A new web-supervised method for image dataset constructions

Yazhou Yao[a,b], Jian Zhang[a,*], Fumin Shen[c], Xiansheng Hua[d], Jingsong Xu[a], Zhenmin Tang[b]

[a] *Global Big Data Technologies Center, University of Technology Sydney, NSW 2007, Australia*
[b] *School of Computer Science and Technology, Nanjing University of Technology, Nanjing 210094, China*
[c] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*
[d] *Alibaba Group, Hangzhou 310052, China*

Abstract

The goal of this work is to automatically collect a large number of highly relevant images from Internet for given queries. A novel automatic image dataset construction framework is proposed by employing multiple query expansions. In specific, the given queries are first expanded by searching in the Google Books Ngrams Corpora to obtain a richer text semantic descriptions. Secondly, the visually non-salient and less relevant expansions are filtered. Thirdly, after retrieving images from Internet with filtered expansions, we further filter noisy images through clustering and progressively Convolutional Neural Networks (CNN) based methods. To evaluate the performance of our proposed method for clean image dataset constructions, we build an image dataset with 10 categories. We then run object detections on our image dataset with three other image datasets which were constructed by weak

*Corresponding author
Email address: Jian.Zhang@uts.edu.au (Jian Zhang)

supervised, web supervised and full supervised learning, the experimental results indicated that the effectiveness of our new image dataset construction method is superior to weak supervised and web supervised state-of-the-art methods. In addition, we do a cross-dataset classification to evaluate the performance of our dataset with two publically manual labelled dataset STL-10 and CIFAR-10.
Keywords: image dataset construction, multiple query expansions, web-supervised

1. Introduction

Labelled image datasets have played a critical role in high-level image understanding. For example, *ImageNet* [1] 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 social network like Flickr. However, directly constructing image dataset with the retrieved images from image search engine is not practical. It is mainly due to the number of images downloaded from image search engine for each query (e.g., Google image search engine is 1000 [8]) and the unsatisfactory accuracy of ranking relatively rearward images.

In order to improve the overall accuracy, some authors proposed to re-rank the images returned from image search engine [2][3][4][5]. [2] re-ranked images

by taking into account of the text contents on the original page from which the images were obtained. [3] involved visual clustering of the images using probabilistic Latent Semantic Analysis (pLSA) [6] on a visual vocabulary. [4] used multiple instance learning and iteratively methods to learn the visual models. [5] proposed an incremental learning strategy to learn the visual models. However, all of these methods have a restriction on the total number of images which can be downloaded from the image search engine.

In order to overcome the restriction of downloading number, [7][8] proposed to use web search to obtain a large pool of images instead of image search engine. The method in [7] can be mainly divided into two steps: First, train a classifier with manual intervention. Then, the classifier is used to re-rank the downloaded images. The advantages of this method are overcoming the restriction of downloading number, as well as avoiding the problem of polysemy and providing relatively high accuracy images for the given query. However, due to the needs of manual intervention, the cost of this method is high which results in the scale problem. [8] adopt text information to re-rank images retrieved from web search and used these top-ranked images to learn visual models to re-rank images once again. The advantage is eliminating the need of manual intervention. The accuracy of image dataset constructed by this method is relatively low. The main reason is the low accuracy of images returned from web search.

In order to leverage the high accuracy as well as overcome the downloading restrictions of image search engine, we propose a novel image dataset constructing framework, through which a large of highly relevant images are automatically extracted from the Internet. The framework can be divided

into three major steps: namely, the step 1: query (text) expanding, the step 2: noisy query (text) expansions filtering and the step 3: noisy images (e.g., the incorrect image contents from the semantic point of view) filtering. The critical technical challenges are in the step 2 and 3. Specifically, by searching in the Google Books Ngrams Corpora (GBNC), the given query is firstly expanded to a set of text semantically rich expansions, the noisy query expansions are then filtered by exploiting both the word-word and visual-visual similarity. Secondly, candidate images are retrieved by using these filtered expansions from image search engine. Thirdly, clustering and progressively CNN based methods are applied to further filter those noisy images from the semantic point of view.

To verify the effectiveness of our proposed method, we construct an image dataset with 10 categories AutoImgSet-10. We evaluate the object detection ability of our image dataset with three other image datasets which were constructed by weak supervised, web supervised and full supervised learning [9] [10][11].In addition, the cross-dataset generalization ability was evaluated on our dataset AutoImgSet-10 and two manually labelled image datasets STL-10 and CIFAR-10.

Our contributions in these paper mainly are:

1. We are the first to use query expansions in the process of image dataset constructions. By expanding query to a set of query expansions, we get a richer text semantic descriptions for the given query. Using multiple expansions to retrieve images can effectively overcome the restriction of downloading number from image search engine.
2. We propose three different filtering mechanisms for three different kinds

of noisy images in the process of image dataset constructions. Using these filtering mechanisms can effectively improve the overall accuracy of image dataset.

