Semantic Object Selection

Ejaz Ahmed[1], Scott Cohen[2], and Brian Price[2]
[1]University of Maryland, College Park and [2]Adobe Research
ejaz@umd.edu,{scohen,bprice}@adobe.com

Abstract

Interactive object segmentation has great practical importance in computer vision. Many interactive methods have been proposed utilizing user input in the form of mouse clicks and mouse strokes, and often requiring a lot of user intervention. In this paper, we present a system with a far simpler input method: the user needs only give the name of the desired object. With the tag provided by the user we do a text query of an image database to gather exemplars of the object. Using object proposals and borrowing ideas from image retrieval and object detection, the object is localized in the target image. An appearance model generated from the exemplars and the location prior are used in an energy minimization framework to select the object. Our method outperforms the state-of-the-art on existing datasets and on a more challenging dataset we collected.

1. Introduction

Object segmentation is of great practical importance in computer vision, especially in image editing tasks where operations are restricted to a single object. An important goal in segmentation is to minimize the effort required to select a desired object.

A common approach to object selection is to require the user to provide mouse (or touch) input to indicate the desired object. Magic Wand [1] requires the user to click on the image and then it selects all pixels with some tolerance. With Intelligent Scissors [18], the user traces the boundary of the object. Graph Cut [3] methods typically require the user to stroke over the object and background. Grabcut [20], a well-known exception to this, rather requires the user to draw a bounding box around the object, and only needs strokes to fix any mistakes. Stroke-based methods have been applied to cosegmentation [2], which also require the collection of similar images with a common object. Matting methods [5, 14, 27] require a trimap to be specified. Due to the complexity of natural scenes, overlapping object and background color distributions, and complicated object boundaries, each of these methods often require a lot of tedious user interaction to accurately select the object.

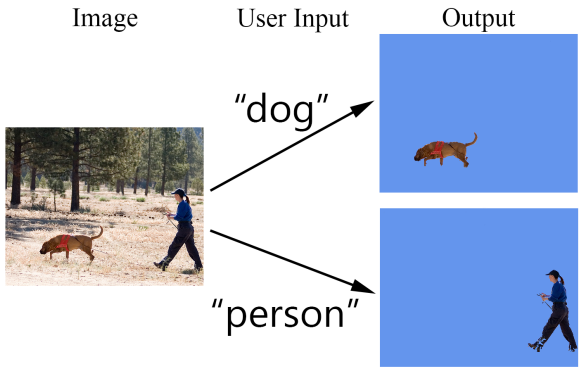

Figure 1. Given an image, the user simply provides the name of the object that he/she wants to select. The specified object is segmented by our method without further user input.

Another approach to segmentation is to perform the selection automatically. Semantic segmentation methods [15, 25] strive to label each pixel in an image with the correct object type. This requires the collection of a large dataset and a known fixed vocabulary, and often needs considerable training time. It is not directed toward an object of user interest, but rather operates on the whole image. Saliency methods [28, 4] are another approach to automatic object selection, where the object which is most visually salient is selected. This usually requires the object to be distinct from the background and quite large in the image. These methods work well if the desired object is in fact the salient object in the image, but this often is not the case.

The goal of this paper is to greatly reduce the user effort required to select an object. This is done by enabling the user to simply name the desired object (Fig. 1), either verbally as part of a natural language image processing engine, like PixelTone [13] or by typing it into a search box. For example, if the user makes a PixelTone request such as "Make the *cat* brighter", our method can be used to identify the cat pixels to be made brighter without any further input from the user (in PixelTone, a user has to paint on

the image to mark the cat, name the selection, and then the user can issue semantic editing requests that mention "cat"). With the current proliferation of natural language interfaces for all sorts of tasks (e.g. Siri), we think our method will be very important for advancing image editing via natural language input. This method is more directed than semantic segmentation and does not require a large trained database. Unlike saliency methods, our method can select objects that are small and potentially not salient in the input image as well as objects in images with several salient objects.

We introduce the new problem of Semantic Object Selection in which a user simply specifies the class of the object to select in an image. We propose a solution that scales well with the number of classes, as we do *not* need to train a detector for each class we wish to recognize. At the core of our system is a novel, robust method using two types of image search for (i) classifying object proposals in the input image as containing the selection class or not and (ii) providing localization information and appearance models for the objects to select. More specifically, a text-based image search, e.g. Google or Microsoft Bing, is used to provide positive images containing instances of the selection class. Negative images unlikely to contain the selection class are also gathered. Object proposals are then computed and used as a query for an image-based search of the positive and negative examples. The object proposals likely to correspond to the desired object are used to compute localization and appearance models that are combined in an energy minimization framework to compute a final selection.

