Enriching Visual Knowledge Bases via Object Discovery and Segmentation

Xinlei Chen, Abhinav Shrivastava, Abhinav Gupta
Carnegie Mellon University

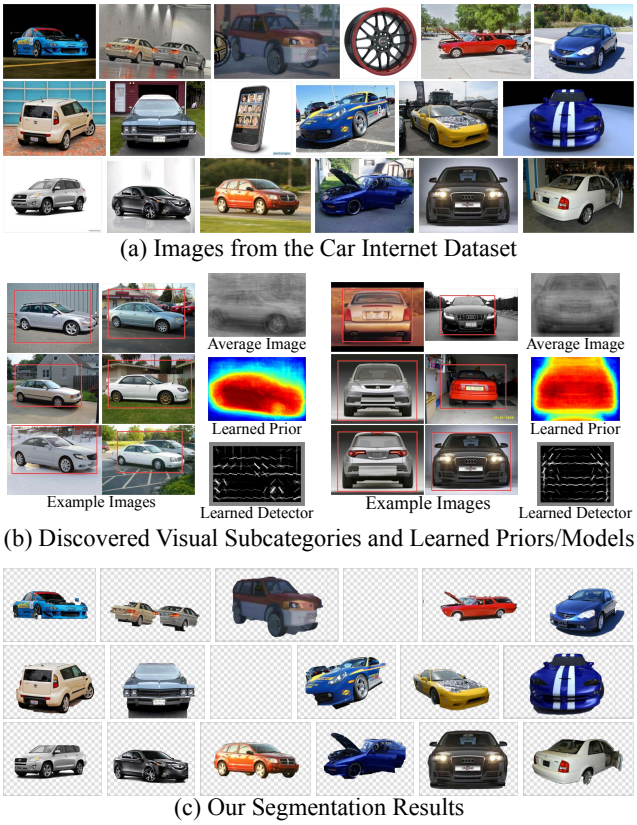
Abstract

There have been some recent efforts to build visual knowledge bases from Internet images. But most of these approaches have focused on bounding box representation of objects. In this paper, we propose to enrich these knowledge bases by automatically discovering objects and their segmentations from noisy Internet images. Specifically, our approach combines the power of generative modeling for segmentation with the effectiveness of discriminative models for detection. The key idea behind our approach is to learn and exploit top-down segmentation priors based on visual subcategories. The strong priors learned from these visual subcategories are then combined with discriminatively trained detectors and bottom up cues to produce clean object segmentations. Our experimental results indicate state-of-the-art performance on the difficult dataset introduced by [29]. We have integrated our algorithm in NEIL for enriching its knowledge base [5]. As of 14^{th} April 2014, NEIL has automatically generated approximately 500K segmentations using web data.

1. Introduction

Object recognition remains one of the most stubborn problems in the field of computer vision. There have been two major directions of research. The first is inspired by Marr's vision [25] and involves using bottom-up cues such as color and contrast to group pixels into segments and then recognize objects. The second and more popular approach is to use sliding windows and solve a binary classification problem of whether an object is present or not. While the segmentation-based approach seems more intuitive and even draws support from psychological theories, the second direction seems to be empirically outperforming the first. Why is that?

We believe that the empirical boost for sliding window based approaches comes from the "magic of data." These approaches have been able to exploit the power of data due to increasing amount of available data in the form of both positive and negative examples. For example, our own effort, NEIL [5] has been automatically labeling data with bounding boxes since July 2013. It has generated approximately 800K bounding box labels in last eight months. On the other hand, segmentation based recognition approaches have still struggled to exploit big data due to the unscalable nature of the required supervision: while it requires only a few seconds to label a bounding-box, hand-labeling a good segmentation is a more labor intensive task. Even crowd sourcing based tools have not been able to generate huge segmentation datasets.



(a) Images from the Car Internet Dataset

(b) Discovered Visual Subcategories and Learned Priors/Models

(c) Our Segmentation Results

Figure 1. We propose an approach to discover objects and perform segmentation in noisy Internet images (a). Our approach builds upon the advances in discriminative object detection to discover visual subcategories and build top-down priors for segmentation (b). These top-down priors, discriminative detectors and bottom-up cues are finally combined to obtain segmentations (c).

