Structural Kernel Learning for Large Scale Multiclass Object Co-Detection

Zeeshan Hayder[1,2], Xuming He[2,1]
[1]Australian National University & [2]NICTA *
{zeeshan.hayder, xuming.he}@anu.edu.au

Mathieu Salzmann[2,3]
[3]CVLab, EPFL, Switzerland
mathieu.salzmann@epfl.ch

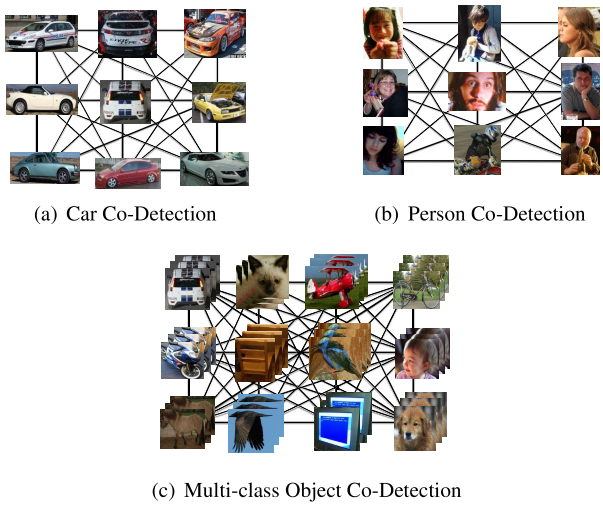
Abstract

Exploiting contextual relationships across images has recently proven key to improve object detection. The resulting object co-detection algorithms, however, fail to exploit the correlations between multiple classes and, for scalability reasons are limited to modeling object instance similarity with relatively low-dimensional hand-crafted features. Here, we address the problem of multiclass object co-detection for large scale datasets. To this end, we formulate co-detection as the joint multiclass labeling of object candidates obtained in a class-independent manner. To exploit the correlations between objects, we build a fully-connected CRF on the candidates, which explicitly incorporates both geometric layout relations across object classes and similarity relations across multiple images. We then introduce a structural boosting algorithm that lets us exploit rich, high-dimensional deep network features to learn object similarity within our fully-connected CRF. Our experiments on PASCAL VOC 2007 and 2012 evidence the benefits of our approach over object detection with RCNN, single-image CRF methods and state-of-the-art co-detection algorithms.

1. Introduction

Exploring contextual relations is one of the key factors to improve object detection under challenging viewing conditions and to scale up recognition to large numbers of object classes. Object co-detection, which jointly detects object instances in a set of related images, constitutes an important step towards utilizing large-scale context beyond individual images [3]. Recent efforts have achieved promising results on challenging detection benchmarks by learning instance similarity within object classes [13, 23, 12].

Despite this progress, most existing object co-detection methods focus on single-class object detection and treat the multiclass scenario as a set of unrelated tasks. As such, they are unable to capture the correlations between co-occurring object categories, or exploit their spatial and semantic relations. In addition, due to their approach to learning instance similarities, these techniques are confined to employing relatively low-dimensional hand-crafted object features, such as color or LBP histograms. Unfortunately, such simple feature descriptors lack the necessary representation power to capture rich characteristics and similarities between instances of multiple object categories.



(a) Car Co-Detection (b) Person Co-Detection

(c) Multi-class Object Co-Detection

Figure 1. Conventional single-class object co-detection vs. our multi-class object co-detection approach.

In this paper, as illustrated in Fig. 1, we tackle the multiclass object co-detection problem at large scale. To this end, we take a hypothesize-and-classify approach and introduce a joint labeling framework that addresses the limitations of the single-class co-detection. Given a large set of class-independent object candidates, we formulate multiclass object co-detection as the task of assigning each object candidate to either one of the target classes, or background. To exploit the correlation between objects, we explicitly incorporate two types of object relations: geometric layout relations between object classes within an image; and similarity relations between object instances across multiple images. To capture complex relationships between objects, we make use of learned CNN features, and introduce a principled approach to learning similarities with such high-dimensional

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the ARC through the ICT Centre of Excellence program.

descriptors.

More specifically, we generate class-independent object candidates based on an objectness measure [26] that does not require any pre-trained detectors, and build a fully-connected Conditional Random Field (CRF) on these candidates. The unary potentials of this CRF are then obtained via a deep convolutional neural network trained to generate class confidence scores [9]. Furthermore, we make use of two types of edge potentials: one that encodes the class-dependent spatial relations between two object instances in an image [5], and one that encourages a similarity-based label smoothness between two object instances across the entire set of candidates.

An important focus of this work is the design of an object similarity measure, and more precisely, the study of how high-dimensional deep learning features, that have proven highly discriminative, can be effectively employed to model the similarity of object instances in our pairwise potential. Indeed, while inference in fully-connected CRFs can be performed efficiently by making use of Gaussian kernels in the pairwise potentials, this strategy scales poorly with the dimensionality of the features used in the kernels. To address this issue, we adopt a structural boosting approach to learning our pairwise potentials. Our learning strategy incrementally adds low-dimensional Gaussian kernels to the CRF model so as to optimize the overall labeling performance. We develop two structural boosting algorithms that encode different measures of such performance: one based on a max-margin criterion with Hamming loss, and one that minimizes the KL-divergence between the marginals of the CRF and the overlap ratios of the object candidates with the ground-truth. The resulting pairwise potential, in the form of a weighted sum of low-dimensional Gaussian kernels, allows us to perform inference efficiently, which is critical for multiclass object co-detection.

We evaluate our method on two large-scale object detection datasets: PASCAL VOC 2007 and PASCAL VOC 2012. Our experiments demonstrate that our approach outperforms state-of-the-art methods on several standard metrics. This evidences the importance of learning rich similarity measures to account for the contextual relations across object classes and instances.