3. Using multiple query expansions to retrieve images and construct the image dataset can effectively reduces failure due to the statistical domain adaptation problem.

The rest of the paper is organized as follows: In Section 2, a brief discussion of related works are given. The proposed algorithm including query expanding, noisy expansions filtering and noisy images filtering is described in Section 3. We evaluate the performance of the proposed algorithm with several other methods in Section 4. Finally the conclusion and future work are offered in Section 5.

2. Related works

To our knowledge, there are three principal methods of constructing image dataset: manual annotation, semi-automatic method and automatic method. Manual annotation has a high accuracy but is labor intensive. For example, it has taken several years to construct the *ImageNet*[1]. In order to reduce the cost of manual annotation, some works also focused on active learning (a special case of semi-supervised method) [12][13][14]. [12] randomly label some images as seed images to learn visual classifiers. Then the learned visual classifiers are applied to do image classifications on unlabeled images to find out images which have unconfident scores for manual labelling. The process is iterated until sufficient classification accuracy is obtained. [13] presented an active learning framework to simultaneously learn contextual

models for scene understanding tasks (multi-class classification). [14] presented an approach for on-line learning of object detectors, in which the system automatically refines its models by actively requesting crowd-sourced annotations on images crawled from the Web. 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 image dataset.

To further reduce the cost of manual annotation, automatic methods have attracted more and more people's attention. [5] leveraged the first few images returned from image search engine to train image classifier (based on the fact that the first few images returned from image search engine tend to be positive), classifying images as positive or negative. When the image is classified as a positive sample, the classifier uses incremental learning to refine its model. With the increase of classifier accepting more positive images, the trained classifier will reach a robust level for this query. [15] proposed to use clustering based method to filter noisy "group" images (e.g.; the incorrect image contents from the semantic point of view) and propagation based method to filter relatively small noisy images.

Other works relate to the step of filtering noisy query expansions and filtering noisy images. [16] is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus and achieves the state-of-the-art performance on word-word similarity computing. [17] represented images and annotations jointly in a low dimensional embedding space for similarity evaluation, it's limited by the used low level visual representations. [18] mapped images into a semantic space via word embedding. However, the

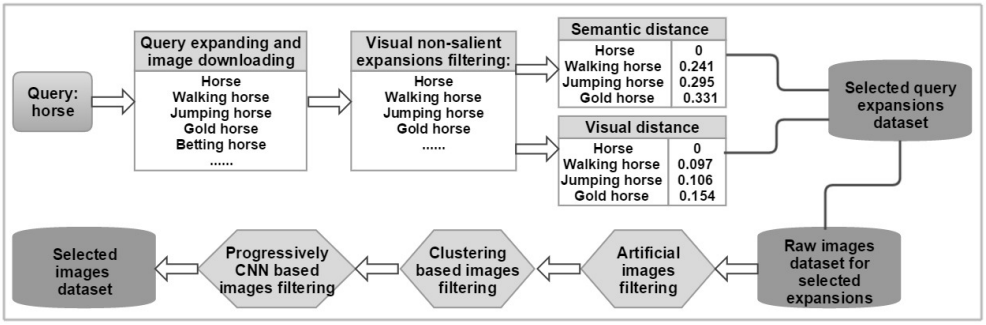


Figure 1: System overview.

semantic space constructed from text corpus is suitable for natural language processing tasks while not fully reflects visual similarity.

Our method is largely inspired by the following works. A visual concept learning system was recently proposed in [10] and achieved impressive performance for object detection. It firstly find all the related parts of the object, then train a mixture DPM [11] detector for the object. The main difference between us is that [10] is formulated for object detection while our method is to automatic construct image datasets for the given queries.

3. System framework and methods

We are targeting at constructing image datasets in a scalable way while ensuring accuracy. Fig.1 shows the process of our proposed method. In order to overcome the number limitation of image downloading through image search engine (e.g., Google Image), we expand the given text query to a set of query expansions. Although, such expanding will bring useful expansions, it brings some noisy text query expansions as well. We filter these noisy expansions based on the similarity distance of the text query. Similarly,

due to the complexity of Internet, using filtered text query expansions to download images from image search engine may also bring some noisy images (e.g.; the incorrect image contents from the semantic point of view). To further improve the accuracy of image dataset, we take clustering based and progressively CNN based methods to filter noisy images in the raw image dataset. The following subsections describe the details of our method.