In this paper, we focus on generating a large segmentation knowledge base which we believe is also the next step in enriching visual knowledge bases such as NEIL. Given a large collection of noisy Internet images of some object

class (say "car"), our goal is to automatically discover the object instances and their segmentations. There has been some recent work on joint segmentation of multiple images. Most of these approaches focus on using generative models for extracting recurring patterns in images. On the other hand, much of the advancement in the field of object detection has come from learning discriminative models using large quantities of visual data. In this work, we propose a conceptually simple yet powerful approach that combines the power of generative modeling for segmentation with the effectiveness of discriminative models for detection to segment objects from noisy Internet images.

The central idea behind our approach is to learn top-down priors and use these priors to perform joint segmentation. But how do we develop top-down priors? Approaches such as Class-cut [1], Collect-cut [18] and [15] develop top-down priors based on semantic classes: *i.e.,* they build appearance models for semantic classes such as cars, airplanes etc. and use them in a graph-based optimization formulation. But are these semantic classes the right way to develop top-down priors? In recent years, we have learned that the high intra-class variations within a semantic class leads to weak priors and these priors fail to significantly improve performance. On the other hand, clustering the data into visual subcategories [6, 5] followed by learning priors on these visual subcategories has shown a lot of promise. In this paper, we build upon these ideas and use visual subcategories to build top-down segmentation priors and improve joint segmentation of multiple images. We use the advances in learning exemplar based detectors [24, 10] to discover visual subcategories and "align" the instances in these visual subcategories; these visual subcategories are then exploited to build strong top-down priors which are combined with image evidence based likelihoods to perform segmentation on multiple images. Figure 1 shows how our approach can extract aligned visual subcategories and develop strong priors for segmentation. Our experimental results indicate that generating priors via visual subcategories indeed leads to state-of-the-art performance in joint segmentation of an object class on standard datasets [29]. But more importantly, we have integrated our algorithm in NEIL and it has generated approximately 500K segmentations using web data.

2. Related Work

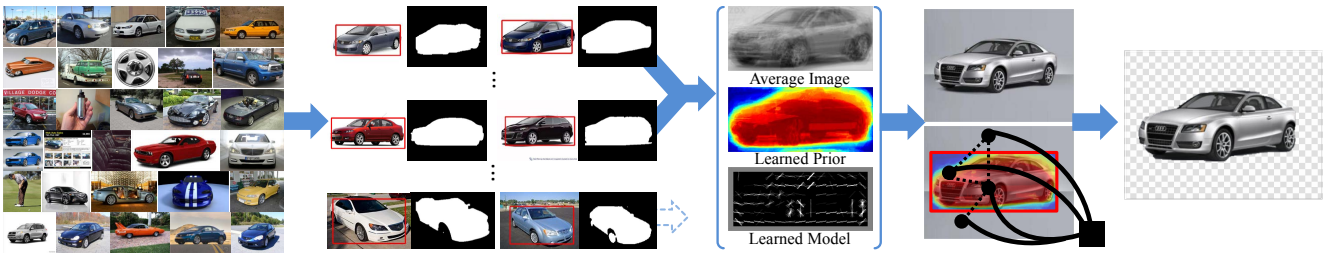
Segmentation is a fundamental problem in computer vision. Early works focused on generating low-level or bottom-up groupings that follow Gestalt laws – the classic pipeline was to use low-level features (such as color, texture, etc. [23]) as input to segmentation or clustering methods [32, 14]. However, for real-world images they fail to produce consistent object segmentation. One of the main reasons for the failure of pure bottom up segmentation is that an object is a complex concept. Generally object segmentation requires combining multiple visually-consistent clusters, which turns out to be too difficult for the vanilla bottom-up segmentation algorithms.

One way to incorporate top-down information is to learn priors in terms of semantic object categories in a fully supervised manner [16, 26, 22]. To reduce the burden of this annotation, semi- and weakly-supervised approaches have been developed. For example, [1] uses image-level object annotation to learn priors. Another popular way to reduce annotation is to use interactive supervision in terms of a few simple scribbles [4, 3, 27, 21]. Finally, approaches have tried using other kind of priors including bounding boxes [20], context [19, 18], saliency [8] and object probability [2, 17, 15]. However, most of these priors are still learned on semantic object classes which often leads to weak priors due to intra-class and pose variations.

