Well Begun Is Half Done: Generating High-Quality Seeds for Automatic Image Dataset Construction from Web

Yan Xia[1], Xudong Cao[2], Fang Wen[2], and Jian Sun[2]

[1] University of Science and Technology of China
[2] Microsoft Research Asia, Beijing, China

Abstract. We present a fully automatic approach to construct a large-scale, high-precision dataset from noisy web images. Within the entire pipeline, we focus on generating high quality seed images for subsequent dataset growing. High quality seeds are essential as we revealed, but they have received relatively less attention in previous works with respect to how to automatically generate them. In this work, we propose a density score based on rank-order distance to identify positive seed images. The basic idea is images relevant to a concept typically are tightly clustered, while the outliers are widely scattered. Through adaptive thresholding, we guarantee the selected seeds as numerous and accurate as possible. Starting with the high quality seeds, we grow a high quality dataset by dividing seeds and conducting iterative negative and positive mining. Our system can automatically collect thousands of images for one concept/class, with a precision rate of 95% or more. Comparisons with recent state-of-the-arts also demonstrate our method's superior performance.

1 Introduction

High quality datasets (e.g. LabelMe [15] and ImageNet [7]) play vital roles in the tasks of computer vision and push relevant researches forward [12]. However extensive human effort is required in order to label tens of millions of images, and existing datasets can not cover all tasks or user specific classes such as "wearing glasses". In this context, automatic dataset construction has emerged: given noisy images crawled from search engines, the images which belong to the query concept/class such as "waterfall", are labeled as positive samples (the rest are negative outliers/distractors) in un- or semi-supervised way.

The pipeline of existing methods [10,16,14,13,5,3] for automatic dataset construction can be summarized into two steps: generating a group of labeled images i.e. seed images or seeds; and growing the dataset from the starting seeds by iterative "self training". But previous works mainly focus on the second step, paying relative less attention to the first step – the seeds are either labeled by human [10,14,13,5] (high cost) or labeled according to top-ranked images [14,13] or noisy surrounding text [1,16] (low quality).

In this work, we study the impacts of the first step, i.e., generating seeds, on automatic dataset construction. We find that high quality seeds are critical to achieving

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 387–400, 2014.

high recall and precision for the constructed dataset (see Section 2). With this mind, we propose a novel and fully automatic approach to generate high quality seeds. The basic observation/assumption underline of our approach is that the positive samples are densely clustered while the negative outliers are widely scattered. In this context, we can identify the images with high density as positive seeds.

To realize this basic idea, we need to handle the heterogeneous problem: the L2 distance is incomparable across different concepts. For example, the average distance between "human" images is usually larger than the average distance between "waterfall" images. We handle this problem by transforming the L2 distance to rank-order distance [19], which is more robust to the heterogeneous problem. Based on the rank-order distance, we define the density of an image as the number of neighbors within a certain distance, and choose images with high densities above a certain threshold as the seeds. For various concepts, we adaptively set the thresholds, because the proportions of the positive samples are concept-dependent. The threshold is automatically determined by balancing the following considerations: the average density of selected seeds should be high, while the selected seeds should be similar to each other but dissimilar from the outliers.

We grow the dataset from the seeds using iterative negative mining [6,9] and positive mining [14]. We develop seed dividing as a preprocessing step in the creation of the dataset to alleviate the multi-modal problem in negative and positive mining. We exploit k-means to divide the seeds into multiple groups and train separated classifiers on each of them. Substantial improvements are observed in the multi-modal case.

We apply the features obtained by deep learning [12,8,18] in the task of automatic dataset construction. As deep learning features are very capable of capturing the semantic meaning of images, we expect substantial improvement and revisit the task of automatic dataset construction.

2 Seeds Quality Matters

Intuitively, the larger the amount and the higher the precision of seeds, the better the quality of the constructed dataset. We measure the seeds' quality using seeds ratio and seeds precision. *Seeds ratio* measures the amount of seeds relative to the total number of the crawled images. *Seeds precision* measures the percentage of the true positive relative to the total number of selected seeds.

To quantitatively study the dataset quality as a function of seeds quality, we artificially build a dataset of "crawled" images: the positive samples are from one class of ImageNet and the negative outliers are randomly crawled web images. In this way, we quantitatively vary the seeds ratio and seeds precision, and study their influences on the quality of the constructed dataset.

