Multi-instance Object Segmentation with Occlusion Handling

Yi-Ting Chen[1] Xiaokai Liu[1,2] Ming-Hsuan Yang[1]
University of California at Merced[1] Dalian University of Technology[2]

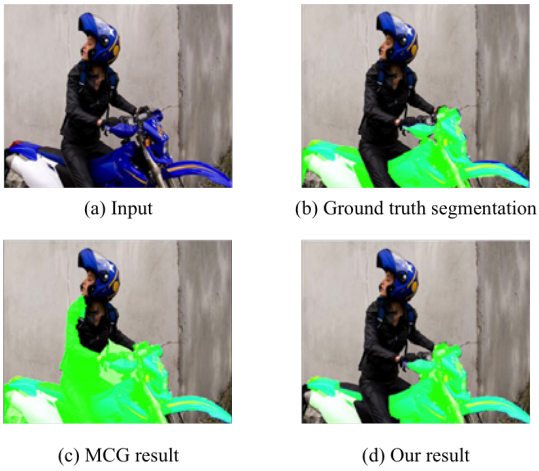
Abstract

We present a multi-instance object segmentation algorithm to tackle occlusions. As an object is split into two parts by an occluder, it is nearly impossible to group the two separate regions into an instance by purely bottom-up schemes. To address this problem, we propose to incorporate top-down category specific reasoning and shape prediction through exemplars into an intuitive energy minimization framework. We perform extensive evaluations of our method on the challenging PASCAL VOC 2012 segmentation set. The proposed algorithm achieves favorable results on the joint detection and segmentation task against the state-of-the-art method both quantitatively and qualitatively.

1. Introduction

Object detection and semantic segmentation are core tasks in computer vision. Object detection aims to localize and recognize every instance marked by a bounding box. However, bounding boxes can only provide coarse positions of detected objects. On the other hand, semantic segmentation assigns a category label to each pixel in an image, which provides more accurate locations of objects. However, semantic segmentation does not provide the instance information (e.g., number of instances). Intuitively, it is beneficial to jointly tackle the object detection and semantic segmentation. However, this is challenging due to occlusions, shape deformation, texture and color within one object and obscured boundaries with respect to other image parts in real-world scenes.

Occlusion is the main challenge in providing accurate segmentation results. A typical semantic segmentation [3, 6, 10] starts with generating segmentation hypotheses by a category-independent bottom-up segmentation algorithm [5, 7, 4] followed by class-specific classifiers. In many cases, bottom-up object segmentation algorithms cannot correctly handle occlusions where an object is spilt into two separate regions since they lack top-down information. Figure 1(a) shows such an example where a motorbike is occluded by the leg of a person and is split into two parts. Here, the best hypothesis (with respect to highest intersection-over-union (IoU) score) generated by the top performing segmentation algorithm (Multiscale Combinatorial Grouping, MCG [5]) fails to parse the motorbike correctly as shown in Figure 1(c).



(a) Input (b) Ground truth segmentation (c) MCG result (d) Our result

Figure 1: **Segmentation quality comparison.** Given an image (a), our method (d) can handle occlusions caused by the leg of the person while MCG [5] (c) includes the leg of the person as part of the motorbike. Note that our result has an IoU of 0.85 and the MCG result has an IoU of 0.61. Moreover, the segment in (c) is classified as a *bicycle* using class-specific classifiers whereas our segment can be classified correctly as a *motorbike*.

In this work, we address this issue by developing an algorithm suited to handle occlusions. To tackle occlusions, we incorporate both top-down and bottom-up information to achieve accurate segmentations under occlusions. We start by finding the occluding regions (i.e., the overlap between two instances). In case of Figure 1, finding the overlap between the person and motorbike gives the occluding region, i.e., leg of the person. To find these regions, we need to parse and categorize the two overlapping instances. Recently, a large scale convolutional neural network (CNN) is applied to obtain highly discriminative features for training class-specific classifiers [16] (i.e., R-CNN). The classifiers can categorize the object in a bounding box with a high

accuracy on the challenging PASCAL VOC dataset [11]. Based on R-CNN, Hariharan *et al*. [19] propose a simultaneous detection and segmentation (SDS) algorithm. Unlike R-CNN, SDS inputs both bounding boxes and segmentation foreground masks to a modified CNN architecture to extract CNN features. Afterward, features are used to train class-specific classifiers. This framework shows a significant improvement in the segmentation classification task. This classification capability provides us a powerful top-down category specific reasoning to tackle occlusions.

We use the categorized segmentation hypotheses obtained by SDS to infer occluding regions by checking if two of the top-scoring categorized segmentation proposals are overlapped. If they overlap, we record this occluding region into the occluding region set. On the other hand, the classification capability are used to generate class-specific likelihood maps and to find the corresponding category specific exemplar sets to get better shape predictions. Then, the inferred occluded regions, shape predictions and class-specific likelihood maps are formulated into an energy minimization framework [30, 27, 1] to obtain the desired segmentation candidates (e.g., Figure 1(d)). Finally, we score all the segmentation candidates by using the class-specific classifiers.

