SHAPE-BASED WEB IMAGE CLUSTERING FOR UNSUPERVISED OBJECT DETECTION[1]

Wei Zheng[2,3], *Changhu Wang*[1], *and Xilin Chen*[2]
[1]Microsoft Research Asia, Beijing, P.R.China, 100190
[2]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences(CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[3]Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China
{wzheng, xlchen}@jdl.ac.cn, chw@microsoft.com

ABSTRACT

Automatic object detection for an arbitrary class is an important but very challenging problem, due to the countless kinds of objects in the world and the large amount of labeling work for each object. In this work, we target at solving the problem of automatic object detection for an arbitrary class without the laborious human effort. Motivated by the explosive growth of Web images and the phenomenal success of search techniques, we develop an unsupervised object detection framework by automatically training the object detector on the top returns of certain image search engine queried by the name of the object class. In order to automatically isolate the objects from the Web images for training, only clipart images with simple background are used, which keep most of the shape information of the objects. A two-stage shape-based clustering algorithm is proposed to mine typical shapes of the object, in which the inner-class variance of object shapes is considered and undesired images are filtered out. In order to reduce the gap between clipart images and real-world images, we introduce an efficient algorithm to synthesize the real-world images from clipart images, and only shape feature is used in the detector training part. Finally, the synthetic images could be used to train object detectors by an off-the-shelf discriminative algorithm, e.g., boosting and SVM. Extensive experiments show the effectiveness of the proposed framework on objects with simple and representative shapes, and the proposed framework could be considered as a good beginning of solving this challenging problem.

1. INTRODUCTION

Object detection is a basic problem in computer vision. Currently, most of object detection systems are trained based on a huge amount of labeled samples, which are difficult to generalize to an arbitrary class since there are countless objects in the world and the labeling work for each object is very laborious and time-consuming.

In order to reduce human labor, learning algorithms such as online learning [1] [3] and semi-supervised learning algorithms [9] are proposed to use in object detection works. Online learning algorithms try to incrementally utilize the incoming unlabeled samples based on an automatic labeler, while semi-supervised learning algorithms are also well known for leveraging a large amount of unlabeled data as well as a small amount of labeled data to improve classification performance. However, the above works need labeled sets, which still require much human supervision when considering to design detectors for unlimited classes. Unsupervised category learning methods [10] [11] [22] make it possible to collect the training samples without human supervision. Recently, [4] proposed to discover shapes from unlabeled image collections and showed promising results on the Caltech-101 dataset [5]. This kind approach requires the number of the categories (object classes). This kind of work might not work well on the web images since it is hard to specify this number for web images with countless objects. The probabilistic latent semantic analysis was introduced to solve object discovering problem in noisy web images [17] [18]. However, this method mainly focused on a classification task (i.e., object present/absent within image), making little attempt to localize the object.

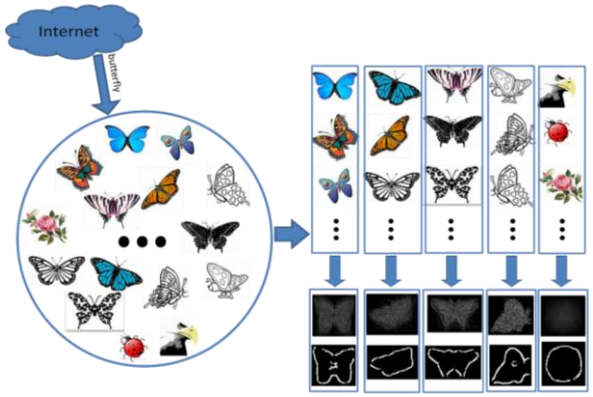


Fig.1. Mining typical shapes of the objects from Internet. The top returned clipart images of certain image search engine are clustered based on shape matching. In the right bottom of the figure, the average edge map in each cluster and its major curve are shown.

[1]This work was performed at Microsoft Research Asia.

In this work, we study the problem of automatic object detection for an arbitrary class with little human supervision. Motivated by the explosive growth of Web images and the success of search engines, our objective is to use the top results of an image search engine queried by the object name as a noisy database related to the object, based on which the object detector is hoped to be obtained. However, it is still very challenging to convert this noisy collection to the final object detector due to the following three issues. First, Web images usually contain cluttered background and it is difficult to isolate the object from the background without human supervision. Second, the objects themselves have various appearances (see Fig.1 for example) due to different factors, such as shape differences, view-point changes, rotation variance, and translation variance. It is difficult to train an effective detector without considering the inner-class variance. Third, the top results of mainstream image search engines usually contain some irrelevant images and thus could be considered as a very noisy training set.

