AUTOMATIC IMAGE DATASET CONSTRUCTION WITH MULTIPLE TEXTUAL METADATA

Anonymous ICME submission

ABSTRACT

The goal of this work is to automatically collect a large number of highly relevant images from the Internet for given queries. A novel image dataset construction framework is proposed by employing multiple textual metadata. In specific, the given query is first expanded by searching in the Google Books Ngrams Corpora to obtain a richer semantic description, from which the visually non-salient and less relevant expansions are then filtered. After retrieving the relevant images from the Internet, we further filter these noisy images by clustering and progressively Convolutional Neural Networks (CNN). To verify the effectiveness of our proposed method, we construct a dataset with 10 categories, which is not only much larger than but also have comparable cross-dataset generalization ability with the manually labelled dataset STL-10 and CIFAR-10. What's more, our method achieves a higher average precision than previous works.

Index Terms— Automatic Image Dataset Construction, Multiple textual metadata, Clustering, Progressively CNN

1. INTRODUCTION

Labelled image datasets have played a critical role in high-level image understanding and drive the progress of feature designing. For example, ImageNet has acted as one of the most important factors in the recent advance of developing and deploying visual representation learning models (e.g., deep CNN). However, the process of constructing ImageNet is both time consuming and labor intensive. It is consequently a natural idea to leverage image search engine (e.g., Google Image) or social network (e.g., Flickr) to construct the desired image dataset.

Generally, Google Image search engine has a relatively higher accuracy than the social network like Flickr. However, directly constructing database with the retrieved images by Google is not practical. It is mainly due to the download restrictions for each query and the unsatisfactory accuracy of ranking relatively rearward images. In order to tackle this problem, we propose a novel image database constructing framework, through which a large collection of highly relevant images are automatically extracted from the Internet. To build a high-quality image dataset from the Web, we propose to construct the collection for each query by three major steps: query expanding, noisy expansions filtering and noisy images filtering. The critical factors of this framework is to purify noisy expansions and noisy images for building the image dataset. Specifically, by searching in the Google Books Ngrams Corpora (GBNC), we firstly expand the given query to a set of semantically rich expansions, from which the noisy query expansions are then removed by exploiting both the word-word and word-visual similarity. After we obtain the candidate images by retrieving these filtered expansions with the search engine, as an important step, clustering and progressively CNN based methods are applied to further remove these noisy images.

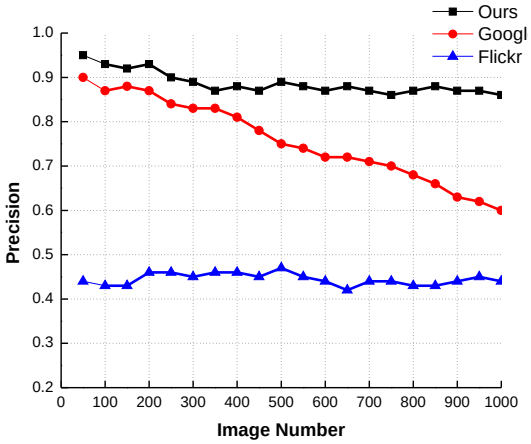


Fig. 1: The average precision of top 1000 images in Google image, Flickr and our dataset for 10 queries.

To verify effectiveness of the proposed automatic image database construction method, we build a image dataset with 10 categories named AutoImgSet-10. We evaluate its precision by comparing with methods [2], [3] and [4]. In addition, we also evaluate the cross-dataset generalization ability by comparing with two manually labeled image datasets STL-10 and CIFAR-10. Fig.1 demonstrates the improvement achieved by our method over the initially downloaded images from Google and Flickr.

2. RELATED WORK

To our knowledge, there are three principal methods of constructing image database: manual annotation, semi-automatic method and automatic method. Manual annotation has a high