In order to learn priors with little or no annotations, recent approaches have also tried using object discovery to extract segments from images automatically, followed by learning of priors (see [35] for an overview). A common approach [34, 30] is to treat the unlabeled images as documents and objects as topics, and use generative topic-model approaches such as Latent Dirichlet Allocation (LDA) and Hierarchical Pitman-Yor (HPY) to learn the distribution and segmentation of multiple classes of objects simultaneously. However, completely unsupervised object discovery and learning of segmentation prior often tends to be non-robust due to the problem being under-constrained.

In this paper, we follow the regime of using web-based supervision to learn segmentation priors [29]. We use query terms to obtain noisy image sets from Internet and then learn models and segmentation priors from these images. However, instead of modeling segmentation priors and constraints based on semantic classes, we model them based on visual subcategories, which are visually homogeneous clusters and have much less intra-class variations.

Our work is also related to co-segmentation, where the task is to simultaneously segment visually similar objects from multiple images at the same time [29, 28, 3, 11, 13, 36]. Most of these approaches assume that all images have very similar objects with distinct backgrounds, and they try to learn a common appearance model to segment these images. However, the biggest drawback with these approaches is that they are either susceptible to noisy data or assume that an object of interest is present in every image of the dataset. The closest work to our approach is the recent paper by [29], which proposes to use a pixel correspondence-based method for object discovery. They model the sparsity and saliency properties of the common object in images, and construct a large-scale graphical model to jointly infer an object mask for each image. Instead of using pairwise similarities, our approach builds upon recent success of discriminative models and exploits visual subcategories. Our discriminative machinery allows us to local-



(a) Car Internet Images (b) Aligned Homogeneous Clusters (c) Visual Subcategories (d) Segmentation

Figure 2. Overview of our approach

ize the object in the scene and the strong segmentation priors help us achieve state-of-the-art performance on the benchmark dataset. Finally, we believe our approach is more scalable than other co-segmentation approaches (including [29]) since we never attempt to solve a global joint segmentation problem, but instead only perform joint segmentation on subsets of the data.

3. Our Approach

Our goal is to extract objects and their segments from large, noisy image collections in an unsupervised manner. We assume that the collection is obtained as a query result from search engines, photo albums, etc. and therefore, a majority of these images contain the object of interest. However, we still want to reject the images which are noisy and do not have the object instance. While one can use approaches like graph-cut with center prior to discover the object segments, such an approach fails due to the weak center prior in case of Internet images. What we need is some top-down information, which can be obtained by jointly segmenting the whole collection. Most approaches build class-based appearance models from the entire image collection to guide the segmentation of individual instances. However, in this work, we argue that due to high intra-class and pose variations such priors are still weak and do not improve the results significantly. Instead, we build priors based on visual subcategories where each subcategory corresponds to a 'visually homogeneous' cluster in the data (low intra-class variations) [6, 5]. For example, for an airplane, some of the visual subcategories could be commercial plane in front view, passenger plane in side view, fighter plane in front view etc. But how does one seed segmentations for these visual subcategories before learning segmentation priors?

In this work, instead of directly discovering disjoint visual subcategories, we first cluster the visual data into overlapping and redundant clusters (an instance can belong to one or more clusters). These overlapping clusters are built using the recent work in training instance based detectors and then using these detectors to find similar instances in the training data [24, 10, 7, 33]. Because we run these detectors in a sliding window manner, our clusters have nicely aligned visual instances. Exploiting the fact that images in these clusters are well aligned, we run a joint co-segmentation algorithm on each cluster by introducing an extra constraint that pixels in the same location should have similar foreground-background labels. Introducing this extra constraint in conjunction with high-quality clusters leads to clean segmentation labels for the images.

