Efficient Scene Layout Aware Object Detection for Traffic Surveillance

Tao Wang
Minjiang University
taowang2600@gmail.com

Xuming He
ShanghaiTech University
xmhe.cs@gmail.com

Songzhi Su
Xiamen University
ssz@xmu.edu.cn

Yin Guan
NetDragon Inc.
niynaug@foxmail.com

Abstract

We present an efficient scene layout aware object detection method for traffic surveillance. Given an input image, our approach first estimates its scene layout by transferring object annotations in a large dataset to the target image based on nonparametric label transfer. The transferred annotations are then integrated with object hypotheses generated by the state-of-the-art object detectors. We propose an approximate nearest neighbor search scheme for efficient inference in the scene layout estimation. Experiments verified that this simple and efficient approach provides consistent performance improvements to the state-of-the-art object detection baselines on all object categories in the TSWC-2017 localization challenge.

1. Introduction

Consider the object detection problem as depicted in Figure 1. As humans, we are able to estimate the scene layout at the very first glance and then know where to look for a given object category. For instance, cars will mostly likely appear on paved areas and pedestrians are usually found on sidewalks. On the contrary, most object detection algorithms produce scores for densely sampled object locations and scales, or a few hundreds to thousands of “blobby” object proposals. While these approaches have merit in terms of straightforwardly building a strong model of object appearance, they usually lack an understanding of the scene layout and act quite differently from what a human would do for the same task.

In this paper, we seek to exploit the spatial context for efficient object detection in traffic surveillance images. A key feature of these data is that they exhibit strong regularities in terms of scene layout that are useful for localizing objects of interest. This general idea has long been proven effective in the computer vision community, with seminal works from Torralba et al. [43, 32, 45], and later Hoiem, Efros and Hebert [19], plus a few more [47, 35, 4] as prominent examples. More recently, the modeling of spatial context has been extended to 3D scenarios [40, 2, 41, 6, 26, 15, 29] as high quality co-registered depth and color images have become more easily accessible. Most existing approaches assume a parameterized model for the scene layout, such as the piecewise planar assumption [9], blocks world assumption [14], or the Manhattan world assumption [18, 24, 23, 5]. These priors are indeed necessary when annotated data is scarce and expensive to obtain. However, in this work we seek to explore scene layout estimation from an alternative perspective. Specifically, we are interested in improving object detection through a nonparametric implicit scene layout model that predicts potential object locations and scales, as shown in Figure 1. Our method crucially depends on the availability of large-scale databases that cover objects of different sizes and at various locations. In particular, surveillance images are well-suited for our approach because their scene layouts provide strong priors for localizing objects. More importantly, large-scale databases such as the MIO-TCD dataset [1] containing more than a hundred thousand images and millions of object instances are becoming publicly accessible. Datasets at this scale allow

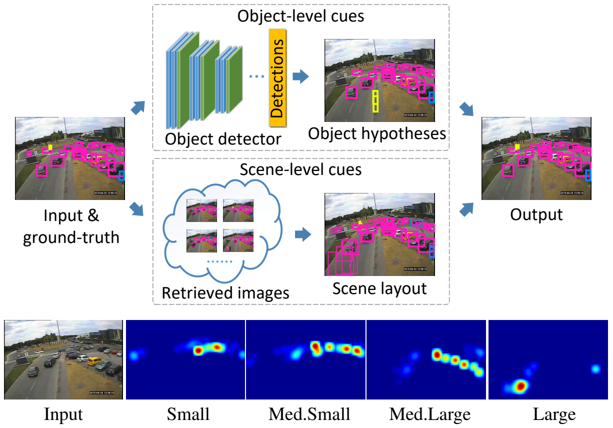

Figure 1. **Upper panel**: Illustration of our method. We incorporate scene-level cues for object detection by nonparametric label transfer. Color keys: *car* □, *pedestrain* □, *motorized vehicle* □. Note that the false alarms at the center bottom of the image are removed for the *pedestrian* category. In addition, two pedestrians at the distant roundabout are detected. **Lower panel**: Possible object locations for the *car* category. We show four different object scales here, from small (nearby) to large (farther away).

for high quality object proposals with a simple K-neareset neighbor search, as illustrated in Figure 3.

The benefit of adopting a nonparametric scene layout model is twofold. Firstly, since we retrieve object layout from the nearest neighbors these models can naturally handle diverse scene layouts, as shown in Figure 5. In addition, similar to other nonparametric knowledge-transfer methods (e.g., [27, 42]) ours is also simple and efficient. Our primary contribution includes a scene layout transfer method to model the spatial context for object detection, and an approximate nearest neighbor search scheme for efficient inference. The proposed method is backed by a consistent performance boost to all object categories in the TSWC-2017 [1] localization challenge, when paired with state-of-the-art object detection algorithms including Faster RCNN [37] and SSD [28]. Our best-performing model achieves a mean AP of 77.19% in the official challenge.