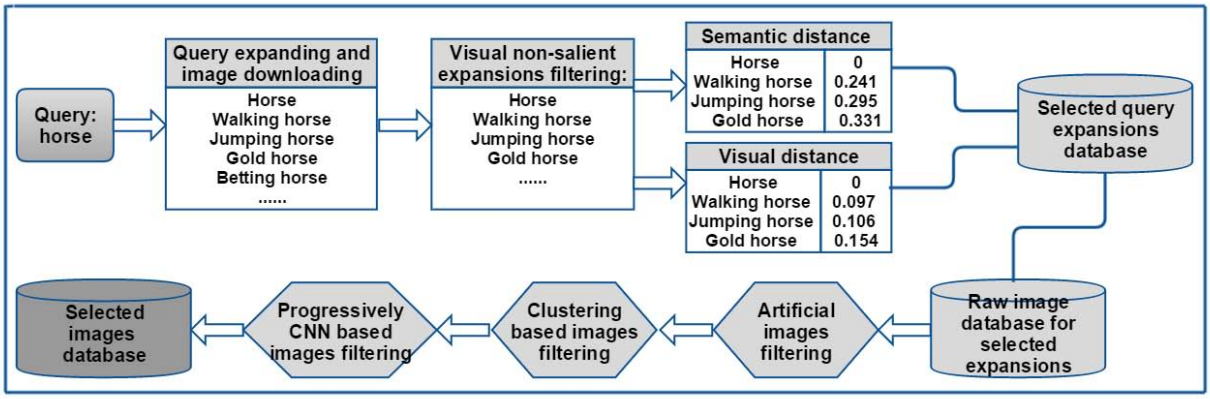


Fig. 2: System overview.

level of accuracy but is labor intensive. For example, it has taken several years to construct the ImageNet. To reduce the cost of manual annotation, some works also focus on active learning (a special case of semi-supervised method). [1] randomly label some images as seed images and these seed images are used to learn visual classifiers. Then the learned visual classifiers are applied to do image classifications on unlabeled images to find out unconfident images for manual labeling. The process is iterated until sufficient classification accuracy is obtained. However, both of manual annotation and active learning require pre-existing annotations which results in one of the biggest limitations to construct a large scale dataset.

To further reduce the cost of manual annotation, automatic methods have attracted more and more people's attention. [2] leverage the first few images returned by search engine to train image classifier (based on the fact that the first few images returned by search engine tend to be positive), classifying images as positive or negative. When the image is classified positive, the classifier uses incremental learning to refine its model. With the increase of classifier accepts more positive images, the classifier can get a better description of this query. [3] employs text information to re-rank images retrieved from search engine and uses these top-ranked images to learn visual models to re-rank images once again. [4] propose to use clustering based method to filter noisy "group" images and propagation based method to filter relatively small noisy images.

Compared to previous methods using one target query for image collection and image purifying, our method leveraging textual metadata in the process of dataset construction achieves a higher precision.

3. SYSTEM FRAMEWORK AND METHODS

We are targeting at constructing image dataset in a scalable way while ensuring accuracy. Fig.2 shows the process of how we construct image dataset for the given query. The basic idea is to leverage the high accuracy of first few images returned by search engine. In order to increase high accuracy images for the dataset, we expand the given query to a set of query expansions. However, query expanding not only take all the useful expansions, but also some noise. We take combined text and image processing methods to filter these noisy expansions. Due to the complexity of Internet, although we just take the first 100 images returned by search engine for each query expansion, we still have lots of chance to get noisy images. To further improve the accuracy, we take clustering based and progressively CNN based methods to filter these noisy images. The following subsections describe the details of our method.

3.1. Query expanding

Images returned by image search engine tend to have a higher accuracy than social network, but downloads are restricted to a certain number. Besides, the accuracy of ranking relatively rearward is also unsatisfactory. In order to obtain a large number of images with a high accuracy for the given query, we expand query to a set of query expansions and then download only few ranking forward images for these query expansions. GBNC [5] cover almost all related queries for any query at the text level. It's much more general and richer than WordNet [6]. We use GBNC to discover query expansions for the given query with Parts-Of-Speech (POS), specifically with NOUN, VERB, ADJECTIVE and ADVERB. Using GBNC helps us cover all expansions for any query the human race has ever written down in books. In addition, POS tag helps us to partially purify these query expansions. Table 1 shows query expanding precisions for ten queries and expanding details are shown in supplementary material.

3.2. Query filtering

Through query expanding, we get a richer semantic description for the given query. However, query expanding also brings some noisy expansions(e.g., "horse power",'betting horse" and "sea horse"). These noisy expansions are mainly divided into two types: (1) visual non-salient and (2) less relevant.

Table 1: Query expanding and noise filtering details for ten queries.

