Elliptical ASIFT Agglomeration in Class Prototype for Logo Detection

Raluca Boia
rboia@imag.pub.ro;
Corneliu Florea
corneliu.florea@upb.ro;
Laura Florea
laura.florea@upb.ro;

Image Processing and Analysis Laboratory
University "Politehnica" of Bucharest
Bucharest, Romania

Abstract

Logo localization and recognition is difficult in natural images due to perspective deformations, varying background, possible occlusions, scaling variability; furthermore, the re–branding induces changes in the logo color palette and spatial distribution. To address this task, we locate keypoints using Affine Difference of Gaussian described by SIFT elliptical features; we construct the class prototypes by analyzing the graph of homographic matching between examples of the same class. The interconnections graph is developed for each class and the representative points from the non central examples are added to the class model. Potentially, an inverted secondary model is built for classes containing color inverted logos. Finally, each class is depicted by the reunion of the suitable keypoints and descriptors. The logo integrated detection (localization and classification) system is tested on multiple databases leading to state of the art accuracy.

1 Introduction

Logo (graphic entity that contains colors, shapes, textures and identifies organizations, goods, etc.) localization and recognition is a subproblem of object detection and recognition and a challenging pattern recognition task. The necessity of automatic detection comes from marketing industry that is aiming for methods to evaluate and increase the impact of a marketing campaign [12]. Applications can be tracked in the automotive industry, sports transmissions, legal (to counteract trademark infringement) or feedback for advertising.

Related Work Into the problem of logo recognition, we note two main directions: specific domain recognition (such as vehicle logo or document logo recognition) and general logo recognition in natural images. In the vehicle logo recognition category we note the work of Psyllos et al. [19] who used a Bag of visual Words (BoW) derived solution, and test it on small databases with few classes. Yet such categories (e.g. vehicle or document) are affected only by limited perspective distortion and limited warping as the background material is rigid.

The problem of *logos in natural images* is tackled from two sides: *retrieval* (i.e. given an image query containing logos, nominate other images with similar content) and *integrated*

Table 1: Logo search (retrieval or detection) systems in natural images.

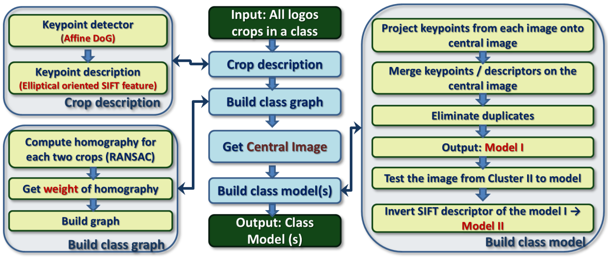
Meth.	Descriptor	Learning	Particularities	Retriev./Det.
[23]	SIFT	BoW	Bundling	Retriev.+Det.
[9]	SURF	BoW	Triangulation	Det.
[20]	SIFT	BoW	Weighting SIFT	Retriev.
[22]	SIFT	BoW	Bundling+Weighting	Retriev.
[24]	HoG	CDS	SIFT Context	Det.
[17]	HoG	SVM	2-Level	Retriev. + Det.
[14]	HoG+ASIFT	SVM /NN	HoG Prefiltering	Retriev. + Det.
[11]	SIFT	BoW	Class Prototype	Retriev.
[21]	HoG	SVM	CCCP	Retriev.
[28]	SIFT+GSC	BOW	Logo Density	Det.
[2]	SIFT	RANSAC	Class Prototype	Det.
[27]	SIFT	BoW	Reweighting in Retrieval	Retriev.
[15]	PCA-SIFT	RANSAC	Logo Heuristics	Det.
Proposed	*ASIFT*	*RANSAC*	*Elliptical SIFT* *1 or 2 Prototypes/Class*	*Det.*

detection (i.e. place a bounding box around the logo - localization, followed by naming its class - recognition). A summary of the more recent and relevant works is presented in the table 1. Many of the previous systems address the retrieval aspect, while for giving feedback and evaluating the performance of advertising campaigns, the detection is more relevant. For instance, in TV transmissions detection allows better quantification of impact. Our proposal goes into the detection category.