The rest of this paper is organized as follows. Section 2 briefly reviews the related literature on object detection, context modeling and nonparametric transfer. We then describe details of our method in Section 3. Afterwards, Section 4 discusses details of our experiments, followed by closing remarks in Section 5. Source codes of our method are available from `https://github.com/realwecan/traffic-context-detection`.

2. Related work

Object detection. Recent years witnessed a huge success of Convolutional Neural Network (CNN) based object detection algorithms over conventional methods based on handcrafted features and a shallow object grammar-based architecture such as the Deformable Parts Model (DPM) [10]. Some of the most prominent examples include sliding-window based OverFeat [38] and object proposal based R-CNN [13] and its faster variants [16, 12, 37]. These methods are directly inspired by the success of CNN for image classification. The latter, proposal-based methods seek to exploit the strong representation power of deep networks to classify and make refinements to a relatively small set (typically hundreds to a few thousands) of potential object regions. Another line of work attempts to make direct predictions using a deep network without the object proposal step. Examples include YOLO [36] and SSD [28] and we note that these methods are generally more efficient and is comparably better suited for real-time detection. In this work, we choose Faster RCNN [37] and SSD [28] as our baseline object detectors and explore how to improve their results via incorporating scene-level context cues.

Context modeling. Context-aware object detection has been well studied, and many context-aware object detection methods have been proposed (e.g., [43, 44, 47, 35, 19, 21, 4, 30, 34]). See [47] for a review and [7] for an empirical study of earlier work in the literature. More recently, Yang et al. [48] have shown that reasoning about a 2.1D layered object representation in a scene can positively impact object detection. Yao et al. [49] propose a holistic scene understanding model which jointly solves object detection, segmentation and scene classification. Mottaghi et al. [31] exploit both the local and global contexts by reasoning about the presence of contextual classes, and propose a context-aware improvement to the DPM. Zhu et al. [51] use CNNs to obtain contextual scores for object hypotheses, in addition to scores obtained with object appearance. Batzer et al. [3] propose a context-aware voting scheme for small and distant object detection. Other works have extended context modeling to 3D scenarios. For example, Bao, Sun and Savarse propose a parameterized 3D surface layout model and combine it with object detectors [2, 41]. Geiger, Wojek and Urtasun [11] propose a generative model for joint inference of scene topology, geometry and 3D object locations. Choi et al. [6] learn latent 3D geometric phrases to jointly solve object detection and scene layout estimation. Similarly, Lin et al. [26] use a CRF model to integrate various contextual relations for holistic scene understanding. Other later works include [15] and [29]. Our work differs from the methods above in the sense that we propose a nonparametric, knowledge-transfer based approach to model the spatial context for object detection, and exploit the regularities in terms of scene layouts in traffic surveillance images.

Nonparametric transfer. Recently, the emergence of large databases of images allows researchers to build nonparametric models for label prediction in various vision tasks. The basic idea is to explain an image by matching its parts to other images from a database. For example, Liu, Yuen and Torralba [27] address the semantic segmentation problem by first retrieving nearest neighbors of a query image with distance derived from global scene descriptors such as GIST [33] and the spatial pyramid intersection of HOG visual words [22]. This is followed by a coarse-to-fine SIFT flow algorithm to establish dense pairwise correspondences between the query scene and each of its nearest neighbors. Similarly, Tighe and Lazebnik propose SuperParsing [42] which performs label transfer at the superpixel level to avoid the expensive inference via SIFT flow. Similar ideas have been used in label propagation in videos [8] and glass object segmentation [46]. Unlike their methods, our goal is to transfer layout as a scene-specific context prior for object detection. Perhaps closest to our work is [50] which also proposes a nonparametric method for scene layout estimation. However, they use a column-based tiered model which is only applicable to a specific viewpoint, while our method has no such restriction and is able to deal with large viewpoint variations. Furthermore, we propose an approximate nearest neighbor search scheme and demonstrate that our method is able to efficiently transfer scene layouts in databases with more than a hundred thousand images.

3. Our approach

The proposed scene layout transfer method can be used in conjunction with any object detection algorithm that outputs bounding boxes. For an input image, scene layout transfer essentially produces a score for any given object hypothesis. The score is then combined with the output of an off-the-shelf object detector to obtain a final output.

More formally, suppose we have an image I and an object class of interest o. Let the object hypothesis be $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is the object pose space. To simplify the notation, we assume each hypothesis is $\mathbf{x} = (\mathbf{x}_c, a_s, a_r)$ where $\mathbf{x}_c = (a_x, a_y)$ is the image coordinate location of the object center, a_s a scale, and a_r an aspect ratio. Note that each $\mathbf{x}$ now implies a bounding box as well. Object detection algorithms define a scoring function $S_d(\mathbf{x}, o)$ for each valid object hypothesis $\mathbf{x}$ and a given object class o. For example, this score is implemented as a two-class softmax score for each object class in Faster RCNN, i.e., $S_d(\mathbf{x}, o) = p(\mathbf{x}, o|I)$.

