Automatic Collecting Representative Logo Images from the Internet

Xiaobing Liu
State Key Laboratory for Intelligent Technology and System, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Bo Zhang
State Key Laboratory for Intelligent Technology and System, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Follow this and additional works at: https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology

Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Tsinghua Science and Technology by an authorized editor of Tsinghua University Press: Journals Publishing.
Automatic Collecting Representative Logo Images from the Internet

Xiaobing Liu and Bo Zhang*

Abstract: With the explosive growth of commercial logos, high quality logo images are needed for training logo detection or recognition systems, especially for famous logos or new commercial brands. This paper focuses on automatic collecting representative logo images from the internet without any human labeling or seed images. We propose multiple dictionary invariant sparse coding to solve this problem. This work can automatically provide prototypes, representative images, or weak labeled training images for logo detection, logo recognition, trademark infringement detection, brand protection, and ad-targeting. The experiment results show that our method increases the mean average precision for 25 types of logos to 80.07% whereas the original search engine results only have 32% representative logo images. The top images collected by our method are accurate and reliable enough for practical applications in the future.

Key words: logo image; sparse coding; scale invariant; shift invariant; multiple dictionary

1 Introduction

Logos have great commercial and social value. There are numerous types of logos all over the world. Logo detection[1,2] and recognition[3-5] techniques are used in applications, such as trademark infringement and brand protection. They are also useful for mobile phone applications. For example, imagine you see a Starbucks on the other side of the street and capture a photograph of the logo on your cell phone. In a second, the Starbucks logo will be detected and the menu will be listed on your phone. Thus, these logo detection techniques are important for these mobile applications. Detecting, removing or inserting a channel logo has also become useful for web videos and IPTV in recent years[6,7]. Logo indexing is another valuable source to support more services. Ad-targeting can be more accurate when a trademark logo is detected in a picture or a video.

Logo detection or recognition systems require high quality logo images for training. (a) These training images should be representative enough to avoid biased training. (Otherwise, for instance, if you train a Starbucks detector using coffee cups with Starbucks logo, other coffee cups may give false alarms with this detector.) (b) These images require accurate labeling or human labeling. (c) New logos have to be updated frequently into the logo dataset to keep up with the explosive growth of new commercial brands. Human collection and labeling cannot keep up with these requirements. This study aims to automatically collect representative logos from the internet without any human labeling. Only the logo name is required as input.

A representative logo image is an image which contains only one logo with no other objects or cluttered background. For instance, Fig. 1 shows some example images found by searching for Starbucks in the Google image search engine. Only the top left image is a representative image (positive sample) for Starbucks. All the others are negative samples. All these images were selected from the search results of the
Google image search engine which returns not only the representative logos of Starbucks, but also outside and inside views of Starbucks shops, coffee cups, and fake logos. Therefore, collecting representative logo image is not easy.

Image search engines are still not accurate enough to collect representative logos, although they can return many relevant images for the logo name. Most public search engines rely on the file name or surrounding text information to retrieve the images. Just returning relevant images is enough for a query like Starbucks, but many of these images are not representative. Statistics on the 25 typical logos (Refer to Section 4 for more details) show that only about 32% of the images were representative in search engine results. The precision may be even worse for some other rare brands or ambiguous logo names (e.g., Washington, Jordan, Lincoln, Bobcat, Transformer, Sun, GPA, KDD, SVM, BTV, and QQ). Therefore, the images returned by the search engines must be filtered to get representative logos.

In this work, we propose Multiple Dictionary Invariant Sparse Coding (MDISC) algorithm to collect representative logo images from the internet. Figure 2 gives an overview of this approach. Only the logo name is required as input to this framework. Neither an initial image nor human labeling is needed. The algorithm outputs a ranking list of representative images. An image search engine converts the logo name to a set of candidate logo images. Then, Scale And Shift Invariant Sparse Coding (SASISC)\cite{8} is used to reconstruct each candidate image with a dictionary built from other candidate images. This coding can handle the scale and shift variances between different versions of the same logo. The reconstruction coefficients represent the relationships between each candidate image and the dictionary elements. Multiple Dictionary Voting (MDV) is proposed to analysis the candidate images and calculate the rank scores using their reconstruction coefficients. The performance of this approach is evaluated using 25 world-famous logos. The results show that this approach can automatically collect logo images and accurately rank the representative logos at top of the result lists.