3.1. Query expanding

Images returned from image search engine (e.g.: Google Image) tend to have a higher accuracy than social network (e.g.: Flickr), but downloads are restricted to a certain number. Besides, the accuracy of ranking relatively rearward is also unsatisfactory. To overcome such restriction, synonyms are often used to expand a query to a set of expansions for more image downloading from the image search engine. However, this method only works well for queries defined from existing ontology such as WordNet [19]. In order to get rid of existing ontology dependence and generalize to queries which have not been compiled into existing ontology, we automatically expand query by searching in the GBNC [20]. GBNC covers almost all related query expansions for any query at the text level. 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 possible query the human being has ever written down in books. In addition, POS tag helps us to partially purify these query expansions. Table 1 shows query expanding details for ten queries.

Table 1: Query expanding details for ten queries.

	Found query expansions			
Query	Total	Correct	Noisy	Precision
horse	811	446	365	0.55
bird	401	265	136	0.66
bus	347	212	135	0.61
airplane	696	480	216	0.69
sheep	276	218	58	0.79
train	314	132	182	0.42
cat	242	119	123	0.49
cow	171	144	27	0.84
dog	437	293	144	0.67
motorcycle	61	51	10	0.84

3.2. Noisy query expansions filtering

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

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

Table 2: The average recall and precision for ten queries corresponding to S_i

S_i	0.9	0.8	0.7	0.6	0.5	0.4	0.3
Recall	0%	40.66%	87.85%	97.42%	100%	100%	100%
Precision	0%	87.06%	78.90%	71.18%	68.74%	65.60%	65.60%

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 S_i. The classification results and its corresponding recalls and precisions are shown in Table 2. After considering the recall and precision, we choose a query expansion i to be visually salient if the classification results S_i giving a relatively high score (0.7). The reason is that we want to get a relatively higher precision while ensuring an acceptable recall rate.

3.2.2. Less relevant expansions filtering

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

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[21]. For a search term x and search term y (just the name for a query rather than the query itself), NGD is defined by:

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

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 text 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 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: $D_{xy} = \frac{NGD(x,y)+NGD(y,x)}{2}$. 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 K-means 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 text 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 text semantic distance and visual distance, a subset of

these positive examples may be "noisy". However, to get the n_- negative training samples, we chose these negative simples directly from the different queries (e.g., "horse" and "cow") rather than from the expansions in the same query. The reason is that these expansions (in the same query) have a higher probability to be positive than different queries. Then the problem can be formulated to calculate the importance weight w for feature V to determine whether the expansion is relevant or not. Given that (1) the feature dimension and training data is relatively small, (2) the training data still have some noises. We choose to use SVM in our experiment. The training process can be formulated into the following optimization problem:

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

$$s.t. \quad \forall k : y_k \left[\vec{w} \cdot \vec{V_k} + b\right] \geq 1 - \xi_k \tag{3}$$

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 [22]. 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. Algorithm 1 shows the process of query expanding and noisy expansions filtering. The images corresponding to the filtered query expansions are then used to construct so-called the raw image dataset for the given query. As

Algorithm 1 Query expanding and noisy expansions filtering

Input:

$X = \{x_0\}$, a given query

1: Expand given query in GBNC with POS and get a set of query expansions $X = \{x_0, x_1, x_2....x_n\}$

2: Delete visual non-salient x_i from X if $S_i \leqslant 0.7$

3: Calculate word-word similarity D_{xy} and visual-visual similarity E_{xy} between x_0 and $\{x_1, x_2....x_n\}$

4: Construct a new relevant feature $V = [D_{xy};E_{xy}]$ and train a relevant classification model based on feature V

5: Delete less relevant x_j from X if x_j is classified into negative category

Output:

A relatively clean expansion for the given query

shown in Table 3, our method is not able to remove the noisy expansions thoroughly in most of the cases. Nevertheless, the raw image datasets constructed by those filtered expansions still achieves a much higher accuracy than directly using the Flickr or Google image data. To have further purifying on the raw image datasets, we will remove those noisy images in the next section.

3.3. Noisy images 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.2,

Table 3: Query expanding and noise filtering details for ten queries.

	after visual non-salient filtering				after less relevant filtering			
Query	Total	Correct	Noisy	Precision	Total	Correct	Noisy	Precision
horse	545	398	147	0.73	285	272	13	0.95
bird	313	246	67	0.79	236	232	4	0.98
bus	250	183	67	0.73	167	157	10	0.94
airplane	524	452	72	0.86	377	362	15	0.96
sheep	232	204	28	0.88	181	176	5	0.97
train	189	125	64	0.66	116	107	9	0.92
cat	175	110	65	0.63	113	106	7	0.94
cow	140	132	8	0.94	130	130	0	1
dog	353	275	78	0.78	248	242	6	0.98
motorcycle	57	51	6	0.89	50	50	0	1

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 images 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