To the best of our knowledge, this tag-based selection system is the first in the literature. Thus we cannot perform direct comparisons. We have compared our method against various other state-of-the-art methods for related problems. We also implemented our own baselines which are competitive by themselves. We have shown results on various classes of the MSRC dataset and the recently introduced Object Discovery dataset. We have also collected more realistic and challenging dataset from imageNet containing dogs. We are comparable to the state-of-the-art on MSRC [24] and beat the state-of-the-art on Object Discovery [21] and our new dataset by a large margin.

2. Related Work

While we are unaware of previous work that addresses the problem of selecting a named object without further interaction, there are several lines of work that could be used to approach this problem.

Saliency methods aim to select the main object in an image by determining which image regions are most "salient". In [28], the saliency is determined by optimizing an energy function which encourages pixels to be salient if they are contained in regions that have high contrast to all other regions. The method in [4] uses similar contrast and location terms, and then computes a binary segmentation by including the computed saliency in a variant of GrabCut [20]. Since these methods do not select object of interest but rather select whichever region stands out, they cannot address the general semantic object selection problem where the object of interest is not the most "salient" object.

Semantic segmentation approaches [15, 25] attempt to automatically segment every object in an image. This could be extended to our problem by labeling every pixel and selecting the pixels corresponding to the named object. There are several drawbacks to this approach. Semantic segmentation methods solve a much large problem than needed, and do not focus on the object of interest. They require a large amount of pre-labeled data and require a predetermined label set that may not contain the desired object label. If the label set contains the desired object, it still may not be found in the image. Many require training classifiers for every label, which for a sufficiently large label set requires excessive computation. Exceptions to this are the non-parametric approaches. These use image retrieval to pull images from the training set for use in transferring labels to the image. For example, in [15] the nearest neighbors in the training set are retrieved and SIFT flow is used to transfer labels to the query image. In [25], globally-similar images are retrieved and the likelihood of each superpixel in query image belonging in each class according to the retrieved set is computed and used in an MRF to compute a segmentation.

There has been much work in object detection, for example [6, 17]. Since object detection localizes a object with a bounding box, it could easily be used to provide a bounding box around a desired object to initialize a segmentation process using a method such as Grabcut [20]. We propose this as a baseline method and compare to it in our results.

Cosegmentation methods [9, 8, 10, 19] operate on multiple input images and select in each image a common object. Such methods could be adapted to our problem by performing an Internet search on the query object and performing cosegmentation on the results together with the query image. In fact, Rubenstein *et al*. [21] propose a method designed to cosegment sets of images collected from an Internet search. It computes a segmentation by optimizing over a function with terms emphasizing sparseness and saliency. Because this method heavily relies on saliency, it is largely restricted to working well on images where saliency methods also work well.

A method which is related to ours is [23]. In this work, a user takes a relatively clean, close-up picture of a product and the goal is to find a similar product to the one in the image. This method iterates between localized image retrieval and selection estimation. The product image database has associated object masks, and the masks of retrieved images are transferred to the query image to estimate the location of the product within the query image. The selection is then

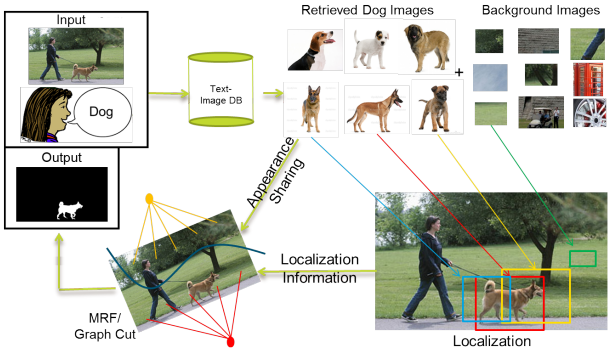

Figure 2. Overview of our system: User starts by providing the name of the object to segment. Text-based image search is performed to gather positive exemplars. Positive exemplars along with generalized negatives are then used localize object in the image. This is done with the help of our object retrieval based detection framework. Localization information along with appearance sharing from positive exemplars is used to formulate the segmentation problem as energy minimization. Graph cut is applied on the constructed graph to obtain the desired segmentation.

computed using a voting scheme based on the image search localizations and is refined using [20]. Note that [23] has a similar goal but simplified input (fairly rigid objects, large and centered in the image with little viewpoint variation).

Unlike [23], our method does not require a database of images with ground truth masks. Our method can handle object classes with larger appearance variations and objects that are less rigid than typical product objects since our method warps the retrieved images to the query image. Our use of object proposals also distinguishes our selection algorithm from the product search work. Clean images where the object of the image is quite large are required in [23] since in cluttered scenes it is difficult to retrieve images that match the object of interest. Our method uses object proposals to make good estimates of where the object may be, which helps avoid background clutter and allows it to handle more general photos of the world.