We propose an additional scene layout score $S_l(\mathbf{x}, o)$ (in the logarithmic space) for any given object hypothesis $\mathbf{x}$ and class o. The final detection score is a weighted sum of the two scores:

$$S(\mathbf{x}, o) = S_d(\mathbf{x}, o) + \theta \log S_l(\mathbf{x}, o) \tag{1}$$

where θ is a hyperparameter for the relative importance between the two terms. The scene layout score $S_l(\mathbf{x}, o)$ is obtained in a nonparametric fashion, as detailed in the next section.

3.1. Scene layout transfer

Similar to other nonparametric label transfer approaches, the scene layout transfer score $S_l(\mathbf{x}, o)$ is obtained by investigating a local neighborhood $\mathcal{N}_I$ of the input image I defined on an appearance feature manifold. This neighborhood is also referred to as *the retrieval set* in the literature.

Concretely, let $I_j \in \mathcal{N}_I$ be a neighbor image of I, and $\mathbf{f}, \mathbf{f}_j$ be the image-level feature vectors of I and I_j that give rise to the neighborhood relations. Note that the retrieval set is typically an annotated database, and is *the training set* in our case. Therefore, each image I_j contains a number of ground-truth object hypotheses given an object class o. We denote these object hypotheses as $\mathbf{y} \in \mathcal{Y}_j$. Our scene layout transfer score $S_l(\mathbf{x}, o)$ is based on the retrieval set $\mathcal{N}_I$ and can be written as:

$$S_l(\mathbf{x}, o|\mathcal{N}_I) = \sum_{j \in \mathcal{N}_I} k^{(1)}(\mathbf{f}, \mathbf{f}_j) \sum_{\mathbf{y} \in \mathcal{Y}_j} k^{(2)}(\mathbf{x}, \mathbf{y}) \tag{2}$$

where $\mathbf{f}$ and $\mathbf{f}_j$ are 2048-D features extracted from the *pool5* layer of a ResNet-50 [17] network applied on images I and I_j respectively. In addition, $k^{(i)}(\cdot, \cdot), i \in \{1, 2\}$ are heat kernels of the following form:

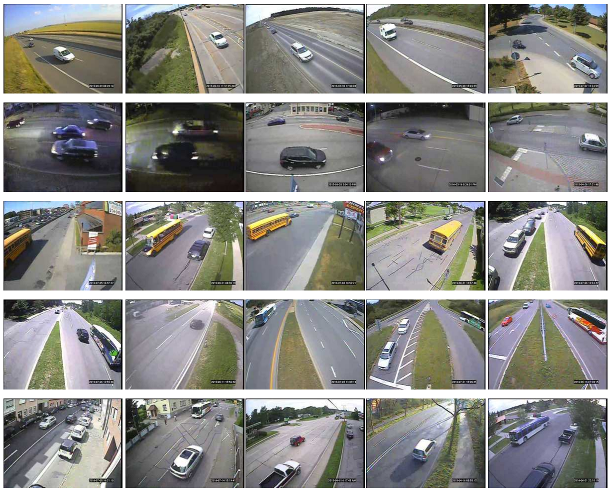

Figure 2. Example images in the neighborhood $\mathcal{N}_I$. The leftmost column shows the query image I. The four columns to the right show examples of neighbour images in $\mathcal{N}_I$ from different cameras with similar views.

$$k^{(i)}(\mathbf{z}_1, \mathbf{z}_2) = \exp\left(-\frac{d^{(i)}(\mathbf{z}_1, \mathbf{z}_2)}{\sigma_i^2}\right) \tag{3}$$

where $d^{(i)}(\cdot, \cdot)$ is a distance metric and σ_i is the kernel width. In this work, we choose the cosine distance between two feature vectors for $d^{(1)}(\cdot, \cdot)$ as it was found to outperform the Euclidean distance. We use the Jaccard index (i.e., the IoU overlap between two bounding boxes) for $d^{(2)}(\cdot, \cdot)$:

$$d^{(2)}(\mathbf{x}, \mathbf{y}) = \frac{area(\mathbf{x} \cap \mathbf{y})}{area(\mathbf{x} \cup \mathbf{y})} \tag{4}$$

Definition of the neighborhood. The most common definition of the neighborhood $\mathcal{N}_I$ of the image I consists of taking the K nearest neighbors (K-NN). In addition, ϵ-NN is another widely adopted neighborhood definition that considers all of the neighbors within $(1+\epsilon)$ times the minimum distance from the image I. Following [27] we adopt the $\langle K, \epsilon \rangle$-NN neighborhood as $\mathcal{N}_I$ for our input image I:

$$\begin{aligned} \mathcal{N}_I = \{I_j | d^{(1)}(\mathbf{f}, \mathbf{f}_j) \leq (1+\epsilon) d^{(1)}(\mathbf{f}, \mathbf{f}_1), \\ \mathbf{f}_1 = \operatorname{argmin} d^{(1)}(\mathbf{f}, \mathbf{f}_j), j = 1 \ldots K\} \end{aligned} \tag{5}$$