2. Related work

Putting objects into context has been widely studied to improve the robustness of detectors by exploiting the co-occurrence of objects and scene properties within an image [15, 8]. In particular, object-object relations have been integrated into several multiclass object detection systems. For example, Desai et al. [5] propose to jointly detect multiple object classes by defining a CRF on top of DPMs, in which the relative geometric relationships among 20 classes are captured. Choi et al. [4] build a tree-structured model to encode both the co-occurrence statistics and relative spatial locations of multiple object classes. While these methods have shown the benefits of object context, they focus on modeling the context from a single image. Our approach incorporates contextual information from all the images.

Object co-detection was first introduced in [3] to exploit the collective power of a set of images in object detection. Bao et al. [3] take an energy minimization approach that integrates potential object correspondences at object and part level, and exhaustively searches for matched object instances in a set of object candidates. While they consider general settings for both 2D and 3D object models at category and instance levels, their method only handles pairs of images. Recent work [12, 23, 13] extend object co-detection to the multi-image setting, ranging from a few frames (e.g., [23]) to the large-scale PASCAL VOC dataset [12]. One key aspect of object co-detection is the modeling of object similarity across images. Guo et al. [12] introduce a robust approach to co-detection that builds a shared low-rank representation of the object instances in multiple feature spaces, such as SIFT and LBP histogram. Shi et al. [23] incrementally learn a Gaussian Process classifier to measure instance similarity. In [13], we propose to learn a category-level similarity function based on color and LBP histograms. These similarity learning approaches, however, scale poorly to the dimensionality of the features, and are thus mostly limited to using relatively low-dimensional hand-crafted features. Instead, here, we design a learning framework that lets us make use of a rich representation of objects from different classes from a deep neural network. More importantly, while all existing co-detection methods consider a single object class at a time, we also model the object relations across multiple classes.

Our work is inspired by the fully-connected CRF model and its learning algorithm [16, 17, 28, 29]. The fully-connected CRF restricts the functional form of the weights in its pairwise potential to a weighted mixture of Gaussian kernels, which allows efficient inference based on fast Gaussian convolution [2]. In practice, the efficiency critically depends on the dimensionality of the input feature space and deteriorates quickly with higher dimensional features. Our work develops a structural boosting approach based on functional gradient descent [19, 21], in which we incrementally learn a set of weighted Gaussian kernels defined on low-dimensional feature spaces. In [13], we also learn a mixture of weighted kernels for a fully connected CRF. However, in that work, learning is treated as a separate regression problem without considering the CRF framework. Furthermore, our previous learning strategy does not scale up to large-scale multiclass object co-detection. Note that, here, our goal is not to build a kernel-based similarity classifier as in [11, 27, 23], since this would not yield a mixture of Gaussian kernels adapted to our fully-connected

CRF framework.

Built on the recent success of deep learning in object recognition [18, 9], our approach exploits the output of the RCNN [9] as unary potentials. Note, however, that other context-free object detectors or deep network classifiers [25, 24, 20, 22] could also be adopted in our framework, and may further improve the performance. Finally, [30] also proposes to incorporate context information within a deep network based detection framework. However, the resulting method focuses only on exploiting the context around an object hypothesis, and thus does not exploit the collective information contained in a set of images.

3. Large-scale object co-detection

Given a set of images, we aim to jointly detect object instances of multiple classes in all the images. To this end, we employ a two-stage strategy: We first generate a set of object hypotheses in each image, and then formulate co-detection as the problem of jointly labeling the hypotheses from all the images as either one of the target object classes or background. To address the labeling task, we build a fully-connected Conditional Random Field (CRF) that captures the spatial relationships of the objects within each image and the object similarity across all images. Our object similarity measure exploits high-dimensional features from which we learn a compact pairwise potential that allows efficient inference with a large number of images and of object hypotheses. We introduce our CRF model in the remainder of this section, and discuss our structural boosting approach to learning its pairwise potential in Section 4.

3.1. Object hypotheses generation

For each image I_m in a given image set $\mathcal{I}$, we generate a set of object hypotheses $\mathcal{X}^m$ based on a class-independent objectness measure [6, 1]. Specifically, we adopt the Selective Search method [26] to extract N_m object hypotheses in I_m, represented by a set of bounding boxes, with a high-recall rate. We then apply a fine-tuned RCNN model [9] to prune down the number of hypotheses based on the SVM score. Finally, following [9], we train a regression model to adjust the position of each remaining bounding box. We denote by $\mathcal{X} = \cup_m \mathcal{X}^m = \{\mathbf{X}_1, \cdots, \mathbf{X}_N\}$ the set of all hypotheses, where $\mathbf{X}_i$ represents the bounding box parameters of the i-th object hypothesis, and $N = \sum_m N_m$. We then extract an object feature descriptor $\mathbf{f}_i \in \mathbb{R}^K$ for each bounding box. In particular, we make use of the $f7$-layer features of the fine-tuned RCNN mentioned above.

3.2. CRF for multiclass object co-detection

Given the set of object hypotheses $\mathcal{X}$, our goal is to classify each object candidate into either one of the foreground object classes $\mathcal{C}$ or background $\emptyset$. We introduce a label variable $y_i \in \{\mathcal{C} \cup \{\emptyset\}\}$ for each object candidate $\mathbf{X}_i$. Our method jointly predicts the labels of all the object candidates, denoted by $\mathcal{Y} = \{y_1, \cdots, y_N\}$, to exploit the dependencies among them. To this end, we build a fully-connected Conditional Random Field (CRF) on the object label variables $\mathcal{Y}$. Each node in the CRF corresponds to the label of one object candidate, and any two candidates are connected by an edge.