2 Related Works

Object dataset collection from the internet is related to the work in this paper. Most of the previous works focus on collecting images for generic object recognition\cite{9-16}. They initially build a model for each category with the search result images or seed images. Then they recursively update the model and the collected image dataset. However, these methods have rarely used for automatic logo image collection. In this paper, we focus on logo images but not generic objects or scenes. Representative logo images are more valuable than generic object images in commercial applications, and logo images have many specific properties. (a) Color feature is important for logos, but is not essential for generic object categories (e.g., cars, airplanes, and bikes). (b) The spatial configuration is fixed in logos, but is variable for many object classes (e.g., grass, sky, sea, and flowers). (c) Logos have few viewpoint changes, while common objects are often captured with different viewpoints. Thus, a color pixel based representation is sufficient for this task. Many other complex techniques, such as the bags-of-words representation and SIFT-like local features\cite{17} are used.
to tolerate differences between objects in the same categories, but they may also allow noise negative samples, especially fake logo images (e.g., the top right image in Fig. 1). Therefore, only using RGB pixel representation is needed here for each image with more attention paid to deal with the scale and position variance and to rank the candidate images.

Object classification has also been an important topic in recent years\(^\text{[18-22]}\). Logo recognition and detection has also attracted much attention\(^\text{[1-5]}\). However this work significantly differs from generic object recognition or logo detections, since human labeling is not used to identify positive and negative samples. This automatic method is accurate enough to collect representative logo images for training logo detection models.

Image retrieval from the web is also popular with long history in multimedia applications\(^\text{[23-28]}\). Some studies also have investigated on logo retrieval\(^\text{[29, 30]}\). From image retrieval point of view, this paper provides high quality image retrieval for logos. Since logo has special properties on color and spatial structure, these properties are deliberately considered in this work. Moreover, a highly ranked negative sample is acceptable for the image retrieval since the user has other options on the first page. A wrong labeled logo image can be a disaster when used for training a detection model. However, since this approach aims to provide unsupervised collection of representative logo images, the algorithm is very concerned about accuracy.

3 Multiple Dictionary Invariant Sparse Coding

MDISC contains (a) an image search engine, (b) SASISC, and (c) MDV as illustrated in Fig. 2. At the first step, we directly type the logo name (e.g., Starbucks) into the image search engine to collect the candidate image set from the top pages of the search results. SASISC handles the scale and shift variance of the logo images and returns the coefficient relationships between the candidate images and the dictionary bases. MDV then re-ranks the candidate images to find the most representative logos. This component filters out noisy images or un-representative logos. Thus the search engine converts a word (logo name) into images, SASISC converts the images into reconstruction relationships, and MDV converts relationships into ranking scores.

3.1 Scale and shift invariant sparse coding

Without any supervised information, the algorithm can only rely on the visual similarities or reconstruction relationships between the candidate images. These relationships are described here by sparse coding coefficients. Sparse coding or similar algorithms can reconstruct each candidate image from other candidate images. Thus we can directly use the candidate images as the dictionary for the sparse coding algorithm. Larger coefficients represent closer relationships between the target image and the corresponding basis image. There are many other options to compute the image similarity, such as color features, texture statistics, shape histograms, and histogram of local features. However most of these methods can only compute the matching similarity for the whole image, whereas logos often occur as a small part of a whole image. Sparse coding can be viewed as a decomposition algorithm when the coefficients represent the component level similarity. Thus reconstruction (sparse coding) algorithms are used here rather than other feature-based methods.