Note that as $\epsilon \to \infty$, $\langle K, \infty \rangle$-NN reduces to K-NN. Conversely, as $K \to \infty$, $\langle \infty, \epsilon \rangle$-NN reduces to ϵ-NN. See Figure 2 for example images in the neighborhood $\mathcal{N}_I$. Note that $\mathcal{N}_I$ contains images taken from different cameras with similar views, not merely different images taken from the same camera. In addition, Figure 3 presents examples of transferred annotations for varying values of K in a K-NN neighborhood. As illustrated, a small value for K gives good recall for objects in the first three images. However, a larger value for K is needed for the remaining examples. The neighborhood definition we adopt this work is flexible at handling feature manifolds with large density variations, which is also relevant to the discussion about an alternative design choice below.

An alternative design choice. In addition to the approach presented above, one easily perceived alternative to handle the image-level similarities is to use clustering methods such as K-means or affinity propagation to obtain scene layout paradigms. Intuitively, these methods would provide an interpretable scene layout representation in terms of clusters. However, in our initial experiments we found it difficult to find a succinct set of universally applicable parameters for these clustering methods due to the highly unstable intra-cluster variations. Our approach addresses this issue by eliminating the need to explicitly form scene layout clusters, and instead infer the scene layout from a $\langle K, \epsilon \rangle$-NN neighborhood. Through experiments, we verified that our design choice outperforms clustering-based methods and is able to reliably transfer scene layouts for object location and scale prediction, as illustrated in Figure 4.

3.2. Efficient approximate inference

One of the key design considerations of recent object detection algorithms is on their efficiency. In particular, state-of-the-art object detectors such as Faster RCNN, YOLO and SSD operate at the speed of tens to hundreds of frames per second. While the scene layout transfer method described in the previous section is efficient when the kernels $k^{(1)}(\cdot, \cdot)$ in Equation 2 are computed, the computation of the pairwise distances of a test image to *all* training images is non-trivial. We now show that with a simple approximate nearest neighbor search technique, the proposed method only brings in a small computation overhead.

More specifically, the training set of the TSWC-2017 localization challenge contains $110,000$ images, and a CPU-based multi-threaded mex implementation to compute the 2048-D pairwise feature distances between a test image and the entire training set takes more than 2.6 seconds on an i7-4790 system. Even with sophisticated GPU acceleration (15x to 30x according to [25] and [20]), the computation time is still comparable to that of a CNN-based object detector. More importantly, the computational cost roughly scales linearly with the size of the training set. To address this issue, we propose an approximate nearest neighbor search scheme for efficient test-time scene layout transfer. The basic idea is to "replace" the query image feature with its approximate nearest neighbor in the training features, so the pairwise distances can be precomputed as part of the training process.

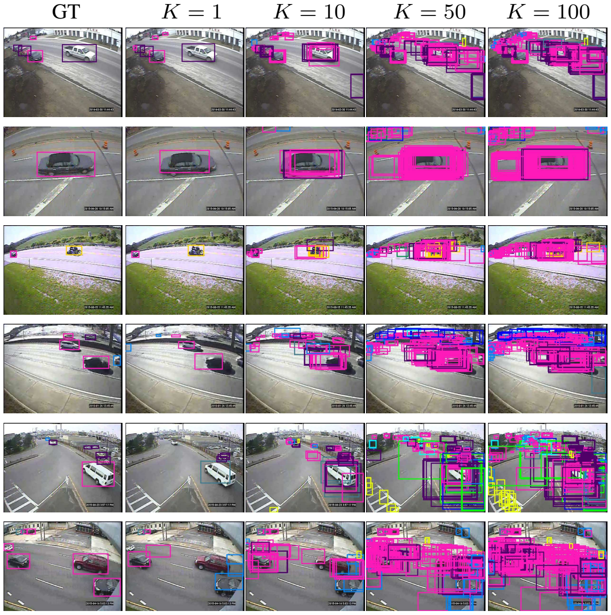

Figure 3. Examples of transferred bounding boxes for varying values of K in a K-NN neighborhood. Images are chosen from a held-out validation set. Ground-truth (**GT**) on the left. See Figure 5 for color keys of the bounding boxes.