We find high seeds precision is critical to dataset quality (Fig. 1(a)). For example, the recall and precision are 91% and 95% given a high seeds precision of 100%, but the recall and precision drop dramatically to 74% and 91% when the seeds precision decreases to 80%. We also find that a certain seeds ratio (e.g. 20%) is necessary for good recall (Fig. 1(b)).

In this context, we need adequate and accurate seeds to benefit the final constructed dataset. We propose our approach with this factor under consideration.

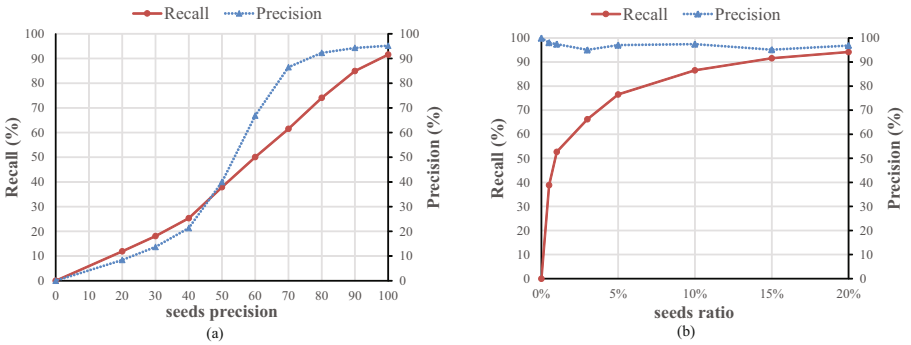

Fig. 1. The impact of seeds quality on the constructed dataset. (a) seeds precision vs. dataset quality. (b) seeds ratio vs. dataset quality. In (a), seeds ratio is fixed to 15%. In (b), seeds precision is fixed to 100%. One concept is taken as a representative example in this figure. Similar patterns are also observed in other concepts.

3 Our Approach

To automatically construct a dataset, we firstly crawl large-scale noisy images from web, and generate clean seed images from the crawled images. Then we grow the dataset starting from the seeds. We will detail these three steps in this section.

3.1 Image Crawling

Herein we describe how we construct the initial set containing tens of thousands images. Since Google and Bing can only return 1000 results at most for one query, which is far from what we need, we enlarge the number of crawled images in two ways: (i) We exploit searching filters provided by search engines. For example, we vary the image layout (square, wide or tall) and size (small, medium or large), and combine these filters to crawl more images. The top 300 results are downloaded in each combination. Then about 2000–4000 images are crawled for one text query. (ii) We utilize automatic query expansion [2]. For example, given a target concept such as "waterfall", we expand 10–20 relevant queries such as "cascading waterfall", "waterfall in Europe" and so on. Then we use the expanded queries to crawl more images.

After downloading, we remove small images (width or height $<$ 160 px) and duplicates. The remaining images form the initial set for subsequent seeds generating and dataset growing.

3.2 Seeds Generating

We exploit the topological structure of crawled images to generate seeds un-supervi sedly. According to the topological structure, most positive images are densely clustered while the outliers are widely scattered. In other words, images in denser areas are more likely to be seeds. Therefore we can rank the crawled images by a certain density

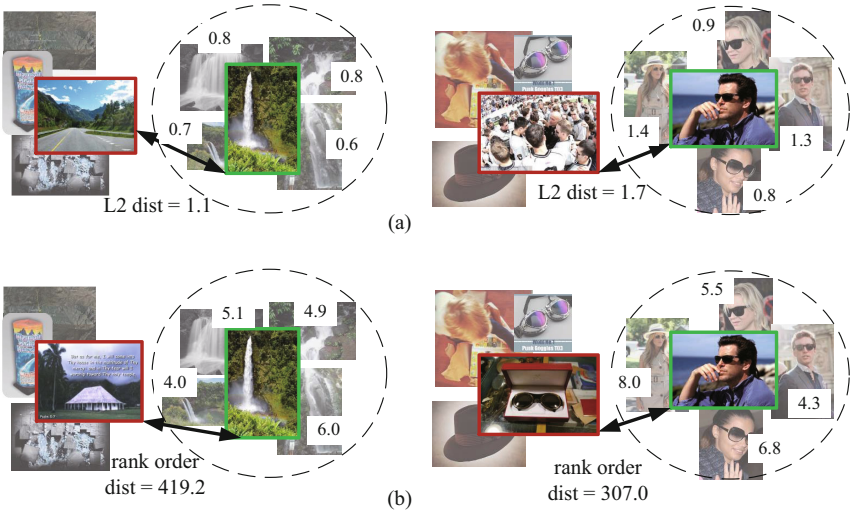

Fig. 2. (a) The L2 distances between the positive image and its nearest outlier are 1.1 and 1.7 for concept "waterfall" and "wearing glasses" respectively. Distance values are shown near images. Note that for concept "wearing glasses", there exists some positive neighbors with L2 distances larger than 1.1. (b) The rank-order distances between images in a dense area are much smaller.

measurement and select the top-ranked samples as the seeds. To translate this basic idea into a good algorithm, we need to deal with two issues.