Formally, we define a joint distribution over the label variables $\mathcal{Y}$ given the observed candidates $\mathcal{X}$ as $P(\mathcal{Y}|\mathcal{X}) = \frac{1}{Z(\mathcal{X})} \exp\left(-E(\mathcal{Y}, \mathcal{X})\right)$, with $Z(\cdot)$ the partition function. The corresponding energy function $E(\cdot)$ is defined as

$$E(\mathcal{Y}, \mathcal{X}) = \sum_{i=1}^{N} \phi(y_i|\mathbf{X}_i) + \sum_{i=1}^{N} \sum_{j>i} \psi(y_i, y_j|\mathbf{X}_i, \mathbf{X}_j), \quad (1)$$

where ϕ and ψ are the unary and pairwise potential functions, respectively. We describe each potential function below.

3.3. Unary potential

The unary potentials $\phi(y_i = c|\mathbf{X}_i)$ encodes the cost of assigning the candidate $\mathbf{X}_i$ to the object class c. To this end, we train a CNN model on normalized bounding boxes with $|\mathcal{C}|+1$ categories and take the output of the $f7$-layer as feature vectors for all the bounding boxes, and then train a set of linear SVM classifiers for $|\mathcal{C}|+1$ classes. The classifier for the background class is trained in an inverted fashion by performing hard-negative mining. In other words, this classifier treats all the ground-truth bounding boxes from the first $|\mathcal{C}|$ classes as positive examples, and we make use of the negative of its output score as a score for the background class. Viewing the scores of all classifiers as log probabilities, we then define our unary term as

$$\phi_u(y_i|\mathbf{X}_i) = -\sum_{c=1}^{|\mathcal{C}|+1} (s_{ic}) \mathbf{1}_{(y_i=c)} \quad (2)$$

where s_{ic} represents the score of class c for each sample bounding-box i.

3.4. Pairwise potential

The pairwise potential $\psi(\cdot)$ captures the relationship between pairs of object hypotheses, and measures the cost of the different possible label assignments for each pair. We incorporate two types of relationship in the pairwise term: (i) the spatial, or geometric, relationships between different object classes within each image; (ii) the similarity between any two object candidates in the image set. We denote these two pairwise potentials as $\psi_g(\cdot)$ and $\psi_s(\cdot)$, respectively.

For ψ_g, we follow the definition of spatial relations of [5], which groups the relative locations of two object hypotheses into D canonical relations. The pairwise potential is defined as

$$\psi_g(y_i, y_j|\mathbf{X}_i, \mathbf{X}_j) = \mathbf{w}_{y_i,y_j}^T \mathbf{d}(\mathbf{X}_i, \mathbf{X}_j), \quad (3)$$

where $\mathbf{w}_{y_i,y_j}$ are weights for all possible (i.e., D) geometric configurations of labels y_i and y_j. The binary vector $\mathbf{d}(\mathbf{X}_i, \mathbf{X}_j)$ encodes the geometric relation of $\mathbf{X}_i$ and $\mathbf{X}_j$ (i.e., 1 for the correct relation, 0 for the other ones). Note that $\mathbf{d} = \mathbf{0}$ when $\mathbf{X}_i$ and $\mathbf{X}_j$ are from different images.

The pairwise potential ψ_s is a data-dependent Potts model, encouraging similar hypotheses to share the same object label. We restrict the data-dependent term to take the form of a weighted mixture of Gaussian kernels defined on image features. This can be written as

$$\psi_s(y_i, y_j|\mathbf{X}_i, \mathbf{X}_j) = \textstyle\sum_{t=1}^{T} w_t K_t(\mathbf{f}_i, \mathbf{f}_j; \sigma_t)\mu(y_i, y_j), \quad (4)$$

where each K_t is a Gaussian kernel with variance σ_t^2 defined on the object features $\mathbf{f}_i$ and $\mathbf{f}_j$, and μ is a label compatibility function, which we define as a Potts model, *i.e.*, $\mu_s(y_i, y_j) = \mathbf{1}_{y_i \neq y_j}$.[1]

In essence, the mixture of Gaussian kernels encodes the appearance similarity between two object candidates. Such a representation was employed by [16] to design an efficient inference algorithm for semantic labeling in a fully-connected CRF. The computational efficiency of this algorithm, however, critically depends on the dimensionality of the object features $\mathbf{f}_i, \mathbf{f}_j$. In this work, we seek to exploit the powerful high-dimensional features extracted by CNNs. Directly employing these features in each kernel would make inference impractically slow. To overcome this issue, we propose to make use of one-dimensional Gaussian kernels of the form

$$K_t(\mathbf{f}_i, \mathbf{f}_j, \sigma_t, \kappa_t) = \exp\left(-\frac{(\mathbf{f}_i^{\kappa_t} - \mathbf{f}_j^{\kappa_t})^2}{\sigma_t^2}\right) \quad (5)$$

where $\mathbf{f}_i^{\kappa_t}$ extracts the κ_t-th dimension of the feature vector. While each kernel can now be efficiently evaluated, with the kind of CNN features that we would like to utilize, considering all feature dimensions would yield several thousands of kernels. This would therefore still make inference intractable. We address this issue in Section 4, where we introduce an approach to learning the kernels that are important for our task.

3.5. Efficient inference for co-detection

Efficient inference in our fully-connected CRF remains challenging since each image contains a large number of object candidates and the spatial pairwise potential ψ_g does not have a form that allows us to use the efficient algorithm of [16]. We therefore design a two-step cascaded inference procedure to obtain the marginal posterior probabilities of each object candidate.

We first start with a model that does not contain the potential ψ_s defined on the dense connections. The resulting model thus decomposes into M individual CRFs, one for each image. We make use of the max-margin procedure of [5] to learn the parameters of this model (i.e., a single set of parameters that will be used for all M images). At inference, we apply the greedy forward search algorithm of [5] and compute the marginals of each object node, which we denote by $\phi_g(y_i|\mathbf{X}_i)$.