In the sparse coding framework, an image can be viewed as a linear combination of several bases and their corresponding coefficients. The RGB pixel value is directly used as one unit of the image. Thus, each image \(X\) or each basis patch \(B_t\) (the \(t\)-th element of the dictionary) is represented as a tensor or 3-layer matrix. The three axes of the tensor represent the image height, image width, and RGB channels.

In many previous studies\(^\text{[31, 32]}\), the sparse coding algorithm must have the target images and the bases share the same predefined size, such as 10 \(\times\) 10 pixels. Large images have to be split into the predefined size patches, as shown in the top part of Fig. 3. However.
this splitting strategy destroys the spatial configurations of the image. A few pixels shift or scale changes do not significantly change the image content, but these small graphical changes can completely alter the sparse coding results. If the bases are used as the common patches for the target image, sparse coding with such splitting cannot reasonably decompose the target image into meaningful components.

SASISC\cite{SASISC} is used to more effectively preserve the spatial properties of logo images to better reveal the relationships between candidate images. SASISC can analyze large size target images and different size bases. A basis can be scaled and placed anywhere in the target image. The bottom part of Fig. 3 shows how the SASISC method reconstructs an image from the given bases. With both types of sparse coding, an image is approximately reconstructed as the linear combination of these components. The typical sparse coding method (at the top of the figure) only allows the same size images and bases, so a large image needs to be split into small patches and reconstructed. However, the SASISC dictionary contains different size bases, as well as their scaled copies. When reconstructing a large image, a dictionary component can occur at any position with the proper scale.

The reconstruction coefficients are computed by minimizing the SASISC loss function as

$$\min \alpha \| X - \sum_{t,s,z} \alpha_{t,s,z} \Phi(B_t, s, z) \|_2^2 + \mu \sum_{t,s,z} |\alpha_{t,s,z}|$$

The image $X$ is then reconstructed by a linear combination of the basis features $B = \{B_t\}$ with their coefficients $\alpha = \{\alpha_{t,s,z}\}$. Here, $t$ is the basis index which ranges from 1 to $T$, where $T$ is the number of bases. The real value $s$ is the scale ratio, with $s$ having 14 scale levels in this algorithm with 0.03 scale steps between levels. $z$ is the pixel-level location of the scaled basis patch in the image. Therefore, the coefficient set covers all the conditions for any basis occurring at any place with any scale. Denote $\Phi(B_t, s, z)$ as the transformation which scales the basis $B_t$ with the scale rate of $s$ and then pastes that patch at location $z$ on an empty image with size $X$. The coefficients are sparse due to the L1 penalty. $\mu$ is the coefficient regularization weight.

Actually, the loss function in Eq. (1) is very similar to that of sparse coding. Formula (1) is the L1 regularized least squares representation of the image, which is convex over the coefficients $\alpha$. For each $\alpha_{t,s,z}, X_t = \Phi(B_t, s, z)$ is a basis in the original sparse coding framework. Thus the mathematical properties of SASISC and sparse coding are very similar. However, there are substantial differences between these two methods when analyzing. The original sparse coding pastes the basis after splitting the entire target image, whereas SASISC can place a basis at any position with any scale in a large image. Overlap is also allowed for the pasted bases. Sparse regularization then pushes most of the coefficients to be zero. This property pushes all the bases to choose their proper positions and scales in the large target image.

SASISC has many good properties as an important component in the MDISC framework. SASISC can tolerate shift and scale variance, which are very common in logo images, and is the reason for not using Shift Invariant Sparse Coding (SISC)\cite{SISC} or the Sparse Coding (SC)\cite{SC}. (SISC cannot support scale invariance, and SC supports neither scale invariance nor shift invariance.) SASISC cannot handle rotations, but rotations are not significant for most logo images. SASISC can handle different size bases. So it is suitable for logo images from the internet that may have various aspect ratios. Thus, SASISC can handle all the graphical properties of logo images, including scale, shift, aspect ratio, and spatial configuration. After the SASISC component, the image properties of the logos are converted to reconstruction relationships between the logo images.