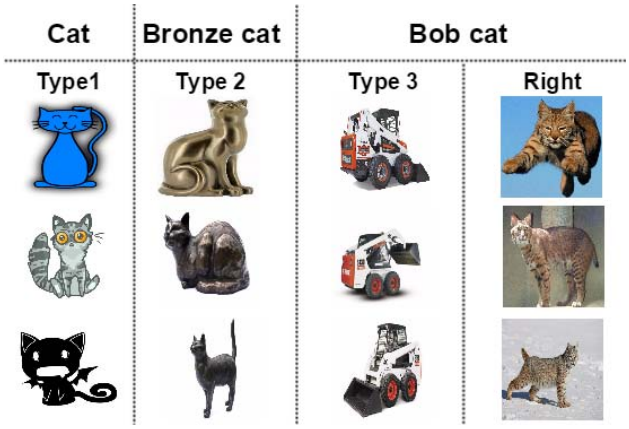


Figure 2: Three types of noisy images in the raw image dataset.

selected (3000). When the SVM model was learned, it can be used to filter out noisy artificial images on the entire raw image dataset.

In order to validate our ideas, we randomly select 1000 artificial images and 1000 natural images from the raw image dataset. We separately select nBins (16, 32, 64) for each color channel, so the dimension of color histogram feature is $nBins^3$. We use the learned SVM model to do image classification on these 2000 images which are selected from the raw image dataset. Table 4 shows the loss of natural images and filtered artificial images. By experimental observations, we choose 16 as the final nBins for each color channel. Although it has a little higher loss of natural images, it can much more effectively filter out artificial images than others.

Table 4: The loss of natural images and filtered artificial images

nBins	16	32	64
Loss of natural images	4.9%	3.8%	3.2%
Filtered artificial images	67.5%	54.7%	47.7%

3.3.2. Clustering based noisy images filtering

In order to further purify Type 3 noisy images in Figure 2, we take clustering based images filtering method. Our motivation is to focus on whether a group of images are sharing similar visual patterns which is relevant to a query. Then the problem can be converted to three mainly problems: feature selecting, cluster number determining and relevant clusters choosing.

Feature selecting Generally speaking, the bigger clusters and visually consistent clusters are, the higher probability will be relevant to the query. In our data, since our images are downloaded from image search engine with index number, clusters including those ranked as 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:

$$Scores = \sum_{i=1}^{k} w_i I_i \tag{4}$$

where w_i represents the weight of ranking i_{th} image, I_i represents the i_{th} image and k represents 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.

Cluster number determining Due to the complexity of Internet data, we can't set a specific cluster number for all the images data which corresponding to query expansions. We try to use automatic methods like Affinity Propagation to get the cluster number k, but the results are not satisfactory. Instead, we propose an automatically method to determine the cluster num-

ber k for different query expansions image data.

For each query expansion image data, we take spectral clustering and start with cluster number $k = 1$. We increase the cluster number when the data appear not to meet Gaussian distribution. Each iteration of our method splits into two cluster numbers. If the data currently assigned to a k cluster center appear to meet Gaussian distribution, then we stop the splitting. In our experiment, when k = 15 and the data still does not appear to meet Gaussian distribution, we also stop the splitting. The hypotheses test to verify whether the assigned data to a center is sampled from Gaussian distribution is as follows:

- H_0: The d dimensions data $X = \{x_1, x_2..., x_k\}$ around the center c are sampled from a Gaussian
- H_1: The d dimensions data $X = \{x_1, x_2..., x_k\}$ around the center c are not sampled from a Gaussian

If the null hypothesis H_0 is accepted, then it is believed that the one center is sufficient for modelling its data. The center splitting is not necessary. and we should not split the cluster into two sub-clusters. If H_0 is rejected and H_1 is accepted, the cluster will be spitted into two sub-clusters.

The test we use is based on the Anderson-Darling (AD) statistic [23]. As the AD statistic test is one-dimensional test, we need to transform d dimensions data X into one dimension representation $X' = \{x_1', x_2'..., x_k'\}$. Given a subset of data X in d dimensions which belongs to center c, we firstly assume k = 2 and run spectral clustering in X. Then we get two centers c_1 and c_2. We construct d dimensions vector $v = c_1$ - c_2 and project d dimensions

data $X = \{x_1, x_2..., x_k\}$ onto vector v by [23]:

$$x_i^{'} = \langle x_i, \mathrm{v} \rangle / ||v||^2 \tag{5}$$

to get one dimension data $X^{'} = \{x_1^{'}, x_2^{'}..., x_k^{'}\}$ which has mean 0 and variance 1. We let $z_i = F(x_i)$, where F is the *N(0,1)* cumulative distribution function. Then the statistic is [23]:

$$A_*^2(Z) = A^2(Z)(1 + 4/n - 25/(n^2)) \tag{6}$$

where $A^2(Z)$ is defined as

$$A^2(Z) = -\frac{1}{n}\sum_{i=1}^{n}(2i-1)[log(z_i) + log(1 - z_{n+1-i})] - n \tag{7}$$

In our experiments, if $A_*^2(Z)$ is in the range of non-critical values at confidence level α (0.0001), then accept H_0, keep the original center and discard $\{c_1, c_2\}$. Otherwise, we reject H_0 and keep $\{c_1, c_2\}$ in place of the original center. The process of determining the cluster number are shown in Algorithm 2.