	Found query expansions				after visual non-salient filtering				after less relevant filtering			
Query	Total	Correct	Noisy	Precision	Total	Correct	Noisy	Precision	Total	Correct	Noisy	Precision
horse	811	446	365	0.55	545	398	147	0.73	285	272	13	0.95
bird	401	265	136	0.66	313	246	67	0.79	236	232	4	0.98
bus	347	212	135	0.61	250	183	67	0.73	167	157	10	0.94
airplane	696	480	216	0.69	524	452	72	0.86	377	362	15	0.96
sheep	276	218	58	0.79	232	204	28	0.88	181	176	5	0.97
train	314	132	182	0.42	189	125	64	0.66	116	107	9	0.92
cat	242	119	123	0.49	175	110	65	0.63	113	106	7	0.94
cow	171	144	27	0.84	140	132	8	0.94	130	130	0	1
dog	437	293	144	0.67	353	275	78	0.78	248	242	6	0.98
motorcycle	61	51	10	0.84	57	51	6	0.89	50	50	0	1

3.2.1. visual non-salient expansions filtering

From the perspective of visual, we want to identify visual salient query expansions and eliminate visual non-salient query expansions in this step. The intuition is that visual salient expansions should exhibit predictable visual patterns. We use image-classifier based filtering method.

For each query expansion, we directly download the first 100 images from Google image search engine as positive images; then randomly split these images into a training set (75 images) and validation set (25 images) $I_i = \{I_i^t, I_i^v\}$, we gather a random pool of negative images (50 images) and split them into a training set (25 images) and validation set (25 images) $\overline{I} = \{\overline{I}^t, \overline{I}^v\}$; We train a linear SVM C_i with I_i^t and $\overline{I}^t$ using dense HOG features and then use $\{I_i^v, \overline{I}^v\}$ as validation images to calculate the classification results. We declare an query expansion *i* to be visually salient if the classification results S_i giving a relatively high score (0.7).

3.2.2. less relevant expansions filtering

From the perspective of relevance, we want to find both semantic and visual relevant expansions for the given query. The intuition is that relevant expansions should exhibit a small semantic and visual distance. We use combined semantic and visual distance based filtering method.

Words and phrases acquire meaning from the way they are used in society. For computers, the equivalent of "society" is "database", and the equivalent of "use" is "a way to search the database". Normalized Google Distance (NGD) constructs a method to extract semantic similarity distance from the World Wide Web (WWW) using Google page counts[7]. For a search term *x* and search term *y*, NGD is defined by (1):

$$NGD(x,y) = \frac{max\{logf(x), logf(y)\} - logf(x,y)}{logN - min\{logf(x), logf(y)\}} \quad (1)$$

where *f(x)* denotes the number of pages containing *x*, *f(x,y)* denotes the number of pages containing both *x* and *y* and *N* is the total number of web pages searched by Google.

We denote the semantic distance of all query expansions by a graph $G_g = \{N, D\}$ where each node represents a query expansion and its edge represents the *NGD* between the two nodes. We set the target query as center (x) and other query expansions have a score (D_{xy}) which corresponds to the distance to the target query. It is defined as:

$$D_{xy} = \frac{NGD(x,y) + NGD(y,x)}{2} \quad (2)$$

Similarly, we represent the visual distance of query and expansions by a graph $G_v = \{C, E\}$ where each node represents a query expansion and each edge represents the visual distance between query and expansions. Each node has a center C_y which corresponds to k = 1 kmeans clustering center. The feature is 1000 dimensional Bag of visual words based on SIFT features. The edge weight E_{xy} correspond to the euclidean distance.

The semantic distance and visual distance will be used to construct a new 2 dimensional feature *V = [D_{xy};E_{xy}]*. The label is 1 (positive) or 0 (negative). We select n_+ positive training examples from these expansions which have small semantic distance or visual distance, a subset of these positive examples may be "noisy". The case of negative examples is more favorable: we calculate the semantic distance and visual distance between different query expansions (e.g., "horse" and "cow") and get the n_- negative training examples. We don't choose to select the n_- negative training examples from these expansions which have a little big semantic distance or visual distance because these expansions have a higher probability to be positive than other different query expansions.

Then the problem can be translated to calculate the importance weight *w* for feature *V* to determine whether the expansion is relevant or not. Based on this situation: (1) feature dimensional and traing data is relatively small, (2) training data has a little noise. We choose to use an SVM classifier since it has the potential to train despite noise in the data and it doesn't require too many features and training examples. The SVM training can be translated into the following optimization problem:

$$\min \quad \frac{1}{2}||\vec{w}||^2 + C_+ \sum_{i:y_i=1} \xi_i + C_- \sum_{j:y_j=0} \xi_j \quad (3)$$

$$s.t. \quad \forall k : y_k \left[\vec{w} \cdot \vec{V_k} + b\right] \geq 1 - \xi_k \quad (4)$$