Firstly, it is not effective to directly measure density in the Euclid space, because of the heterogenous problem: the L2 distance is incomparable across different concepts. As shown in Fig. 2(a), for the concept "waterfall", the L2 distance between a positive image and its nearest outlier is 1.1; while this value is 1.7 for concept "wearing glasses". These distance values are measured by deep learning features we extract. This implies that, for a same distance, a sample could be within the concept "wearing glasses", but beyond the concept "waterfall". This makes it hard to define a good density measurement in Euclid space.

Secondly, it is clear that samples with a very high density are positive, and samples with a very small density are negative. But it is not easy to draw a line in the middle to decide how many top images are selected: selecting too many images leads to low seeds precision, while selecting too few leads to low seeds ratio. In addition, the proportion of positive images differs for different concepts, which requires adaptive threshold.

Rank-Order Distance. We use rank-order distance [19] to address the heterogenous problem. We denote the initial set as $X = \{x_i : 1 \leq i \leq N\}$. Rank-order distance is defined as:

Fig. 3. Images and their density scores (shown below images)

$$d(x_i, x_j) = \frac{D(i,j) + D(j,i)}{\min(O_i(j), O_j(i))}, \tag{1}$$

$$D(i,j) = \sum_{k=0}^{O_i(j)} O_j(f_i(k)), \quad 1 \le i, j \le N,$$

where O_i is denoted as an order list ranked by the L2 distance from x_i to other images. $O_i(j)$ is the order of x_j in O_i. $f_i(k)$ returns the k^{th} image in O_i. (see [19] for details)

Rank-order distance is used here because it is defined on the structure of neighborhood. If two images have similar neighbors, then the distance between the two images is small, otherwise the distance is large. Because of the topological structures of different concepts are similar, rank-order distance is comparable across different concepts. Moreover, rank-order distance is more robust to outliers. The distances between images in a dense area are around 100 times smaller than those between outliers, but this difference is insignificant when measured in Euclid space, as illustrated in Fig. 2(b).

Density Score. With the rank-order distance, we define the measurement of density as the number of neighbors within a certain distance ($d = 15$ in our implementation). Formally, the neighborhood graph on X is defined as:

$$h(x_i, x_j) = \begin{cases} 1, & d(x_i, x_j) < d \\ 0, & d(x_i, x_j) > d \end{cases} \tag{2}$$

and the density score of x is $v(x) = \sum_{i=1\ldots N} h(x_i, x)$.

To understand how well the density score works, we visualize the crawled images under the concept "waterfall" in Fig. 3, along with their density scores. We find that for a high density (around 4,000), all images belong to "waterfall"; for a medium density (around 1,000), most images belong to "waterfall", and the rest belong to relevant concepts such as "fast-flow river" or "small waterfall"; for a small density (about 0), most images are irrelevant. It is interesting to mention that the image with a 107 density score (in the third row, third from left) says that "a long ponytail" looks similar to "waterfall".

Adaptively Thresholding. Denote the set of selected seeds as $S_t = \{x : v(x) \geq t\}$, and outliers $\overline{S}_t = X - S_t$. For various concepts, we propose choosing the optimal threshold t^* by maximizing the following objective function:

$$t^* = \operatorname*{argmax}_t E_u + E_i - E_e \tag{3}$$

$$\text{where:} \quad E_u = \frac{1}{|S_t|} \sum_{x \in S_t} v(x),$$

$$E_i = A(S_t, S_t),$$

$$E_e = \frac{1}{2}(A(S_t, \overline{S}_t) + A(\overline{S}_t, S_t)).$$

Here $A(S_t, R_t)$ is the similarity between two sets, which is defined hierarchically: first we define the similarity between two samples x and y as the amount of their common neighbors, i.e., $h(x,:)h(y,:)^T$; then we define the similarity between a sample and a set as the maximum similarity between this sample and the samples in the set, i.e., $g(x, R_t) = \max_{y \in R_t, y \neq x} h(x,:)h(y,:)^T$; finally we define the similarity between two sets as the average similarity between the samples of one set to the other set, i.e., $A(S_t, R_t) = \frac{1}{|S_t|} \sum_{x \in S_t} g(x, R_t)$.