We demonstrate the effectiveness of the proposed algorithm by comparing with SDS on the challenging PASCAL VOC segmentation dataset [11]. The experimental results show that the proposed algorithm achieves favorable performance both quantitatively and qualitatively; moreover, suggest that high quality segmentations improve the detection accuracy significantly. For example, the segment in Figure 1(c) is classified as a *bicycle* whereas our segment in Figure 1(d) can be classified correctly as a *motorbike*.

2. Related Work and Problem Context

In this section, we discuss the most relevant work on object detection, object segmentation, occlusion modeling and shape prediction.

Object Detection. Object detection algorithms aim to localize and recognize every instance marked by a bounding box. [12, 34, 16]. Felzenszwalb *et al*. [12] propose a deformable part model that infers the deformations among parts of the object by latent variables learned through a discriminative training. The "Regionlet" algorithm [34] detects an object by learning a cascaded boosting classifier with the most discriminative features extracted from subparts of regions, i.e., the regionlets. Most recently, a R-CNN detector [16] facilitate a large scale CNN network [26] to tackle detection and outperforms the state-of-the-art with a large margin on the challenging PASCAL VOC dataset.

Object Segmentation. Recent years have witnessed significant progress in bottom-up object segmentation algorithms [25, 7, 4, 35]. Arbelaez *et al*. [4] develop a unified approach to contour detection and image segmentation based on the gPb contour detector [4], the oriented watershed transform and the ultrametric contour map [2]. Carreira and Sminchisescu [7] generate segmentation hypotheses by solving a sequence of constrained parametric min-cut problems (CPMC) with various seeds and unary terms. Kim and Grauman [25] introduce a shape sharing concept, a category-independent top-down cue, for object segmentation. Specifically, they transfer a shape prior to an object in the test image from an exemplar database based on the local region matching algorithm [24]. Most recently, the SCALPEL [35] framework that integrates bottom-up cues and top-down priors such as object layout, class and scale into a cascade bottom-up segmentation scheme to generate object segments is proposed.

Semantic segmentation [3, 6, 28, 16, 31] assigns a category label to each segment generated by a bottom-up segmentation algorithm. Arbelaez *et al*. [3] first generate segmentations using the gPb framework [4]. Then, rich feature representations are extracted for training class-specific classifiers. Carreira *et al*. [6] starts with the CPMC algorithm to generate hypotheses. Then, they propose a second order pooling (O_2P) scheme to encode local features into a global descriptor. Then, they train linear support linear regressors on top of the pooled features. On the other hand, Girshick *et al*. [16] extract CNN features from the CPMC segmentation proposals and then apply the same procedure as in O_2P framework to tackle semantic segmentation. Most recently, Tao *et al*. [31] integrate a new categorization cost, based on the discriminative sparse dictionary learning, into the conditional random field model for semantic segmentation. A similar work that also utilizes the estimated statistics of mutually overlapping mid-level object segmentation proposals to predict optimal full-image semantic segmentation is proposed [28]. On the other hand, the proposed algorithm incorporates both category specific classification and shape predictions from mid-level segmentation proposals in an energy minimization framework to tackle occlusions.

Occlusion Modeling. Approaches for handling occlusion have been studied extensively [32, 36, 15, 37, 21, 23, 14]. Tighe *et al*. [32] handle occlusions in the scene parsing task by inferring the occlusion ordering based on a histogram given the probability for the class c_1 to be occluded by the class c_2 with overlap score. Winn and Shotton [36] handle occlusions by using a layout consistent random field, which models the object parts using a hidden random field where pairwise potentials are asymmetric. Ghiasi *et al*. [15] model occlusion patterns by learning a pictorial structure with local mixtures using large scale synthetic data. Gao *et al*. [14] propose a segmentation-aware model that handles occlusions by introducing binary variables to denote the visibility of the object in each cell of a bounding box. The assignment