3. Overview

Our method takes as the user input the name of the object as shown in Fig. 2. Using this tag we do a text query into an image database to gather exemplars corresponding to the object. Along with positive exemplars we also gather generalized negative examples and put them in an image retrieval database. We then divide our image into object proposals [26]. Each object proposal queries the image retrieval database (using [22]) to validate the presence or absence of the object in a given object proposal.

Once we have found object proposals potentially containing the desired object and their corresponding exemplars, we estimate the location of the object in the corresponding exemplar. We transfer this information onto an object proposal using SIFT flow based image warping [16] to produce a *location prior*. We use this location prior to obtain image specific object and background models. We then combine the image specific appearance model and location prior in a graph cut energy minimization framework.

Figure 3. Positive exemplar database: Objects on white background and exemplars from PASCAL VOC (last 2 columns).

Note that any state-of-the-art object detectors like DPM or exemplar-SVM [6, 17] followed by our segmentation algorithm cannot be used here since such object detectors usually have an extremely expensive training phase, involving bootstrapping and hard negative mining phases. Since our goal is to deal with large number of object classes, we cannot use pre-trained models. Also in case of DPM, exemplar-based mask transfer cannot be used since various positive examples are clubbed together to build a model. Moreover, in our experiments we show that our method performs better than a DPM-based segmentation algorithm.

4. Localization

Given the tag of the object, our first challenge is to find the location of the object in the image. Our goal is to obtain an object location prior for the target image. We first collect positive exemplars corresponding to the object along with some generalized negative exemplars to build an image retrieval database [22]. We then break the target image into object proposals using [26] and validate the presence of the object in the object proposals. We use SIFT flow to transfer the location associated with validating exemplar areas to the target image. The accumulated location information provides a location prior.

4.1. Exemplar Retrieval Database

Once the tag of the corresponding object is provided by the user, our system needs to learn what our object looks like. We collect positive examples on white background for different classes by querying Google with "<object> on white background". This gave us a large database of positive exemplars (Fig. 3). We append this dataset with other publicly available positive sources such as PASCAL VOC.

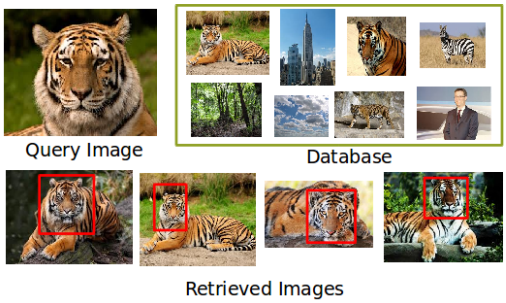

Figure 4. Object retrieval with localization: We use object retrieval system of [22] which returns ranked retrieved images along with the bounding box around the matched object.

We also use some generalized negative images from INRIA pedestrian dataset. Note that performance increases if the negative exemplars are representative of the background in the target image. Although it is not necessary to accurately represent all background objects, by identifying some likely background regions we can eliminate some potential false positives and improve the localization of the foreground.

Having positive and negative exemplars makes this a typical detection problem. Many state-of-the-art detection systems like [6, 17] try to solve this problem by learning a classifier between positives and negatives. While such a method is promising, it has its limitations in this scenario as they require a very expensive training step. We instead leverage concepts from image retrieval to obtain the location of the object in an image.

We use the retrieval system from [22] which uses a spatially-constrained similarity measure to handle rotation, scaling, view point change and appearance deformation. The similarity measure is calculated by geometrically aligning SIFT visual words indexing the query and database images; achieving object retrieval and localization simultaneously (Fig, 4). We put positive and negative exemplar images into an *exemplar database* for localized search using [22]. This involves computation of SIFT features and creation of an inverted file that stores feature locations for faster voting map generation during retrieval.

4.2. Detection via Object Proposal Validation

For object detection, the current state-of-the-art is based on exhaustive search. However, to enable the use of more expensive features and classifiers a selective search is more desired. We use object proposal method proposed in [26]. They have reported a recall of 96.7% with around 1500 windows per image on PASCAL VOC 2007. We divide the target image using object proposals. These object proposals contain desired object, other objects and even background. Our goal is to classify each object proposal as containing the desired object or not. To solve this, we use the exemplar database created in the previous step.