In this work, we circumvent this first problem by leveraging the clipart images from Internet. The clipart images preserve the shape "essence" of the object class to a large extent and also have simple background. It is easy to isolate the objects from the background for the clipart images. We propose a two-stage clustering algorithm to handle the second and third issues. In the first stage, similar shapes of the corresponding object are grouped together based on a basic shape matching technique, i.e., chamfer matching, and then the undesired images are removed. In this stage, due to rotation, translation and scale variance, some similar shapes might be grouped into different clusters. Thus we propose to use an invariant chamfer matching approach in the second-stage clustering to handle these variances. In order to reduce the gap between clipart images and real-world images, we introduce an efficient algorithm to synthesize the real-world images from clipart images, and only shape feature is used in the detector training part. Finally, a well-known boosting framework [23] is used to train object detectors. The experiments on public datasets have shown promising results on the object detection tasks.

2. PROPROSED UNSUPERVISED OBJECT DETECTION FRAMEWORK

In this section, we introduce the proposed unsupervised object detection framework. Using the top results retrieved from Google clipart image search engine queried by the name of the target object. As shown in Fig.2, first, we extract the major contours of clipart images. Then, a two-stage shape-based clustering approach is introduced to mine typical shapes of the target object and remove noisy images. Finally, an efficient algorithm is used to synthesize the real-world images from clipart images, based on which the object detector could be trained.

2.1. Shape extraction and matching

Shape is one of the most important cues for object recognition and the clipart images preserve the shape "essence" of the object class to a large extent. First, we generate the edge maps of the clipart images using canny edge detector [19]. Since objects usually have large and continuous contours, we then crop the main objects by finding the major contours in the edge maps. The continuous contours can be found using the well-known flood fill algorithm. The cropped parts are then normalized to be a size with maximal side being 100 pixels.

Then, we measure the similarity (distance) among the shapes of the target object for the further shape-based clustering. The chamfer distance [20] is widely used for measuring the similarity of two contours. Given the edge maps p and q, the basic chamfer distance is given by

$$d^A(p,q) = \frac{1}{\|p\|}\sum_{e_i \in p} \min_{e_j \in q} \| e_i - e_j \|_2 , \qquad (1)$$

where e_i is the i^{th} edgel (edge pixel) in p and $\|p\|$ is the number of edgels in p.

The time complexity of Eqn. 1 is $O(\|p\|\times\|q\|)$. Distance transform, which can be calculated in linear time [14], can reduce this complexity of Eqn. 1 to $O(\|p\|)$. [12] proposed to extend the basic chamfer matching by dividing the edge map into multiple sub-maps according to the orientations of edgels and sum the chamfer distances of all the sub-maps as the final score. The oriented chamfer matching is represented by

$$d^o(p,q) = \frac{1}{n}\sum\nolimits_{o\in(1,2,\ldots,n)} d^A(p^o,q^o), \qquad (2)$$

where p^o is the o^{th} sub -map of p.

Since the basic chamfer matching and oriented chamfer matching are sensitive to rotation, position and scale, we introduce an invariant chamfer distance, which is invariant to rotation, position and scale:

$$d^I(p,q) = \min_{t_x\in X, t_y\in Y, s\in S, r\in R} d^o(p(t_x,t_y,s,r),q) \qquad (3)$$

where $p(t_x,t_y,s,r)$ is a new edge map, and can be obtained by translating p by (t_x,t_y), then resizing to scale s and finally rotating with angle of r. In the invariant chamfer distance, q is fixed and we change p to find the best match between these two contours. Obviously, the invariant chamfer distance is more expensive than the oriented chamfer matching in terms of computation complexity and not very practical to handle a large number of samples. Therefore, we only use the invariant chamfer matching in the second stage of the clustering while use the oriented chamfer matching in the first stage. It should be noted that the above chamfer distances are asymmetric. It is easy to obtain the symmetric chamfer distance as follows:

