Feature Fusion by Similarity Regression for Logo Retrieval

Fan Yang
University of Maryland College Park
fyang@umiacs.umd.edu

Mayank Bansal
SRI International
mayank.bansal@sri.com

Abstract

We propose a simple yet effective multi-feature fusion approach based on regression models for logo retrieval. Rather than fusing original features, we focus on similarities between pairs of images from multiple features, where only an annotation of similar/dissimilar pairs of images is needed. For each pair of images, a new vector is constructed by concatenating the similarities between the image pair from multiple features. A regression model is fitted on the new set of vectors with similar/dissimilar annotations as labels. Similarities from multiple features between the query and database images can then be converted to a new similarity score using the learned regression model. Initially retrieved database images are then re-ranked using the similarities predicted by the regression model. Logo class information from the training samples can also be included in the training process by learning an ensemble of regression models for individual logo classes. Extensive experiments on public logo datasets FlickrLogo32 and BelgaLogo demonstrate the effectiveness and superior generalization ability of our approach for fusing various features.

1. Introduction

Logo retrieval from a large dataset is an important topic in content-based image retrieval for various academic and commercial applications, such as logo and trademark detection, brand advertising and automatic logo annotation. The task of logo retrieval focuses on searching same/similar logos given a query at the *instance-level*, where different logos should be well discriminated, rather than at the *category-level*, where we only need to differentiate the logo class from other object categories, such as person, animal and scenes. Although the standard bag of words (BoW) approach [26] can be readily applied to this task, it is not robust enough due to the fact that logos are usually blurred due to camera motion or occupy only a small portion of the entire image. In these cases, only a limited number of or even no keypoints can be extracted, which makes the BoW approach vulnerable. Other factors, such as viewpoint change, rotation and distortion, make accurate logo retrieval more challenging.

Figure 1. Samples of *pepsi* and *apple* logos. Note that the *pepsi* logos exhibit various scale and rotational changes but the color distribution is relatively constant. In contrast, the *apple* logos exhibit varied colors, but consistent shape.

Nevertheless, a special property of logos which makes it different from other image retrieval problems is that they exhibit synthetic patterns and fixed or prominent color distributions. For example, as shown in Fig. 1, the *pepsi* logo has a distinct color distribution that is composed of blue, red and white, although it exhibits various scale and rotational changes. In this case, as a global feature, color is more powerful to capture higher level information compared to local features, which may help us locate the correct logos accurately and retrieve them effectively. Therefore, a single feature may not effectively handle all the different variations and thus combining multiple complementary features is a way to exploit the information that cannot be found by a single feature alone.

However, how to combine multiple features still remains an open question. Usually, to better capture distinctive local and global patterns of logos from a large collection of images, the dimensionality of feature vectors has to be extremely high. One has to use millions of visual words for constructing BoW vectors or tens of thousands of dimensions for Fisher Vectors (FV) [18] to obtain good performance. It is prohibitively expensive both to store all feature vectors for a database containing millions of images, as well as to learn weights from those features using any

Supported by NSF EAGER grant: IIS1359900, Scalable Video Retrieval, Office of Naval Research (ONR) MURI Grant N000141010934 and the Air Force Research Laboratory (AFRL) through contract number FA8750-12-C-0103. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution statement "A" (Approved for Public Release, Distribution Unlimited).

classifiers. In addition, due to the large variation of dimensionality among different features, it is even more challenging to determine the relative importance of individual features if they are simply concatenated, since the performance of the concatenated feature is prone to be dominated by high dimensional features. Furthermore, for retrieval tasks, we can only obtain a limited amount of labeled samples because manual annotation for millions of images is impractical, while the appearance of database images can be quite diverse. Moreover, we do not have any prior information of and cannot make any assumption on the characteristics of queries, which might be very different from the database images. Learning on a small set of annotated samples, which do not sufficiently represent the entire database and queries, is likely to generalize poorly.

We address the aforementioned problems by a simple yet effective regression-based framework for multi-feature fusion, which is only based on pairwise similarities between images and does not rely on original feature vectors. Given a training database with annotated similar/dissimilar pairs of images, we compute the similarities using individual features and convert them into a new data sample for each pair of images, of which each dimension represents the similarity corresponding to a specific feature type. A regression model is fitted based on the set of new (training) data samples to generate a weight vector, which describes the importance of individual features from the similarity perspective. For a query, we first obtain initial ranked lists of retrieved images using individual features, and then apply the learned regression model to the similarity vectors directly. The output of the regression model corresponds to the fused similarity which is then used for reranking the initial ranked lists. To further increase the discriminative ability of the regression model, we propose to utilize additional class label information, and fit a regression model for each logo class to obtain an ensemble of regression models. For each pair of images, multiple new similarities are calculated as outputs of the ensemble of regression models, which focuses on the difference between similar/dissimilar images at a finer level. The final similarity is further inferred by optimally combining these new ensemble-based similarities. Learning is done offline while the inference of new similarities is efficient and can be performed online in real time. Extensive experiments demonstrate that our regression-based multi-feature fusion approach is very robust against several unreliable features and is able to exploit the effectiveness of individual best-performing features. Moreover, it generalizes surprisingly well even when we use a relatively small training set and can handle heterogeneous features as well without any modifications.