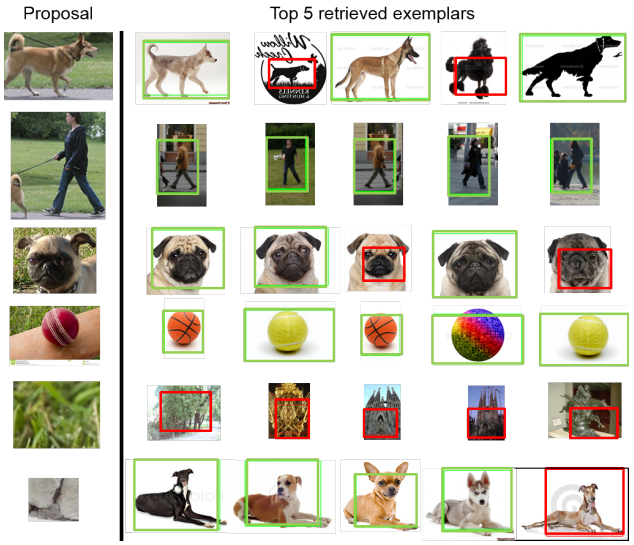

Figure 5. Validation: Each row shows an object proposal and its top 5 retrieved exemplars. Retrieved exemplars also contain the bounding box around the matched object. The color of the bounding box specifies whether the exemplar is considered as positive (green) or negative (red). If the box is not centered, e.g. in 1^{st} row 4^{th} exemplar, the exemplar is considered negative. Majority voting decides whether the object proposal contains the specified object or not. The last row shows an example of a false positive where an object proposal is incorrectly validated as a dog. The positive class for each query from top to bottom is dog, person, pug, ball, person and dog.

We query each object proposal into the exemplar database. When calculating the voting map for retrieval, we follow the general retrieval framework of [22], i.e., for each visual word k in the query, retrieve the image IDs and locations of k in these images through the inverted files. Object center locations and scores are then determined and votes are casted on corresponding voting maps. This results in ranking by similarity score of all the exemplars in the database along with the potential location of the object in the exemplars. We consider the top t exemplars for validation. Recall that each image in the retrieval database is known to be the object of interest or the background. Some of the exemplars in top t belong to the object (tag), while others might belong to background. It might also happen that the exemplar belongs to object (tag), but the bounding box returned by localization is not centered on the exemplar. In this case the exemplar is also considered as negative. From these top t exemplars, majority voting is performed and the object proposal is classified as belonging to the object (tag) if most of the exemplars in top t are positive. See Fig. 5.

4.3. Location Prior

For each positive retrieved exemplar, we desire to transfer the location of each pixel belonging to the object to the

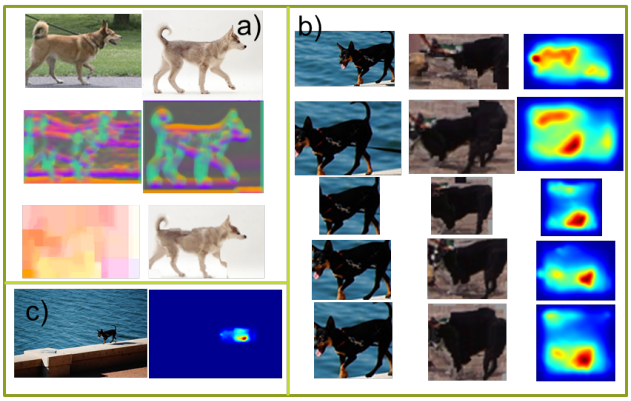

Figure 6. Mask Transfer: a) Warping of an exemplar (top right) onto the object proposal (top left). 2^{nd} row shows sift features for object proposal and exemplar. 3^{rd} column shows the sift flow correspondence(left) and warping of exemplar onto the object proposal(right). b) Top 1^{st} column shows object proposals, 2^{nd} column shows best exemplar warped onto the object proposal, and 3^{rd}column shows the saliency mask for the warped exemplars. c) Input image and aggregated location prior.

object proposal. Our retrieved images have no associated ground truth masks so we must estimate the location of the object in the retrieved images. We use saliency to estimate the object location. Since our object proposals are not usually not cluttered and often match to uncluttered retrieved images and since we explicitly search for objects on a white background to collect the retrieval set, we find that saliency works sufficiently well in this constrained use case.

For each object proposal containing the object, its best positive exemplar (according to retrieval score) is considered for segmentation transfer. We obtain soft segmentation mask on the exemplar image by computing saliency map of [7], which gives a score in $[0, 1]$ to each pixel in the exemplar image. We transfer this mask to the corresponding object proposal by SIFT flow warping [16]. Note that many positive object proposals can be shifted versions of the same object and hence their masks can be overlapping on the target image. All masks are aggregated on the target image and re-normalized to lie between $[0, 1]$. Fig. 6 shows this process.

5. Segmentation

Given the retrieved positive exemplars for each positive object proposal and the location prior, we compute a binary segmentation of the desired object. We pose the segmentation problem in a classic energy minimization framework [12, 20, 11]. Our unary terms consist of an image specific appearance model, an appearance model shared from exemplars, and a location prior. We iteratively minimize the energy, updating our models in each iteration.