$$d^{S(o)}(p,q) = (d^o(p,q) + d^o(p,q))/2, \qquad (4.1)$$

$$d^{S(I)}(p,q) = (d^I(p,q) + d^I(p,q))/2. \qquad (4.2)$$

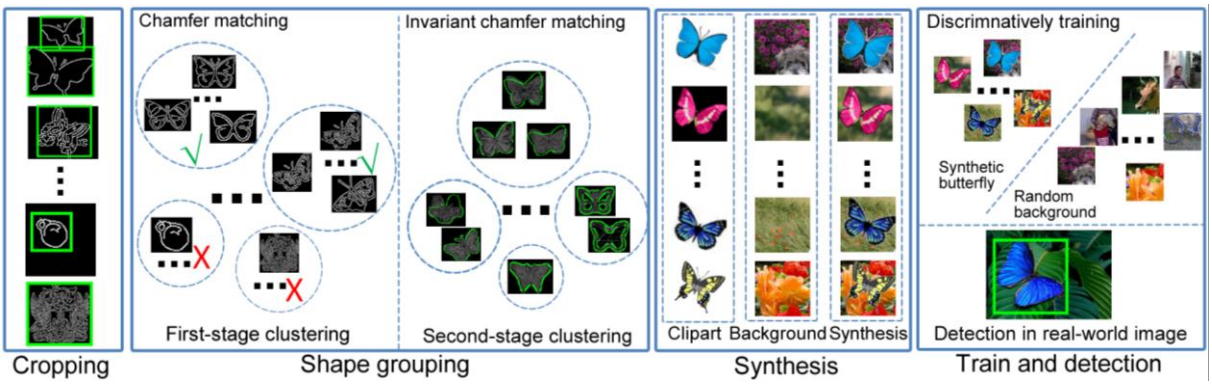


Fig.2. The framework of the unsupervised object detection. First, we extract the edges from clipart images and crop the objects by finding large and consistent edges. Then, a two-stage shape-based clustering algorithm is proposed to mine the typical shapes of the target object. In the first stage, we group the objects based on oriented chamfer matching [12] and remove the noisy images. In the second stage, we group the clusters again based on invariant chamfer matching. In the synthesis phase, we isolate the objects from clipart images on pixel level and overlay the objects on the background image. Finally, the synthetic images are used to train the object detectors.

2.2. Two-stage shape-based clustering

Lots of clustering algorithms can be used for grouping the shapes of a target object, e.g., K-Means, spectral clustering, and hierarchical clustering. In practice, it is not easy to tell the typical shape number of an object class before clustering. Thus, in this work we adopt the complete linkage clustering algorithm [21], which does not need this number in advance. The complete linkage clustering is a kind of the well-known agglomerative hierarchical clustering algorithms, which regards each sample as an individual cluster at the beginning and gives the hierarchical clustering result by progressively merging clusters. The distance between two clusters C_1 and C_2 is measured as

$$D(C_1, C_2) = \max_{x_i \in C_1, x_j \in C_2} d(x_i, x_j), \quad (5)$$

where x_i, x_j are the samples in cluster C_1, C_2 and $d(,)$ is a certain distance measure.

The clustering algorithm progressively merges two clusters with the minimum distance (Eqn. 5) until there remains only one cluster. In practice, we terminate the clustering process if the minimum distance is larger than a threshold $\boldsymbol{\tau}$.

In the first stage, the shape distance is calculated by Eqn. 4.1. The clusters with more than N samples will be reserved, while others will be considered as noisy images or images with non-typical shapes. For example, in the "butterfly" collections as shown in Fig. 1, the cluster containing ladybugs and eagles will be removed since it is an outlier cluster and thus its size would be small. From the average edge maps of the clusters in Fig.1, we can see that there are common silhouettes and repeatable inner edges, namely major curves (right bottom of Fig.1). The major curves can be obtained by extracting edges from the average edge maps.

Similar shapes with different rotations, scales and positions might be grouped into different clusters in the first stage. In the second stage, we need to further merge these clusters together. For each sample, we calculate the total distance with other samples in the same cluster and choose a representative shape that has the minimum distance for each cluster. In the second stage, the complete linkage clustering is also used to group the clusters (representative shapes) by matching the major curves of the average edge maps based on invariant chamfer distance in Eqn. 4.2.