The scale and shift transformations increase the computational complexity, but there are some specific techniques to speed up the SASISC algorithms. For each basis patch, the algorithm pre-computes a scale pyramid to cover several scale levels and avoid repeated scaling. FFT-based (Fast Fourier Transform) or GPU-based convolution operations can also speed up the computations of the shift factor. Moreover, the coefficients are very sparse, with an average of only 51.2 non-zero reconstruction coefficients ($\alpha_{t,s,z}$) for each image. Moreover, the feature-sign search algorithm\cite{algorithm} is used to speed up the coefficients calculations. With these techniques, an image can be processed in one second. The computational complexity is acceptable, since the calculations are off-line, and precision is more important than speed in automatic logo collection.
3.2 Multiple dictionary voting

We propose MDV method to compute the rank scores for all the candidate images from the SASISC reconstruction results. Some of the candidate images returned by the image search engine are randomly selected to serve as sparse coding bases. These bases are divided into several subsets with each subset called a dictionary for SASISC.

Figure 4 illustrates an example when using 2 dictionaries with 4 bases per dictionary on the left side. The candidate images are ranked on the right side by their ranking scores. The middle column shows the coefficient histograms which reconstruct the candidate images from the basis dictionaries.

We make an assumption that each dictionary should have at least one representative logo image basis. Therefore a logo image should be well-reconstructed by the dictionary with the majority of the coefficients concentrated on the representative logo bases. For a non-representative logo image, the coefficients will be uniformly distributed on all the bases. For a dataset containing both types of images, the large coefficients will have a higher probability of representing the actual logo bases.

The SASISC algorithm computes the coefficients $\alpha_{k,t,s,z}$ for each candidate image and each dictionary, where $\alpha_{k,t,s,z}$ is the coefficient of basis $B_t$ with scale ratio $s$ and occurring at position $z$ of image $X_k$. These coefficients decide a binary graph between the candidate images and the basis images. There are many algorithms for this type of binary graph, however, few of the algorithms are appropriate for this ranking task. Thus a method was developed to compute the rank scores for all the candidate images.

The coefficients corresponding to the same basis at different positions and scales are summed and normalized as

$$w_{k,i} = \frac{\sum_{t \in \text{dict}(i)} |\alpha_{k,t,s,z}|}{\sum_{t \in \text{dict}(i)} \sum_{s,z} |\alpha_{k,t,s,z}|}$$  \hspace{1cm} (2)

where $i$ is the index of the bases and $\text{dict}(i)$ is the dictionary containing $B_i$. The normalization is taken over all the coefficients for each SASISC processing. Thus, $w_{k,i}$ is the normalized coefficients of the basis $B_i$ over image $X_k$.

The voting score, $h_i$, of basis $B_i$ is the powered sum of the normalized coefficients from all the candidate images:

$$h_i = \left( \sum_k w_{k,i} \right)^\gamma$$  \hspace{1cm} (3)

The basis vote parameter $\gamma$ is a positive real number greater than one. Most of the small coefficients will be filtered out by this operation. Therefore, a large $h_i$ basis requires many large coefficients from different images. Representative bases more easily have large $h_i$.

The final ranking score, $f_k$, of the candidate image $X_k$ is the weighted sum of all the relevant basis score:

$$f_k = \sum_i w_{k,i} h_i$$  \hspace{1cm} (4)

where the weights are the normalized coefficients. Thus, an image reconstructed from many representative bases with large $h_i$ and large normalized coefficient $w_{k,i}$, has a higher probability to be a representative logo image.

Multiple dictionaries are used to get more stable voting results than only one dictionary as described in Table 1. With only one dictionary with many bases, the sparse coding generates only a small fraction of non-zero coefficients. Since the candidate images are collected from the top pages of the noisy search

![Fig. 4 Illustration of MDV (Best viewed in color).](image)

Table 1: MAP for different numbers of dictionaries for 25 logos. The total numbers of bases are similar.

