1	**41.0**	34.3	49.6	33.2
+ FGS	No	44.3	55.5	28.9	19.1	22.9	56.9	57.6	37.8	19.6	35.7	**31.9**	38.1	**43.0**	42.7	30.3	9.8	**42.3**	33.3	43.4	55.4	37.4
+ StructObj + FGS	No	43.5	56.1	30.9	18.7	24.9	55.2	57.6	38.9	20.7	38.6	28.4	37.7	38.7	**46.3**	30.9	8.4	37.6	37.0	42.2	51.3	37.2
+ StructObj-FT + FGS	No	**46.3**	**58.1**	**31.1**	**21.6**	**25.8**	**57.1**	**58.2**	**43.5**	**23.0**	**46.4**	29.0	**40.7**	40.6	**46.3**	**33.4**	10.6	41.3	40.9	**45.8**	**56.3**	**39.8**
R-CNN (AlexNet)	Yes	47.6	48.7	25.3	**25.0**	17.3	53.4	54.6	36.8	16.7	42.3	31.6	35.8	38.0	41.8	24.5	14.3	38.8	28.9	34.0	49.0	35.2
R-CNN (VGG)	Yes	45.1	48.6	26.0	18.2	21.2	57.2	52.4	37.3	20.1	33.7	31.9	38.8	39.6	36.3	26.5	9.2	37.8	33.4	39.4	50.7	35.2
+ StructObj	Yes	49.4	56.5	**36.5**	21.3	23.3	61.0	58.1	44.3	20.8	47.4	33.3	39.8	40.7	45.9	31.0	**14.7**	39.6	42.9	45.7	56.9	40.5
+ StructObj-FT	Yes	49.3	58.1	35.4	23.3	24.4	62.3	60.1	45.8	21.8	48.7	32.4	41.8	43.2	45.7	32.0	14.4	44.6	**45.1**	48.6	59.8	41.8
+ FGS	Yes	50.9	59.8	34.4	20.9	**31.6**	**66.1**	62.3	44.9	22.0	46.5	**36.8**	42.5	51.4	46.8	34.1	13.5	**44.7**	39.1	48.9	57.7	42.7
+ StructObj + FGS	Yes	**53.6**	60.7	32.1	19.9	31.3	63.2	63.2	46.4	23.6	**53.0**	34.9	40.4	**53.6**	**49.9**	34.6	10.2	42.2	40.1	48.3	58.3	43.0
+ StructObj-FT + FGS	Yes	47.1	**61.8**	35.2	18.1	29.7	66.0	**64.7**	**48.0**	**25.3**	50.4	34.9	**43.7**	50.8	49.4	**36.8**	13.7	**44.7**	43.6	**49.8**	**60.5**	**43.7**

Table 2: Test set mAP of VOC 2007 with IoU > 0.7. The entries with the best APs for each object category are bold-faced.

We also report similar experimental analysis for the real detector trained with the proposed structured objective in the supplementary materials (Sec. S6).

5.2. PASCAL VOC 2007

In this section, we demonstrate the performance of our proposed methods on PASCAL VOC 2007 [12] detection task (comp4), a standard benchmark for object detection problem. Similarly to the training pipeline of R-CNN [18], we finetuned the CNN models (with softmax classification layer) pretrained on ImageNet database using images from both train and validation sets of VOC 2007 and further trained the network with linear SVM (baseline) or the proposed structured SVM objective. We evaluated on the test set using the proposed FGS algorithm. For post-processing, we performed NMS and bounding box regression [18].

Figure 3 shows representative examples of successful detection using our method. For these cases, our method can localize objects accurately even if the initial bounding box proposals don't have good overlaps with the ground truth. We show more examples (including the failure cases) in the supplementary materials (Sec. S9, S10, S11).

The summary results are in Table 1 with IoU criteria of 0.5 and Table 2 with 0.7. We report the performance with the AlexNet [25] and the VGGNet (16 layers) [39], a deeper CNN model than AlexNet that showed a significantly better recognition performance and achieved the best performance on object localization task in ILSVRC 2014.[9] First of all, we observed the significant performance improvement by simply having a better CNN model. Building upon the VGGNet, the FGS improved the performance by 4.2% and 1.8% in mAP without and with bounding box regression (Table 1). It becomes much more significant when we consider IoU criteria of 0.7 (Table 2), improving upon the baseline model by 6.6% and 7.5% in mAP without and with bounding box regression. The results demonstrate that our FGS algorithm is effective in accurately localizing the bounding box of an object.