In the final cluster after the second stage, all shapes will be aligned with the representative shape in this cluster. For example, for a shape M_i, it will be aligned to be $M_i(t_x^*, t_y^*, s^*, r^*)$, and the alignment parameters is represented by

$$(t_x^*, t_y^*, s^*, r^*) = \min_{t_x \in X, t_y \in Y, s \in S, r \in R} d^o(M_i(t_x, t_y, s, r), M_i^{rep}), \quad (6)$$

where M_i^{rep} is the representative shape of the cluster containing M_i. As shown in Fig.2, some butterflies with similar shapes may be grouped into different clusters in the first stage due to rotation variance. However, these butterflies will be grouped into one cluster and aligned to the same angle in the second stage.

2.3. From clipart images to real-world images

In order to reduce the gap between clipart images and real-world images, we adopt a heuristic and efficient algorithm to synthesize the real-world image from clipart images. We observe that there are two main differences between clipart images and real-world images. First, most of the clipart images have very simple background. Second, some of the clipart images are cartoons or drawings, and it means that the shape and texture may be different from those of real-world images. Some examples are shown in Fig.3.

Overlaying the objects cropped from clipart image set on the real-world background is able to reduce the "background gap". To achieve this target, we first crop the object from clipart background on pixel level. As shown in Fig.3, the background of the clipart images is simple and usually contains only one color. Therefore, we can use the color information to distinguish the foreground and background. However, the foreground may have similar color with the background. As shown in Fig.4, the middle of the car has similar color with the white background, and this region will be considered as background only using color information.

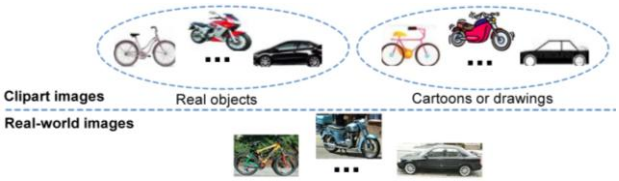


Fig.3. Clipart images vs. real-world images.

Thus it is unreliable to segment the objects only based on color information. Therefore, we propose a segmentation algorithm by leveraging both color and shape information for clipart images. Since most of the objects locate in the center of the cropped image patch, we use the median color (r_m,g_m,b_m) of the four corner regions (see Fig.4) to estimate the background color. The initial background is obtain by

$$\{(r,g,b) \mid r-r_m<T, g-g_m<T, b-b_m<T\}, \quad (7)$$

where T is a threshold. We can obtain the shape prior is by averaging all the initial segmentation results. As shown in Fig.4, the shape prior is represented as a gray level image. Each point of the shape prior can be mapped to a probability of being foreground (shape evidence p_s). Obviously, a larger p_s means more confident to be foreground. We assume that the background color is in Gaussian distribution for each individual color channel. For each pixel, we can calculate the probability of being foreground (color evidence p_c) from the color value. For a pixel, the final probability of being foreground is given by

$$p = \arg\max(p_s, p_c). \quad (8)$$

Intuitively, the probability of being a foreground is determined by the more confident evidence, either p_s or p_c. A pixel is considered as foreground if p is larger than 0.5. Thus, we can crop the object from background on pixel level using the above method, and then overlay the objects on the background images, which can be obtained from Internet.

2.4. Discriminative training for object detection

We use the synthetic images as positive training samples and collect negative samples from the Web images of other objects. The positive and negative samples are normalized to a standard size, e.g., 100×40 for side-view car. We can use the average size of the synthetic images as this standard size. Then, many off-the-shelf classifiers can be applied on these training samples, such as boosting [23] and SVM [8].

We adopt the Histogram of Oriented Gradient (HOG) based boosting framework [16] to train object detectors. HOG reflects the statistical information of the object contour, whose effectiveness and efficiency has been shown in object detection tasks. Using HOG feature could somewhat reduce the gap between clipart images and real-world images, since the clipart images may preserve the "essence" of contour information. Please refer to [16] for details.

3. EXPERIMENTS

We first provide the shape clustering results for the 101 objects in the Caltech-101 dataset [5], followed by object detection experiments on UIUC car dataset [7] and VOC2006 dataset [6].

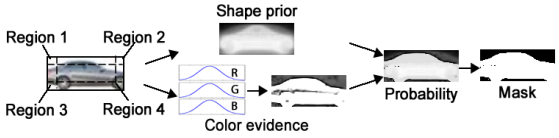


Fig.4. Clipart images segmentation.