Mathematically, let M be the number of images in our training set and D be the feature dimension for the image-level appearance features (i.e., $D = 2048$ for our ResNet *pool5* features). We can perform K-means clustering with N clusters for the training feature matrix $\mathbf{F}^{tr} \in \mathbb{R}^{D \times M}$, and denote $\mathbf{C} \in \mathbb{R}^{D \times N}$ as the cluster centers. Here we use bold uppercase letters to denote matrices and the corresponding lowercase letters to denote their column vectors. For example, $\mathbf{f}^{tr}_m, m = 1 \dots M$ are features for each training image and $\mathbf{c}_n, n = 1 \dots N$ are individual cluster centers. Additionally, let $\mathbf{F}^{tr,n} \in \mathbb{R}^{D \times M_n}$ and its columns $\mathbf{f}^{tr,n}_m, m = 1 \dots M_n$ denote features in the n-th cluster. At test time, we approximate $d^{(1)}(\mathbf{f}, \mathbf{f}_j)$ with $\tilde{d}^{(1)}(\mathbf{f}, \mathbf{f}_j)$ defined as follows:

$$\begin{aligned} &\tilde{d}^{(1)}(\mathbf{f}, \mathbf{f}_j) := d^{(1)}(\mathbf{f}^{tr,n}_m, \mathbf{f}_j), \\ &m = \operatorname{argmin} d^{(1)}(\mathbf{f}, \mathbf{f}^{tr,n}_m), m = 1 \dots M_n, \\ &n = \operatorname{argmin} d^{(1)}(\mathbf{f}, \mathbf{c}_n), n = 1 \dots N. \end{aligned} \tag{6}$$

Here we note that both features in the pairwise distance $d^{(1)}(\mathbf{f}^{tr,n}_m, \mathbf{f}_j)$ on the right hand side of Equation 6 belong to the training set and can be precomputed. Therefore, the computation reduces to working out the two $\operatorname{argmin}(\cdot)$ operators in Equation 6. In our experiments, we set $N = 200$ and M_n are typically in the hundreds (the mean of M_n is 495 and 98.3% of all clusters have 1000 members or less). We note, however, it makes sense to further set an upper

Algorithm 1: Efficient approximation of $d^{(1)}(\mathbf{f}, \mathbf{f}_j)$.

```
Initialization: Precompute pairwise distance on $\mathbf{F}^{tr}$
and perform K-means clustering to obtain $\mathbf{C}$.
Input: Features $\mathbf{f}, \mathbf{f}_j$; Number of clusters $N$, Number
    of nearest clusters to search $T$.
for $t = 1 : T$ do
    1. Find the $t$-th nearest cluster:
       $n_t \leftarrow \text{argmin}\, d^{(1)}(\mathbf{f}, \mathbf{c}_n)$,
           $n \in \{1 \ldots N\}$ if $t = 1$,
           $n \in \{1 \ldots N\} \backslash \{n_1 \ldots n_{t-1}\}$ otherwise;
    2. Find the nearest feature in this cluster:
       $m_t \leftarrow \text{argmin}\, d^{(1)}(\mathbf{f}, \mathbf{f}_m^{tr,n_t}), m = 1 \ldots M_{n_t}$;
end
Output: $\tilde{d}^{(1)}(\mathbf{f}, \mathbf{f}_j) \leftarrow \text{argmin}\, d^{(1)}(\mathbf{f}_{m_t}^{tr,n_t}, \mathbf{f}_j)$,
                $t = 1 \ldots T$.
```

bound to M_n so that the worst case time complexity can be guaranteed. In practice, we can additionally evaluate features in the T nearest clusters instead of only one, as this was found to narrow the performance gap between the approximate inference and the exact inference to a negligible level. See Section 4.2 for details. The inference procedure is summarized in Algorithm 1.

4. Experimental evaluation

In this section, we describe some details in relation to the TSWC-2017 localization challenge and our experiments. The TSWC-2017 introduces a new large-scale database of traffic surveillance images: the MIOvision Traffic Camera Dataset (MIO-TCD). The images in the localization challenge is partitioned into a training set with $110,000$ images and a test set of $27,743$ images. All our quantitative results reported in Section 4.2 are obtained on the test set by uploading our algorithm outputs to the challenge website. In addition, we put aside $11,000$ images from the training set and use them as a held-out validation set.

The two baseline object detection algorithms we trained include Faster RCNN and SSD. We use the stock training settings and parameters shipped with their respective source codes without any changes. We choose the alternating optimization variant of Faster RCNN and SSD-512 in our experiments. For our efficient approximate inference, we empirically choose $N = 200$ for the number of clusters and $T = 3$ for the number of nearest clusters to search, and note that the results are not sensitive to these specific values.

Some of the model parameters are learned with grid search on a held-out validation set. This includes the weight θ for the scene layout term in Equation 1, K and ϵ in the $\langle K, \epsilon \rangle$-NN neighborhood, and the kernel widths σ_1 and σ_2 in Equation 3.