Further improvement has been made by training a classifier with structured SVM objective; we obtained $\mathbf{68.5}\%$ mAP in IoU criteria of 0.5, which, to our knowledge, is higher than the best published results, and $\mathbf{43.0}\%$ mAP in IoU criteria of 0.7 with FGS and bounding box regression by training the classification layer only ("StructObj"). By finetuning the whole CNN classifiers ("StructObj-FT"), we observed extra improvement for most cases; for example, we obtained $\mathbf{43.7}\%$ mAP in IOU criteria of 0.7, which im-

[9] The 16-layer VGGNet can be downloaded from: https://gist.github.com/ksimonyan/211839e770f7b538e2d8.

Model	BBoxReg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN (AlexNet)	No	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6	49.6
R-CNN (VGGNet)	No	76.3	69.8	57.9	40.2	37.2	64.0	63.7	80.2	36.1	63.6	47.3	81.1	71.2	73.8	59.5	30.9	64.2	52.2	62.4	58.7	59.5
R-CNN (AlexNet)	Yes	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1	53.3
R-CNN (VGGNet)	Yes	79.2	72.3	62.9	43.7	45.1	67.7	66.7	83.0	39.3	66.2	51.7	82.2	73.2	76.5	64.2	33.7	66.7	56.1	68.3	61.0	63.0
+ StructObj	Yes	80.9	74.8	62.7	42.6	46.2	70.2	68.6	84.0	42.2	68.2	54.1	82.2	74.2	79.8	66.6	39.3	67.6	**61.0**	71.3	65.2	65.1
+ FGS	Yes	80.5	73.5	**64.1**	**45.3**	48.7	66.5	68.3	82.8	39.8	68.2	52.7	82.1	75.1	76.6	66.3	35.5	66.9	56.8	68.7	61.6	64.0
+ StructObj + FGS	Yes	**82.9**	**76.1**	**64.1**	44.6	**49.4**	**70.3**	**71.2**	**84.6**	**42.7**	68.6	**55.8**	**82.7**	**77.1**	**79.9**	**68.7**	**41.4**	**69.0**	60.0	**72.0**	**66.2**	**66.4**
NIN [29]	-	80.2	73.8	61.9	43.7	43.0	**70.3**	67.6	80.7	41.9	**69.7**	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3	63.8

Table 3: Test set mAP of VOC 2012 with IoU > 0.5. The entries with the best APs for each object category are bold-faced.

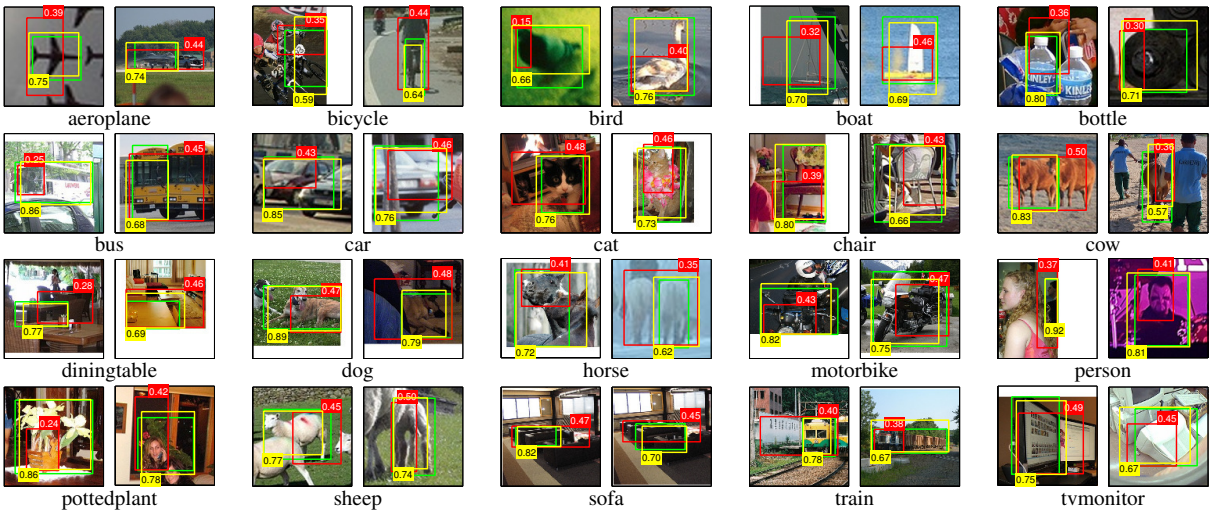

Figure 3: Detection examples from PASCAL VOC 2007 test set. Two examples from 20 object categories are shown, with the ground truth bounding boxes (green), the boxes obtained by baseline R-CNN (VGGNet) (red), and those obtained by the proposed R-CNN + StructObj + FGS (yellow). The numbers near the bounding boxes denote the IoU with the ground truth.