3.1. Shape clustering results for Caltech-101 objects

For each object category of Caltech-101 dataset, we use the object name as the query to retrieve the top 1,000 clipart images from Google clipart image search engine. The two-stage shape clustering algorithm is used to cluster the major contours of the top 1000 images for each object. The average edge maps of the biggest clusters of all 101 objects are shown in Fig.5. From Fig.5 we can see that, the clustering algorithm achieves good results for the objects with simple and typical shape structures, such as anchor, bass, butterfly and so on. However, there are also some objects that we cannot see clear shapes from the average edge maps, such as brain, mayfly and airplane side. The undesired results might be caused by the following three reasons. First, the object may not have representative shapes and it is not suitable to describe the objects using contours, such as brain and ant. Second, the image search results of some objects may be unsatisfactory, and thus it is not easy to obtain the representative shapes for the corresponding objects. For example, the retrieval results of cougar body are very noisy and contain many humans and cars, and thus our approach fails to find the typical shapes. Third, some keywords used to retrieve images may be somewhat ambiguous. For example, the first cluster of ibis shows the shape of bicycle, since IBIS is a well-known brand of mountain bicycles.

For a detailed analysis, we select 12 classes and show the largest four clusters in Fig.6. From Fig.6 we can see that, the clustering algorithm can find several representative shapes of the objects but cannot guarantee that the clusters are all meaningful. The major reason may be that the retrieved images by the search engine sometimes contain too many undesired objects, and it is not easy to remove all noisy images in the first-stage clustering. However, we hope the clustering results could become better once the retrieval quality of image search engines is improved.

3.2. Detection experiments on UIUC car dataset

UIUC car dataset consists of two sub-datasets, i.e., single scale dataset containing 200 side-view cars and multi-scale dataset containing 139 side-view cars. We adopt the Equal-Precision-Recall (EPR) rate to evaluate the performance. Six unsupervised methods are compared, namely *Proposed*, *ProposedNoSyn*, *Top300Image*, *Top300Clipart*, *Top1000Image*, and *Top1000Clipart*. *Proposed* is the proposed method. The only difference between *ProposedNoSyn* and *Proposed* is that there is no synthetic stage in *ProposedNoSyn*, i.e., the clipart images are directly

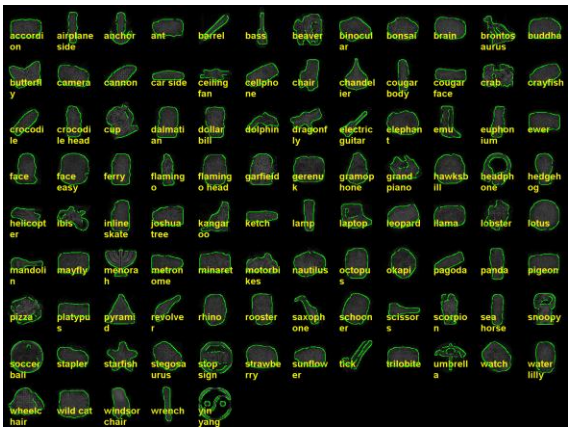


Fig.5. Average edge maps and major contours of the biggest clusters of the 101 objects in Caltech-101 dataset.

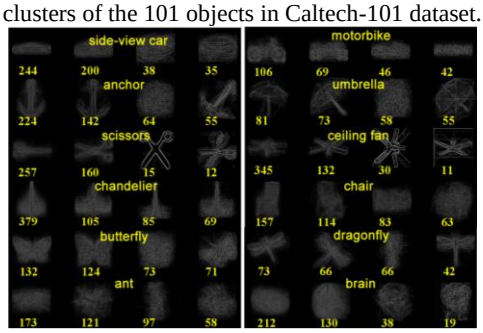


Fig.6. Average edge maps of the largest four clusters.

used for training the classifier in *ProposedNoSyn*, which is used to test the usefulness of the synthetic stage. For the other four methods, the synthetic images for training are replaced by different strategies, in which the effectiveness of the proposed framework, the usefulness of the typical shape mining algorithm, and the rationality of the use of clipart images are evaluated. In *Top300Image*, we use the top 300 images returned by Google image search engine to replace the synthetic images. In *Top300Clipart*, we adopt the top 300 clipart images returned by Google clipart image search engine. The top 1000 images are also tried in *Top1000Image and Top1000Clipart.*

The results of the above six unsupervised methods are listed in Table 1. We can see that *Top300Image* and *Top1000Image* produced quite bad performance. It shows that we cannot directly use the top results of Google image search engine (real-image engine) as the positive training set to automatically train object detectors without human supervision. This is because that there are too many noisy images in the retrieval results and the objects in the natural images are not well-aligned. As a result, the training algorithm cannot learn useful information from the noisy image collection. But by leveraging clipart images, we can find the objects and crop them from the images. Comparing to the top returned natural images, the cropped clipart images are well aligned in scale. That is why *Top300Clipart* and *Top1000Clipart* achieved better performance than *Top300Image* and *Top1000Image*. Moreover, the significant

Table 1. Performance comparison on UIUC car dataset.