Our clusters are tight (low recall, high precision) with very few instances, and therefore some of the clusters are noisy, which capture the repetition in the noisy images. For example, 5 motorbikes in the car collection group together to form a cluster. To clean-up the noisy clusters, we merge these overlapping and redundant clusters to form visual subcategories. The subcategories belonging to the underlying categories find enough repetition in the data that they can be merged together. On the other hand, the noisy clusters fail to cluster together and are dropped. Once we have these large subcategories, we pool in the segmentation results from the previous step to create top-down segmentation priors. We also train a discriminative Latent-SVM detector [9] for each of the cluster. These trained detectors are then used to detect instances of object across all the images. We also generate a segmentation mask for each detection by simply transferring the average segmentation for each visual subcategory. Finally, these transferred masks are used as the top-down prior and a graph-cut algorithm is applied to extract the final segment for each image. The outline of our approach is shown in Figure 2.

3.1. Discovering Aligned and Homogenous Clusters

To build segmentation priors, we first need to initialize and segment a few images in the collection. We propose to discover strongly coherent and visually aligned clusters (high precision, low recall). Once we have visually homogeneous and aligned clusters, we propose to run a co-segmentation approach with strong co-location constraints and obtain seed segments in the dataset. But how do we discover visually coherent and aligned clusters? One naïve approach would be to sample a random set of patches and then cluster these patches using standard k-means. However, in the case of random patches it is extremely unlikely to hit multiple occurrences of the same object in a well-aligned manner unless we sample hundreds of thousands of windows per image. On this scale, clustering approaches tend to give incoherent clusters as shown by recent approaches [7]. Motivated by recent work on discriminative clustering via detection [7, 33, 5], we propose an approach

Figure 3. (Top) Examples of strongly aligned and visually coherent clusters that we discovered. (Bottom) We also show the result of our modified co-segmentation approach on these clusters.

to create coherent, aligned but overlapping and redundant clusters in the data.

Our approach is as follows: we first use each image as a cluster seed and we build clusters by detecting similar patches in the rest of the data. Specifically, we train an exemplar detector [10, 24] (eLDA in our case) based on Color-HOG (CHOG) features [31]. Once we have an eLDA detector for each cropped image, we use this detector to detect similar objects on all the images in the collection and select the top k detections with highest scores. Since CHOG feature focuses on shapes/contours, the resulting clusters are well aligned, which serves as the basis for the following joint segmentation and subcategory discovery step. Note that since we develop a cluster for each image and some images are noisy (do not contain any objects), some of the clusters tend to be noisy as well. Figure 3(top) shows some examples of the well aligned clusters extracted using the above approach.

3.2. Generating Seed Segmentations

The discovered visually coherent and overlapping clusters in the previous step are aligned due to sliding window search, and they are aligned up to the level of a CHOG grid cell. We can use this strong alignment to constrain the co-segmentation problem and jointly segment the foreground in all images, in the same cluster, using a graph-cut based approach. Notice that objects can occur in different environments and have backgrounds with various conditions. The benefits of segmenting all the images at once is that some instances can be more easily segmented out (e.g., product images with clean, uniformly colored background), and those segmentations can help in segmenting the hard images (e.g., images taken with a low-resolution camera, real-world images with multiple objects, overlaps and occlusions).

Mathematically, we cast the problem as a classical graph cut problem to label every pixel in every image patch as foreground or background. Suppose we have n image patches $I_1, I_2, \ldots, I_n$ that belong to one cluster, each pixel-feature $x_{i,p}$ (for the pixel p) should be labeled as either foreground $c_{i,p} = 1$ or background $c_{i,p} = 0$, where p denotes its location in image i. A labeling C of all the pixels corresponds to a segmentation. We define an energy function over pixels and labels, and the optimal labeling is the one with minimum energy.

The energy function E has four terms, leveraging the instance-level cues and cluster-level cues in a coherent way. The first term $E(i, p; A_i)$ is the unary potential from an appearance model specific to image i, and the second term $E(i, p; A_S)$ is the unary potential from an appearance model shared between all images in the cluster. An instance based appearance model A_i consists of two Gaussian mixture models (GMM), one for the foreground (used when $c_{i,p} = 1$) and one for the background (used when $c_{i,p} = 0$). Each component is a full-covariance Gaussian over the RGB color space. We learn the foreground and background appearance models using the pixels inside and outside the bounding box generated from detections during clustering step.