Let x_p be the label of the pixel p in the image and $\mathbf{x}$ be the vector of all x_p. The energy function given the appearance model A and exemplar data X_E can be given by

$$E(\mathbf{x}; A, X_E) = \sum_p E_p(x_p; A, \mathcal{X}_\mathcal{E}) + \sum_{p,q} E_{pq}(x_p, x_q) \quad (1)$$

In this the pairwise potential is given by

$$E_{pq}(x_p, x_q) = \delta(x_p \neq x_q) \cdot d(p, q)^{-1} \cdot \exp(-\gamma \|c_p - c_q\|^2), \quad (2)$$

where c_p is the color at pixel p. This potential encourages smoothness by penalizing neighboring pixels taking different labels. The penalty depends on the color contrast between pixels, being smaller in regions around image edges (high contrast). We consider an 8-connected pixel grid.

Our unary term is a linear combination of three terms:

$$E_p(x_p, \mathcal{A}, \mathcal{X}_\mathcal{E}) = -\alpha_I \log p(x_p; c_p, \mathbf{A_I}) \quad (3)$$
$$-\alpha_{\mathcal{X}_\mathcal{E}} \log p(x_p; c_p, \mathbf{A}_{\mathcal{X}_\mathcal{E}}) - \alpha_M \log M_p(x_p; \mathcal{X}_\mathcal{E}).$$

Each potential $p(x_p; c_p, \mathbf{A})$ evaluates how likely a pixel of color c_p is to take label x_p, according to the appearance model $\mathbf{A}$. The first term uses an image specific image prior $\mathbf{A_I}$. The foreground and background appearances are each separately modeled using a 5 component GMM. The foreground and background are initialized using the location prior; all pixels whose location prior is below some threshold γ_B or above some γ_F are assumed to be background or foreground respectively and are included in the respective appearance model.

The appearance model $\mathbf{A}_{\mathcal{X}_\mathcal{E}}$ is obtained from the positive exemplars used to compute the location prior. This appearance model is useful in sharing information from exemplars and is particularly useful when segmenting object classes whose appearance does not change over exemplars (particular breed of dog, e.g. brown Labradors).

We obtain the location prior $\mathbf{M_p}$ using exemplar-based image retrieval in the previous step. It is a soft segmentation between $[0, 1]$ and has probabilistic nature. Thus we directly use $\mathbf{M_p}(x_p; \mathcal{X}_\mathcal{E}) = \mathbf{M_p}^{x_p}(1-\mathbf{M_p})^{1-x_p}$ as a unary potential in Eq. 3.

Our segmentation framework, shown in Figure 7, is inspired by [20, 12]. A graph is constructed with a node for each pixel and using unary and binary potentials from Eq. 1. Graph cut is then used to computed a binary segmentation. The image-specific appearance model is updated given the new foreground. We iterate (5 times) between solving the energy function using graph cut and updating the models.

6. Results

We present both qualitative and quantitative results on various datasets. We set $\gamma_B = 0.05$, $\gamma_F = 0.8$, $\alpha_I = 0.6$, and $\alpha_M = 0.4$. The trade off between the unary term and binary term is $\lambda = 50$. While for objects with consistent appearance, the exemplar-specific appearance model can be very useful, we largely tested on objects with a large variation in appearance and thus set $\alpha_{\mathcal{X}_\mathcal{E}} = 0$. To report qualitative results we use Jaccard similarity, i.e. intersection over

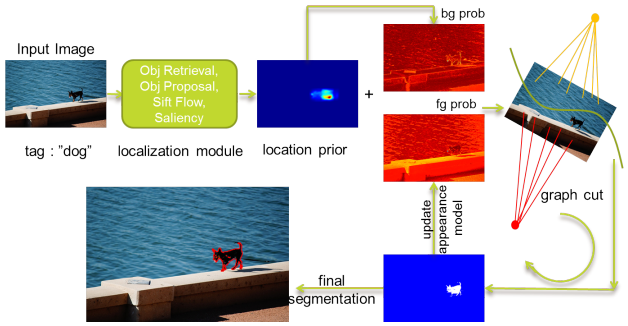

Figure 7. Segmentation Framework: Given the input image and the tag, object retrieval based localization is performed to obtain a location prior. Using this location prior, fg and bg probabilities are obtained. These probabilities along with the location prior are used to set the weights of the graph. Graph cut is applied to obtain intermediate segmentation which is used to update our models. After a few iterations a final selection is obtained.