So in Equation(3), the three terms E_u, E_i, E_e mean the following:

(1) E_u: the unary term. It equals to the average density of all selected seeds.
(2) E_i: the intra-seeds similarity term. It is the self similarities of the set of seeds.
(3) E_e: the extra-seeds similarity term. It is the cross similarity between the set of seeds and the set of outliers.

As there is only one unknown variable in the optimization, we use binary search to efficiently find the optimal threshold t^* and select S_{t^*} as seeds.

3.3 Dataset Growing

Given the generated seeds, we can grow the dataset in an iterative self-training way: (i) train a classifier using the seeds as the positive and an irrelevant reference set as the negative; (ii) apply the classifier to all crawled images; (iii) add images with high scores and remove those with low scores; (iv) iterate. Here an irrelevant reference set is introduced to anchor the iterative training and screen out outliers.

With this in mind, in order to reduce training cost, we first do **negative mining**: we train a linear svm classifier on the positive seeds and the negative reference set; apply the classifier to the negative set only; the hard negative samples are selected to form a new negative set; iterate. By doing this, we obtain hard negative samples, which are more effective in identifying the boundary of positive samples. We next do **positive mining**. This procedure is similar to the aforementioned self-training process, expect that the negative samples are not the whole reference set, but the hard negative samples.

In our implementation, we use NUS-WIDE [4] as the reference set. This dataset contains about 250K images crawled from Flicker, and we find 100K of these are enough for good performance. Usually 2%∼6% (i.e. 2K∼6K) images are selected as hard negative samples in negative mining.

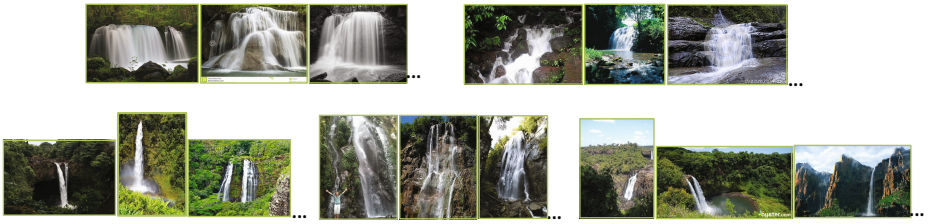

Fig. 4. Five groups of seeds after seeds dividing

Seeds Dividing. Generally, images under one concept may be multi-modal due to different scenes, viewpoints, illumination and so on. This fact will degrade the negative and positive mining because linear svm does not handle multi-modal problem so well. Therefore we propose dividing the seeds into multiple groups as a pre-processing step to alleviate the multi-modal problem. Given the divided groups, we train classifiers on each group separately.

We use k-means to divide the seeds into m groups ($m = 5$ in our experiments as a tradeoff between cost and performance). Take "waterfall" as an example again. In Fig. 4, seeds of "waterfall" are divided into five groups. Between different groups, there are apparent appearance gaps/differences, but within the same group the appearance variations are relatively smaller. Intuitively, we can tell the proposed seed dividing simplifies the classification task. With simpler tasks, linear svm can fit better and generalize better, resulting in better performance.

4 Experiments

In this section, we compare the proposed method with recently state-of-the-arts and demonstrate our performance in both quantitative and qualitative ways. We also experimentally investigate different components of our method to reveal more understandings.

Deep Learning Features. Recently, deep learning features trained on ImageNet [7] have been successfully applied in many vision tasks such as image classification [12,8,18] and object detection [17]. We first apply deep learning [12,8,18] features to the task of automatic dataset construction. Similar to [12], we train a seven-layer convolutional neural network on ImageNet 2010 and use it to extract features from images – 2048-dimensional features are extracted from the first fully-connected layer.

Evaluation Datasets. We use two datasets to quantitatively compare and analyze our method. The first one is Web-23 [14], containing 23 concepts with positive images and outliers. We also construct a synthetic dataset: we choose 16 concepts and gather corresponding images from ImageNet as the positive samples (see Table 1), and use randomly crawled web images as the negative outliers. Note these subsets are excluded from the training of the deep neural network. Each concept in the synthetic dataset is formed by mixing positive and negative samples ar a ratio of 1:1.