<table>
<thead>
<tr>
<th>Number of dictionaries</th>
<th>Number of bases per dictionary</th>
<th>Number of bases in total</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>50</td>
<td>0.5856</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>51</td>
<td>0.7132</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>50</td>
<td>0.8007</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>49</td>
<td>0.7641</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>50</td>
<td>0.7728</td>
</tr>
</tbody>
</table>

Note: MAP is the mean average precision.
results, even non-representative images may be related to similar images or even duplicate copies. Non-representative logo images may then have large coefficients from accidentally similar bases. However, with multiple small dictionaries, non-representative images are less likely to get support from all the dictionaries. Representative logos are then more likely to rank high in the candidate set. This strategy further reduces the influence of accidental voting from a weird dictionary of candidate images and reduces the computational complexity of the sparse coding calculation. MDV can also be generalized to automatically collect representative samples of other types of images (e.g., objects or scenes), especially unlabeled images from search results.

The dictionary size (the number of bases in the dictionary) is crucial to accurate results in this method. If the dictionary is too small, the dictionary may not contain any representative logo bases and will provide little evidence to discriminate logo and none-logo images. If the dictionary is too large, perhaps with duplicated bases of none-logo candidates, the voting algorithm will mistakenly give a high confidence value to none-logo bases. Thus, the dictionary size should be set according to the positive sample percentage of candidate images. For example, the positive sample percentage was about 30% in this work, and the dictionary size was set as 10. The probability of all 10 bases being negative images is then $(1 - 30\%)^{10} \approx 0.02825$. Therefore, the probability is about 97% that a dictionary has at least one representative logo basis.

The MDV output is the ranking scores for each candidate image. If a threshold is used to select the top part of the ranking list, the method will automatically output a set of representative images or binary predictions for each image. Precision and recall are good measures with this type of results. However, the threshold-selection operation is very sensitive to the threshold. Also, it is hard to compare two collection methods, because the thresholds may be unfair for different methods. The ranking scores are more useful since they contain more information than a binary label output. Thus, a ranking list will provide a better comparison of different methods than binary predictions. Knowing that an image is more representative than another is enough for most applications. Therefore, the ranking score is a better indicator than binary predictions, although it is not very intuitive. In this paper, we adopt ranking score as the output format of the MDISC framework.

4 Experiment Results

4.1 Dataset and evaluation

The name of each logo was entered into the Google and Baidu image search engines to begin the process. Then about 100 candidate images were downloaded from the top search result pages. 25 famous logos were used in our experiments (Adobe, Apple, AT&T, BMW, CNN, DHL, Facebook, Google, HP, IBM, Intel, MasterCard, McDonalds, Microsoft, NBA, NBC, Nike, Nokia, Olympic, Puma, Starbucks, Superman, Target, Yahoo, and YouTube) with 2855 candidate images collected for the evaluations in total.

The method was evaluated by manually labeling all the candidate images, but these labels were not used for training. Since the objective is to collect the most representative images for the logo, only images with large, correct logos were labeled as positive samples. All the other images were labeled as negative samples, such as images with other objects, clutter background, commercial products or fake structures. Statistically, only 31.2% of the images were positive samples. Therefore, the task is very difficult.

The rank order information of the search engine result images was not used, and no query expansion techniques were used to obtain high precision candidate image sets. The objective was to avoid the effects of specific search engine settings, including (a) the specific image search engine, (b) the precision of the search engine results, (c) the performance of easy task, and (d) the specific query expansion technique. Not all the image search engines provided high quality results, and not all the logos were famous enough to have numerous high-quality images ranked in the top pages. Thus, the method was evaluated on a difficult, noisy candidate image set. If the method can perform well in the difficult environment, it will certainly perform better with a stronger image search engine on an easier task.

Average Precision (AP) was used as the evaluation measure.

$$\text{AP} = \frac{1}{n} \sum_{k=1}^{n} \frac{k}{\text{rank}(k)} \quad (5)$$

where $n$ is the number of positive samples and rank$(k)$ is the rank of the $k$-th positive sample from the ranking list. Average precision is an approximation of the integrated area under the precision-recall curve. If our
method has high quality ranking list of the candidate images, then high quality representative image subsets can be easily obtained by setting the proper threshold with the subset having good precision and recall.