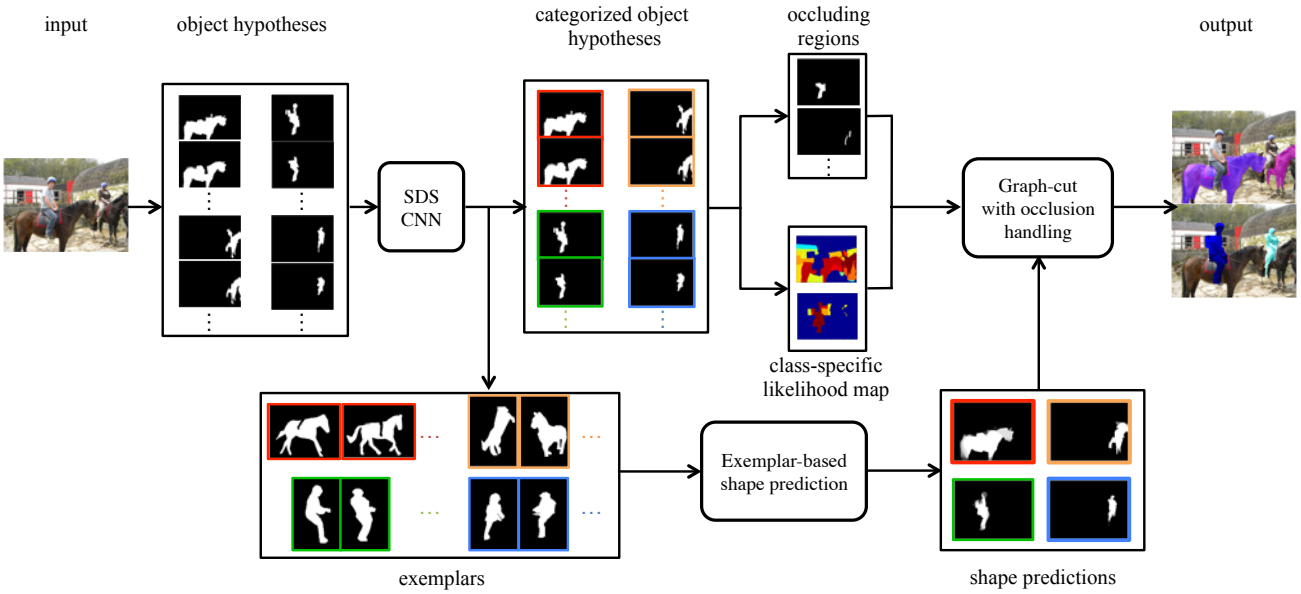


Figure 2: **Overall framework.** The framework starts by generating object hypotheses using MCG [5]. Then, a SDS CNN architecture [19] extracts CNN features for each object hypothesis, and subsequently the extracted features are fed into class-specific classifiers to obtain the categories of object hypotheses. Categorized segmentation hypotheses are used to obtain class-specific likelihood maps, and top-scoring segmentation proposals are used to infer occluding regions. Meanwhile, these exemplars serve as inputs to the proposed exemplar-based shape predictor to obtain a better shape estimation of an object. Finally, the inferred occluding regions, shape predictions, class-specific likelihood maps are formulated into an energy minimization problem to obtain the desired segmentation.

of a binary variable represents a particular occlusion pattern and this assignment is different from [33], which only models occlusions due to image boundaries (e.g., finite field of view). Hsiao and Hebert [23] take a data driven approach to reason occlusions by modeling the interaction of objects in 3D space. On the other hand, Yang *et al*. [37] tackle occlusions by learning a layered model. Specifically, this layered model infers the relative depth ordering of objects using the outputs of the object detector. In this work, we tackle occlusions by incorporating top-down category specific reasoning and shape prediction through exemplars, and bottom-up segments into an energy minimization framework.

Shape Prediction. Shape is an effective object descriptor due to the invariance property to lighting conditions and color. Several recent works have attempted to use the shape prior to guide the segmentation inference. Yang *et al*. [37] use the detection results to generate shape predictions based on the content of the bounding boxes. Gu *et al*. [17] aggregate posetlet activations and obtain the spatial distribution of contours within an image cell. He and Gould [21] apply the exemplar SVM to get a rough location and scale of candidate objects, and subsequently project the shape masks as an initial object hypothesis. In this paper, we obtain fine-grained shape priors by evaluating the similarities between the segmentation proposals and the exemplar templates.

3. Proposed Algorithm

3.1. Overview

In this section, we present the detail of the proposed multiple-instance object segmentation algorithm with occlusion handling in details. We first introduce the joint detection and segmentation framework and then our approach to tackle occlusions. Figure 2 illustrates the proposed algorithm.

3.2. Joint Detection and Segmentation

In this section, we briefly review SDS algorithm proposed by Hariharan *et al*. [19]. SDS consists of the following four steps. First, they generate category-independent segmentation proposals based on MCG [5]. Then, these segments are fed into a CNN network to extract features, and this CNN network is based on the R-CNN framework [16]. the CNN architecture of SDS is shown in Figure 3. This architecture consists of two paths, box and region. The box pathway is the same network as the R-CNN framework. The R-CNN has been shown to be effective in classifying the object proposals in the detection task. However, the R-CNN does not perceive the foreground shape directly. Hariharan *et al*. adopt the idea proposed by Girshick *et al*. by computing another CNN features on another

bounding box where it only has the foreground contents. Third, The two resulting CNN feature vectors are concatenated and the result is given as the input to train class-specific classifiers. Note that the two pathways are trained jointly in this framework. These classifiers assign scores for each category to each segmentation proposal. Finally, a refinement step is conducted to boost the performance. More details about the SDS algorithm can be found in [19].

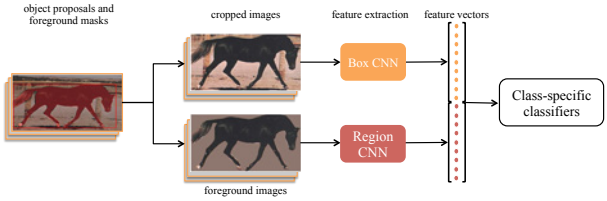


Figure 3: **SDS CNN architecture** [19]. SDS first applies MCG [5] to obtain foreground masks and the corresponding bounding boxes. Foreground images and cropped images are fed into Region and Box CNN respectively to jointly train the CNN network. Finally, the grouped CNN features are used to train class-specific classifiers.

3.3. Class-specific Likelihood Map

From SDS, we obtain a set of categorized segmentation hypotheses $\{h_k\}_{k=1}^{K}$ and scores $\{s_{h_k}^{c_j}\}_{k=1}^{K}$, where $c_j \in \mathcal{C}$ and $\mathcal{C}$ is a set of target classes. We use superpixel to represent an image I. For each superpixel sp covered by h_k, we record the corresponding category and score. After examining all the segmentation proposals, each superpixel has a list $\{s_{sp}^{c_j}\}_{sp\in I}^{c_j\in\mathcal{C}}$ indicating the score of a superpixel belonging to the class c_j. Then, the class-specific likelihood map is defined as the mean over the scores of all superpixels being the class c_j. Figure 4(b) and Figure 4(c) are the person and horse likelihood maps respectively.