Relevant clusters choosing After choosing features and cluster numbers, we label a set of clusters to learn a SVM classifier that determine whether the cluster is relevant to query or not. In our experiments, we randomly select 1000 relevant clusters and 1000 less-relevant clusters as the positive and negative training samples. The labelling work only need to be done once for all queries and the learned classifier can be applied on all the clusters. To verify the performance of our method, we then select 2500 relevant clusters and 2500 less-relevant clusters as the test data. We compare our method with K-means based clustering (as the baseline) and Affinity Propagation (AP) based clustering. For K-means based clustering and Affinity

Algorithm 2 Cluster number determination process

Input:

$X = \{x_1, x_2..., x_k\}$, a set of images corresponding to query expansion

1: Let C as the initial set of centers for data X

2: $C \leftarrow$ spectral clustering(C,X)

3: Let $\{x_i|class(x_i) = j\}$ as the set of datapoints assigned to center c_j

4: Use AD test to calculate if each $\{x_i|class(x_i) = j\}$ meets the Gaussian distribution at the confidence level α (0.0001)

5: If the data meets Gaussian distribution, keep c_j. Otherwise, split c_j with two centers.

6: Repeat from step 2 until no more centers are added.

Output:

Cluster number k and corresponding images

Propagation, we use the same feature and relevant clusters choosing mechanism. The difference is in the process of cluster number generating. Table 5 shows the loss of relevant clusters and filtered less-relevant clusters.

3.3.3. Progressively CNN based noisy images filtering

By observing the experimental results, we found that clustering based images filtering method can not filter the *Type 2* noisy images efficiently. In order to further purify the image dataset, we take a purifying method similar to [24]. The difference is that we do not train a CNN model from the beginning, instead, we directly fine-tune a CNN model using filtered images on a trained model "$bvlc\_reference\_caffenet$" [25]. Then all of the filtered images are used to do image classification using the fine-tuned model. We

Table 5: The loss of relevant clusters and filtered less-relevant clusters

Method	K-means	AP	Spectral clustering
Loss of relevant clusters	16.4%	20.8%	12.3%
Filtered less-relevant clusters	78.6%	62.5%	83.4%

take the probabilistic sampling algorithm to select the new training sample images according to the classification scores on the training data itself. The intuition is we want to keep images with distinct sentiment scores between classes with high probability. We use the new selected sample images to further fine-tune the previous model, repeat the above steps until reach the pre-set iteration value (in our experiment is 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 the new selected training instance with probability $P(i)$ given by:

$$P(i) = 1 - max(0, 2 - exp(|V_{i1} - V_{i2}|)) \tag{8}$$

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. The process of filtering noisy images are shown in Algorithm 3. In our experiments, we found that the *Type 2* noisy images can be effectively filtered through this CNN method. The reason for this is that after the filtering process, the number of noisy images in the raw image dataset are relatively small compared with the whole image dataset for the particular target query.

Algorithm 3 Noisy images filtering algorithm

Input:

$X = \{I_1, I_2..., I_k\}$, a set of images corresponding to query expansion

1: Filter artificial images (*Type 1*) from X using the color histogram features + SVM framework

2: Filter *Type 3* noisy images from X using clustering based filtering method

3: Fine-tune a learned CNN model with filtered images

4: Calculate *Scores* as the sentiment scores for image I_i using fine-tuned model

5: **for** $I_i \subset X$ do

6: Keep I_i as new training samples with probability *P(i)*

7: **end for**

8: Repeat from step 3 until reaching the pre-set iteration numbers

Output:

Return the remaining images $X^{'} \subset X$ as the final filtered images

4. Experiments

In this section, several state-of-the-art methods are compared with our methods in the process of noisy query expansions filtering and noisy images filtering. We choose ten of twenty categories in PASCAL VOC 2007 dataset as the target queries, then we do the query expanding, noisy query expansions filtering and noisy images filtering to construct the image dataset AutoImgSet-10. Table 6 shows the detailed scale for each query in AutoImgSet-10.

Table 6: 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

4.1. Performance evaluation on 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 nearest query expansions by comparing it with method [16]. The source code for method [16] is obtained from the author's website: http://nlp.stanford.edu/projects/glove/.