This measure focuses on the rank order and the top results, with the top results being most important in this application. If the top results are accurate and stable, they can be used as the most representative images for the logo. A highly ranked negative sample will dramatically reduce the average precision, so the average precision is a strict measure for this task.

4.2 Parameter settings

For most of the experiments in this paper, the parameters were set as follows. The basis vote parameter, $\gamma$, was 8. We used 5 dictionaries with 10 bases per dictionary. The coefficient loss weight, $\mu$, was 0.1 since this value gives an absolute value of the reconstructed image that was similar to the original image. The SASISC scale level was 14. But the scale level was 1 for the original sparse coding or SISC methods. All the downloaded images were resized to $50 \times 50$ pixels keeping the original width-height ratio and setting the side margins as white pixels.

The influence of the basis vote parameter is evaluated in Fig. 5. The performance is stable for basis vote parameters larger than 6 with the best result achieved for a basis vote parameter of 8.

Table 1 lists the results for different numbers of dictionaries with similar total numbers of bases. The results show that multiple dictionary are better than single dictionary, and that the best number of dictionaries should not be too large or too small. Fewer dictionaries reduce the ranking score stability while more dictionaries with fewer bases per dictionary break the assumption that each dictionary has at least one representative logo basis.

Table 2 shows the results for different numbers of dictionaries with similar numbers of bases per dictionary. Here, the MAP increases with the number of dictionaries. The best results use 5 basis dictionaries with 10 images per dictionary. This setting is used in the following experiments.

4.3 Comparisons

Table 3 shows the average results for the 25 logo images. Five methods are compared here. Random ranking ranks the candidate images using a random permutation. LossRate (SASISC+LossRate) ranks the candidate images by the value of the loss function in Formula (1). The last three methods use the MDV method for the ranking score with the representation part using the classical SC method, the SISC method, and the SAS ISC method.

These results show that the MDISC (MDV+SASISC) method has a significantly higher MAP of 0.8007 which significantly outperforms the other methods. The random ranking result (randomly ranking the candidate images) has a MAP of only 0.3233, which indicates the difficulty of this task. Actually about 68% of the top-ranked images returned by the search engine were negative samples (31% unrepresentative logo images and 37% none-logo images). Without any supervised training, the MDISC (SASISC+MDV) accurately re-ranked the logo candidates. The shift invariance (SISC+MDV), increased MAP by 136%.

Table 2 Mean average precision for different numbers of dictionaries for 25 logos. The numbers of bases in each dictionary are the same.

<table>
<thead>
<tr>
<th>Number of dictionaries</th>
<th>Number of bases per dictionary</th>
<th>Number of bases in total</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0.6671</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>0.7552</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>30</td>
<td>0.7726</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>40</td>
<td>0.7897</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>50</td>
<td>0.8007</td>
</tr>
</tbody>
</table>

Table 3 Mean average precision for logo image ranking.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random ranking</td>
<td>0.3233</td>
</tr>
<tr>
<td>SASISC + LossRate</td>
<td>0.4224</td>
</tr>
<tr>
<td>SC + MDV</td>
<td>0.4628</td>
</tr>
<tr>
<td>SISC + MDV</td>
<td>0.6311</td>
</tr>
<tr>
<td>SASISC + MDV</td>
<td>0.8007</td>
</tr>
</tbody>
</table>
over the original sparse coding with the MDV strategy (SC+MDV). The scale invariance (SASISC+MDV) increased MAP by an additional 126%. Thus, shift and scale invariance are crucial for the sparse coding part in this framework to properly represent the images before ranking. The experiment also shows that ranking the images based on their relative reconstruction loss (SASISC+loss rate) is significantly worse than the MDV method, which implies that the reconstruction relationships have more information than the value of the loss function in this task.