Due to the high computational load of the gPb edge detector [4], we generate the superpixel map by using [8]. The resulting superpixel maps are shown in Figure 4(a).

(a) Superpixel (b) Person (c) Horse

Figure 4: The superpixel map and class-specific likelihood maps.

3.4. Exemplar-Based Shape Predictor

Bottom-up segmentation proposals tend to undershoot (e.g., missing parts of an object) and overshoot (e.g., containing background clutter). Thus, we propose an exemplar-based shape predictor to better estimate the shape of an object. The framework of the proposed shape predictor is shown in Figure 5.

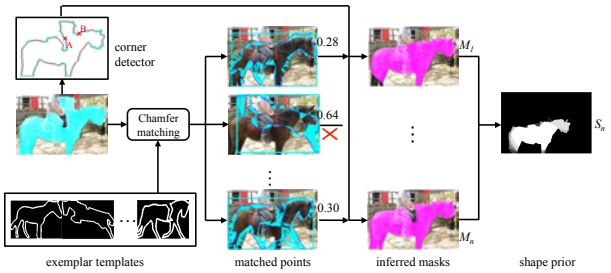


Figure 5: Overview of the exemplar-based shape predictor. This figure shows an example that the shape predictor uses the top-down class-specific shape information to remove the overshooting on the back of the horse.

We assume that segmentation proposals can provide instance-level information to a certain extent whereas these proposals may be undershooting or overshooting in reality. We aim to remove these issues according to the global shape cues and simultaneously recover the object shape. We thus propose a non-parametric, data-driven shape predictor based on the chamfer matching (CM). Given a proposal, we identify and modify strong matches locally based on the chamfer distance to every possible exemplar template. After aggregating all the matches for the best-scoring segmentation proposals, the matches are automatically clustered into a sequence of shape priors. Note that exemplar templates are selected from the VOC 2012 segmentation training set.

We first choose the top 20 scoring segmentation proposals from each class. Given a proposal, we first slightly enlarges 1.2x width and height of the segmentation proposal as the search area. Then, we start with placing an exemplar template at the top left corner of the enlarged search area with a step size of 5 pixels. A fast CM algorithm [29] is applied to evaluate the distance between the contour of the proposal and the contour of the exemplar template. CM provides a fairly robust distance measure between two contours and can tolerate small rotations, misalignments, occlusions and deformations to a certain extent.

The chamfer distance between the contour of the proposal U and the contour of the exemplar template T is given by the average of the distance between each point $\mathbf{t}_i \in T$ and its nearest point $\mathbf{u}_j$ in U as

$$d_{CM}(T,U) = \frac{1}{|T|}\sum_{\mathbf{t}_i\in T}\min_{\mathbf{u}_j\in U}|\mathbf{t}_i - \mathbf{u}_j|\,, \qquad (1)$$

where $|T|$ indicates the number of the points on the contour of the exemplar template T and we use boldface to

represent a vector. The matching cost can be computed efficiently via a distance transformation image $DT_U(\mathbf{x}) = \min_{\mathbf{u}_j \in U} |\mathbf{x} - \mathbf{u}_j|$, which specifies the distance from each point to the nearest edge pixel in U. Then, (1) can be efficiently calculated via $d_{CM}(T, U) = \frac{1}{n} \sum_{\mathbf{t}_i \in T} DT_U(\mathbf{t}_i)$. Based on the instance-level assumption, for a segmentation proposal of size w, we limit the searching scale in $[w \times 1.1^{-3}, w]$. By searching over the scale space, we select the one with the minimum distance as the shape candidate U^*. Among all the exemplar templates, we choose the top 5 matches for a proposal.

However, CM only provides discrete matched points. We need to infer a closed contour given all matched points. Moreover, CM cannot handle undershooting and overshooting effectively. Therefore, we propose a two-stage approach to solve these issues. First, we eliminate the effects of the large distances in the $DT_U(\mathbf{x})$ due to undershooting and overshooting by truncating those $DT_U(\mathbf{x})$ that are above τ to τ. Second, undershooting and overshooting always lead to contour inconsistency. We conduct the following processes to remove the contour inconsistency. We first apply the Harris corner detector [20] to detect inconsistent points (blue dots in Figure 5). We choose three inconsistent points and check the number of matched points on the adjacent contour segments that is formed by inconsistent points. If less than 20% of points on segments are matched with a template, we index the common inconsistent point, the middle one of the three inconsistent points. We then choose another three inconsistent points and conduct the aforementioned process. Finally, we remove the indexed inconsistent points from the inconsistent point set. In this way, we are able to effectively remove those odd contours and obtain a better object shape estimate.