Table 1. The 16 concepts in the synthetic dataset and the number of positive images within

concept	bird	tower	building window	bus	car	car mirror	wheel	fireplace
#images	7196	9439	7799	11037	7046	3288	1981	1359
concept	bread	dish	hamburger	pizza	porridge	corn	garden	tree
#images	6850	10216	2473	2337	8879	1157	2388	2081

4.1 Comparison Experiments

Seeds Comparison. We conduct experiments to compare different seeds generating methods and their impacts on the constructed dataset. We compare our method with the recent state-of-the-art [3] on the synthetic dataset. [3] uses exemplar-LDA [11] classifiers to vote the seeds, on which multiple classifiers are trained to find more positive samples. We call this method ELDA for simplicity.

In our method, a density score based on rank-order distance is used for identifying seeds. A more straight forward approach is to define the density score based on L2 distance. We use this naive approach as a baseline. We also compare with ELDA which is a highly competitive method. For various seeds ratios (5%, 10% and 20%), we compare the seeds precision of different methods. Average results of 16 concepts are reported in Table 2, from which we can tell: (i) our method is significantly better than the baseline approach, because rank-order distance is more robust than L2 distance; (ii) compared to ELDA, the precision of our method is still 6%$\sim$9% higher, see visualized comparison in Fig. 5. We also apply our dataset growing method to both our and ELDA's seeds to isolate the quality of seeds for study. As shown in Table 2, our recall is 5%$\sim$7% higher than ELDA while the precision of both methods are comparable.

If adaptive thresholding is applied, we will have 98% seeds precision and 18% seeds recall. This result presents good balance between seeds ratio and seeds precision. Such a good balance translates into high dataset quality: 74.2% recall and 98.3% precision, better than any pre-defined threshold (i.e. selecting top 5%, 10%, and 20%).

To further investigate the impact of seeds on constructed datasets, we splice seeds generating and dataset growing in different methods. As shown in Table. 3, by replacing the seeds of ELDA with ours, recall/precision increase from 70.3%/91.9% to 72.4%/97.7%. If we replace the seeds in our method with ELDA's, recall/precision drop from 74.2%/98.3% to 72.1%/93.8%. These results imply that: (i) better seeds lead to better system performance; (ii)our seeds generalize well to ELDA's dataset growing process.

Table 2. Seeds comparison. The datasets are constructed by corresponding seeds and our dataset growing method.

Seeds Ratio	Seeds Precision			Recall/Precision of constructed dataset	
	L2	ELDA [3]	Ours	ELDA [3]	Ours
5%	75.7	90.9	99.7	52.0/95.4	59.0/98.5
10%	72.6	89.6	98.9	61.8/94.0	68.8/96.6
20%	70.2	88.0	94.2	73.3/93.8	78.2/92.2

Fig. 5. Seeds comparison visualization. We first generate image rank lists for our and ELDA method. Then we randomly sample images at the 20% quantile position of the rank lists.

System Comparison. We conduct system comparison with OPTIMOL([14]) and ELDA([3]). Results on Web-23 are shown in Fig. 6. The values of OPTIMOL are from [14,13], and the seeds' amount K in ELDA is set to be the same with ours. From Fig. 6, it can be seen that our method collects more images than the other methods, with a higher level of precision.

We also compare with ELDA on the synthetic dataset. In this experiment, we manually set the K so that the corresponding seeds ratio are 5%, 10% and 20% respectively. The comparison results are shown in Table 4 , from which we can see that: (i) our method consistently outperforms ELDA, no matter measured by recall or precision; (ii) larger seeds ratios generally lead to higher recall and lower precision.

4.2 Components Evaluation

Here we study the contributions of different components to our entire system. Basically our system contains three components: seeds generating, negative and positive mining and seeds dividing. We break down those components and design three experiments to analyze their contributions to the entire system.

- *Seeds*: only seeds generating, with dataset growing.
- *Seeds+NegPosMining*: negative and positive mining without seeds dividing.
- *Seeds+Divide+NegPosMining*: our full system, including all components.

Table 3. Seeds' impact on constructed datasets. The amount of ELDA's seeds is set to be the same with ours.