class	ours	[21]	[8]	[10]	[9]	[19]
bike	**55.3**	54.1	43.3	29.9	42.3	42.8
bird	64.6	**67.3**	47.7	29.9	33.2	–
car	**66.8**	66.7	59.7	37.1	59	52.5
cat	**70.7**	66.2	31.9	28.7	37.6	39.4
chair	60.3	**62.2**	39.6	28.7	37.6	39.4
cow	78.5	**79.4**	52.7	33.5	45	26.1
dog	**69.1**	67.5	41.8	33	41.3	–
plane	**58.8**	56.7	21.6	25.1	21.7	33.4
sheep	**81.2**	78.9	66.3	60.8	60.4	45.7
average	**67.3**	66.5	45.0	34.1	42.0	39.9

Table 1. Results on MSRC dataset. We compare against Object Discovery [21], Joulin *et al*. [8], Kim *et al*. [10], Joulin *et al*. [9] and Mukherjee *et al*. [19]. Our method is slightly better or comparable to Object Discovery which is state-of-the-art on MSRC.

Methods	OD airplane	OD car	OD horse	ImageNet dog
Ours	**64.27**	**71.84**	**55.08**	**69.91**
OD [21]	55.81	64.42	51.65	–
Joulin *et al*. [9]	15.36	37.15	30.16	28.65
Joulin *et al*. [8]	11.72	35.15	29.53	24.69
DPM+Grabcut	39.47	68.00	50.12	48.24
CEN+Grabcut	37.29	64.96	48.89	34.53
GT+Grabcut (Upper bound)	50.87	80.82	65.99	79.52

Table 2. Results on Object Discovery(OD) and ImageNet Dog. On the Object Discovery dataset [21] we perform better than the state-of-the-art by a significant margin. We also compare against our DPM-based segmentation baseline method and outperform it by a significant margin. Note that we beat the upper bound (using ground-truth bounding boxes) on the airplane category. On ImageNet-dog we perform much better than DPM+Grabcut.

union of the result and ground truth segmentation. More results can be found in the supplementary material.

6.1. Results on MSRC Dataset

We report results on the MSRC dataset [24]. We search Google to get objects on white background as positive exemplars, and append this list with PASCAL VOC 2010 training positive examples for each class. 9 of the 14 classes of MSRC are present in PASCAL VOC 2010, we thus compute results on 9 classes of MSRC (around 30 images per class). We compare our performance with [10, 19, 9, 8] as reported in [8]. We also compare against the recent Object Discovery work [21] which uses dense correspondences between images to capture the visual variability of common object. This method works well when the object is salient in the image. The cosegmentation methods use all the test images as input to the system while the input to our system is just one label and one image. The quantitative results are given in Table 1.

Our method is significantly better than [10, 19, 9, 8]. It is also slightly better or comparable to Object Discovery [21]. The closeness in performance is due to the fact that the MSRC dataset contains images with a salient target object and uniform background. This acts as a boon to Object Discovery's approach which is tuned to work well in cases where object is the most salient object in the image. Our approach is a more general approach which works well in this scenario but is not limited to images with salient objects only. Qualitative results can be found in Fig. 10.

6.2. Result on Object Discovery Dataset

To prove our claim that our method is more general and works well when the object is not the only salient object in the image, we test the performance of our method on the Object Discovery dataset. This dataset was introduced in [21] and consists of images downloaded from the Internet. There is large variation in style, color, texture, pose, scale, position and viewing angle. The dataset consists of three classes, car, horse, and airplane, with around 100 images in each category.

In order to further prove effectiveness of our approach we implemented our own baselines. For each image, we initialize a centered bounding box covering 25% of the area of the image and initialize Grabcut [20] using this bounding box. We call this approach CEN+Grabcut. Next, we compared our approach with a detector-based method. We trained discriminative part based detectors [6] on the car, horse, and airplane categories from PASCAL VOC 2010. In order to select a detection threshold we obtained the PR-Curves and selected a threshold corresponding to f1 score. The detection bounding boxes obtained by running the detector are used to initialize Grabcut. We call this method as DPM+Grabcut. Finally, we initialize Grabcut with the ground truth bounding box of the objects in the image. We call this GT+Grabcut. Note that this is an upper bound of a detection plus Grabcut approach. Since [6] uses models

trained on PASCAL VOC 2010, we only use PASCAL VOC 2010 training images as positive exemplars so that the comparison is fair. We also compared against [21, 9, 8].

The quantitative results can be found in Table 2. Since the images contain objects which are not salient in the image (more realistic images), our approach performs better than Object Discovery. It performs better than detector-based segmentation DPM+Grabcut, which has an expensive training phase and is not practical in our scenario. Also note that for airplanes our approach even performs better than the detector-based method upper bound. This is evidence of the high quality of our location prior as the initial GMM foreground and background color models derived from the location prior lead to better results than initializing color models from the correct tight bounding box input. Qualitative comparisons can be found in Fig. 8 and more of our results in Fig. 9.