proves by 0.7% in mAP over the method without finetuning. However, for IoU$>$0.5 criterion, the overall improvement due to finetuning was relatively small, especially when using bounding box regression. In this case, considering the high computational cost for finetuning, we found that training only the classification layer is practically a sufficient way to learn a good localization-aware classifier.

We provide in-depth analysis of our proposed methods in the supplementary materials. Specifically, we report the precision-recall curves of different combinations of the proposed methods (Sec. S7), the performance of FGS with different GP iterations (Sec. S5), the analysis of localization accuracy (Sec. S8), and more detection examples.

5.3. PASCAL VOC 2012

We also evaluate the performance of the proposed methods on PASCAL VOC 2012 [14]. As the data statistics are similar to VOC 2007, we used the same hyperparameters as described in Section 5.2 for this experiment. We report the test set mAP over 20 object categories in Table 3. Our proposed method shows improvement by 2.1% with R-CNN + StructObj and 1.0% with R-CNN + FGS over baseline R-CNN using VGGNet. Finally, we obtained $\mathbf{66.4}\%$ mAP by combining the two methods, which significantly improved upon the baseline R-CNN model and the previously published results on the leaderboard.

6. Conclusion

In this work, we proposed two complementary methods to improve the performance of object detection in R-CNN framework with 1) fine-grained search algorithm in a Bayesian optimization framework to refine the region proposals and 2) a CNN classifier trained with structured SVM objective to improve localization. We demonstrated the state-of-the-art detection performance on PASCAL VOC 2007 and 2012 benchmarks under standard localization requirements. Our methods showed more significant improvement with higher IoU evaluation criteria (e.g., IoU $= 0.7$), and hold promise for mission-critical applications that require highly precise localization, such as autonomous driving, robotic surgery and manipulation.

Acknowledgments

This work was supported by Samsung Digital Media and Communication Lab, Google Faculty Research Award, ONR grant N00014-13-1-0762, China Scholarship Council, and Rackham Merit Fellowship. We also acknowledge NVIDIA for the donation of GPUs. Finally, we thank Scott

Reed, Junhyuk Oh, Xinchen Yan, Roni Mittelman, Wenling Shang, and Ye Liu for helpful discussions.

References

[1] T. Ahonen, A. Hadid, and M. Pietikinen. Face recognition with local binary patterns. In *ECCV*, 2004. 2

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, Nov 2012. 6

[3] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. In *NIPS*, 2007. 1

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, Aug 2013. 1

[5] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008. 2, 5

[6] Y.-l. Boureau, Y. L. Cun, et al. Sparse feature learning for deep belief networks. In *NIPS*, pages 1185–1192, 2008. 1

[7] Q. Dai and D. Hoiem. Learning to localize detected objects. In *CVPR*, 2012. 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2

[9] 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. 6

[10] 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. *CoRR*, abs/1310.1531, 2013. 2

[11] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014. 1, 2

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007. 1, 6, 7

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010. 1

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012. 6, 8

[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 2

[16] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 2

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3, 5, 6, 7

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

[20] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv:1406.4729*, 2014. 1

[21] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504–507, 2006. 1

[22] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012. 1

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014. 6

[24] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001. 3

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 6, 7

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 1

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011. 1

[28] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. In *Robotics Science and Systems*, 2013. 1

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

[30] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91–110, 2004. 2

[31] J. Mockus, V. Tiesis, and A. Zilinskas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978. 1

[32] B. Pepikj, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *CVPR*, 2013. 2

[33] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. 4, 5

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1

[35] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. 1

[36] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof. Accurate object detection with joint classification-regression random forests. In *CVPR*, 2014. 2

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

[38] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 1, 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2, 6, 7

[40] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *NIPS*, 2012. 1

[41] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013. 1, 2

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1

[43] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, (6): 1453–1484, 2005. 5

[44] J. R. R. Uijlings, K. E. A. Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 1, 2, 3, 6

[45] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, pages 17–24, Dec 2013. 2

[46] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 1

[47] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. *CoRR*, abs/1504.03293, 2015. URL http://arxiv.org/abs/1504.03293. 4