The third term $E(i, p, q; c_{i,p}, c_{i,q})$ is the pairwise potential where we define:

$$E(i, p, q; c_{i,p}, c_{i,q}) = \delta(c_{i,p} \neq c_{i,q}) e^{-\beta \mathrm{P}_E(x_p, x_q)}, \quad (1)$$

as the pairwise compatibility function between labels of pixels (p and q) based on the probability of having an intervening contour (IC) between them [32]. Intuitively, this term penalizes two pixels getting different labels if they do not have an IC between them.

Finally, we want the segmentation masks across the cluster to be aligned and consistent. In our approach, it is modeled as a prior over the pixels: $\mathrm{P}_M(c_p | L_S, p)$ where L_S is the average segmentation mask across the aligned cluster. This denotes the prior probability that each pixel belongs to foreground or background, given the pixel location and the average cluster mask. In terms of energy, the fourth term can be defined as:

$$E(i, p; L_S) = -\log(\mathrm{P}_M(c_p | L_S, p)). \quad (2)$$

Since we do not know the segmentation prior (L_S) and appearance models before segmentation, we iterate between the global optimal graph cut step for each image and re-estimating the model parameters and location prior (by taking the mean) until the algorithm converges. Figure 3(bottom) shows some examples of segmentations obtained for the visually coherent clusters.

3.3. From Clusters to Visual Subcategories

In the last step, we used a standard co-segmentation approach to segment the object of interest in strongly aligned clusters. While one can pool-in results from all such clusters to compute final segmentations, this naive approach will not work because internet data is noisy, especially for images returned by search engines which are still mainly dependent on text-based information retrieval. Therefore,

Figure 4. Examples of visual subcategories obtained after merging clusters. We show few instances, the average images, learned Latent SVM model and the segmentation prior for each subcategory.

some clusters still correspond to noise (e.g., a bike cluster is created from car data). But more importantly, our initial clustering operates in the high-precision, low-recall regime to generate very coherent clusters. In this regime, the clustering is strongly discriminative and focuses on using only part of the data. Therefore, as a next step we create larger clusters which will increase the recall of bounding boxes. To compute the segmentations, we exploit the top-down segmentation priors from the previous step.

Specifically, we merge these aligned clusters and create visual subcategories which are still visually homogeneous but avoid over fitting and allow better recall. This clustering step also helps to get rid of noise in the data as the smaller and less consistent (noisy) clusters find it difficult to group and create visual subcategories. One way to merge clusters would be based on similarity of cluster members. However, in our case, we represent each cluster in terms of the detector and create the signature of the detector based on the

Table 1. Performance Evaluation on the Entire Internet Dataset

	Car		Horse		Airplane	
	P	J	P	J	P	J
[29]	83.38	63.36	83.69	53.89	86.14	55.62
eLDA	85.56	**70.61**	85.86	56.98	85.25	55.31
K-Means	82.11	54.35	87.02	52.99	86.08	51.18
NEIL subcategories	85.49	63.09	82.98	51.49	85.23	50.02
Ours	**87.09**	64.67	**89.00**	**57.58**	**90.24**	**59.97**

detector score on randomly sampled patches. Therefore, we first create a detector-detection matrix $S \in \mathbb{R}^{N \times M}$ (where N is the number of detectors and M is the number of detections), with each entry $S_{i,j}$ filled by the detection score of detector i firing on patch j. Each row i in this matrix can be viewed as a signature of the detector. We then cluster the detectors based on these detection signatures. After normalization, we take the eigenvectors that correspond to the largest eigenvalues of the normalized S and apply k-means clustering to get the cluster index for detectors. Finally, we learn a LSVM detector for each merged cluster.

3.4. Generating Segmentations From Subcategories

In the final step, we bring together the discriminative visual subcategory detectors, the top-down segmentation priors learned for each subcategory and the local image evidence to create final segmentation per image. Given the discovered visual subcategories we learn a LSVM detector without the parts [9] for each subcategory. We use these trained detectors to detect objects throughout the dataset. Finally, we transfer the pooled segmentation mask for each subcategory to initialize the grab-cut algorithm. The result of the grab-cut algorithm is the final segmentation of each instance. The experiments demonstrate that this simple combination is quite powerful and leads to state-of-the-art results on the challenging Internet Dataset [29].