2. Related Work

Logo retrieval. *Instance-level* object retrieval has been a popular topic for years. Various algorithms have been proposed and shown good performance on landmarks, scenes and generic objects [2, 17, 9, 19]. Logo and trademark retrieval has also been extensively studied in recent years. Joly and Buisson [11] proposed a contrario normalization of geometric consistency score for adaptively determining the threshold of matching scores used for spatial verification. In [14], a trademark and logo retrieval system was presented. The MSER detector is incorporated with DoG concept, so that it detects an interest region with both shape and orientation information preserved, on which a shape descriptor is further extracted. Retrieval is done by matching descriptors of queries to a collection of stable regions generated by a training set. Fu *et al*. [6] combined SIFT, shape and patch features with adaptive weights for logo retrieval. It assumed that top 5 to 10 retrieved images are correct and can be used to infer the importance of individual features, which is not well justified. [23] used shape context descriptors which are indexed by locality-sensitive hashing (LSH) to improve the speed of k-NN search for queries.

Since logos from the same class usually have stable geometric patterns, previous works also attempted to encode the spatial information particularly for logos. [12] proposed to use multi-scale Delaunay triangulation to encode spatial relationships of interest points close to each other, and represent them by signatures. Matching is performed by comparing the similarity between signatures of the query and database images. Bundling min-hash [21] was also proposed to group locally close keypoints and encode them using min-hash. A statistical model was learned in [20] to down-weight the scores of keypoints which are frequently matched in incorrect detections. More recently, [24] presented a logo recognition framework, where local features are grouped as constellations and matched by minimizing an energy function which considers the quality of feature matching and co-occurrence of features.

Multi-feature for retrieval. There is abundant work on fusing multiple features to improve retrieval performance. In [4], multiple attribute features are combined by averaging outputs of SVM classifiers. The score vector is then concatenated with Fisher Vectors after normalization and dimensionality reduction. Graph-based techniques are also widely used in the literature. [29] proposed a graph-based approach with relevance feedback to fuse multiple features for image retrieval. Weights of individual features are learned statistically from the retrieved results given a large set of queries, and thus this method is not flexible if we do not have any information of queries beforehand. [33] converted initial ranked lists by individual features to graphs and combined them together. Similarities between images are evaluated by Jaccard similarity, and graphs are equally summed up. Image attributes were used in [3] as labels to search for anchors in the graph which are further used

for graph alignment. A complicated multi-graph learning algorithm was also applied to learn a weight matrix from multiple graphs. All graph-based works require similarities between database images, which are not always available. Similarly, [34] also utilized attributes learned from a large dataset apart from the retrieval database. These attributes provide additional information to refine the inverted file that is originally constructed by SIFT visual words. Recently, [35] constructed a 2D indexing file using SIFT and color visual words. To our knowledge, there is no work on logo retrieval that fuses multiple features without relying on the inter-relationship between database images.

Multi-feature learning. Although numerous feature fusion algorithms are available, we limit our focus only to a few of them closely related to our work. Multi-kernel learning (MKL) [13, 8, 7] was widely used to find the optimal combination of kernels for image classification, where each feature type can be mapped to different kernels. Partial Least Squares (PLS) analysis [25] was applied to dimension reduction of a high dimensional vector formed by multiple feature vectors, which implicitly selects the most important features. Canonical Correlation Analysis (CCA) [28] was also effective to learn relationships of two sets of features. A hierarchical regression algorithm was proposed in [30] to exploit the information from individual features, where the manifold structure of different feature spaces is preserved. For cartoon image retrieval, [31] proposed a bi-distance metric learning algorithm to learn a distance metric from heterogeneous features. [32] decomposed multiple score matrices by multiple features as a low rank matrix plus feature-specific sparse errors. [5] proposed to learn logistic regression models with sparsity regularization to determine weights for visual words from multiple dictionaries for image classification. Note that most of these approaches deal with original feature vectors or require complicated optimization and matrix operations.