6.3. Results on Imagenet Dog

In order to test on a difficult real-world dataset where the object of interest is often small and not salient, we collected 100 images from ImageNet containing dogs. We show segmentation results on this dataset in Table 2. PASCAL VOC 2010 dog training positives were used for training DPM+Grabcut. Qualitative results can be found in Fig. 10.

7. Conclusion

In this paper we have proposed a new system for object selection. Our system has a far simpler interface for object selection, taking only the object name as input. In order to solve this problem we propose a exemplar-based localization method which relies on object retrieval. We break the image into object proposals and validate the presence of the object in the proposal. Location priors obtained in this way are then used to get an image specific appearance model and both are used to solve the segmentation problem in an MRF framework. We have introduced our own imageNet dog dataset and we outperform the state-of-the-art on a number of other datasets.

References

[1] Adobe system incorp. 2002. adobe photoshop user guide.

[2] D. Batra, C. M. Univerity, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, volume 1, pages 105–112, 2001.

[4] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.

[5] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *CVPR*, 2001.

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

[7] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS 19*, pages 545–552, 2007.

[8] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[9] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[10] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. *ICCV*, pages 169–176, 2011.

[11] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012.

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

[13] G. P. Laput, M. Dontcheva, G. Wilensky, W. Chang, A. Agarwala, J. Linder, and E. Adar. Pixeltone: a multimodal interface for image editing. In *SIGCHI*, 2013.

[14] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR*, 2006.

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

[16] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, pages 28–42, 2008.

[17] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[18] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *SIGGRAPH*, pages 191–198, 1995.

[19] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. *CVPR*, pages 1881–1888, 2011.

[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, volume 23, pages 309–314, 2004.

[21] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. *CVPR*, June 2013.

[22] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 2012.

[23] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, 2012.

[24] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

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

[26] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[27] J. Wang and M. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007.

[28] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013.

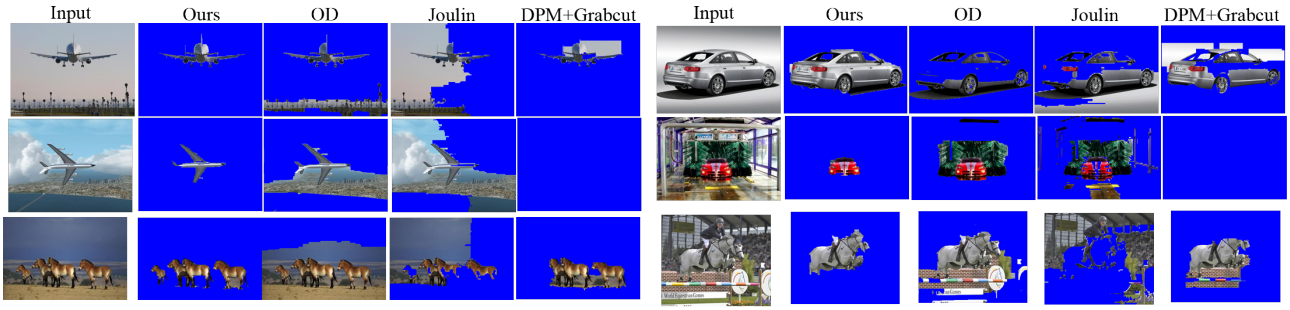

Figure 8. Comparison on the Object Discovery (OD) dataset of our method, OD, Joulin *et al*. [9], and DPM+Grabcut. Note how our method is able to segment non-salient objects while OD picks other areas apart from the object. DPM is unable to detect some objects.