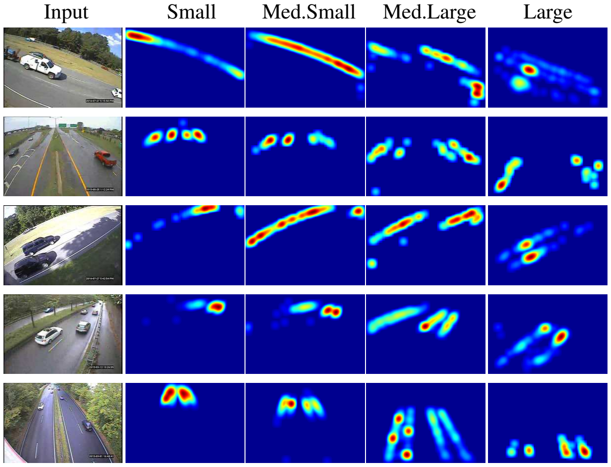

Figure 4. Possible object locations for the *car* category inferred from the transferred scene layouts. Input images are shown in the leftmost column, with possible locations for small (farthest), medium-small (far), medium-large (close), and large (closest) objects shown in the four columns to the right.

We report the results of three variants of our method. The first is *Context (No Detector)* which is obtained by switching off the object detector term $S_d(\mathbf{x}, o)$ in Equation 1. The second and the third are termed *Faster RCNN+Context* and *SSD+Context*, which are obtained by adding the scene layout transfer score $S_l(\mathbf{x}, o)$ to the Faster RCNN and SSD baselines respectively.

4.1. Qualitative studies

In order to closely examine the scene layouts we obtained from data, Figure 4 shows examples of possible object locations and scales inferred from the scene layouts being transferred. From these examples we can clearly see potential locations for the smaller and distant objects, as well as for the larger and closer ones.

In addition, Figure 5 shows a side-by-side comparison of detection results obtained with SSD and with *SSD+Context* on the held-out validation set. To allow for an easier comparison, for each class in every image we only show N_g top-scoring detections where N_g is the number of ground-truth objects for that class. In general, our method outperforms the baseline by making the following types of improvements: (1) removal of out-of-context false alarms; (2) removal of multiple detections for a same category at a similar location, but some with incorrect scales; (3) better detection of missed distant objects; (4) better handling of extreme viewpoint variations for difficult objects.

4.2. Quantitative results

We summarize the results that we obtain in the TSWC-2017 localization challenge in Table 1. The three baseline methods are YOLO (Version 1) [36], Faster RCNN [37] and

Object Categories	a.truck	bicyle	bus	car	motorcycle	m.vehicle	n.m.vehicle	pedestrian	p.truck	s.u.truck	workvan	mean AP
Baseline Approaches												
YOLO v1 [36]	82.72	70.02	91.56	77.16	71.43	44.41	20.68	18.08	85.59	58.30	69.26	62.65
Faster RCNN [37]	80.70	70.63	93.45	79.85	74.58	46.48	21.22	19.49	86.71	53.29	67.40	63.07
SSD [28]	91.28	77.36	96.56	93.59	79.53	55.39	56.60	41.58	92.66	72.74	79.40	76.06
Our Approaches												
Context (No Detector)	25.38	9.65	13.40	14.75	38.31	7.80	13.54	5.87	34.16	12.31	14.16	17.21
Faster RCNN+Context	82.40	72.94	93.97	81.22	77.57	49.42	30.20	20.84	87.19	56.53	68.65	65.54
	(+1.70)	(+2.31)	(+0.52)	(+1.37)	(+2.99)	(+2.94)	(+8.98)	(+1.35)	(+0.48)	(+3.24)	(+1.25)	(+2.47)
SSD+Context	**91.62**	**79.90**	**96.77**	**93.80**	**83.63**	**56.40**	**58.24**	**42.61**	**92.75**	**73.80**	**79.56**	**77.19**
	(+0.34)	(+2.54)	(+0.21)	(+0.21)	(+4.10)	(+1.01)	(+1.64)	(+1.03)	(+0.09)	(+1.06)	(+0.16)	(+1.13)

Table 1. Per-class and mean average precision values (in %) we obtained in the TSWC-2017 localization challenge. Note that our method improves performance on all categories for both the Faster RCNN and the SSD baselines.

	SSD (ms)	ResNet-50 (ms)	NN search (ms)	Others (ms)	Total (ms)	mean AP (%)
Exact	53	35	2626	18	2732	77.19
Approximate ($T = 3$)	53	35	45	18	151	77.13

Table 2. Average per-image runtime statistics for the exact and the approximate inference methods. The efficient inference is about 18 times faster. System specs: i7-4790 CPU, 32GB DDR3 RAM, GTX TITAN X Pascal GPU. Test batch size set to 1. See text for details.

SSD [28]. As expected, SSD outperforms Faster RCNN and YOLO by a clear margin, and the performance difference between the latter two is small. The results reported here are obtained without using the approximate nearest neighbor search scheme. We note that the approximate nearest neighbor search only affects the performance slightly (mAP of 77.13% for approximate search v.s. 77.19% for exact search). A comparison on the computational costs is reported in Section 4.3.