3. Multi-feature Fusion

3.1. Problem formulation

Given an image database consisting of N images, we can extract M visual features, each of which focuses on a specific aspect of images and thus are complementary with each other. Our aim is to effectively fuse these features to improve the retrieval performance. Denote the feature vector obtained by m-th feature for an image I_i as f_i^m[1], we obtain a set of feature vectors $\mathcal{F}^m = \{f_1^m, f_2^m, ..., f_n^m, ..., f_N^m\}$. On the other hand, for each image I_i, a set of feature vectors $\mathcal{F}_i = \{f_i^1, f_i^2, ..., f_i^m, ..., f_i^M\}$ are used to describe the image from different perspectives. By fusing features in $\mathcal{F}_i$, we hope to obtain a better image representation. The easiest way is to concatenate all feature vectors in $\mathcal{F}_i$ to form a single long vector, and directly use it for retrieval. However, concatenation is not always a sensible way due to disparate scaling and dimensionality of different feature vectors. Another straightforward approach is to learn the weight for each feature using original feature vectors in $\mathcal{F}_i$. In this way, feature vectors are required for a learning model. Although widely used in classification and recognition, it is not practical for our logo retrieval task, where feature vectors are usually high dimensional, *i.e.*, millions of dimensions, which cannot be easily stored and fed into the learning model. Also, we do not store an explicit BoW representation. Instead, it is more efficient to directly compute a distance between two images by computing a tf-idf score from an index tree representing the database. Instead of relying on the original feature vectors, we utilize the similarities between similar/dissimilar pairs of images to learn the weight for each individual feature effectively and efficiently.

[1]For point features like SIFT, a BoW representation is not explicitly created, so f_i^m can be seen as a collection of SIFT features.

3.2. Training data derivation

For a pair of images I_i and I_j, we have two sets of feature vectors $\mathcal{F}_i = \{f_i^1, f_i^2, ..., f_i^m, ..., f_i^M\}$ and $\mathcal{F}_j = \{f_j^1, f_j^2, ..., f_j^m, ..., f_j^M\}$, respectively. We employ a similarity metric $\phi_m(f_i^m, f_j^m)$ to quantitatively measure the similarity between images I_i and I_j in terms of m-th feature. The metric $\phi_m(f_i^m, f_j^m)$ can be any functions which convert two feature vectors into a normalized scalar with a fixed range [0,1], such as histogram intersection, $L2$ distance imposed a Gaussian kernel and tf-idf distance from a vocabulary tree/feature index, etc. The similarity $\Phi(i,j)$ between images I_i and I_j is then defined as

$$\Phi(i,j) = [\phi_1(f_i^1, f_j^1), \phi_2(f_i^2, f_j^2), .., \phi_M(f_i^M, f_j^M)]^\top \quad (1)$$

For each pair of images, we compute the similarity by Equation (1). Original features are not required anymore once we have the similarities for annotated similar/dissimilar pairs.

For clarity, we replace $\phi_m(f_i^m, f_j^m)$ and $\Phi(i,j)$ by $x_{i,j}^m$ and $\mathbf{x}_{i,j}$. Equation (1) can be re-written as $\mathbf{x}_{i,j} = (x_{i,j}^1, x_{i,j}^2, ..., x_{i,j}^m, ..., x_{i,j}^M)^\top$, where each dimension of $\mathbf{x}_{i,j}$ corresponds to a single feature. Essentially, this newly derived vector represents the inter-relationship between two images in terms of similarities from multiple features. Therefore, the multi-feature fusion problem is converted from learning weights for original feature vectors to determining the importance of each dimension of $\mathbf{x}_{i,j}$. Since the number of features used for the retrieval task is usually limited, $\mathbf{x}_{i,j}$ is of manageable dimensionality and can be easily processed.

Similarities from each individual feature are normalized, so that 1 means the two images are the same while 0 means that they are totally different. Suppose similar/dissimilar

annotations for P pairs of images are provided, we obtain a set of new samples $\mathbf{X} \in \mathbb{R}^{M \times P}$, where each column of $\mathbf{X}$ represents the similarity vector between a pair of images. We further assign labels to $\mathbf{X}$ so that $y = 1$ for similar image pairs and $y = 0$ for dissimilar pairs. With training data and labels, we will introduce our learning approach based on regression models.

3.3. Regression model for weight learning

Although a binary classifier can be readily applied to the training set, we are more interested in estimating the "similarity level" given a new data point. In addition, the labels do not have clear categorical meaning or strictly differentiate similar pairs of images from dissimilar ones. Therefore, we propose to fit a regression model on the training set rather than classifying it into two disjoint classes. The general form of a regression model is

$$y = R(\mathbf{w}, \mathbf{x}), \tag{2}$$

where $\mathbf{x}$ is the derived feature vector and $\mathbf{w}$ is the weight vector. y is the predicted value measuring the similarity level of a given data $\mathbf{x}$: how similar the two images are? Larger y indicates higher probability that the two images are similar. We aim to find a good weight vector $\mathbf{w}$ that fits the training data and generalizes well on unseen images. We have experimented with two regression functions: linear regression and logistic regression.