After collecting all the strong matches for those segmentation candidates with high classification scores, we apply Affinity Propagation (AP) [13] to find representative shape priors $\{S_n\}_{n=1}^N$, where N is the number of clusters and is determined automatically. A shape prior corresponds to a cluster $cls(n)$. The n-th shape prior is defined as the weighted mean of every matched inferred mask M_m in the cluster $cls(n)$:

$$S_n = \frac{1}{|cls(n)|} \sum_{M_m \in cls(n)} s_{h_k}^{c_j} \cdot s_{h_k}^{CM} \cdot M_m \,, \qquad (2)$$

where $s_{h_k}^{c_j}$ is the classification score of the proposal h_k for the class c_j. The parameter $s_{h_k}^{CM} = \exp(-\frac{d_{CM}^2}{\sigma^2})$ is the chamfer matching score between the contour of the proposal and the contour of the exemplar template. Note that shape priors $\{S_n\}_{n=1}^N$ are probabilistic. We threshold the shape prior by an empirically chosen number (i.e., 0.6 in the experiments) to form the corresponding foreground mask. We denote the thresholded shape priors as $\{\widetilde{S}_n\}_{n=1}^N$. Finally, we form a set of foreground masks $\mathcal{F} = \{\widetilde{S}_1, \widetilde{S}_2, \cdots, \widetilde{S}_N, h_1, h_2, \cdots, h_K\}$ by concatenating thresholded shape priors and segmentation proposals.

3.5. Graph Cut with Occlusion Handling

In this section, we introduce the proposed graph cut formulation to address occlusions. Specifically, we infer the occluding regions (i.e., the overlap between two instances) based on segmentation proposals with top classification scores. We formulate the occluding regions into the energy minimization framework.

Let y_p denote the label of a pixel p in an image and $\mathbf{y}$ denote a vector of all y_p. The energy function given the foreground-specific appearance model $\mathcal{A}_i$ is defined as

$$E(\mathbf{y}; \mathcal{A}_i) = \sum_{p \in \mathcal{P}} U_p(y_p; \mathcal{A}_i) + \sum_{p,q \in \mathcal{N}} V_{p,q}(y_p, y_q) \,, \qquad (3)$$

where $\mathcal{P}$ denotes all pixels in an image, $\mathcal{N}$ denotes pairs of adjacent pixels, $U_p(\cdot)$ is the unary term and $V_{p,q}(\cdot)$ is the pairwise term. Our unary term $U_p(\cdot)$ is the linear combination of several terms and is written as

$$\begin{aligned} U_p(y_p; \mathcal{A}_i) \;=\; & -\alpha_{\mathcal{A}_i} \log \mathrm{p}(y_p; c_p, \mathcal{A}_i) - \alpha_{\mathcal{O}} \log \mathrm{p}(y_p; \mathcal{O}) \\ & -\alpha_{\mathcal{P}_{c_j}} \log \mathrm{p}(y_p; \mathcal{P}_{c_j}) \,. \end{aligned} \qquad (4)$$

For the pairwise term $V_{p,q}(y_p, y_q)$, we follow the definition as Grabcut [30].

The first potential $p(y_p; c_p, \mathcal{A}_i)$ evaluates how likely a pixel of color c_p is to take label y_p based on a foreground-specific appearance model $\mathcal{A}_i$. As in [30], an appearance model $\mathcal{A}_i$ consists of two Gaussian mixture models (GMM), one for the foreground ($y_p = 1$) and another for the background ($y_p = 0$). Each GMM has 5 components and each component is a full-covariance Gaussian over the RGB color space. Each foreground-specific appearance model $\mathcal{A}_i$ corresponds to the foreground and background models initialized using one of the elements in the $\mathcal{F}$. Note that the element in the set $\mathcal{F}$ is denoted as f_i.

The second potential $p(y_p; \mathcal{O})$ accounts for the occlusion handling in the proposed graph cut framework. To find occluding regions in a given image I, we first choose segmentation proposals with the top 10 scores from each category. Then, we check whether a pair of proposals overlaps or not. If they overlap, we record this occluding region into the occluding region set $\mathcal{O}$. We use classification scores to determine the energy of the pixel in the occluding regions. Thus, we define the second energy term $-\log \mathrm{p}(y_p; \mathcal{O})$ as

$$\text{-}\log \mathrm{p}(y_p; \mathcal{O}) = \begin{cases} \text{-}\log s_{f_i, y_p}^{c_j} + (2y_p\text{-}1)\gamma & \text{if } p \in \mathcal{O}^* \text{ and} \\ & s_{f_i \setminus \mathcal{O}^*}^{c_j} > s_{f_i}^{c_j} \\ \text{-}\log s_{f_i, y_p}^{c_j} & \text{otherwise} \end{cases} , \qquad (5)$$

Table 1: Per-class results of the joint detection and segmentation task using AP^r metric over 20 classes at 0.5 IoU on the VOC PASCAL 2012 segmentation validation set. All number are %.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	plant	sheep	sofa	train	TV	avg
SDS [19]	58.8	**0.5**	60.1	34.4	29.5	60.6	40.0	73.6	6.5	**52.4**	**31.7**	62.0	**49.1**	45.6	47.9	22.6	**43.5**	26.9	66.2	66.1	43.8
Ours	**63.6**	0.3	**61.5**	**43.9**	**33.8**	**67.3**	**46.9**	**74.4**	**8.6**	52.3	31.3	**63.5**	48.8	**47.9**	**48.3**	**26.3**	40.1	**33.5**	**66.7**	**67.8**	**46.3**