The first generic logo retrieval approaches [1], [8] were limited in handling large image collections. Later methods [10] retrieved logos by performing frequent item-set mining to discover association rules in spatial pyramids of visual words. Revaud et al. [20] used a BoW based approach coupled with learned weights to penalize correlated keypoints while computing distances. Romberg et al. [23] enhanced the BoW system by embedding spatial knowledge into the cascaded index. Romberg and Lienhart [22] extended the BoW by bundling on the min-hashing of SIFT-based visual words. Ries et al. [21] trained a SVM with Convex-concave Procedure over HoG features extracted from logos. Krapac et al. [11] used a voting scheme to build class prototypes. Yang and Bansal [27] reweighted each pair of images found to be similar by regression over an iteratively increasing set of training examples.

Logo detection is reported by Kalantidis et al. [9] who described logos by the histogram of triangles between SURF keypoints. Sahbi et al. [24] described logos with SIFT and their spatial interaction with Context Dependent Similarity (CDS) to construct the class model and match with query image keypoints. Zhang et al. [28] evaluated the SIFT and generalized shape contexts (GSC) keypoints density to get cues for branching their logo system. Lu et al. [17] stacked a set of K-D trees to discriminate among the feature vectors extracted with a first level of an ensemble of linear detectors. Li et al. [14] used a SVM to select the HoG described potential windows of interest and further classify them with Affine SIFT (ASFIT) and nearest neighbor (NN). Boia and Florea [2] constructed a class graph by aggregating the homographies between standard SIFT keypoints descriptors. Liu and Chen [15] recog-


(a)



(b)

Figure 1: (a) Examples of logos: top - NVidia from FlickrLogos-32 (note the variability and the color invertion) and bottom - Puma from BelgaLogos (note the very small size of the logo). (b) The schematic of the proposed system while describing the classes. We underlined with red specific choices necessary for greater performance.

nized storefront logos (in a there–introduced database) by describing them with PCA-SIFT, identifying and applying some heuristics on the keypoint neighborhood and building class prototype by RANSAC.

Contributions The challenges in a logo detection task are due to: perspective deformations, background variability, occlusions, scaling (from resolutions of 1000×1000 to 20×20), warping of the support object (in sports). This problem differs from the near-duplicate retrieval approaches by high intra-class variability, as a certain brand logo can have variations in the used colors or even in shape (as exemplified in figure 1 (a)).

In this paper we contribute by: (1) a new *class prototyping* method based on a central image extracted by analyzing the homographies graph and reprojecting the keypoints on that image (2) a logo detection system that exhibits great performance. The main conceptual difference to previous systems (as we will specifically point in the text) is that they manually branched their process to deal with corner-cases, while we perform the branching automatically, proposing a compact and self-adjusting system.

The paper structure is as follows: in section 2 we present the feature extraction, the keypoint description, the class homography graph and the method to construct the class models. In section 3 we present the testing procedure, the databases used for evaluation and we discuss the achieved results. The paper ends with short conclusions.

2 Class Description by Class Model

In the training phase (detailed in Fig. 1 (b)) models (prototypes) are built to describe classes. In the next paragraphs we will present the particular choices at each step.

Feature extraction The logo classes are described by a derivation of the SIFT algorithm: the Affine Difference-of-Gausssians (ADoG) [18] locator of keypoints is followed by the description of the keypoints using oriented SIFT elliptical local features.

The choice of ADoG differs from the standard DoG (in standard SIFT [16]) typically used [23], [22], [20], [19], [2] by the introduction of the affine part which aims to increase robustness to scaling, rotation and shearing of the logos. Practically, more keypoints are

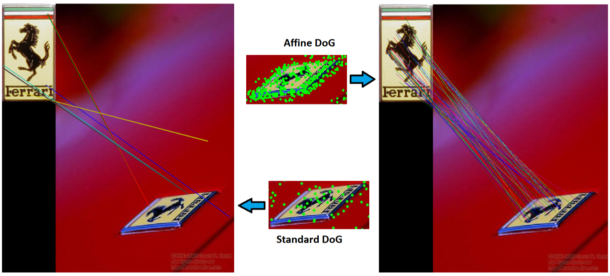


Figure 2: Comparison between matching performance using standard SIFT keypoint detector (left hand side) and Affine SIFT detector (right hand side). Note that using SIFT too few keypoints are found and the further homographic match is erroneous.

found using ASIFT than using standard SIFT, as exemplified in figure 2. A keypoint is described by gradient computation in *elliptical*, oriented neighborhoods. This choice contrasts with the standard circular vicinity used in [22], [20], etc. and is motivated by its capability to indicate keypoint orientation for circular (symmetrical) logos such as Apple or BMW. The benefits of the elliptical neighborhood are detailed by Li and Ma [13].