Linear regression Linear regression is the simplest regression function for linearly separable data. The formulation of linear regression is

$$y = \mathbf{w}^\top \mathbf{x} + b, \tag{3}$$

where b is the intercept. Least square analysis is often used to find the optimal $\mathbf{w}$. Regularization on $\mathbf{w}$ can also be included. Linear regression assumes that the output y is linearly correlated to the input data, which may not always hold. Additionally, due to different scaling of multiple features, the unbounded output is prone to be dominated by a single dimension of the input data, while the effect of other dimensions is diluted.

Logistic regression To handle the abovementioned situation, logistic regression is used. Basically, it is a normalization by mapping the output of the linear regression to $[0, 1]$ using a sigmoid function. In this way, noise is suppressed and the output is bounded. The formulation of logistic regression is

$$y = 1/(1 + exp(-\mathbf{w}^\top \mathbf{x} + b)). \tag{4}$$

Gradient descent is usually adopted to find the optimal $\mathbf{w}$. The output y here has specific meaning: it indicates the probability that the input data $\mathbf{x}$ has label 1. It is particularly suitable in our task where we need to measure the probability of similarity between two images.

After obtaining the optimal regression model R from the training data, we can use it during retrieval stage to improve the performance. Given a query image I_q, we initially obtain a set of ranked lists, $\mathcal{L}_q = \{L_q^1, L_q^2, ..., L_q^M\}$, by comparing the similarities of original feature vectors with respect to individual features. Each element in $\mathcal{L}$ is a list of retrieved images ordered by the similarities between feature vectors of the query and dataset images. Suppose T database images are initially retrieved. Similar to the training set, for each dataset image I_i, we have a vector $\mathbf{x}_{q,i} = (x_{q,i}^1, x_{q,i}^2, ..., x_{q,i}^m, ..., x_{q,i}^M)^\top$ measuring its similarity to the query from multi-feature perspective. If I_i is not initially retrieved by m-th feature, $x_{q,i}^m = 0$. Applying the learned regression model to $\mathbf{x}_{q,i}$, we obtain the probability, $y_{q,i} = R(\mathbf{w}, \mathbf{x}_{q,i})$, of the similarity between I_i and I_q. According to the new similarities $\{y_{q,1}, y_{q,2}, ..., y_{q,T}\}$ between every pair of the query and the dataset images, we rerank the initially retrieved images to obtain the refined results. Since the computation of new similarities only involves simple operations, the reranking process is extremely fast and can be easily incorporated into real-time retrieval systems. In addition, the weights for individual features are implicitly encoded by the weight vector $\mathbf{w}$ in the regression model. They are not affected by dimensionality or scaling of original feature vectors, so tend to be more robust against noise and unreliable features.

3.4. Including class information

We have learned a single regression model on the entire training data, where only annotations of similar/dissimilar image pairs are needed. However, in some cases, training samples have class labels annotated. For example, we may have the labels *adidas*, *dhl*, and *coca-cola* for training images from three logo classes. Moreover, images from diversified classes are much different from each other, where the complex distributions of original similarities from multiple features cannot be easily fitted by a single regression model. Under such circumstances, we revise our original formulation to introduce an ensemble of regression models utilizing class labels of training data.

Suppose training data is from C disjoint logo classes, $\{N_1, N_2, ..., N_c, ..., N_C\}$, where N_c denotes the number of images in class c, $\sum_c N_c = N$. For class c, we assign label 1 to pairs of images within the same class, and label 0 to pairs of images between c and all other classes. A class-specific regression model R_c is then learned from the training data. From C classes, we obtain an ensemble of class-specific regression models $\mathcal{R} = \{R_1, R_2, ..., R_c, ..., R_C\}$, which model the similarity distribution of each logo class.

Given a query I_q, same process in Sec. 3.3 is performed to obtain a new similarity $y_{q,i}^c$ regarding the pair of I_q and a dataset image I_i by each regression model R_c in $\mathcal{R}$. In total, we have C new similarities with respect to the pair (I_q, I_i),

denoted as the vector $\mathbf{y}_{q,i} = (y^1_{q,i}, y^2_{q,i}, ..., y^c_{q,i}, ..., y^C_{q,i})^\top$. Then our task is to infer a single similarity $\widetilde{y}_{q,i}$ from $\mathbf{y}_{q,i}$ so that $\widetilde{y}_{q,i}$ is an optimal combination of the elements in $\mathbf{y}_{q,i}$.