Detailed precision results for different rank positions are shown in Fig. 6. Each point on the line is the mean precision over all 25 logos at the specific rank position. Although MDISC (SASISC+MDV) is unsupervised, the precision for the first rank results is 100% for 25 logos. And the precision for the top 10 results is 91.2%. Thus, the top result images ranked by MDISC can be used as accurate representative images of the logo.

Figure 7 illustrates the average precision of MDISC for each category of logos. The IBM logo has the highest average precision of 97.3%, while the Facebook logo has the lowest average precision of 36.1%. Seven of these logos have precision higher than 90%. For all the logos, the MDISC method significantly outperforms the original results returned by the image search engine.

4.4 Top weighted dictionary basis images

Ten example images from the dictionaries for each logo are shown in Fig. 8. All these basis images were selected using the top ranked weights, $h_i$, computed by Eq. (3). A small fraction of these images are not representative logos, because they are common patches that occur in the search engine results. That is why the dictionary weight are not directly used as the final rank score for candidate images.

4.5 Top example images

The top images ranked by MDV with SASISC are illustrated in Fig. 9. Each row on the left side shows the top 15 images ranked by MDISC method for each logo from left to right. Most of these top images are very representative of the corresponding logo categories. The last 5 ranked images are also listed on the right to demonstrate some examples of negative samples. They are related to the logo, which is why they were returned by the image search engine, but many are not representative logos. These bottom images contain various types of negative samples such as products, fake images, and low-quality logo images. These show the difficulty of this unsupervised task, as well as the effectiveness of our method.

5 Conclusions

In this paper, we propose MDISC to automatically collect representative logo images from the internet without any manually labeled training examples. Although only one third of the images from the search engines are representative logos in the experiments, this method achieves a high average precision of 80.1%. All the first ranked images in the rank list of each logo category are representative. The
experiments show that MDISC accurately identifies representative logos without dependence on a high precision image search engine. The precision will be better with a more accurate image search engine or query expansions. The top images ranked by the proposed method are very representative, and can be used as prototypes or training images for logo detection, trademark infringement detection, and brand protection.

Future work will focus on applying this method to more types of logos and collecting more images for logo detection systems. For practical logo detection system trademark infringement detectors, the accuracy of the collected logos is crucial due to the high commercial value. This work is only a first step on this task. Further research will also focus on improving voting methods to achieve higher precision. The proposed method will also be extended to more general images such as generic objects or scenes by replacing the RGB pixel feature with SIFT-like local features or HMAX-like features\cite{17,36,37}.

Acknowledgements

This work was supported by the National Key Basic Research and Development (973) Program of China (Nos. 2012CB316301 and 2013CB329403), the National Natural Science Foundation of China (No. 91120011), Tsinghua University Initiative Scientific Research Program (No. 20121088071), and Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList). The authors thank Dong Wang, Jianmin Li, Jun Zhu, and Xiaolin Hu for helpful discussions.

References

Fig. 9  Top and bottom example images ranked by the MDISC for 25 types of logos.


with scale and shift invariant sparse coding, in Proc. of 18th International Conference on Image Processing (ICIP), Brussels, Belgium, 2011, pp. 1225-1228.


**Xiaobing Liu** received his BEng in computer science and technology from Tsinghua University in 2006. He is currently a PhD candidate in Department of Computer Science and Technology in Tsinghua University. His research interests include image processing, object recognition, and video retrieval. He has published 8 research papers on international conferences.

**Bo Zhang** is now a professor of Computer Science and Technology Department of Tsinghua University, the fellow of Chinese Academy of Sciences. In 1958 he received his BEng degree from Tsinghua University in 1958, and became a faculty member since then. From 1980/02 to 1982/02 he visited University of Illinois at Urbana-Champaign, USA as a scholar. He is now the chairman of steering committee of Research Institute of Information Technology, Tsinghua University, the technical advisor of Fujian government, and the member of Technical Advisory Board of Microsoft Research Asia. His research interests include artificial intelligent, granular computing, machine learning, and image and video retrieval.