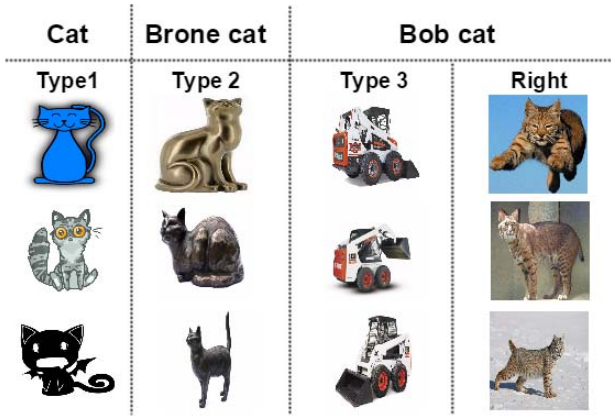


Fig. 3: Three types of noisy images in the raw image dataset.

where V_k is the feature vector of example i and $y_k \in \{1, 0\}$ is the class label. C_+ and C_- are the false classification penalties for the positive and negative expansions with ξ being the corresponding slack variables. We solve this optimization problem with publicly available SVM software LIBSVM. All experiments towards finding an appropriate representation were done on the training set using linear SVMs. Three parameters w, C_+ and C_- are optimized by using 10-fold cross validation on the training set. Finally, the trained SVM is used to filter out noisy expansions based on the semantic distance and visual distance towards to the target query.

Filtered expansions are then used to download the top 100 images from search engine to construct the raw image dataset for the target query. As shown in Table 1, our method is not able to remove noisy expansions thoroughly in most of the cases. However, the raw image dataset still achieves a much higher accuracy than directly using the Flickr or Google image data. To further purify the raw image dataset, we take a series of methods to remove noisy images in the next section.

3.3. Image filtering

Although Google image search engine has ranked the returned images, some noisy images are still included. The reason is that Google image is a text based search engine. In addition, a few unfiltered noisy expansions will also bring some noisy images to the raw image dataset. As shown in Fig.3, these noisy images can be divided into three categories: artificial images (type 1), noisy images brought by noisy expansions (type 2) and noisy images which don't match query (type 3).

3.3.1. artificial images filtering

We remove artificial images as we are just interested in building natural image dataset. Artificial images contain: sketches, drawings, cartoons, charts, comics and so on. All of these images tend to have a few colors in large areas. Based on this motivation, we train a radial basis function SVM using color histogram features. The artificial images were obtained by using key words: "sketch", "drawings","cartoons" and "charts" to download from Google image search engine (1000), natural images were obtained by manual selected (1000). When the SVM model was learned, it can be used to filter out noisy artificial images on the entire raw image dataset. Although the color histogram features+SVM framework that we use is not the prevailing state-of-the-art image classification method, we found our method to be effective in removing this type of noisy images.

Table 2: The scale of image dataset AutoImgSet-10.

Query	Data scale	Query	Data scale
Horse	22K	Bus	13K
Bird	21K	Sheep	14K
Dog	20K	Train	8K
Cat	9K	Cow	11K
Airplane	30K	Motorcycle	49K

3.3.2. Clustering based images filtering

In order to further purify type 3 noisy images, we take clustering based images filtering method. The motivation is: it's much easier for computers to decide whether a group of images are sharing similar visual patterns than determine whether an individual image is relevant to an query. Due to the complexity of Internet data, we can't set a specific cluster number for all the query expansions image data. We cluster the images for each query expansion using Affinity Propagation based on their visual similarities. Then the problem is converted to how to choose the relevant clusters.

Generally speaking, bigger clusters and visually consistent clusters have higher probability to be relevant to the query. In our data, as our images were downloaded from search engine with index number, clusters with lots of ranked relatively rearward images also have higher chance to be relevant to the query. Based on this motivation, we add weight w_i to each image according to their ranking index number. Then the scores of each cluster can be calculated by:

$$Scores = \sum_{i=1}^{k} w_i I_i \tag{5}$$

where w_i represent the weight of ranking i_{th} image, I_i represent the i_{th} image and k represent the numbers of image in the cluster. In summary, we use the following features to discover relevant clusters: (1) scores of the cluster; (2) size and percentage of the cluster; (3) minimum, maximum and average distances of images in the cluster. After choosing features, we label a set of clusters to learn a SVM classifier that determine whether the cluster is relevant to query. The labeling work only need to be done once for all queries and the learned classifier can be used on all the clusters.