Approach	Single-scale	Multi-scale
Top300Image (unsupervised)	9.0%	3.7%
Top300Clipart (unsupervised)	46.5%	30.4%
Top1000Image (unsupervised)	22.0%	3.2%
Top1000Clipart (unsupervised)	50.0%	32.0%
ProposedNoSyn (unsupervised)	77.5%	82.5%
Proposed (unsupervised)	92.5%	92.1%
CascadeHOGClipart [16](supervised)	93.0%	92.4%
CascadeHOGUIUC [16] (supervised)	96.6%	96.0%

improvement of *ProposedNoSyn* over *Top1000Clipart* shows the effectiveness of the two-stage clustering algorithm. Finally, *Proposed* still outperforms *ProposedNoSyn* a lot, which shows the importance of the synthetic stage.

Besides the unsupervised approaches, we also provide the experimental results of two supervised approaches, i.e., *CascadeHOGClipart* and *CascadeHOGUIUC*. In *CascadeHOGClipart*, we manually collect training samples from Google Clipart images and then synthesize real-world images in the same way. In *CascadeHOGUIUC*, we use the training samples provided by UIUC dataset to train the object detector. The object detector training approaches of the two methods are the same as that of *Proposed*. The results are also shown in Table 1. We can see that, the supervised manner in clipart dataset (*CascadeHOGClipart*) only brings a little improvement compared with our unsupervised framework (*Proposed*), and the manually collected real-world training samples (*CascadeHOGUIUC)* also do not bring much improvement (4%), which shows the effectiveness of the proposed unsupervised method.

3.3. Detection experiments on VOC2006 dataset

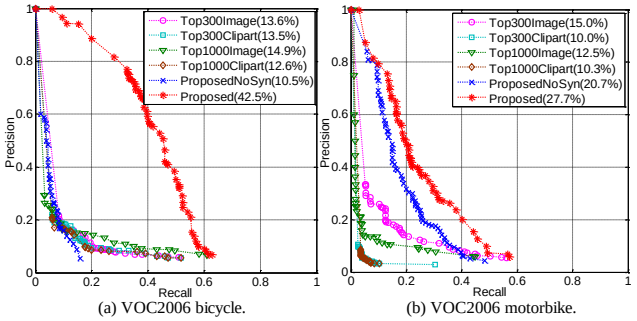
VOC2006 dataset consists of 2686 real-world images with cluttered background, with 326 bicycles and 274 motorbikes. We adopt the Average Precision (AP) measure as in [6], and the precision-recall curves for unsupervised algorithms are also provided. The comparison results of six unsupervised algorithms are shown in Fig.7 and Table 2, from which we can draw similar conclusions as Section 4.2.

For comparing with supervised methods, *Proposed* has similar performance as *CascadeHOGClipart*, which shows the effectiveness of the unsupervised framework. However, the performance gap between *Proposed* and *CascadeHOGClipart* (10%) is larger than that in UIUC dataset (4%). One reason of the accuracy loss is that the VOC2006 dataset is a challenging dataset and many objects of this dataset are partially occluded. However, most of the clipart training samples are not occluded and the final learned detector may fail to detect the occluded objects. The part-based object detection approaches might improve the accuracy on this dataset. For both of UIUC and VOC2006 datasets, online-learning [1] or transfer learning [15] may help to reduce the "background gap" between clipart images and real-world images and further improve the accuracy.

Besides the side-view cars, bicycles, motorbikes, we also trained object detectors for other object classes, such as

Table 2. Performance comparison on VOC2006 dataset.