For ten queries, we randomly select 100 query expansions from each query (except query *motorcycle*). Table 7 shows the top 8 similar query expansions found by our method and [16] for each query.

Our method consider not only the text semantic similarity (Normalized Google Distance), but also the visual semantic similarity (Euclidean distance) in the process of similarity computing. We take the performance metric proposed by [26] to do the comparison:

$$acc@K = \frac{\sum_{k=1}^{K} 1\{(w_k, w)\}}{K} \tag{9}$$

where $1\{\cdot\}$ is an indicator function, so that is equals to 1 if (w_k, w) is visually similar word pair and 0 otherwise. Obviously, our method achieves

Table 7: Top-8 similar query expansions found by our method and method [16].

Query	Found by our method	Found by method [16]
Horse	canter/walking/eating/running horse rear/young/brown/black horse	eating/young/sea/black horse grey/bet/sawing horse, horse stable
Train	fast/modern/underground/metra train powered/tourist/long/electromagnetic train	fast/evening/underground train, train station, group/court/sweeping/overcrowded train,
Sheep	grazing/marsh/australian/goat sheep breeding/desert/eating/fat sheep	grazing/goat/fat/gold sheep game/breeding/market/cooked sheep
Motorcycle	honda/red/police/driving motorcycle white/fast/expensive/yamaha motorcycle	racing/first/police motorcycle, motorcycle hat japanese/honda/driving/fast motorcycle
Dog	angry/wolf/fighting/home dog wild/hunt/smiling/little dog	wolf/fighting/toys/paper dog cartoon/pet/sitting/rough dog
Cow	black/donor/young/bull cow walking/milk/lying/breeding cow	black/donor/milk/breeding cow cow milk, silver/jumping cow, cow steller
Cat	brown/little/wild/rabbit cat bronze/napping/sand/funny cat	missing/angry/funny/sand cat tom/napping/brown cat, cat tails
Bus	public/city/travels/tour bus school/touring/airport/dual bus	city/travels/system/general bus school/fast/monitor/airport bus
Bird	swallow/flight/black/seagull bird swan/silver/eagle/small bird	sing/seagull/small/bee bird bird nest, bat/flapping/angry bird
Airplane	737/france/a320/united airplane p3/british/airbus/flying airplane	jet/latest/war/f16 airplane 737/france/british/airbus airplane

a higher precision. The reason is that we filter noisy query expansions with combined text semantic distance and visual semantic distance which is much more efficient than just using context constraints. Thus our method is more suitable to expand queries for image dataset construction. Fig.3 shows the average accuracy of Top-K query expansions for method [16] and ours.

4.2. Performance evaluation on image dataset precision

Due to both of the size and the species included in the datasets are different, we can't directly compare the precision of a particular category. Instead, we compare the average precision of the constructed dataset with three other automatic methods [5][8][15] in Fig.4. [5] use an iterative framework that simultaneously collect object image datasets. The framework use Bayesian incremental learning as its theoretical base. The disadvantage of this ap-

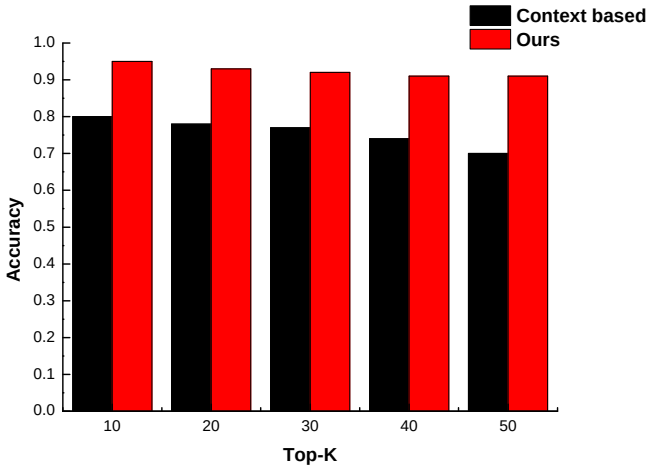


Figure 3: Average accuracy of Top-K similar query expansions.

proach is overly dependent on the accuracy of the initial training data. [8] take text information to re-rank images retrieved from web search and use these top-ranked images to learn visual models to re-rank images once again. [15] use clustering based method to filter noisy "group" images and propagation based method to filter relatively small noisy images. However, the accuracy of image dataset constructed by all of these three methods is still unsatisfactory. The main reason is the raw image dataset constructed by only one query contains too much noise. Our method has a higher precision than previous methods mainly because we use multiple query expansions and the high accuracy of ranking forward images returned from image search engine in the process of dataset construction.

4.3. Performance evaluation on object detection ability

To compare the object detection ability with three other state-of-the-art baseline methods [9][10][11], we train the DPM detector for object detection.