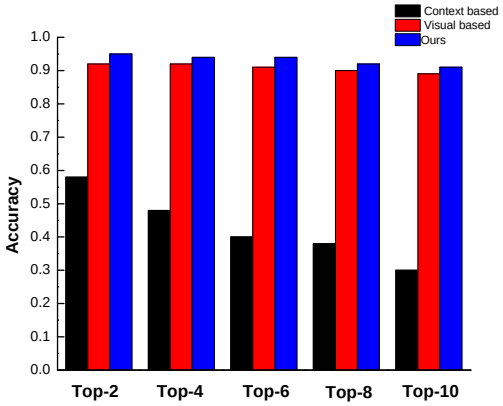


Fig. 4: Average accuracy of Top-K similar query expansions.

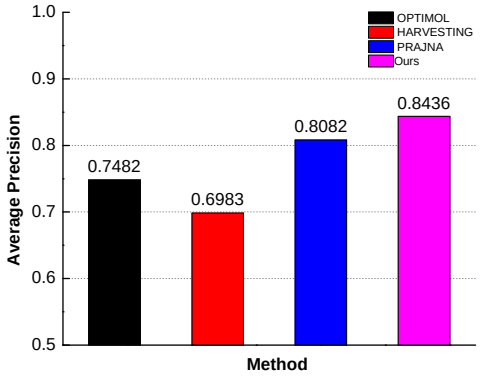


Fig. 5: Average precision of dataset constructed by us and three other methods.

3.3.3. Progressively CNN based images filtering

In order to further purify the image dataset, we take progressively CNN based filtering method. The intuition is we want to keep images with distinct sentiment scores between classes with high probability. We fine-tune a CNN model using filtered images on a trained model "$bvlc\_reference\_caffenet$" [8]. Then all of the filtered images are used to do image classification using the fine-tuned model. We take the probabilistic sampling algorithm to select the new training sample images according to the classification scores on the training data itself. We use the new selected sample images to further fine-tune the previous model, repeat the above steps until reach the preset iteration value (1000).

Let $Scores(i) = (V_{i1}, V_{i2})$ be the classification scores for the first two classes of instance i. We choose to select the training instance i as new selected training instance with probability $P(i)$ given by:

$$P(i) = 1 - max(0, 2 - exp(|V_{i1} - V_{i2}|)) \qquad (6)$$

The training instance will be kept in the training set if the classification scores of one training instance are large enough. Otherwise, the smaller the difference between the classification scores, the large probability that this instance will be removed from the training set. Type 2 and type 3 noisy images can be effectively filtered using this method. The reason for this is that the number of noisy images are relatively small in the whole image dataset for the target query. Table 2 shows the detailed scale for each query in AutoImgSet-10 and our database will be released if our paper was accepted.

4. EXPERIMENTS

In our experiments, one image dataset named AutoImgSet-10 is constructed to verify the effectiveness of our method. We carry three quantitative evaluations for the learned query expansions and Image dataset.

4.1. Query expansions

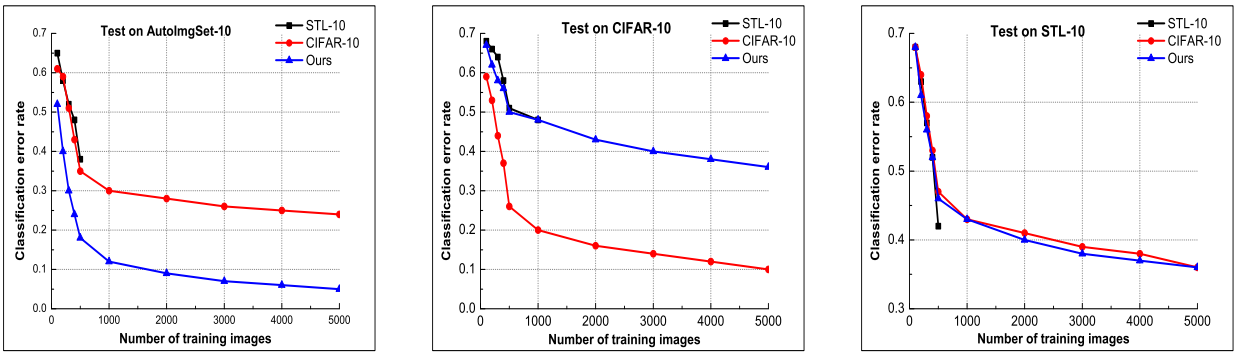
The ground truth of query and expansions are similar if they are sharing similar visual patterns, otherwise not. We carry a quantitative evaluation for the learned query expansions by comparing it with method [10] which filter noisy expansions with context constraints and state-of-the-art method [11] which filter noisy expansions with visual constraints. Our method achieves a higher precision. The reason is that we filter noisy query expansions with combined semantic and visual distance which is much more efficient than just using context or visual constraints. Thus our method is more suitable to expand queries for image dataset construction. Fig.4 shows the average accuracy of Top-K query expansions for method [10], [11] and ours.