Approach	Bicycle	Motorbike
Top300Image(unsupervised)	13.6%	15.0%
Top300Clipart(unsupervised)	13.5%	10.0%
Top1000Image(unsupervised)	14.9%	12.5%
Top1000Clipart(unsupervised)	12.6%	10.3%
ProposedNoSyn (unsupervised)	10.5%	20.7%
Proposed (Unsupervised)	42.5%	27.7%
CascadeHOGClipart [16] (Supervised)	43.0%	28.5%
CascadeHOGVOC [16] (Supervised)	52.7%	35.6%



(a) VOC2006 bicycle. (b) VOC2006 motorbike.

Fig.7. Precision vs. recall curves on VOC2006 dataset.

butterflies, anchors, and so on. Some detection results of these objects are shown in Fig.8.

4. CONCLUSIONS

We proposed a novel framework for unsupervised object detection by leveraging web images. To automatically find and isolate objects from Web images, only clipart images were used. A two-stage clustering algorithm was proposed to group the objects by shape matching and remove noisy images. Our approach could automatically mine the typical shapes of the target object, and save a lot the manually labeling work. We also analyzed the gap between clipart images and real-world images and proposed an image synthetic approach for reducing this gap. Experiments have shown the effectiveness of proposed framework on objects with simple and representative shapes.

Actually, automatically generating object detector for an arbitrary object class is a very challenging problem. In this work, we studied the problem and only gave some primary attempts. Some approaches in this paper may be not optimal and better approaches are worthy to be further studied.

5. REFERENCES

[1] B. Wu and R. Nevatia. Improving part based object detection by unsupervised online boosting. *CVPR*, '07.

[2] V. Nair and J. Clark. An Unsupervised, Online learning framework for moving object detection. *CVPR*, '04.

[3] M. Pham and T. Cham. Online learning asymmetric boosted classifiers for object detection. *CVPR*, '07.

[4] Y. Lee and K. Grauman. Shape discovery from unlabeled image collections. *CVPR*,'09.

[5] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *CVPR*,'04.

[6] M. Everingham and A. Zisserman. The Pascal Visual Object Classes Challenge 2006 (VOC2006) Results.

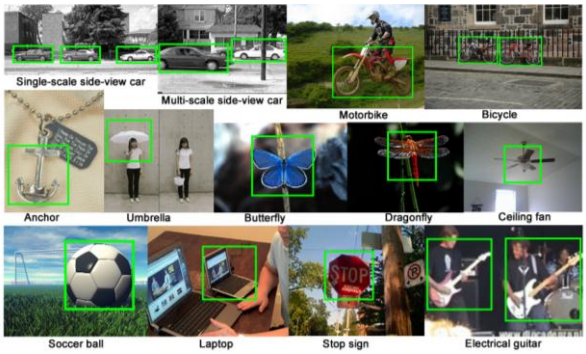


Fig.8. Example results of detection for different objects.

[7] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 2004.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, '05.

[9] W. Wu and J. Yang. Semi-supervised learning of object categories from paired local features. *CIVR*, '08.

[10] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis Techniques. *CVPR*, '08.

[11] D. Liu and T. Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. *ICCV*, '07.

[12] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 2006.

[13] L.Chen, R. S. Feris, and M. Turk. Efficient Partial Shape Matching Using Smith Waterman Algorithm. Workshop on NORDIA'08. *CVPR*,'08.

[14] P.F. Felzenszwalb and D.P. Huttenlocher. Distance transforms of sampled functions. Technical report, 2004.

[15] G. Heitz, G. Elidan, and D. Koller. Transfer learning of object classes: from cartoons to photographs. *NIPS*, '05.

[16] Q. Zhu, S. Avidan, M. Yeh and K. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *CVPR*, '06.

[17] L. Fei-Fei, R. Fergus and P. Perona. and A. Zisserman. Learning object categories from Internet image searches. IEEE Special Issue on Internet Vision, 2010.

[18] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. *ICCV*, '05.

[19] J. Canny. A computational approach to edge detection. *PAMI*, 1986.

[20] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching. In *IJCAI*.

[21] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. The Computer Journal.

[22] R. Fergus, P. Perona, and A.Zisserman. Object class recognition by unsupervised scale-Invariant learning. *CVPR*, '03.

[23] P. Viola, M.J. Jones. Robust Real-Time Face Detection. *IJCV*, '04.

[24] L. Chen, J. J. McAuley, R. S. Feris, T. S. Caetano, and M. Turk. Shape Classification Through Structured Learning of Matching Measures. *CVPR*,'09.