We then integrate the marginals from the first step with our fully-connected pairwise potential ψ_s, which yields a CRF with energy function

$$\tilde{E}(\mathcal{Y}, \mathcal{X}) = \sum_{i=1}^{N} \phi_g(y_i|\mathbf{X}_i) + \sum_{i=1}^{N}\sum_{j>i} \psi_s(y_i, y_j|\mathbf{X}_i, \mathbf{X}_j). \quad (6)$$

The marginals of each object candidate can then be obtained efficiently using the approximate mean-field inference method of [16], which relies on fast Gaussian filtering. As mentioned above, however, inference is efficient only as long as the number of Gaussian kernels remains relatively small. This problem will be addressed in the next section.

4. Kernel learning for fully-connected CRFs

In this section, we introduce an approach to learning the kernels that define the pairwise potential of our fully-connected CRF. We also learn a transformation of unary scores because unary scores are not scaled across different classes. As briefly discussed in Section 3, the efficiency of our inference strategy depends on the compactness of our pairwise potential ψ_s, or, more precisely, since we employ one-dimensional Gaussian kernels, on the number of kernel functions in ψ_s. Unfortunately, the feature descriptors that have proven the most discriminative in practice, such as CNN features, are typically very high dimensional. Employing one kernel per feature would then not be practical. Here, we propose to overcome this issue by learning the kernels that are relevant to our goals. To this end, we introduce a structural boosting approach that allows us to select a small subset of distinctive object features and learn their corresponding kernel functions. In the remainder of this section, we first discuss our structural boosting framework and then describe the two different loss functions to learn our pairwise potentials.

4.1. Structural boosting for fully-connected CRFs

We now present our structural boosting framework. This framework follows the general functional gradient descent approach introduced in [19]. In this general context, however, we design two novel algorithms specialized to the problem of kernel learning in fully-connected CRFs, with emphasis on the task of multi-class object co-detection.

Let $(\hat{\mathcal{X}}, \hat{\mathcal{Y}})$ be a training set containing pairs of object bounding boxes $\mathbf{X}_i$ with corresponding ground-truth label $\hat{y}_i$. Furthermore, let $R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}]$ denote an empirical loss function defined with respect to the pairwise potential ψ_s

[1] General compatibility functions can also be used in our model.

and the training examples. Our approach exploits the fact that, as shown in Eq. 4, our pairwise potential ψ_s has an additive form, i.e., $\psi_s(\cdot) \sim \sum_t w_t h_t(\cdot)$. Each $h_t \in \mathcal{H}$ can thus be thought of as a weak learner belonging to a family $\mathcal{H}$. In our case, $\mathcal{H}$ is defined as a family of pairwise potential functions indexed by (σ, κ), such that each weak learner has the form

$$h(\cdot; \sigma, \kappa) = \sum_{i=1}^{N} \sum_{j>i} K(\mathbf{f}_i, \mathbf{f}_j, \sigma, \kappa) \mu(y_i, y_j), \qquad (7)$$

where $K(\cdot)$ is a one dimensional kernel as shown in Eq. 5.

Our structural boosting algorithm then works as follows. At each step t, we first compute the negative functional gradient $-\nabla_f R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}]$ with respect to ψ_s and evaluate it at the previous pairwise potential estimate ψ_s^{t-1}, and find its maximum projection onto the weak learner space $\mathcal{H}$. This can be expressed as computing

$$h_t^* = \underset{h \in \mathcal{H}}{\operatorname{argmax}} \langle -\nabla_f R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}]\Big|_{\psi_s^{t-1}}, h \rangle. \qquad (8)$$

Given the best weak learner h_t^*, we then use line-search to determine the corresponding weight as

$$w_t^* = \underset{w_t \in \mathbb{R}}{\operatorname{argmin}} R[\psi_s^{t-1} + w_t h_t^*|\hat{\mathcal{X}}, \hat{\mathcal{Y}}]. \qquad (9)$$

To maximize the projection of the functional gradient in Eq. 8, we discretize the parameter space of Gaussian kernel widths and enumerate all combinations of discrete width and feature dimension in $\mathbf{f}$. The resulting optimal kernel at step t is denoted by $K_t(.,.,\sigma_t,\kappa_t)$. Due to the approximate nature of our inference procedure, the functional gradient is also approximate for the boosting process. We stop the structural boosting process when the change in accuracy on a validation set is less than a certain threshold between the current boosting iteration and previous boosting iteration.

Below, we introduce two special cases of the general algorithm described above: One based on a max-margin learning approach, and one based on a direct loss minimization strategy. In each case, we first discuss the corresponding empirical loss functions R. We will derive the functional gradient projection for max-margin learning, propose a numerical approximation for the direct loss minimization. The steps of our approach are outlined in Algorithm 1.

The time complexity of the learning algorithm is $O(KSNT^2)$, where N is the total number of candidate boxes in the pool, and T, K, S are defined in Algorithm 1. The space complexity is $O(NT)$. At test time, the time complexity and the space complexity are $O(NT)$.

4.2. Max-margin structural boosting

Let us first consider the case of max-margin learning. In this scenario, the empirical loss R can be expressed as

$$R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}] = \max_{\mathcal{Y}} \left(-\tilde{E}(\hat{\mathcal{X}}, \mathcal{Y}) + \mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}})\right) + \tilde{E}(\hat{\mathcal{X}}, \hat{\mathcal{Y}}), \quad (10)$$

where $\tilde{E}(\hat{\mathcal{X}}, \mathcal{Y})$ is the energy function defined in Eq. 6, and $\mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}})$ is the Hamming loss between a label configuration $\mathcal{Y}$ and the ground-truth $\hat{\mathcal{Y}}$, i.e., $\mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}}) = \sum_{i=1}^{N} \mathbf{1}_{(\hat{y}_i \neq y_i)}$.