4.2. Image dataset

The image dataset AutoImgSet-10 we constructed has 10 categories. We firstly compare the precision of our dataset with three fully automatic methods [2], [3] and [4]. Then we compare the cross-dataset generalization ability of our dataset with two publicly image dataset STL-10 and CIFAR-10.

Due to both of the sizes and species are different, we can't directly compare the precision of a particular category. Instead, we compare the average precision of dataset constructed by us and three other methods in Fig.5. Our method has a higher precision than previous methods mainly bacause we use multiple textual metadata in the process of constructing dataset.

As we can't get the dataset extracted by [2], [3] and [4], we compare the cross-dataset generalization ability with two publicly available dataset STL-10 and CIFAR-10. Cross-dataset generalization measures the performance of classifiers learned from one dataset on the other dataset [12]. To be fair, we choose these five same categories (horse, bird, airplane, cat and dog) to verify their cross-dataset generalization ability with STL-10 and CIFAR-10. We randomly select 500 training images and 500 testing images for each category in



(a) (b) (c)

Fig. 6: Cross-dataset generalization of HOG+SVM trained on STL-10, CIFAR-10 and AutoImgSet-10, then tested on: (a) AutoImgSet-10, (b) CIFAR-10 and (c) STL-10.

STL-10, CIFAR-10 and our dataset (because the maximum number of training data in STL-10 is 500). Then each dataset was used to learn the image classification model based on same feature (HOG) and learning method (SVM). We use the learned model to do image classification on these three image datasets. The results are shown in Fig. 6.

In all three cases, with the same number of training images, the best performance is achieved by training and testing on the same dataset AutoImgSet-10. Since the smallest dataset STL-10 only has 500 training images per category, we compare the performance of three different dataset at the point of 500 training images, it shows that the generalization ability of these three datasets is very close and our dataset performs slightly better than STL-10 and CIFAR-10. In addition, our dataset is larger than the other two datasets, it achieves the best performance on two testing sets when all training images are used. Note, our dataset was constructed automatically while other datasets were manually labeled.

5. CONCLUSION AND FUTURE WORK

In this work, we presented a new framework for automatically building high-quality image dataset with multiple textual metadata. Three successive modules were employed in the framework including query expanding, noisy expansions filtering and noisy images filtering. Using this method, we constructed a image dataset AutoImgSet-10 with 10 categories. Through our experiments, we found our image dataset constructed by automatically has a higher average precision than automatic methods [2], [3] and [4]. Besides, our dataset can surpasses the manually labeled dataset STL-10 and CIFAR-10 in terms of both scale and cross-dataset generalization ability.

Although good results were obtained in this work by the attempt to make use of textual metadata in the process of building image dataset, there is still room to improve our approach. For example, we can potentially use more sophisticated approaches to purify noisy images and that will be the focus of our future work.

6. REFERENCES

[1] Brendan Collins, Jia Deng, et al., "Towards scalable dataset construction: An active learning approach," in *ECCV 2008*

[2] Li-Jia Li and Li Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," in *IJCV 2010*

[3] Florian Schroff, Antonio Criminisi, et al., "Harvesting image databases from the web," *IEEE PAMI 2011*

[4] Xian-Sheng Hua and Jin Li, "Prajna: Towards recognizing whatever you want from images without image labeling," in *AAAI 2015*,

[5] Yuri Lin, Jean-Baptiste Michel, et al., "Syntactic annotations for the google books ngram corpus," in *ACL 2012*.

[6] George A Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[7] Rudi L Cilibrasi and Paul MB Vitanyi, "The google similarity distance," *IEEE TKDE 2007*

[8] Yangqing Jia, Evan Shelhamer, et al., "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM 2014*

[9] Quanzeng You, Jiebo Luo, et al., "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *AAAI 2015*

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," in *EMNLP 2014*

[11] Yalong Bai, Kuiyuan Yang, et al., "Automatic image dataset construction from click-through logs using deep neural network," in *ACM MM 2015*

[12] Antonio Torralba, Alexei Efros, et al., "Unbiased look at dataset bias," in *IEEE CVPR 2011*