where $s^{c_j}_{f_i,y_p} = (s^{c_j}_{f_i})^{y_p}\left(1-(s^{c_j}_{f_i})^{1-y_p}\right)$ and the penalization $\gamma = -\log\frac{1}{|\mathcal{O}^*|}\sum_{p\in\mathcal{O}^*}(s^{c_j}_{f_i,y_p=0}+e)$. The parameter e is a small number to prevent a logarithm function returning infinity. The variable $\mathcal{O}^* \subset f_i \cap \mathcal{O}$ is one of the possible occluding regions for f_i. Given a foreground mask f_i and its class score $s^{c_j}_{f_i}$, we check the corresponding score of the region $f_i \backslash \mathcal{O}^*$. The score $s^{c_j}_{f_i\backslash\mathcal{O}^*}$ is obtained by applying the classifier of the class c_j and the region $f_i\backslash\mathcal{O}^*$ is obtained by removing the occluding region $\mathcal{O}^*$ from the foreground mask f_i. When $s^{c_j}_{f_i\backslash\mathcal{O}^*} > s^{c_j}_{f_i}$, that means the pixel p in the occluding region $\mathcal{O}^*$ is discouraged to be associated with the foreground mask f_i. In this case, we penalize the energy of the occluding regions by adding the penalization γ when $y_p = 1$. When $y_p = 0$, the energy of the occluding regions is subtracted with the penalization γ.

The third potential $p(y_p; \mathcal{P}_{c_j})$ corresponds to one of the class-specific likelihood map $\mathcal{P}_{c_j}$. Because of the probabilistic nature of class-specific likelihood map $\mathcal{P}_{c_j}$, we set the third potential as $p(y_p; \mathcal{P}_{c_j}) = \mathcal{P}^{y_p}_{c_j}(1-\mathcal{P}^{1-y_p}_{c_j})$. Finally, we iteratively minimize the energy function (3) as in [30]. Parameters of the foreground-specific appearance model will keep updating in each iteration until the energy function converges.

In the experiment, the parameter $\alpha_{\mathcal{A}_i}$ is set to be 1. We vary the parameter $\alpha_{\mathcal{O}}$ from 0.5 to 1 with a step size of 0.1. In addition, the parameter $\alpha_{\mathcal{P}_{c_j}}$ ranges from 0.1 to 0.3 with a step size of 0.1. In the pairwise term, we vary the constant controlling the smoothness degree from 60 to 240 with a step size of 60. We use these combinations of parameters to generate different segmentation candidates for a given foreground mask f_i. Finally, the segmentation candidates of all the foreground masks are applied with class-specific classifiers trained on top of the CNN features extracted from the SDS CNN architecture. Note that We apply the same classifiers as in SDS.

4. Experiments

We present experimental results for the joint detection and segmentation task on the PASCAL VOC 2012 validation segmentation set with the comparison to SDS [19]. There are 1449 images on the PASCAL VOC 2012 segmentation validation set. Moreover, we show our performance on the subset of segmentation validation set to better evaluate our occlusion handling as images in segmentation set mostly contain only one instance.

Table 2: Results of the joint detection and segmentation task using AP^r metric at different IoU thresholds on the VOC PASCAL 2012 segmentation validation set. The top two rows show the AP^r results using all validation images. The bottom two rows show AP^r using the images with *occlusions* between instances. We discuss the selection scheme in the text.

	# of images	IoU Score				
		0.5	0.6	0.7	0.8	0.9
SDS [19]	1449	43.8	34.5	21.3	8.7	0.9
Ours	1449	**46.3**	**38.2**	**27.0**	**13.5**	**2.6**
SDS [19]	309	27.2	19.6	12.5	5.7	1.0
Ours	309	**38.4**	**28.0**	**19.0**	**10.1**	**2.1**

4.1. Results of Joint Detection and Segmentation

Experimental Setting. We use AP^r to evaluate the proposed algorithm against SDS [19] on the joint detection and segmentation task. However, the recent works on object proposal algorithms [9, 22] show that an IoU of 0.5 is not sufficient for different purposes. Thus, Hariharan *et al*. propose to vary IoU scores from 0.1 to 0.9 to show their algorithm can be adopted for different applications. In our application, we aim to provide accurate segmentations, thus we choose thresholds from 0.5 to 0.9 in the experiments.

In addition to the above, we also collect a subset of images from the VOC 2012 validation dataset to form the $\text{VOC}_{occluded}$. Each image in the $\text{VOC}_{occluded}$ dataset satisfies the following: (a) It contains at least two instances (with respect to the VOC object categories) and (b) There is an overlap between two instances in the image. In the end, the $\text{VOC}_{occluded}$ contains 309 images in total and it helps to evaluate the detection performance of our proposed algorithm under occlusions.