The negative functional (sub-)gradient of this loss function with respect to ψ_s can be written as

$$\begin{aligned} -\nabla_f R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}] &= \nabla_f E(\hat{\mathcal{X}}, \mathcal{Y}^*) - \nabla_f E(\hat{\mathcal{X}}, \hat{\mathcal{Y}}) \quad (11) \\ &= \delta_{\mathcal{Y}=\mathcal{Y}^*} - \delta_{\mathcal{Y}=\hat{\mathcal{Y}}} \end{aligned}$$

where δ is the Kronecker delta function, and $\mathcal{Y}^*$ is the label configuration that determines the value of the first term in Eq. 10. In other words,

$$\mathcal{Y}^* = \underset{\mathcal{Y}}{\operatorname{argmax}} \{-E(\hat{\mathcal{X}}, \mathcal{Y}) + \mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}})\}, \qquad (12)$$

and can thus be estimated approximately using loss augmented inference in our CRF. We use the same mean field algorithm with loss augmented unary terms to find the marginal posterior probability and labels $\mathcal{Y}^*$.

The optimal weak learner can then be computed by maximizing the projection of the functional gradient as in Eq. 8, which can be written as

$$\begin{aligned} h_t^* &= \underset{h \in \mathcal{H}}{\operatorname{argmax}} \langle -\nabla_f R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}], h \rangle \quad (13) \\ &= \underset{\sigma,\kappa}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_{j>i} K(\mathbf{f}_i, \mathbf{f}_j, \sigma, \kappa) \left(\mu(y_i^*, y_j^*) - \mu(\hat{y}_i, \hat{y}_j)\right). \end{aligned}$$

In each boosting step, we compute the projection using the same fast Gaussian convolution as for inference, and enumerate all the (σ, κ) pairs.

4.3. Direct-loss structural boosting

Typically, detection algorithms are evaluated using the mean average-precision error, and thus do not only care about the predicted label for each object, but rather about some notion of score obtained for each class. Therefore, the max-margin framework described above might not be optimizing the best loss. To overcome this issue, we introduce a direct-loss minimization approach.

More precisely, instead of predicting a unique class label for each object hypothesis, we aim to estimate the overlap between the proposed object bounding box and the ground-truth annotations of each class. In the following, we refer to this overlap as *the target overlap distribution*. Here, we propose to directly minimize the KL divergence between the target overlap distribution and the marginal probability from our fully-connected CRF. Let us denote the target overlap distribution of object hypothesis $\mathbf{X}_i$ as $\mathbf{p}_i$ and the corresponding marginal output of the CRF as $\mathbf{q}_i$. We then define the empirical loss R as

$$R[\psi_s|\hat{\mathcal{X}}, \hat{\mathcal{Y}}] = \sum_{i=1}^{N} \sum_{k=1}^{|\mathcal{C}|+1} \mathbf{p}_i(k) \log\left(\mathbf{p}_i(k)/\mathbf{q}_i(k)\right) \quad (14)$$

Algorithm 1 Structural Kernel Learning Algorithm

Initialization:

Training data: $\mathbf{D} = (\mathcal{X}, \hat{\mathcal{Y}})$
Unary potentials: $\phi_{\mathbf{g}}$
CNN FC7 feature dimension: $\mathbf{K}$
No. of Iterations: $\mathbf{T}$
Loss Function: $\mathcal{L}$
$\mathbf{f} = \sigma = \mathbf{w} = \emptyset$

Iteration:

for t = 1 ... $\mathbf{T}$ **do**
  Build Dense CRF Graph using Unary $\phi_{\mathbf{g}}$ and $(t-1)$ Structural Gaussian kernels $\mathbf{f}$
  for k = 1 ... $\mathbf{K}$ **do**
    Select a range of $\mathbf{S}$ σ values
    for s = 1 ... $\mathbf{S}$ **do**
      Generate weak learner h_t using feature index k and variance $\sigma^{\mathbf{s}}$
      Efficient Mean-field inference
      Compute Loss using Eq. 10 or Eq. 14
    end for
  end for
  Select best structural kernel h_t^* using variance σ_t^* and feature index f_t^*
  $\sigma = [\sigma \bigcup \sigma_t^*]$
  $\mathbf{f} = [\mathbf{f} \bigcup f_t^*]$
  Line Search for kernel weight $\mathbf{w}^*$, $\mathbf{w} = [\mathbf{w} \bigcup \mathbf{w}^*]$
  $\psi_{\mathbf{s}} \leftarrow [\psi_{\mathbf{s}} \bigcup h_t]$
end for

Output:

Learned boosted structural kernel features: $\mathbf{f}$
Kernel widths: σ
Kernel weights: $\mathbf{w}$

where $\mathbf{q}_i(k)$ is a function of ψ_s, which is recursively defined by the mean field update equation

$$\mathbf{q}_i(k) \propto \exp\left(-\phi_g(k|\mathbf{X}_i) - \sum_{j \neq i} \sum_{y_j} \psi_s(k, y_j|\mathbf{X}_i, \mathbf{X}_j)\mathbf{q}_j(y_j)\right). \quad (15)$$

Computing the functional gradient and its maximum projection is expensive due to the recursion in Eq. 15. We therefore follow a finite difference approach to approximately estimate the functional gradient. At each step t, this translates to searching for the weak learner that maximizes the decrease in the empirical loss according to the finite difference gradient, which can be expressed as

$$h_t^* = \underset{h \in \mathcal{H}}{\operatorname{argmax}}\, R[\psi_s^{t-1}|\hat{\mathcal{X}}, \hat{\mathcal{Y}}] - R[\psi_s^{t-1} + \epsilon h|\hat{\mathcal{X}}, \hat{\mathcal{Y}}], \quad (16)$$

where ϵ is a small constant value. As in the max-margin case, we enumerate all possible h. Since inference can be performed efficiently, this search remains tractable.