4. Experimental Results

We now present experimental results to demonstrate the effectiveness of our approach on both standard datasets and Internet scale data. Traditional co-segmentation datasets like [3] are too small and clean; however our algorithm is specifically suited for large datasets (1000 images or more per-class). Therefore, we use the new challenging Internet dataset [29] for evaluation. This dataset consists of images automatically downloaded from the Internet with query expansion. It has thousands of noisy images for three categories: airplane, horse, and car, with large variations on pose, scale, view angle, etc. Human labeled segmentation masks are also provided for quantitative evaluation.

Figure 5 shows some qualitative results. Notice how our approach can extract nice segments even from cluttered scenarios such as cars. Also, our approach can separately detect multiple instances of the categories in the same image. The last row in each category shows some failure cases which can be attributed to weird poses and rotations that are not frequent in the dataset.

Table 2. Performance Evaluation on the subset of Internet Dataset (100 images per class)

	Car		Horse		Airplane	
	P	J	P	J	P	J
[11]	58.70	37.15	63.84	30.16	49.25	15.36
[12]	59.20	35.15	64.22	29.53	47.48	11.72
[13]	68.85	0.04	75.12	6.43	80.20	7.90
[29]	85.38	64.42	82.81	**51.65**	88.04	**55.81**
Ours	**87.65**	**64.86**	**86.16**	33.39	**90.25**	40.33

4.1. Quantitative Evaluation

We now quantitatively evaluate the performance of our approach and compare against the algorithm of [29]. Note that most co-segmentation algorithms cannot scale to extremely large datasets and hence we focus on comparing against [29]. For our evaluation metric, we use Precision (P) (the average number of pixels correctly labeled) and Jaccard similarity (J) (average intersection-over-union for the foreground objects). Table 1 shows the result on the entire dataset. Our algorithm substantially outperforms the state-of-the-art algorithm [29] on segmenting Internet images.

To understand the importance of each component, we perform detailed ablative analysis. We use the following one-step clustering baselines: (a) No Merging Step (eLDA): Directly using eLDA results followed by pooling the segmentation; (b) No eLDA Step (K-means): Directly using visual subcategories obtained using K-means; (c) No eLDA Step (NEIL subcategories): Using NEIL based clustering [5] to obtain visual subcategories. Our results indicate that the two-step clustering is the key to obtain high performance in joint segmentation. Finally, we also tried using HOG instead of CHOG and it gave almost similar performance (0.5% fall in P and no fall in J).

Our algorithm hinges upon the large dataset size and therefore, as our final experiment, we want to observe the behavior of our experiment as the amount of data decreases. We would like a graceful degradation in this case. For this we use a subset of 100 images used in [29]. This experiment also allows us to compare against the other co-segmentation approaches. Table 2 summarizes the performance comparison. Our algorithm shows competitive results in terms of precision. This indicates that our algorithm not only works best with a large amount of data, but also degrades gracefully. We also outperform most existing approaches for co-segmentation both in terms of Precision and Jaccard measurement. Finally, we would like to point out that while our approach improves the performance with increasing size of data, [29] shows almost no improvement with dataset size. This suggests that the quantitative performance of our approach is more scalable with respect to the dataset size.

NEIL Integration: We have integrated our object discovery and segmentation algorithm in NEIL [5]. As of 14^{th}