For the feature extraction part we rely on the VLFeat library [26]. We use the following adjustments: the peak threshold (the cornerness measure) decreases from 0.1 to 0.001 in order to include more features, while the edge threshold setting the limit of curvature variation ratio increases from 10 to 100 - to force features to be on edges.

Inspired by the work of Revaud et al. [20] and to address the lack of found features in very small logos, the smaller images are upscaled to 200 pixels while keeping the aspect ratio. In [20] all images are upscaled.

Homograpy, RANSAC and Class Graph All the logo crops from the same class are grouped in a graph: the nodes are the logos while an edge is given by the homography between that pair of logos-nodes.

Various logos from the same class differ by some perspective distortion. The perspective is described by the homography transform, $H \in R^{3\times 3}$, which moves a keypoint (x_1, y_1) from the first image plane to the coordinates (x_2, y_2) on the second image:

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T H = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T \tag{1}$$

A homography (see [4] for more details) has 8 degrees of freedom and it is typically estimated by point feature correspondences. A point correspondence provides 2 equations and 4 pairs suffice in solving H. When more than 4 correspondences are available, Brown and Lowe [3] suggested the use of RANSAC to deal with outliers.

The homography between 4 correspondences is found with the direct linear transformation (DLT) [7]. As in our previous work [2], the RANSAC iterates much more than typical 500 cycles [3], [27]. More precisely, we iterate - 20,000 times. The increase is motivated by smaller sized logos with few keypoints case when one needs sufficient sampling cycles so to get a good match. Yet, compared to [2] where circular vicinities were used and 200,000 iterations were required for a good fit, here, due to the use of more informative elliptical neighborhoods, less pairs matched suffice, and thus less iterations are needed for a good consensus. If too few inliers are found (less than 20), we declare no projection (graph link) is doable between a pair of logo crops.

An *error map* is used to determine the quality of matching by the homography of each pair of logo images. In fact, the error map helps to distinguish the areas of correct matching from which the keypoints and features should be kept. Using the found homography, the second image is projected on the first image plane. Each image is described with Dense SIFT (as implemented in VLFeat [26]) and the Hellinger distance is computed as the measure of fit. A pixel in a cell having a good quality value (low distance value) is a point that represents a suitable connection between the images and is not an occlusion or a distortion of the shape, as exemplified in figure 3 (a).

Given n logo crops per class, $n_1 < n(n-1)/2$ image pairs are matched; the difference is caused by occlusions, inverted colors or insufficient features. The matched pairs are grouped together in a graph similar to the idea suggested by Romberg et al. [23]. We differ by using a *penalty weighted* graph (illustrated in figure 4 (a)), where the weight of the link between two logo crops is given by $w_{ij} = \frac{1}{N_{ij}}$, where N_{ij} is number of keypoints matched between logo i and j.

As in the work of Psyllos et al.[19], we do not use the graph to describe the class, but to select a *central image* and construct the class model from it. In contrast to the work of Psyllos et al. [19], where the central image is manually picked, we select it as the image with the most connections; the main difference with respect to the later is that we also, potentially, build a secondary model.

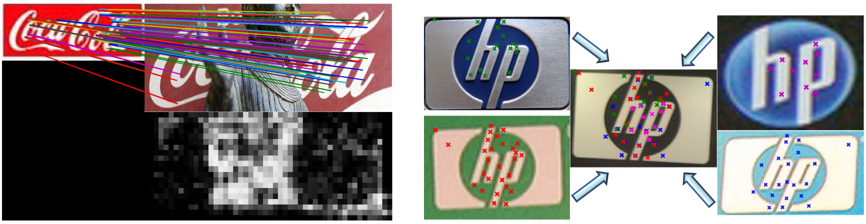
Color Inverted Logos: Secondary Prototype Logo images vary due to illumination conditions while occasional re-branding adds changes in color and spatial distributions of the logo setup. SIFT feature descriptors of the grayscale images may cope with changes in color palette, but can't do it for full color inversion. Full inversion leads to a mirroring (separately on the horizontal and on the vertical) of the gradients orientation, [6] and this aspect must be addressed specifically.