5. Experiments

We now demonstrate the effectiveness of our method on large scale multiclass object co-detection. To this end, we evaluate our approach on two challenging large datasets with multiple object classes, i.e., PASCAL VOC 2007 and 2012, and compare our results with those of the state-of-the-art methods.

5.1. Datasets and setup

The PASCAL VOC dataset [7] contains 20 object classes. We use the standard trainval/test partitions.

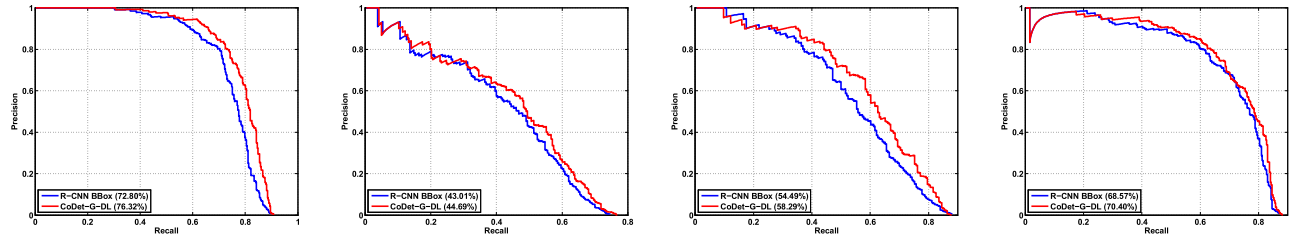
Our training procedure consists of four stages: a) fine-tuning the parameters of a deep network, b) training class-specific SVM classifiers [9], c) learning the per-image layout CRF [5], and d) learning the similarity kernels via structural boosting in our fully-connected CRF. We split the trainval images into a training and a validation set. The training set consists of randomly selected 75% of the trainval images, and the remaining 25% of the trainval images are kept as validation set. We use the training set to fine-tune a pre-trained deep network (AlexNet [18]) and train the 21 class-specific SVMs as in [9]. We learn the layout CRF and the kernels in our fully-connected CRF based on the validation set. The unary and feature computation jointly take 9 sec per image on average and the learning algorithm takes approximately 72 hrs to train the full model.

At test time, we perform inference in our fully-connected CRF, which not only computes the marginal posterior probability but also generate the maximum-a-posteriori labeling for each object bounding box hypothesis. To evaluate per-class performance, we divide our bounding-box pool into different categories according to the MAP labeling and use the marginal probability as the detection scores to generate the precision-recall curves. The PASCAL VOC 2007 and 2012 datasets comprise 4952 and 10991 test images, respectively. In both cases, we jointly perform detection in all the test images via inference in our fully-connected CRF. Performing inference in our fully-connected CRF containing a total of 995937 nodes (VOC 2007) took 9.8052sec.

We compare our approach (CoDeT-G-LA: kernel selection using max-margin learning, and CoDeT-G-DL: kernel selection using direct loss minimization) with the state-of-the-art context-free detector RCNN [9] (R-CNN and R-CNN-BB), the layout CRF [5] with RCNN as unary term (CoDeT-G) and the state-of-the-art single class object co-detection method [13]. For this last baseline, and to have a fair comparison, we used CNN features to generate the unary scores for each individual object detector. However, the approach of [13] to learning object similarity does not scale to CNN features and to large training sets. Therefore, following [13], we employed color and local binary pattern to define the pairwise potentials, and subsampled the training set to learn the similarity. The weight of the pairwise

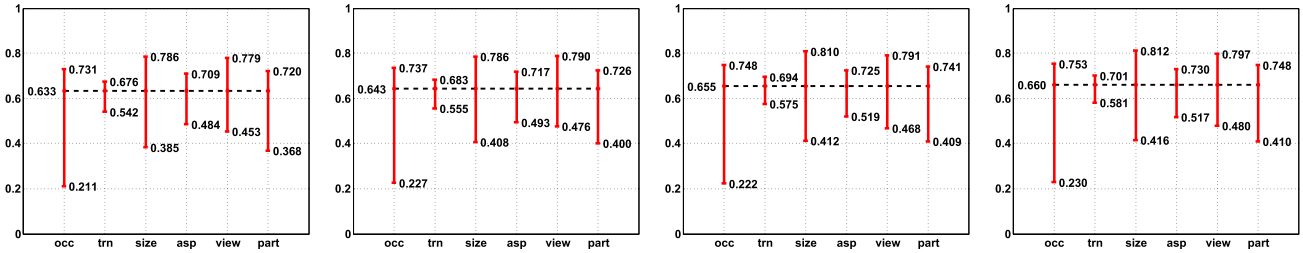
VOC 2007 (test)	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
MLRR [12]	34.1	53.0	12.4	18.9	31.2	43.2	52.7	21.6	22.8	25.0	32.2	10.6	51.7	41.0	38.6	19.2	27.3	32.5	41.3	41.9	32.5
MCOL [5]	28.8	56.2	3.2	14.2	29.4	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	15.2	16.1	20.1	34.2	35.4	27.1
R-CNN (AlexNet) [9]	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 BB (AlexNet) [9]	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
Co-detection (AlexNet) [13]	65.4	71.6	54.2	41.9	34.9	66.7	74.4	67.5	35.1	64.1	53.9	61.4	69.3	68.7	59.6	33.2	62.5	50.8	61.0	63.0	58.0
(Ours) CoDet-G (AlexNet)	70.1	72.4	58.8	43.5	38.7	67.2	74.8	**68.3**	34.1	65.4	56.8	63.4	70.3	68.9	58.9	33.2	63.9	**51.2**	**64.2**	**65.2**	59.5
(Ours) Codet-G-LA (AlexNet)	70.2	74.8	60.7	43.6	45.0	67.0	74.9	64.1	34.2	65.8	**58.7**	62.1	72.0	70.4	60.1	35.4	**65.5**	47.5	61.1	64.0	59.9
(Ours) CoDet-G-DL (AlexNet)	**70.6**	**76.3**	**61.7**	**44.7**	**45.6**	**67.6**	**75.0**	64.9	**34.6**	**66.3**	58.3	**63.7**	**72.3**	**70.4**	**60.4**	**35.6**	65.4	48.6	63.8	64.4	**60.5**
Fast R-CNN (VGG16) [10]	74.5	**78.3**	69.2	53.2	36.6	77.3	78.2	82.0	**40.7**	72.7	**67.9**	79.6	79.2	73.0	69.0	30.1	**65.4**	**70.2**	**75.8**	65.8	66.9
(Ours) CoDet-G-DL (VGG16)	**75.8**	78.1	**69.3**	**53.8**	**36.9**	**77.5**	**79.0**	**82.5**	40.1	**73.5**	67.7	**81.4**	**82.2**	**75.4**	**70.0**	**33.4**	65.4	70.0	74.3	**67.2**	**67.7**