Experimental Results. We use the benchmarking source code provided by Hariharan *et al*. [19] and follow the same protocols to evaluate the proposed algorithm on the joint detection and segmentation task. in Table 1 shows per-class results of the joint detection and segmentation task using AP^r metric (at IoU 0.5) on all the images of the validation set. Our method is highly beneficial for object classes such

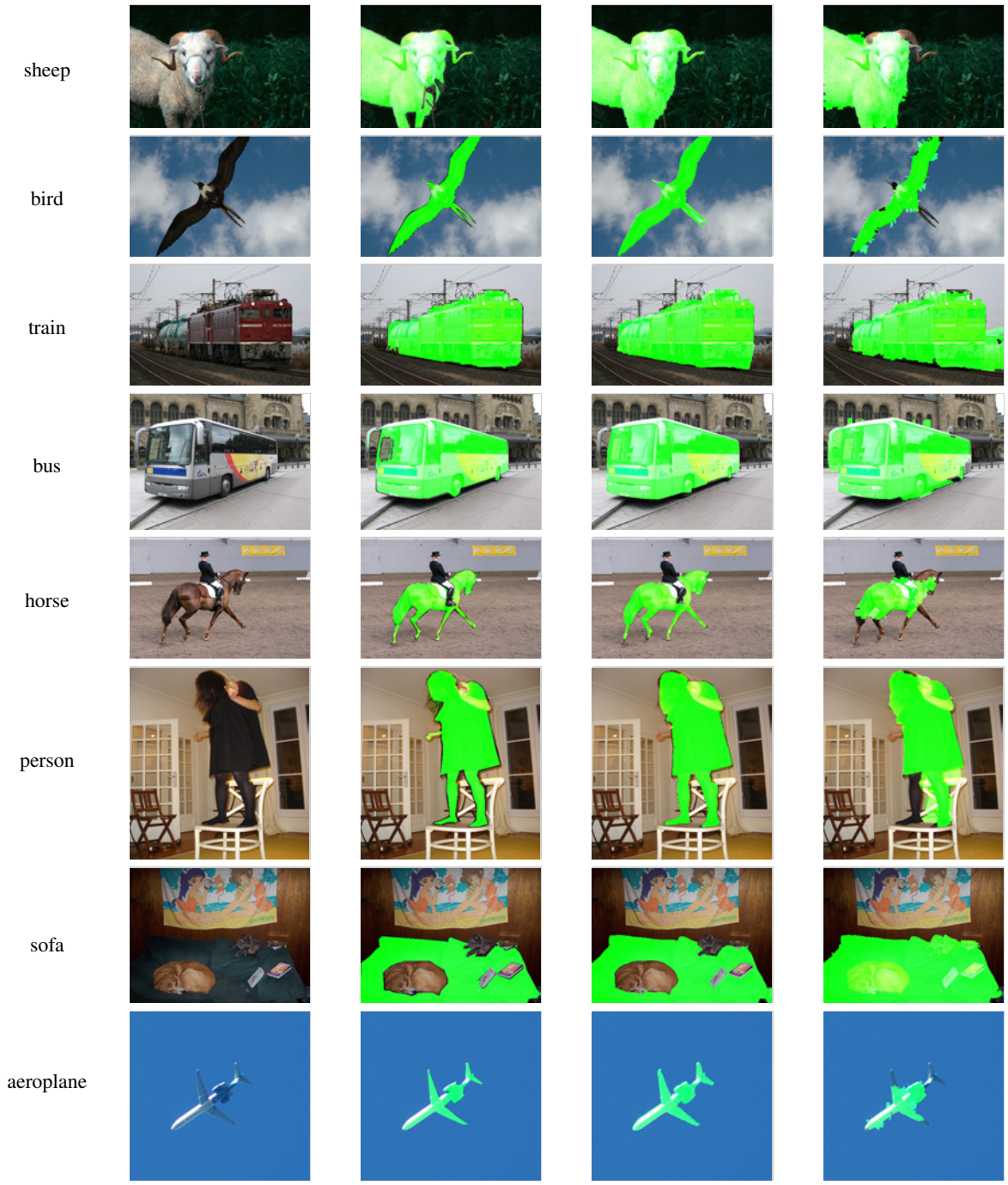
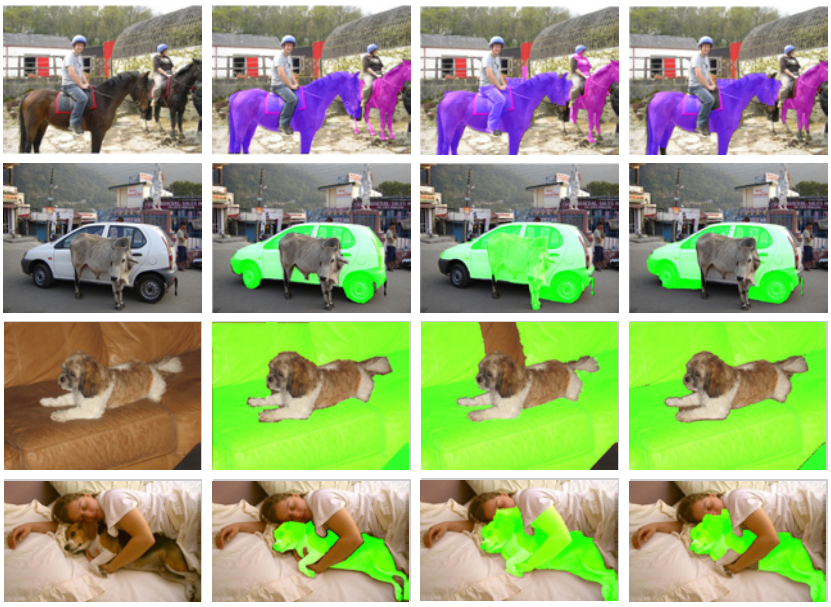