Assuming that the regression models of different classes are independent, mathematically, the inference can be written as:

$$\widetilde{y}_{q,i} = \sum_{c=1}^{C} R_c(\mathbf{w}, \mathbf{x}_{q,i}) \cdot p(R_c|I_q, I_i), \tag{5}$$

where $p(R_c|I_q, I_i)$ represents the probability of the regression model R_c given the pair of images (I_q, I_i). $R_c(\mathbf{w}, \mathbf{x}_{q,i}) = y^c_{q,i}$ is the output from the regression model R_c, given the similarity vector $\mathbf{x}_{q,i}$. The probability $p(R_c|I_q, I_i)$ can be estimated by exploiting the relationship amongst different classes within a Bayesian framework. In this paper, we simplify Equation (5) and express it as a linear combination of the outputs of all C regression models:

$$\widetilde{y}_{q,i} = \mathbf{u}^\top \mathbf{y}_{q,i}, \tag{6}$$

where $\mathbf{u}$ is a weight vector. In practice, we assume that all logo classes have equal probability, so that $u = p(R_c|I_q, I_i) = 1/C$ for $c = 1, 2, ..., C$ for each element u in $\mathbf{u}$. $\widetilde{y}_{q,i}$ is then a simple average of the outputs from all C regression models. We will show in the experiments that this technique leads to an improvement in performance compared to applying a single regression model.

4. Experiments

4.1. Experimental setting

Datasets. We experiment with two logo datasets, **FlickrLogo32** [22] and **BelgaLogo** [11].

FlickrLogo32 contains 32 brand logo classes used for logo detection, recognition and retrieval tasks. Each class contains 70 images showing various scale and viewpoint changes of the logo, of which 40 serve as database images and the remaining are query images. The separation of database and query images is pre-defined and fixed for all experiments. For retrieval tasks, 3000 non-logo images are included in the database. Totally we have 960 query images and 4280 database images.

BelgaLogo dataset includes 10000 images, where an image may contain multiple logos or no logos. Two sets of groundtruth are provided for different evaluation purposes. The global groudtruth contains image level annotations of 26 different logo classes indicating whether a specific logo is present in the images or not. 55 queries with localization of logos (*qset1*) [11] are provided. The local groundtruth includes 37 logo classes with bounding boxes, where 2697 images are used as internal queries (*qset3*) [16] and all the other images serve as database images. The number of images for different logo classes greatly varies.

Figure 2. Samples of cropped logos from **FlickrLogo32-crop**.

Features. We have experimented with the following features. We adopt Hessian affine keypoint detector with and without orientation information, and describe detected keypoints by SIFT descriptors, denoted as *shape2-oriented* and *shape2*. For global features, we extract LAB color histogram from the entire image, denoted as *image-lab*. We also adopt a compact color feature, Named Color (NC) descriptor [27], as an additional feature representation, dubbed as *image-nc*. The NC descriptor is an 11D histogram that is very efficient to compute and compare. Despite of its low dimensionality, the NC descriptor have a good discriminative power. Since logos usually only occupy a small portion of the whole image, the global color features may not be useful in some cases. To compensate this, we additionally apply the BING object detector [1] to an image to first generate a set of potential patches which may include logos, and then extract NC descriptors within each patch, denoted as *bing-nc*. We also include spatial information by dividing each BING box in a spatial pyramid manner, extract NC descriptor for each sub-patch and concatenate them to form a single feature vector. We denote this feature as *bing-nc-sp*.

Parameters. The size of the vocabulary for our (implicit) BoW representation is 1M. We extract 2000 BING boxes on each image, and divide each BING box by a 3×3 grid to include spatial information.

4.2. Results

4.2.1 Cropped logos

We first evaluate the performance of the proposed approach on cropped logos of **FlickrLogo32** dataset. We crop all logos out of the images according to the groundtruth bounding boxes, and obtain a new dataset named **FlickrLogo32-crop**. Note that there are cases where an image may contain multiple logos. Discarding logos which are too small and blurred, we obtain a total of 1802 logo images as the database, and 1347 logo images as the query set. All 3000 non-logo images remain unchanged. Some sample images of cropped logos are shown in Fig. 2. We obtain the similar/dissimilar annotations of all pairs of images from the database containing 4802 images to learn the regression models.

We use *shape2-oriented*, *shape2*, *image-lab* and *image-nc* in this experiment. The performance is evaluated by mean average precision (mAP) over the query set. As a baseline, we evaluate the performance using a simple equal-weight (EW) linear combination of similarities from multi-

Table 1. Performance in terms of mAP (%) using individual features

Dataset	1. *shape2-oriented*	2. *shape2*	3. *image-lab*	4. *image-nc*	5. *bing-nc*	6. *bing-nc-sp*
FlickrLogo32-crop	53.32	46.46	8.06	11.89	-	-
FlickrLogo32	57.15	51.90	-	-	5.21	6.95
BelgaLogo (*qset1*)	22.07	20.06	-	-	3.21	4.93
BelgaLogo (*qset3*)	17.84	14.32	-	-	2.76	4.45

Table 2. Performance in terms of mAP (%) using different fusion methods on **FlickrLogo32-crop**.

Fusion	EW	Linear			Logistic		
		S	A	C	S	A	C
1+2	56.16	55.73	56.20	**56.27**	56.2	56.36	**56.27**
1+2+3	11.48	17.58	20.21	22.85	55.21	55.81	**57.71**
1+2+4	19.47	27.05	28.91	31.10	56.27	57.24	**58.94**
all	13.58	22.96	22.93	25.24	56.33	57.16	**59.07**

Table 3. Performance in terms of mAP (%) using different fusion methods on **FlickrLogo32**.