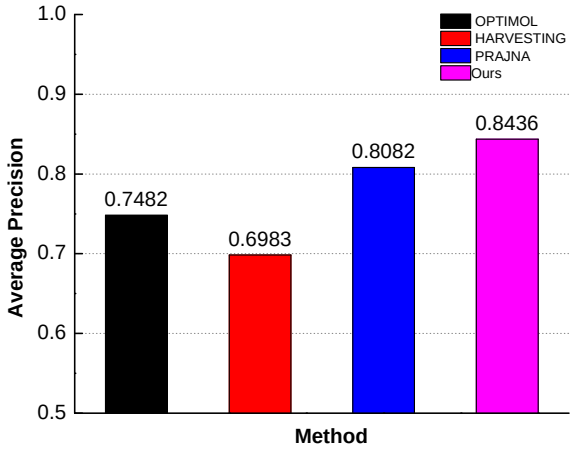


Figure 4: Comparison of the average precision of the four methods of dataset construction.

The code for DPM is obtained from the author's website: http://www.cs.berkeley.edu/rbg/latent/ [11]. We take the same image preprocessing methods: resize images to a maximum of 500 pixels, discard all near-duplicates, and ignore images with extreme aspect ratios (aspect ratio > 2.5 or < 0.4). We initialize our bounding box to a sub-image within the image that ignores the image boundaries. By doing this, we can avoid the two-stage training procedure used in [27]. Then the detector is trained using the default parameters of [11]. We evaluate the performance of our trained detection model for the 10 queries in the PASCAL VOC 2007 test set [28]. We pick this dataset as recent state-of-the-art weakly supervised methods have been evaluated on it. Table 8 displays the results using our own image dataset AutoImgSet-10 and compares them with state-of-the-art baselines [9][10][11].

Compared to [9] which uses weak supervision and [11] which uses full su-

Table 8: Results (Average Precision) on Pascal VOC 2007 (test) object detection

Method	Supervised	bird	train	cat	cow	dog	horse	sheep	plane	bus	mbike	average
[9]	weak	3.1	**34.2**	7.1	9.3	1.5	29.4	0.4	13.4	31.2	**38.3**	16.79
[10]	web	**12.5**	23.5	8.4	17.5	12.9	30.6	**18.8**	14.0	35	27.5	16.62
our	**web**	12.3	25.5	**10.7**	**18.7**	**14.2**	**32.7**	15.3	**14.6**	**36.7**	28.5	**20.92**
[11]	full	10.3	45.2	22.5	24.3	12.6	56.5	20.9	33.2	52.0	48.5	32.6

pervision, our newly proposed method performs better as even the training set does not need to be labelled manually. Nonetheless, our results substantially surpass the previous best results in weakly supervised object detection. Compared to [10] which also uses web supervision, our method surpasses their results in most of the cases. The main reason for this is that our training data generated from the Internet contains much richer and accurate visual descriptions in images. By observing the binding data in Table 3 and Table 8, we found that those concepts which have good performance tend to have sufficient and accurate query expansions for the query. In other words, our approach discovers query expansions that have much more useful linkages to the visual patterns in the corresponding image set.

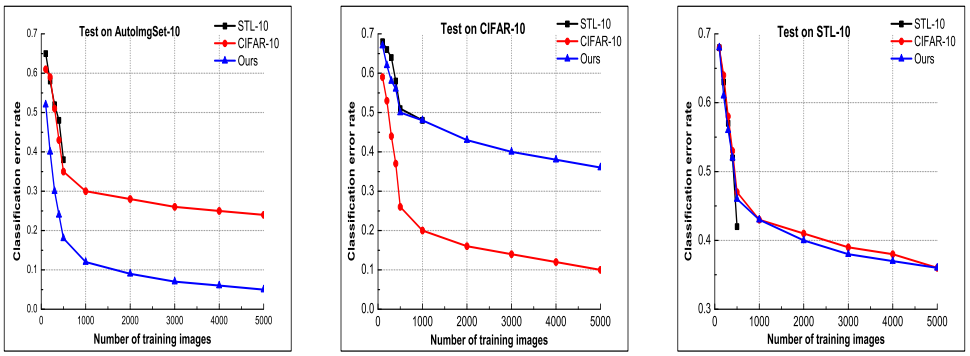
4.4. Performance evaluation on cross-dataset generalization ability

We compare our dataset 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 and tested on the other dataset [29]. It indicates the efficiency and robustness of our proposed method for clean image dataset construction.