Seeds Generating	Dataset Growing	Recall	Precision
Ours	Ours	74.2	98.3
Ours	ELDA's	72.4	97.7
ELDA's	Ours	72.1	93.8
ELDA's	ELDA's	70.3	91.9

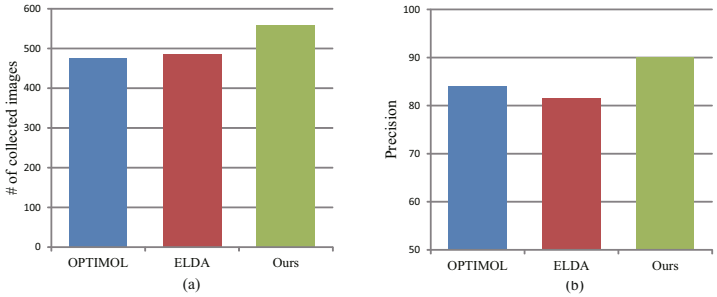

Fig. 6. System comparison on Web-23 dataset.

From the results in Fig. 7, we can see that: *First*, the recall of *Seeds* is around 20%∼40%. This is fairly good. Considering the typical amount of crawled images is more than 10K per concept, such recall can offer us over 2K images with very high precision (95% or more). *Second*, the negative and positive mining effectively improve the recall, which is around 50%∼80% after mining. *Third*, the system is further improved by seeds dividing, which helps the final recall achieve 70%∼90%. This means that we generally can collect more than 7K images for one concept.

We also observe that: for some concepts, like "bird" and "tower", recalls of our system are much higher than that obtained without seed dividing. This is because images under these concepts have diverse visual appearances. This indicates that seeds dividing is essential to our dataset construction system.

Table 4. System comparison on the synthetic dataset

concept	Ours recall/precision	ELDA's recall/precision		
		5% seeds	10% seeds	20% seeds
bird	70.0/98.3	21.2/90.8	32.1/86.3	53.7/82.3
tower	69.9/95.5	22.0/89.1	33.5/85.2	56.6/83.8
building window	89.7/96.3	20.9/85.6	38.3/82.0	51.2/78.7
bus	76.4/98.6	20.9/92.5	39.1/90.2	59.4/87.3
car	82.2/98.0	16.3/93.9	29.3/91.3	56.2/89.7
car mirror	52.6/99.1	18.9/83.6	28.6/81.6	46.9/80.9
wheel	56.1/99.8	9.6/97.4	29.3/93.2	43.7/89.5
fireplace	75.1/99.4	20.2/97.2	36.8/94.9	48.1/92.6
bread	73.8/97.9	17.3/90.3	29.3/88.3	55.0/85.9
dish	70.6/98.1	20.4/92.1	36.0/90.6	59.4/88.6
hamburger	75.7/98.6	10.6/95.3	31.9/94.0	45.2/93.1
pizza	85.5/99.4	21.5/94.7	39.3/94.2	64.7/91.0
porridge	88.1/99.3	17.0/90.6	39.2/88.9	50.7/87.5
corn	65.3/99.2	8.7/98.1	20.3/93.6	37.8/90.5
garden	83.8/98.4	10.1/89.0	29.1/86.2	41.8/81.2
tree	72.9/96.7	11.1/93.2	25.2/90.9	41.1/88.8

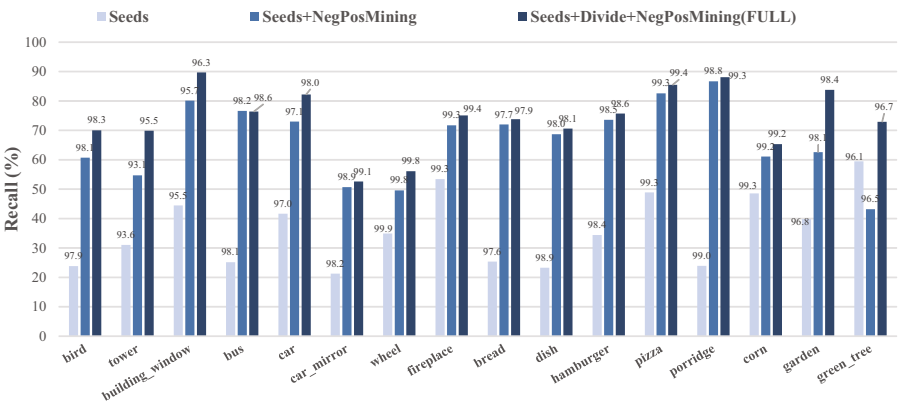

Fig. 7. Components Evaluation. The height of a bar stands for recall and the numbers above the bar is precision. (best viewed in color).