Fusion	EW	Linear			Logistic		
		S	A	C	S	A	C
1+2	60.75	57.23	60.81	**60.94**	60.63	60.63	60.65
1+2+5	16.69	53.09	38.89	39.21	62.17	62.14	**62.40**
1+2+6	20.24	53.73	39.44	39.81	62.25	62.14	**62.63**
all	13.54	54.36	41.16	40.99	62.09	62.22	**62.53**

ple features. In addition, to thoroughly evaluate the performance of our regression based fusion approach, we test three variants of our framework:

- **SINGLE (S)**: only one regression model is learned from all training data as explained in Sec. 3.3.
- **AVG (A)**: we include class labels and learn an ensemble of regression models as in Sec. 3.4. The final similarity value is calculated by Equation (6) with the assumption that all logo classes have equal probability.
- **CLASS-SPEC (C)**: we assume the class label of a query is available, and then apply the corresponding regression model to obtain the new similarity. This variant is not practical in reality, but it provides an upper bound of our approach, helping better understand the performance of our approach.

The results are shown in Tables 1 and 2, from which we can make the following observations. First, EW is good enough if features are not complementary (*shape2-oriented* and *shape2*). Second, regression is always better than EW since weights of features are learned from data. Third, EW and linear regression are easily affected by a single weak feature, while logistic regression is very robust and always improves performance. Fig. 3 shows the retrieval performance for each individual logo class when fusing all four features. This clearly shows that logistic regression based fusion achieves the best results for most classes. For example, *shape2-oriented* achieves the best mAP for *aldi* logos, and *image-nc* is the best for *dhl* logos. In both cases, the logistic regression model successfully captures the best individual features and improves the performance accordingly. We further visualize the weights of individual features learned by the ensemble of logistic regression models on all training logo classes in Fig. 4 where the class-conditional adaptive weighting of individual features is clearly demonstrated.

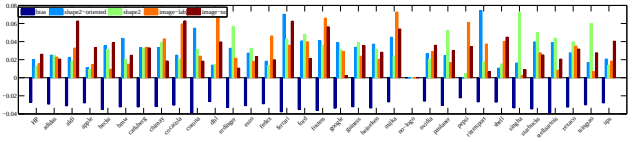

Figure 4. Bar plot of weights learned by the logistic regression models for individual features on all logo classes in **FlickrLogo32-crop**.

4.2.2 Non-cropped logos

We further conduct experiments on the original **FlickrLogo32** dataset. We replace the global color features *image-lab* and *image-nc* by *bing-nc* and *bing-nc-sp*, because a global color descriptor extracted from the whole image containing a large portion of background is unlikely to work well, given that it only achieves around 10% mAP on cropped logos without background. We use the partitions for database images and query images specified in [22]. The only difference between our experiments and [22] is that queries in our experiments are specified by a bounding box enclosing the logo. Since there are multiple logos in a single image, we treat them as independent queries, and again have 1347 queries as in Sec. 4.2.1, while the number of database images is 4280. The maximal similarity between the query bounding box and all BING boxes from a single database image is chosen as the similarity between the query and the database image. Results are shown in Table 1 and 3. Note that our results are not directly comparable to [22] since [22] always uses the entire image as a query.

Qualitative results of retrieved images for three sample queries are shown in Fig. 5. For the query on the left, logistic regression fusion is able to achieve significantly higher AP compared to any of the individual features. For the middle query, *shape2* achieves higher AP than the color features, while *bing-nc* achieves higher AP for the right query. Still, in both cases, our logistic fusion is able to achieve even higher AP while the simple equal-weight (EW) fusion performs poorer than the individual features. Fusion using logistic regression discovers similar images which were previously ranked much lower, and thus significantly improves the quality of retrieved results.

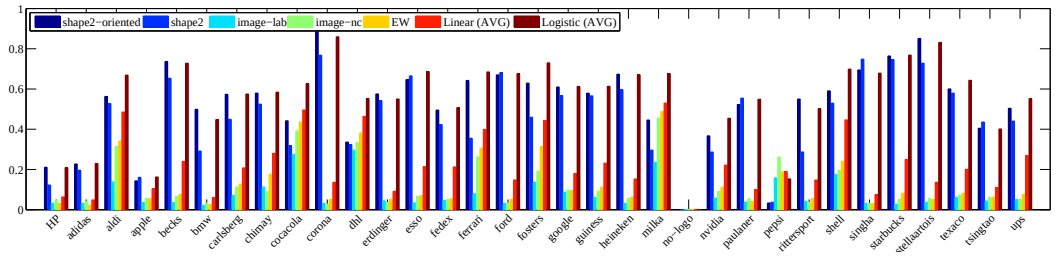

Figure 3. Bar plot of the mAP by individual features and different fusion methods on all logo classes from **FlickrLogo32-crop**. Logistic regression successfully captures and utilizes the best feature in most cases. See text for details.