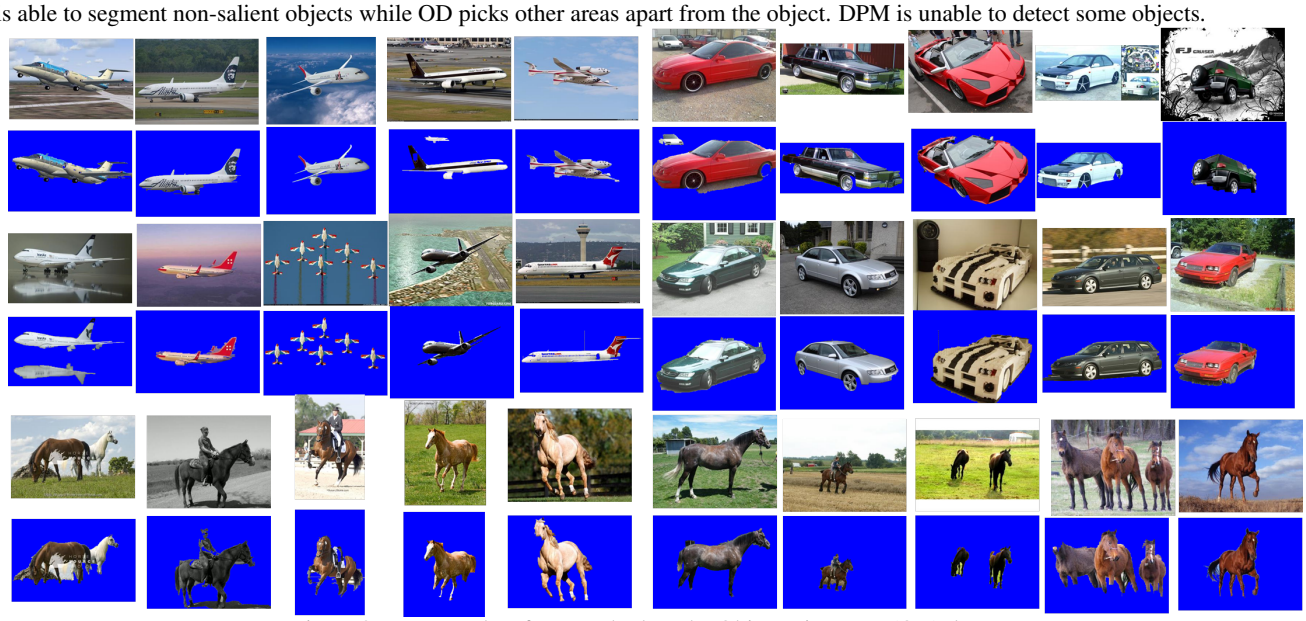

Figure 9. More results of our method on the Object Discovery (OD) dataset.

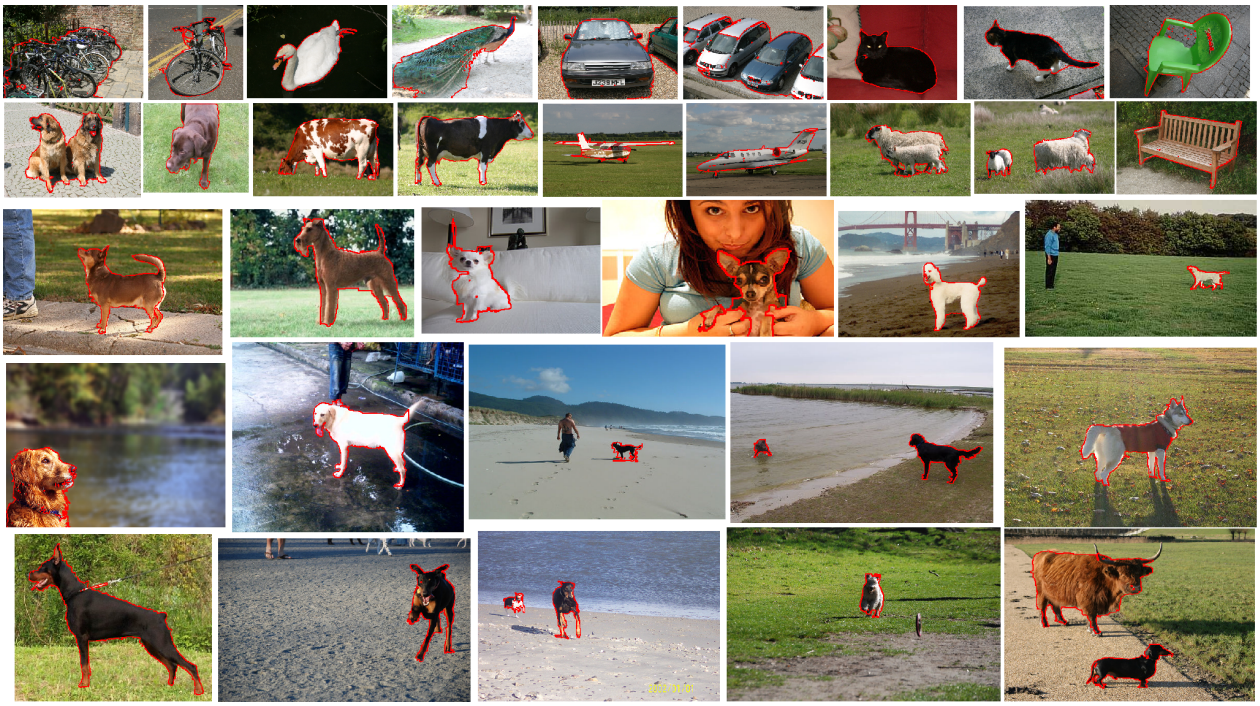

Figure 10. Qualitative Results on MSRC (first two rows) and ImageNet-dog (last three rows).