4.3 Dataset Constructed by Our System

Our system has now constructed a dataset which contains 80 concepts, averaging 7K images per concept. It takes nearly a month, and most of the time is spent on downloading images from Internet. The processing time for seeds generating and dataset growing

Table 5. Some concepts in our constructed dataset and corresponding statistics

	concept	#images crawled	#images collected	precision of collected images	#images in ImageNet
general concepts	bus	35232	16585	94.5	>15590
	train	24485	10087	96.8	>10041
	tree	39457	19885	96.0	>14433
concepts with few images in ImageNet	cabin	16224	6649	96.3	2546
	crowd	23943	9150	99.3	1296
	ferris wheel	10120	5283	98.8	1795
	fried egg	8087	5104	97.5	1295
	meeting room	11119	6681	98.0	1354
	sculpture	37213	15732	98.0	2677
	sun	24855	6816	99.3	1341
	tattoo	33364	21201	97.0	1685
	tree root	16554	8547	97.5	1769
concepts not exist in ImageNet	fly in sky	17813	5141	96.5	0
	team photo	9919	4829	98.5	0
	wearing glasses	15723	5082	92.4	0
	waterfall	18766	9368	98.5	0
	view from plane	3350	1274	98.3	0

Fig. 8. Exemplar images from our constructed dataset

is just around 3 hours per concept. Exemplary images from this dataset are shown in Fig. 8. We also present the statistics of some representative concepts in Table 5, including image amounts and corresponding precisions. We estimate the precision from a manually labeled subset (500 images per concept) of the entire dataset. The average precision is 95%.

Our constructed dataset demonstrates two desired properties which could be a good supplementary to existing large-scale datasets such as ImageNet: (i) apart from a handful of generic concepts such as "bus" and "train", most concepts in ImageNet only contain 1,000∼2,000 images, while our method can enlarge that amount by 5 times on average; (ii) Although a large range of concepts has been covered by ImageNet, the uncovered range is even larger. For example, there is no concept such as "wearing glasses" or "view from sky" in ImageNet, but our system can automatically collect many precise images for these concepts.

5 Conclusion and Future Work

In this work, we present a fully automatic system to construct large-scale, high-precision dataset from noisy web images. The system can collect thousands of images for one concept with very high precision – 95% or more. The superior performance is mainly due to the proposed seeds generating, negative and positive mining, as well as seeds dividing. Currently we only use visual features, and it is worth studying how to appropriately exploit text and web-ranking to further improve the system performance.

References

1. Berg, T.L., Forsyth, D.A.: Animals on the web. In: Computer Vision and Pattern Recognition, vol. 2, pp. 1463–1470 (2006)
2. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems (TOIS) 19(1), 1–27 (2001)
3. Chen, X., Shrivastava, A., Gupta, A.: Neil: Extracting visual knowledge from web data. In: International Conference on Computer Vision, vol. 3 (2013)
4. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: Nus-wide: A real-world web image database from national university of singapore. In: Proc. of ACM Conf. on Image and Video Retrieval (2009)
5. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction: An active learning approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 86–98. Springer, Heidelberg (2008)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, pp. 248–255 (2009)
8. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv:1310.1531 (2013)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010)

10. Feng, H., Chua, T.S.: A bootstrapping approach to annotating large image collection. In: Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 55–62 (2003)
11. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 459–472. Springer, Heidelberg (2012)
12. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)
13. Li, L.J., Fei-Fei, L.: Optimol: automatic online picture collection via incremental model learning. International Journal of Computer Vision 88(2), 147–168 (2010)
14. Li, L.J., Wang, G., Fei-fei, L.: Optimol: automatic online picture collection via incremental model learning. In: Computer Vision and Pattern Recognition (2007)
15. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. International Journal of Computer Vision 77(1-3), 157–173 (2008)
16. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. In: International Conference on Computer Vision, pp. 1–8 (2007)
17. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems, pp. 2553–2561 (2013)
18. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional neural networks. arXiv:1311.2901 (2013)
19. Zhu, C., Wen, F., Sun, J.: A rank-order distance based clustering algorithm for face tagging. In: Computer Vision and Pattern Recognition, pp. 481–488 (2011)