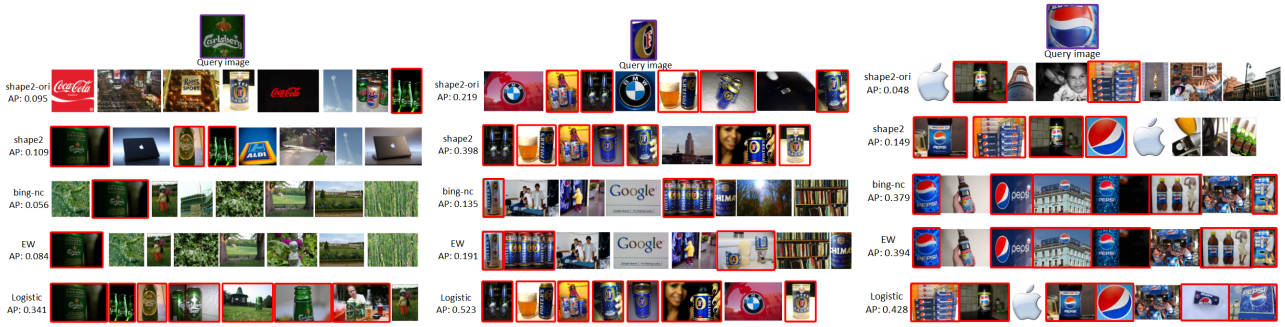

Figure 5. Retrieved results by individual features and fusion by logistic regression (**AVG**) for three queries from **FlickrLogo32**. For each query shown in the first row, the five rows below it show the top 8 retrieved results. We also show the Average Precision (AP) using individual features and different fusion methods. Images with red bounding boxes are the correct matches.

4.2.3 Generalization analysis

Varying train/test splits. The above experiments assume that the query logo belongs to one of the logo classes in the training set. However, it is not a realistic assumption due to the enormous number of potential logo classes which we may not have training data for. In this section, we will evaluate the generalization ability of our fusion method (logistic regression) when the query logo is outside the training logo classes.

We partition the database images from **FlickrLogo32-crop** dataset into two random subsets with disjoint logo classes. The two subsets serve as training and test sets, respectively, so that all images belonging to a logo class go to either the training or the test set. We learn a regression model independently for each logo class in the training set. The query set is also divided into two disjoint sets – "train query" and "test query" – corresponding to the split of the logo classes into training/test sets. Thus, given a "train query", we only retrieve images from the training set. Similarly, we only retrieve images from the test set for a "test query". For both "train query" and "test query", the new similarity is calculated by averaging the output of all regression models. The performance on "train query" and "test query" is evaluated separately. Results averaged across 10 random train/test splits are shown in Table 4.

Our fusion method using logistic regression is very robust: even using only 25% database images for training, we still obtain comparable mAP to the results obtained using 75% database images for training, which means that the regression models trained on 8 logo classes generalize well across the disjoint 24 (test) logo classes. By using logistic regression models, we always improve the performance compared to EW that is sensitive to individual features and train/test splits. Therefore, using the ensemble of logistic regression models with equal probability, we can ensure a performance improvement even when the query logo is not from the logo classes used in the training set.

Table 4. Comparison of different train/test splits in terms of mAP (%). We combine *shape2-oriented*, *shape2* and *image-nc*.

Dataset	Train/Test	EW		Logistic (AVG)	
		Train	Test	Train	Test
FlickLogo32-crop	75%/25%	19.58	19.15	57.26	56.06
	50%/50%	19.59	19.36	55.00	59.49
	25%/75%	21.91	18.66	59.00	56.49

Transfer between datasets. We have evaluated the generalization ability of our fusion method on **FlickrLogo32**. Nevertheless, in realistic scenarios, we cannot always have enough annotated training data for each database. In this case, we aim to learn a model from a database consisting of abundant training data and apply it to another database with limited or no labeled data. Specifically, we train regression models on all images from **FlickrLogo32** dataset and apply them to **BelgaLogo** dataset, where logos only occupy a very small portion of the entire image in most cases. We evaluate the performance in terms of mAP using two sets of queries, *qset1* and *qset3*. Results by individual features and

Table 5. Performance in terms of mAP (%) by different fusion methods by transferring learned models on **FlickrLogo32** to **BelgaLogo**.

Dataset	Fusion	Query	EW	Linear		Logistic	
				S	A	S	A
F → B	all	*qset1*	8.24	21.62	16.85	**26.29**	26.28
		qset3	7.04	18.91	14.24	**21.94**	21.71

Table 6. Comparisons of results by ESR, RVP and logistic regression (**AVG**) in terms of mAP (%) on 6 logo classes of *qset3* from **BelgaLogo**.