Somewhat surprisingly, without using any object detectors we obtained a mean AP of 17.21% with *Context (No Detector)* by scene layout transfer alone. We note that this method should be regarded as more of an object proposal one as it does not aim at predicting the location of any particular object, but possible object locations and scales in general (see Figure 4). Both *Faster RCNN+Context* and *SSD+Context* compare favorably with their respective baselines, providing mean AP improvements of 2.47% and 1.13% respectively. Although SSD has encoded the spatial context for object detection in terms of utilizing feature maps from several different layers in a CNN, the transferred scene-specific layouts are able to further improve its performance. We note that the improvements are consistent for both methods and for all object categories. See Table 1 for detailed per-class AP comparisons.

4.3. Computational efficiency

Table 2 reports a comparison in average per-image runtimes between the exact and the approximate nearest neighbor search methods. The first two components, namely SSD and ResNet-50, are implemented with Caffe, and the rest parts are implemented with MATLAB. When choosing $T = 3$ in the approximate inference, the performance gap in terms of mean AP difference between the two methods is small, yet the efficient inference is about 18 times faster.

In addition to the NN search, another component in our method that may be considered time-consuming is the extraction of ResNet-50 features. A forward pass of ResNet-50 takes 35ms on a TITAN X Pascal. In a real-world application, this feature may be replaced by alternatives such as VGG-16 [39] and subsequently be integrated into detection networks (e.g., SSD), incurring less extra computation.

5. Conclusion

In this paper, we propose an efficient scene layout aware object detection method for traffic surveillance. The nonparametric scene layout transfer in our method provides a general approach to context modeling for object detection that can be used in conjunction with many other detection algorithms not mentioned in this paper. There are two future directions in which we wish to explore. First, we are interested in integrating the contextual model into the detection network, providing a unified model to facilitate end-to-end training. In addition, we wish to explore the correlations among objects of different classes in a single image, as well as among objects from a set of test images.

Acknowledgements

We thank Shuyang Chen and Hanyuan Chen for their help in our initial experiments, the anonymous reviewers for their insightful and detailed comments, and NVIDIA Corporation for the generous GPU donations.