Figure 6: Top detection results (with respect to the ground truth) of SDS [19] and the proposed algorithm on the PASCAL VOC 2012 segmentation validation dataset. Compared with SDS, the proposed algorithm obtains favorable segmentation results for different categories. Best viewed in color.

as boat, bus, car and sofa by boosting the performance by more than 5%. Overall, the proposed algorithm performs the best in 15 out of the 20 categories. Note that in our experiments, SDS obtains a much lower AP^r in the bike category compared to the original scores reported in [19]. This is because we evaluate the performance using the VOC 2012 segmentation annotations from the VOC website instead of annotations from the semantic boundary database



(a) Input (b) Ground truth (c) MCG [5] (d) Ours

Figure 7: Some representative segmentation results with comparisons to MCG [5] on the PASCAL VOC segmentation validation dataset. These results aim to present the occlusion handling capability of the proposed algorithm.

(SBD) [18].

The first two rows of Table 2 show the AP^r at different IoUs on all the VOC images. The results suggest that high quality segmentation candidates boost the detection results at high IoUs. In particular, we achieve more than 5% jump in performance at high IoUs. Moreover, the bottom two rows of Table 2 show the proposed algorithm outperform SDS by a large margin on the *occlusion* images from $\text{VOC}_{occluded}$ dataset. This suggests that an algorithm with occlusion handling can even boost the detection results significantly.

We present qualitative results in Figure 6 and 7. Figure 6 shows the segmentation quality comparisons of the top detection results (with respect to the ground truth). The proposed algorithm obtains favorable segmentation results for different categories. Although we show promising segmentation quality in Figure 6, the segmentation quality of the best detected proposal may not be the best. We further present the Figure 7 to demonstrate that the proposed algorithm generates high quality segmentations. Moreover, it shows the capability to handle occlusions.

5. Ablation studies

In this section, we conduct ablation studies to understand how critical are the exemplar-shape based predictor and occlusion regularization in (5) for the performance of the joint detection and segmentation task. First, we disable the functionality of the occlusion regularization in (5) and perform experiments on the segmentation datasets. The performance drops from 46.3% to 46%. On the other hand, when the functionality of the exemplar-shape based predictor is disabled, the performance drops to 39.3%.

Next, we conduct experiments on the occlusion subset. Without the occlusion regularization, the performance drops from 38.4% to 37.9%. If we turn off the exemplar-shape based predictor, the performance drops to 33.2%. The above studies suggest that exemplar-shape based predictor is more important than the occlusion regularization for the joint detection and segmentation task. We conclude that a better estimate of object shape helps detection significantly.

6. Conclusion

We present a novel multi-instance object segmentation to tackle occlusions. We observe that the bottom-up segmentation approaches cannot correctly handle occlusions. We thus incorporate top-down category specific reasoning and shape predictions through exemplars into an energy minimization framework. Experimental results show that the proposed algorithm generates favorable segmentation candidates on the PASCAL VOC 2012 segmentation validation dataset. Moreover, the results suggest that high quality segmentations improve the detection accuracy significantly especially for those image images with occlusions between objects.

Acknowledgment. This work is supported in part by the NSF CAREER Grant #1149783, NSF IIS Grant #1152576, and a gift from Toyota. X. Liu is sponsored by CSC fellowship. We thank Rachit Dubey and Simon Sáfár for their suggestions and the CVPR reviewers for their feedback on this work.

References

[1] E. Ahmed, S. Cohen, and B. Price. Semantic object selection. In *CVPR*, 2014.

[2] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *POCV*, 2006.

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

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[5] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[6] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012.

[7] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012.

[8] Y.-T. Chen, J. Yang, and M.-H. Yang. Extracting image regions by structured edge prediction. In *WACV*, 2015.

[9] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[10] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and segmentation. In *ECCV*, 2014.

[11] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge (VOC). *IJCV*, 88(2):303–338, 2010.

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

[13] B. J. Frey and D. Dueck. Object detection with discriminatively trained part based models. *Science*, 315(5814):972–976, 2007.

[14] T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011.

[15] G. Ghiasi, Y. Yang, D. Ramana, and C. Fowlkes. Parsing occluded people. In *CVPR*, 2014.

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

[17] C. Gu, P. Arbel, Y. Lin, K. Yu, and J. Malik. Multi-Component Models for Object Detection. In *ECCV*, 2012.

[18] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[20] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, 1988.

[21] X. He and S. Gould. An exemplar-based CRF for multi-instance object segmentation. In *CVPR*, 2014.

[22] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.

[23] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*, 2012.

[24] J. Kim and K. Grauman. Boundary preserving dense local regions. In *CVPR*, 2011.

[25] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[27] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012.

[28] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu. Composite statistical inference for semantic segmentation. In *CVPR*, 2013.

[29] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, 2010.

[30] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[31] L. Tao, F. Porikli, and R. Vidal. Sparse dictionaries for semantic segmentation. In *ECCV*, 2014.

[32] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014.

[33] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial truncation. In *NIPS*, 2009.

[34] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013.

[35] D. Weiss and B. Taskar. Scalpel: Segmentation cascades with localized priors and efficient learning. In *CVPR*, 2013.

[36] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006.

[37] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkess. Layered object models for image segmentation. *PAMI*, 34(9):1731–1743, 2012.