Table 1. Detection average precision(%) on the PASCAL VOC 2007 test set. Rows 1-2 shows the co-detection baselines. Rows 3-4 provide the baseline state-of-the-art results for detection. R-CNN (without bounding-box regression); R-CNN BB (with bounding-box regression). Row 5 provides the co-detection results with deep network unary potentials. Rows 6-8 show co-detection performance. CoDet-G (R-CNN-BB with geometric context model learning); CoDet-G-LA (Kernel selection using max-margin learning); CoDet-G-DL (Kernel selection using direct loss minimization). Rows 9-10 show the results of the Fast-RCNN baseline and of our method using VGG16.



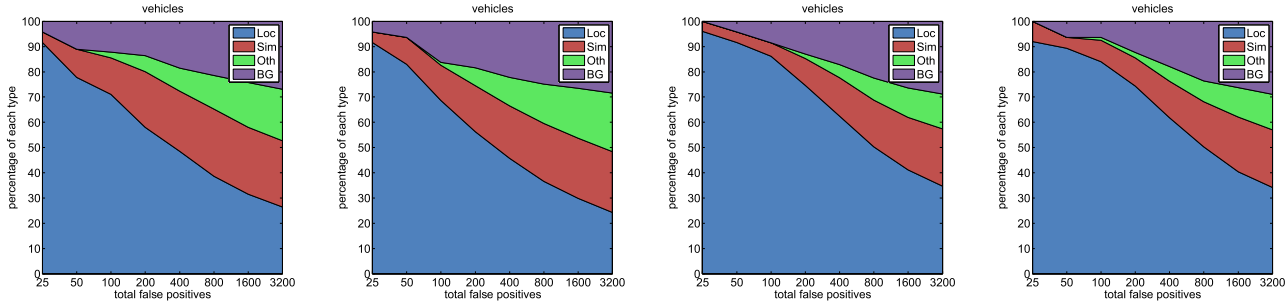
(a) Bike (b) Boat (c) Dining Table (d) Motorbike

Figure 2. Precision/Recall curves performance comparison on four representative categories using the VOC 2007 dataset.



(a) R-CNN FT FC7 BB (b) CoDeT-G (c) CoDeT-G-LA (d) CoDeT-G-DL

Figure 3. Sensitivity and Impact Analysis: Overall detailed performance comparison using different metrics (i.e., occlusion, truncated, size, aspect ratio, view point and part visibility). The black dashed line indicates the overall average normalized precision $\mathrm{AP_N}$.



(a) R-CNN FT FC7 BB (b) CoDeT-G (c) CoDeT-G-LA (d) CoDeT-G-DL

Figure 4. False positive analysis using all the vehicles category.

potential was learned using the validation set.

5.2. Results on VOC 2007

We report our results on the VOC 2007 test set using two different evaluation protocols: the standard per-class metric and the multiclass metric of [5].

Per-class scores We follow the VOC performance evaluation protocol and report the Average Precision (AP) and mean of AP (mAP) for the 20 object classes in Table 1. We can see that CoDet-G outperforms the R-CNN-BB results on 18 out of 20 classes, which suggests that learning a geometric model to incorporate inter-class objects configura-

VOC 2012 (test)	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN [9]	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 BB [9]	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
CoDet-G	**72.5**	64.2	51.4	**34.8**	32.6	62.5	60.3	**71.2**	27.8	52.4	**42.4**	**69.7**	60.5	67.7	58.6	30.0	57.7	40.0	**61.0**	55.6	53.6
CoDet-G-DL	69.8	**67.5**	**52.3**	34.4	**35.7**	**63.0**	**63.4**	68.3	**28.3**	**54.0**	41.1	67.7	**65.2**	**70.7**	**60.6**	**32.5**	**60.0**	**41.1**	54.7	**57.9**	**54.4**

Table 3. Detection average precision(%) on the PASCAL VOC 2012 test set. Rows 1-2 provide the baseline state-of-the-art results. R-CNN (without bounding-box regression); R-CNN BB (with bounding-box regression). Row 3-4 show our co-detection performance. CoDet-G (R-CNN-BB with geometric context model learning); CoDet-G-DL (Kernel selection using direct loss minimization).