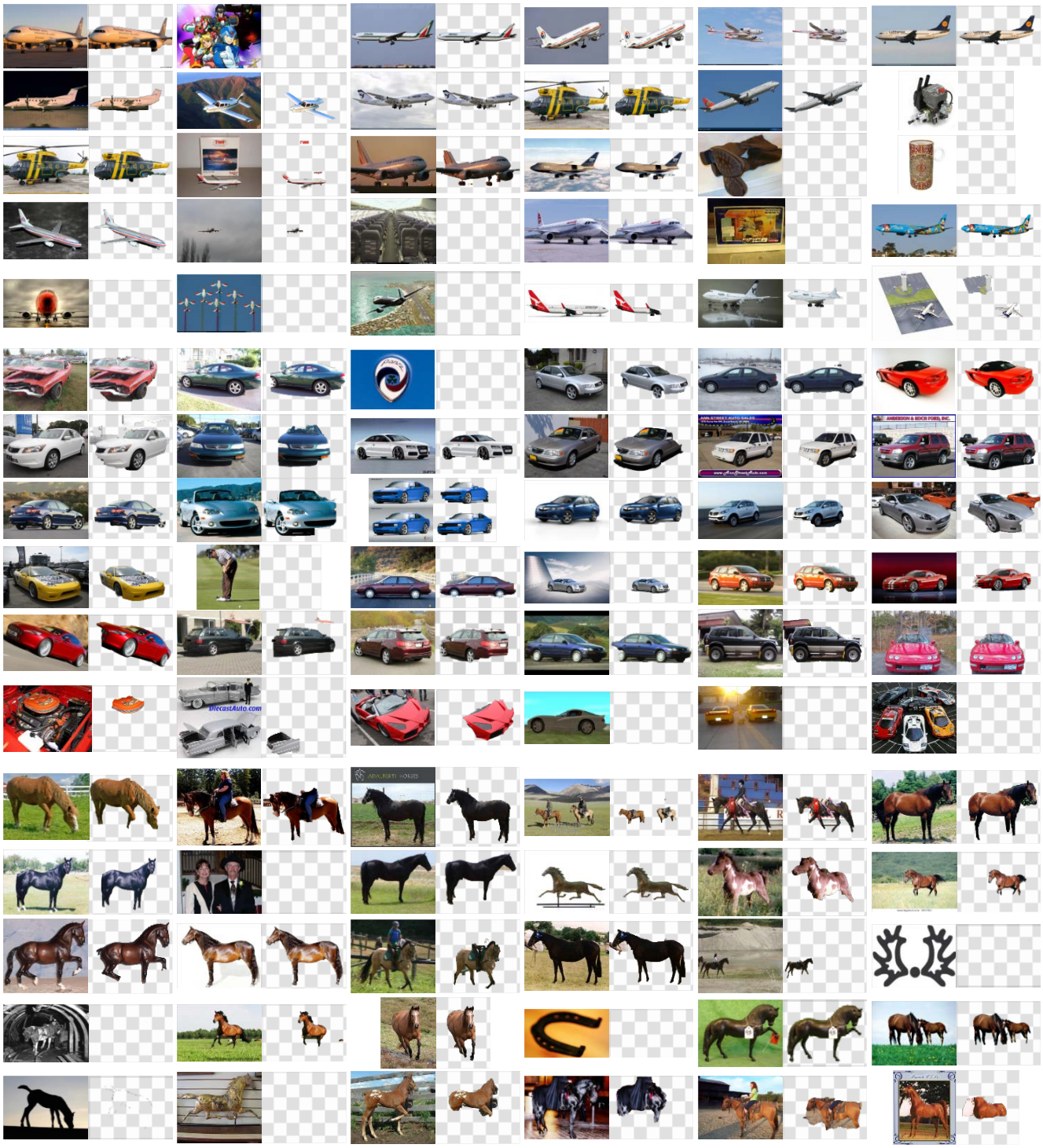


Figure 5. Qualitative results on discovering objects and their segments from noisy Internet images. We show results on three categories: car, horse, and airplane. The last row in each result shows some failure cases.

April 2014, NEIL has automatically generated approximately 500K segmentations using web data. Figure 6 shows some segmentation results from NEIL. All the data and segmentation models are available on the NEIL website (www.neil-kb.com).

Acknowledgments: The authors would like to thank David Fouhey and Ishan Misra for comments and suggestions. This research is supported by supported by ONR MURI N000141010934 and a gift from Google. AG was supported by Bosch Young Faculty Fellowship. The authors would also like to thank Yahoo! for the donation of a computing cluster to the project NEIL.