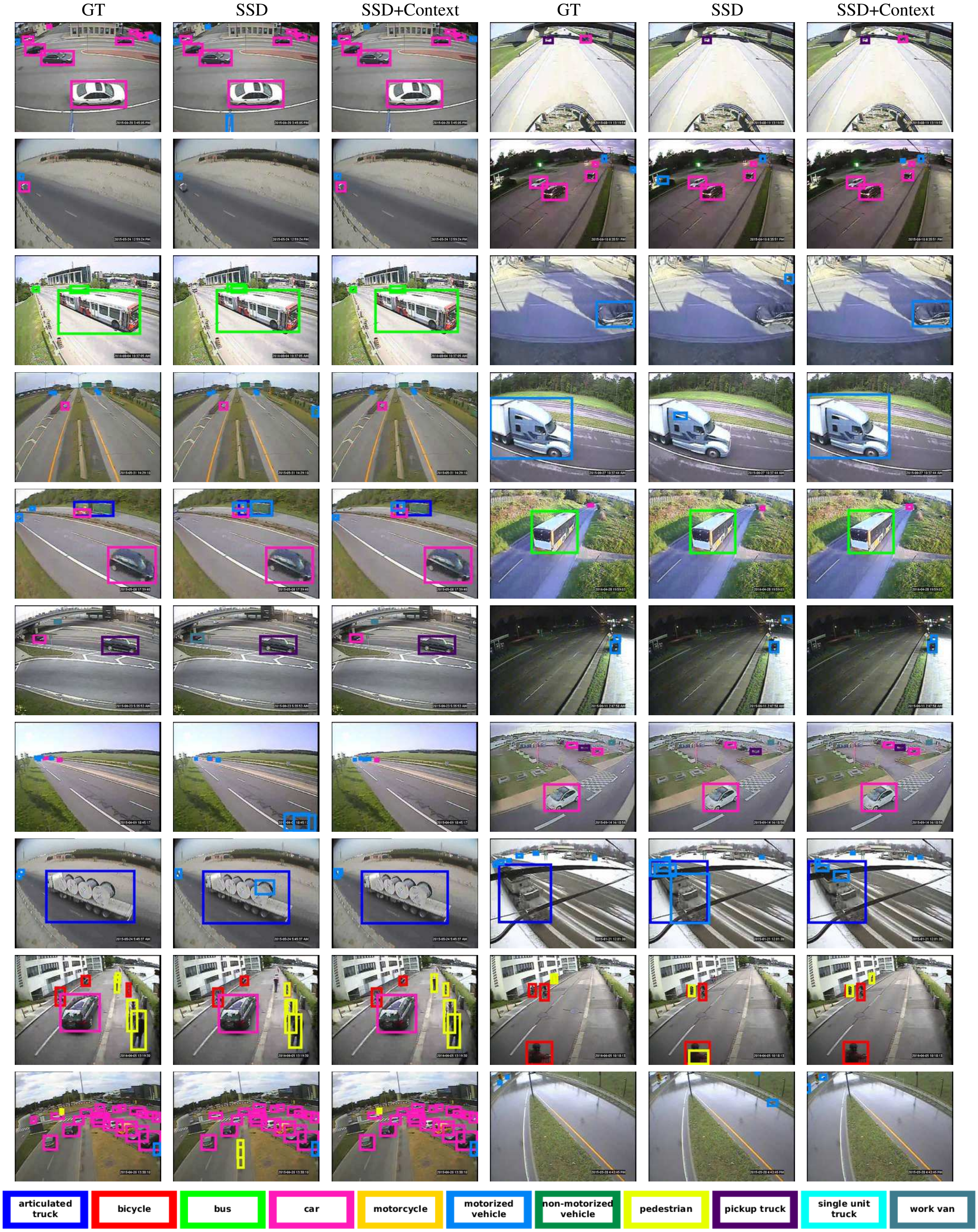

Figure 5. Example detection results on our held-out validation set of the TSWC-2017 localization challenge. Columns: **GT**: Ground-truth. **SSD**: Detections with SSD. **SSD-Context**: Detections with *SSD+Context*. Best viewed electronically, zoomed in.

References

[1] The Traffic Surveillance Workshop and Challenge 2017 (TSWC-2017). http://podoce.dinf.usherbrooke.ca. 1, 2

[2] S. Y. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *CVPR*, 2010. 1, 2

[3] A.-K. Batzer, C. Scharfenberger, M. Karg, S. Lueke, and J. Adamy. Generic hypothesis generation for small and distant objects. In *19th IEEE International Conference on Intelligent Transportation Systems*, 2016. 2

[4] M. Blaschko and C. Lampert. Object localization with global and local context kernels. In *BMVC*, 2009. 1, 2

[5] Y.-W. Chao, W. Choi, C. Pantofaru, and S. Savarese. Layout estimation of highly cluttered indoor scenes using geometric and semantic cues. In *International Conference on Image Analysis and Processing*, 2013. 1

[6] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 1, 2

[7] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009. 2

[8] A. Fathi, M. Balcan, X. Ren, and J. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011. 2

[9] O. D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03):485–508, 1988. 1

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32(9):1627–1645, 2010. 2

[11] A. Geiger, C. Wojek, and R. Urtasun. Joint 3d estimation of objects and scene layout. In *NIPS*, 2011. 2

[12] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[14] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. *ECCV*, 2010. 1

[15] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. 1, 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[18] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1

[19] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008. 1, 2

[20] S. Kim and M. Ouyang. Compute distance matrices with gpu. In *3rd Annual International Conference on Advances in Distributed & Parallel Computing*, 2012. 4

[21] S. Kluckner, T. Mauthner, P. Roth, and H. Bischof. Semantic image classification using consistent regions and individual context. In *BMVC*, 2009. 2

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2

[23] D. C. Lee, A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 1

[24] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 1

[25] Q. Li, V. Kecman, and R. Salman. A chunking method for euclidean distance matrix calculation on large dataset using multi-gpu. In *ICMLA*, 2010. 4

[26] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013. 1, 2

[27] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. PAMI*, 33(12):2368–2382, 2011. 2, 3

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2, 6

[29] W. Liu, R. Ji, and S. Li. Towards 3d object detection with bimodal deep boltzmann machines over rgbd imagery. In *CVPR*, 2015. 1, 2

[30] M. Maire, S. Yu, and P. Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011. 2

[31] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, et al. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 2

[32] K. Murphy, A. Torralba, W. Freeman, et al. Using the forest to see the trees: a graphical model relating features, objects and scenes. *NIPS*, 2003. 1

[33] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 2

[34] J. Pan and T. Kanade. Coherent object detection with 3d geometric context from a single image. In *ICCV*, 2013. 2

[35] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 1, 2

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2, 5, 6

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 5, 6

[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. 2

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

[40] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *CVPR*, 2006. 1

[41] M. Sun, Y. Bao, and S. Savarese. Object detection with geometrical context feedback loop. In *BMVC*, 2010. 1, 2

[42] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 2

[43] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 1, 2

[44] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004. 2

[45] A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin, et al. Context-based vision system for place and object recognition. In *ICCV*, 2003. 1

[46] T. Wang, X. He, and N. Barnes. Glass object segmentation by label transfer on joint depth and appearance manifolds. In *ICIP*, 2013. 2

[47] L. Wolf and S. Bileschi. A critical view of context. *IJCV*, 69(2):251–261, 2006. 1, 2

[48] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object detection for multi-class segmentation. In *CVPR*, 2010. 2

[49] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 2

[50] D. Zhang, X. He, and H. Li. Data-driven street scene layout estimation for distant object detection. In *DICTA*, 2014. 2

[51] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *CVPR*, 2015. 2