	Base	Dexia	Ferrari	Kia	Mercedes	President	Overall
ESR	17.9	11.7	5.2	49.7	18.0	44.6	24.5
RVP	20.8	15.3	1.3	**50.6**	**21.5**	67.5	29.5
Ours	**52.4**	**24.1**	**34.0**	41.2	11.0	**76.4**	**39.9**

fusion methods on the two sets of queries are shown in Table 1 and 5. Our fusion method still significantly improves the performance on **BelgaLogo**, even after training a single logistic regression model on a completely different dataset.

We further compare our results on **BelgaLogo** with ESR [15] and RVP [10], where 6 logo classes are evaluated. mAP is evaluated for all queries in each class and an overall mAP is also computed. Results are shown in Table 6. Our approach by logistic regression (**AVG**) significantly improves the performance on 4 classes and achieves the best overall mAP.

5. Conclusion

We presented a multi-feature fusion by similarity regression for logo retrieval, which only relies on pairwise similarities between images and does not require original feature vectors. For each pair of images, we construct a new sample by concatenating the similarities from multiple features. With annotations of similar/dissimilar pairs of images, a regression model is fitted on the set of new samples. Incorporating logo class labels, we learn an ensemble of regression models to better capture the inter-class variance. A new similarity between a query logo and a database image can be inferred from the learned regression models using original similarities from multiple features. Extensive experiments with two regression functions and various parameter settings have demonstrated that the logistic regression model performs very well. In addition, the regression model generalizes well on unseen logo classes and completely different datasets.

References

[1] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 5

[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8, 2007. 2

[3] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao. Visual reranking through weakly supervised multi-graph learning. In *ICCV*, pages 2600–2607, 2013. 2

[4] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, pages 745–752, 2011. 2

[5] B. Fernando, É. Fromont, D. Muselet, and M. Sebban. Discriminative feature fusion for image classification. In *CVPR*, pages 3434–3441, 2012. 3

[6] J. Fu, J. Wang, and H. Lu. Effective logo retrieval with adaptive local feature selection. In *ACM Multimedia*, pages 971–974, 2010. 2

[7] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009. 3

[8] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011. 3

[9] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008. 2

[10] Y. Jiang, J. Meng, and J. Yuan. Randomized visual phrases for object search. In *CVPR*, pages 3100–3107, 2012. 8

[11] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *ACM Multimedia*, pages 581–584, 2009. 2, 5

[12] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. S. Avrithis. Scalable triangulation-based logo recognition. In *ICMR*, page 20, 2011. 2

[13] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. S. Davis. Incremental multiple kernel learning for object recognition. In *ICCV*, pages 638–645, 2009. 3

[14] M. Kostinger, P. M. Roth, and H. Bischof. Planar trademark and logo retrieval. Technical report, Graz University of Technology, Austria, 02 2010. 2

[15] C. H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *ICCV*, pages 987–994, 2009. 8

[16] P. Letessier, O. Buisson, and A. Joly. Scalable mining of small visual objects. In *ACM Multimedia*, pages 599–608, 2012. 5

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006. 2

[18] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391, 2010. 1

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2

[20] J. Revaud, M. Douze, and C. Schmid. Correlation-based burstiness for logo retrieval. In *ACM Multimedia*, pages 965–968, 2012. 2

[21] S. Romberg and R. Lienhart. Bundle min-hashing. *IJMIR*, 2(4):243–259, 2013. 2

[22] S. Romberg and R. Lienhart. Bundle min-hashing for logo recognition. In *ICMR*, pages 113–120, 2013. 5, 6

[23] M. Rusiñol and J. Lladós. Efficient logo retrieval through hashing shape context descriptors. In *Document Analysis Systems*, pages 215–222, 2010. 2

[24] H. Sahbi, L. Ballan, G. Serra, and A. D. Bimbo. Context-dependent logo matching and recognition. *IEEE Transactions on Image Processing*, 22(3):1018–1031, 2013. 2

[25] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *ICCV*, pages 24–31, 2009. 3

[26] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003. 1

[27] J. van de Weijer, C. Schmid, J. J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009. 5

[28] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, pages 1473–1480, 2002. 3

[29] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal graph-based reranking for web image search. *IEEE Transactions on Image Processing*, 21(11):4649–4661, 2012. 2

[30] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. G. Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia*, 15(3):572–581, 2013. 3

[31] Y. Yang, Y. Zhuang, D. Xu, Y. Pan, D. Tao, and S. J. Maybank. Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning. In *ACM Multimedia*, pages 311–320, 2009. 3

[32] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, pages 3021–3028, 2012. 3

[33] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, pages 660–673, 2012. 2

[34] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian. Semantic-aware co-indexing for image retrieval. In *ICCV*, pages 1673–1680, 2013. 3

[35] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *CVPR*, 2014. 3