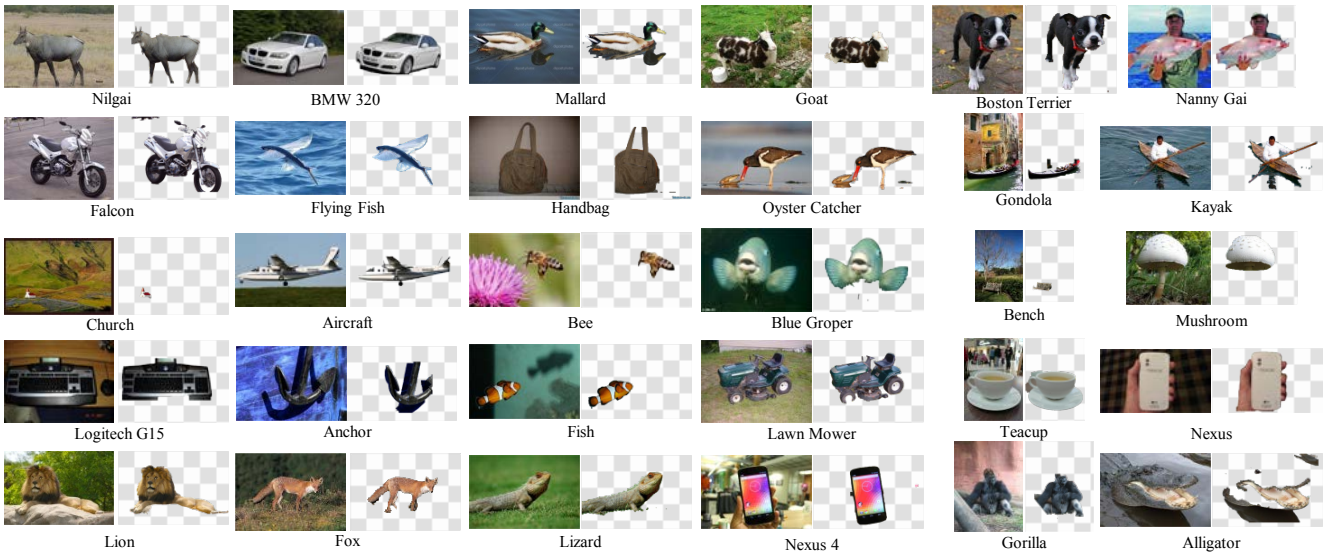


Figure 6. Qualitative results on discovering objects and their segments in NEIL [5]. The last column shows some failure cases.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *ECCV*, 2010. 2

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 2

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 2, 6

[4] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001. 2

[5] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013. 1, 2, 3, 6, 8

[6] S. K. Divvala, A. A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *Parts and Attributes Workshop, ECCV*, 2012. 2, 3

[7] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM TOG (SIGGRAPH)*, 2012. 3

[8] A. Faktor and M. Irani. "clustering by composition" - unsupervised discovery of image categories. In *ECCV*, 2012. 2

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 3, 6

[10] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 2, 3, 4

[11] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2, 6

[12] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 6

[13] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion. In *ICCV*, 2011. 2, 6

[14] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 2004. 2

[15] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012. 2

[16] M. P. Kumar, P. Torr, and A. Zisserman. OBJCUT: Efficient segmentation using top-down and bottom-up cues. *PAMI*, 2010. 2

[17] D. Küttel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012. 2

[18] Y. J. Lee and K. Grauman. Collect-cut: Segmentation with top-down cues discovered in multi-object images. In *CVPR*, 2010. 2

[19] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, 2010. 2

[20] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 2

[21] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM TOG (SIGGRAPH)*, 2004. 2

[22] T. Ma and L. J. Latecki. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *CVPR*, 2013. 2

[23] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001. 2

[24] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 2, 3, 4

[25] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. 1982. 1

[26] B. Packer, S. Gould, and D. Koller. In *ECCV*, 2010. 2

[27] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (SIGGRAPH)*, 2004. 2

[28] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *CVPR*, 2006. 2

[29] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013. 1, 2, 3, 6

[30] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[31] F. Shahbaz Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez. Color attributes for object detection. In *CVPR*, 2012. 4

[32] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 1997. 2, 4

[33] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 3

[34] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 2

[35] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 2010. 2

[36] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 2