VOC 2007 (test)	overall AP	Max-A-Posteriori Acc.
R-CNN BB [9]	61.46	57.05
CoDet-G-LA	62.13	62.94
CoDet-G-DL	62.60	64.49

Table 2. Multi-class average precision(%) on the PASCAL VOC 2007 test set. CoDet-G-DL (Kernel selection using direct loss minimization). We constructed the baseline curve for R-CNN-BB (with bounding-box regression) by pooling the detections across all object classes and images when computing the PR curves. Our model clearly provides a noticeable boost in overall performance.

tion is vital to object detection when we have multiple objects appearing in the same image. By contrast, we observed that the results of the co-detection method of [13] yields virtually no improvement over the R-CNN-BB results. We conjecture that this is due to the limited scalability of this method, which forced us to subsample the training data when learning the object similarity and to employ less informative, but lower-dimensional, features. Our CoDet-G-LA algorithm yields a further small improvement over CoDet-G on 14 classes, thus outperforming R-CNN-BB by 1.4%. Our CoDet-G-DL algorithm yields an improvement of 2% over the R-CNN BB baseline results. Recently the Feature Edit method [22] outperformed the R-CNN BB baseline by a margin of 1.6%. We also evaluated our method using the VGG16 [24] network. We obtained an improvement of 0.8% over the Fast-RCNN VGG16 baseline [10]. In other words, our CoDet-G-DL algorithm achieves state-of-the-art results on VOC 2007 (based on the same kind of deep network structure).

We also computed Precision-Recall (PR) curves for comparison with the R-CNN-BB baseline. Fig. 2 shows the precision-recall curves for bicycle, boat, dining table and motorbike. These curves clearly indicate that our method improves over R-CNN-BB in the high-recall low-precision region. Following [14], we provide a detailed comparison of different performance criteria in Fig. 3. Our method outperforms the R-CNN-BB baseline by 2.7% in average normalized precision $\mathrm{AP_N}$. Finally, in Fig. 4, we also provide an analysis of the false-positives using the vehicles category group. False positives with confusion across similar object categories and different object categories are significantly reduced by our approach. This shows the strength of our multiclass object co-detection approach, which benefits from intra-class and inter-class similarity to help reduce the false positives.

Multi-class scores Multi-class object detection performance is difficult to measure using per-class AP. Since our main goal is multi-class object co-detection, we also report the results according to the multiclass detection metric of [5]. Following [5], we pool the detections across all the classes and all the images and generate a single precision-recall curve from which we compute the overall AP. We also compute the labeling accuracy based on the Maximum-A-Posteriori (MAP) estimation. The results are summarized in Table 2. We can see that our method achieves a large improvement over the baseline.

5.3. Results on VOC 2012

In Table 3, we report our results on the VOC 2012 test dataset using the standard per-class metric based on the summary statistics from the evaluation server. Precision-recall curves and multi-class performance metrics cannot be generated here, since we do not have access to the original labeling of the VOC 2012 test dataset. Our CoDeT-G-DL algorithm outperforms the R-CNN-BB baseline in 15 object categories, which, altogether, yields 1.1% overall gain in the mean average precision on VOC 2012 test dataset. This indicates that our approach achieves a consistent improvement over the baseline method. Note that, to obtain our results on VOC 2012, we simply re-used the kernels learned using the VOC 2007 trainval dataset.

6. Conclusion

In this work, we have introduced a novel large scale object co-detection method that simultaneously considers multiple object classes. Our approach based on a fully-connected CRF allows us to incorporate contextual information within and across images, as well as within and across the classes. Furthermore, our structural boosting strategy lets us benefit from rich, high-dimensional features to learn the object relationships within our CRF framework. Our experiments have demonstrated the benefits of our approach over the state-of-the-art methods on PASCAL VOC 2007 and 2012, where we obtained state-of-the-art detection accuracies. In the future, we intend to learn the geometric model and instance similarity jointly as well as to exploit the object segmentations within the bounding boxes to improve the accuracy of object co-detection.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[2] J. Baek, A. Adams, and J. Dolson. Lattice-based high-dimensional gaussian filtering and the permutohedral lattice. *JMIV*, 2013.

[3] S. Bao, Y. Xiang, and S. Savarese. Object co-detection. In *ECCV*, 2012.

[4] M. J. Choi, A. Torralba, and A. S. Willsky. A tree-based context model for object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):240–252, 2012.

[5] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[6] I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*. 2010.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *CVIU*, 114:712–722, 2010.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[10] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[11] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *JMLR*, 12, 2011.

[12] X. Guo, D. Liu, B. Jou, M. Zhu, A. Cai, and S.-F. Chang. Robust Object Co-detection. In *CVPR*, 2013.

[13] Z. Hayder, M. Salzmann, and X. He. Object co-detection via efficient inference in a fully-connected crf. In *Computer Vision–ECCV 2014*, pages 330–345. Springer, 2014.

[14] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *Computer Vision–ECCV 2012*, pages 340–353. Springer, 2012.

[15] D. Hoiem, A. a. Efros, and M. Hebert. Putting Objects in Perspective. *IJCV*, 80:3–15, 2008.

[16] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[17] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 513–521, 2013.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 975–982. IEEE, 2009.

[20] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. *arXiv preprint arXiv:1412.5661*, 2014.

[21] N. D. Ratliff, D. Silver, and J. A. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[22] Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. *arXiv preprint arXiv:1409.6911*, 2014.

[23] J. Shi, R. Liao, and J. Jia. CoDeL: An efficient human co-detection and labeling framework. In *ICCV*, 2013.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] 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.

[26] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013.

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

[28] V. Vineet, J. Warrell, and P. H. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. In *ECCV*. 2012.

[29] Y. Zhang and T. Chen. Efficient inference for fully-connected crfs with stationarity. In *CVPR*, 2012.

[30] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *arXiv:1502.04275*, 2015.