To be fair, we only choose the five same categories (horse, bird, airplane, cat and dog) which are included in three datasets to verify their cross-dataset

generalization ability with STL-10 and CIFAR-10. Specifically, we randomly select 500 training images and 500 testing images for each category in STL-10, CIFAR-10 and our dataset (because the maximum number of training data in STL-10 is 500). We resize all images to $[64, 64]$ and convert all images to grayscale images. When training the image classification model, we set the same options for three datasets. Setting the type of SVM to be C-SVC, the type of kernel to be radial basis function and all other options to be the default LIBSVM options [22]. Then datasets are used to learn the image classification model based on same feature (HOG[30]) and learning method (SVM). We use the learned model to do image classification on these three image datasets. The results are shown in Fig. 5.

In two of three cases, with the same number of training images, the best performance of classification is achieved by using the dataset AutoImgSet-10. Since the dataset STL-10 has only 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 much 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 labelled. In addition, the image datasets STL-10 and CIFAR-10 constructed by one query for image collection tend to have the domain adaptation problem [31]. However, our method using multiple query expansions can effectively ease the domain adaptation problem.



(a) (b) (c)

Figure 5: 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.

5. Conclusion and future work

In this work, we presented a new framework for automatic web-supervised image dataset construction. Three successive modules were employed in the framework including query expanding, noisy expansions filtering and noisy images filtering. To verify the effectiveness of our proposed method, we construct an 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 [5], [8] and [15]. Besides, we evaluate the object detection ability with methods [9] [10][11]. The results shows our method can surpass [9][10] in most of the cases. Finally, we evaluate the cross-dataset generation ability of our dataset with manually labeled dataset STL-10 and CIFAR-10, it shows our image 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 multiple query expansions in the process of constructing 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] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.

[2] W.-H. Lin, R. Jin, A. Hauptmann, Web image retrieval re-ranking with relevance model, in: Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on, IEEE, 2003, pp. 242–248.

[3] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from google's image search, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2, IEEE, 2005, pp. 1816–1823.

[4] S. Vijayanarasimhan, K. Grauman, Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[5] L.-J. Li, L. Fei-Fei, Optimol: automatic online picture collection via incremental model learning, International journal of computer vision 88 (2) (2010) 147–168.

[6] T. Hofmann, Probabilistic latent semantic analysis, in: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[7] T. L. Berg, D. Forsyth, et al., Animals on the web, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006, pp. 1463–1470.

[8] F. Schroff, A. Criminisi, A. Zisserman, Harvesting image databases from the web, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (4) (2011) 754–766.

[9] P. Siva, T. Xiang, Weakly supervised object detector learning with model drift detection, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 343–350.

[10] S. K. Divvala, A. Farhadi, C. Guestrin, Learning everything about anything: Webly-supervised visual concept learning, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 3270–3277.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1627–1645.

[12] B. Collins, J. Deng, K. Li, L. Fei-Fei, Towards scalable dataset construction: An active learning approach, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 86–98.

[13] B. Siddiquie, A. Gupta, Beyond active noun tagging: Modeling contextual interactions for multi-class active learning, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2979–2986.

[14] S. Vijayanarasimhan, K. Grauman, Large-scale live active learning: Training object detectors with crawled data and crowds, International Journal of Computer Vision 108 (1-2) (2014) 97–114.

[15] X.-S. Hua, J. Li, Prajna: Towards recognizing whatever you want from images without image labeling, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[16] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12 (2014) 1532–1543.

[17] J. Weston, S. Bengio, N. Usunier, Wsabie: Scaling up to large vocabulary image annotation, in: IJCAI, Vol. 11, 2011, pp. 2764–2770.

[18] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.

[19] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (11) (1995) 39–41.

[20] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, S. Petrov, Syntactic annotations for the google books ngram corpus, in: Proceedings of the ACL 2012 system demonstrations, Association for Computational Linguistics, 2012, pp. 169–174.

[21] R. L. Cilibrasi, P. M. Vitanyi, The google similarity distance, Knowledge and Data Engineering, IEEE Transactions on 19 (3) (2007) 370–383.

[22] C.-C. Chang, C.-J. Lin, Libsvm: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (3) (2011) 27.

[23] C. Sinclair, B. Spurr, M. Ahmad, Modified anderson darling test, Communications in Statistics-Theory and Methods 19 (10) (1990) 3677–3686.

[24] Q. You, J. Luo, H. Jin, J. Yang, Robust image sentiment analysis using progressively trained and domain transferred deep networks, in: The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI), 2015.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.

[26] Y. Bai, K. Yang, W. Yu, C. Xu, W.-Y. Ma, T. Zhao, Automatic image dataset construction from click-through logs using deep neural network, in: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM, 2015, pp. 441–450.

[27] M. Pandey, S. Lazebnik, Scene recognition and weakly supervised object localization with deformable part-based models, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1307–1314.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[29] A. Torralba, A. Efros, et al., Unbiased look at dataset bias, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1521–1528.

[30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886–893.

[31] A. Bergamo, L. Torresani, Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach, in: Advances in Neural Information Processing Systems, 2010, pp. 181–189.