Compared to the work of Revaud et al. [20] where classes with negative color (e.g. Adidas, HP, NVidia, etc.) are manually selected and the inverted class model is build, we create a secondary model automatically, by evaluating the connectivity degree of the components in the class graph using the Tarjan's algorithm [25].

The training images with similar luminance match to each other through homographies, while the inverted luminance ones cluster separately. Images not connected to the central image are compared with the *inverted* central image descriptors and if the match is good, then the secondary model is needed and built. We avoid building directly a secondary model from the images marked as belonging to the secondary cluster since they are too few. Instead, the secondary model is determined by the inversion of the descriptors of the main model (see [6] for an explanation), while the keypoints position is kept unaltered. The secondary model will be used as a full, separate model in the testing phase.

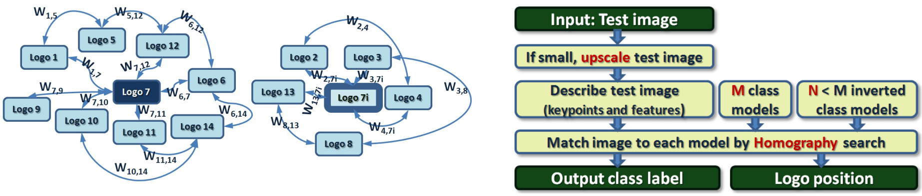
The Class Model and Descriptor The class model is built by agglomerating onto the central image the keypoints and their description of all the logo images found to be in the main cluster of the class graph, as illustrated in figure 4 (a).

Keypoints in images (associate with a good matching onto the central image according to the error map) directly connected to the central image are back-projected (by inverting the matching homography) on the central image. We note that SIFT feature descriptors are not affected by projection into the plane of the central image. The equivalent homography between images that are not directly connected to the central image is determined by com-



(a) (b)

Figure 3: (a) The error map for an occluded logo; note that on the occluded part the errors in the map are high. (b) The process of agglomerating keypoints from examples onto the central image. The keypoints on the central image are not retained directly but implicitly as they matched keypoints from other images in the class.



(a) (b)

Figure 4: (a) Illustration of a class graph. For the main cluster, the *central image* has the most connections (dark blue). The right hand cluster is not connected with the first one (it forms the secondary cluster) and signals the need for a secondary model obtained by inverting colors of the main model. (b) The schematic of the system used to locate and classify a logo in a testing image.

posing the homographies placed on the path between that image and the central one. Yet, the homographic projection is imperfect and it adds an error.

While the shortest path means composing the fewest homographies, thus accumulating fewest terms for the error, this is not necessary the smallest error. It is intuitive that a homography projection error (in the case when the poorest homographic fits are discarded) is inversely proportional with the number of pairs matched, so these were turned into graph weights. Thus, the chosen patch is the one that ensures the largest volume of information for the smallest error, since a larger number of pairs means more data to be accumulated in the model.

The model of the class consists from the reunion of all the suitable points and descriptors in the class, projected in the plane of the central image and accumulated on top of it. In figure 3 (b) we illustrate the result of the aggregation of the keypoints from four images on the central image, showing that each matching process reveals different pairs of keypoints that must be merged in order to obtain the best representation of the logo.

In the process of agglomerating, some of the keypoints may be too close and become redundant. To avoid this redundancy we use a quantization step based on K-D trees over the keypoint descriptors [3].

3 Implementation and Results

The feature extraction and the matching part is build upon VLFeat open source library [26], while the rest of the system is implemented in Matlab.

Evaluation Procedure Given a query image, we count a true detection if the found logo is indeed in the image and if the intersection-over-union (Jaccard index), is above 50% as defined in Pascal VOC protocol [6].

3.1 Databases. Training sets

FlickrLogos-32. To test the robustness of the method, we first choose the FlickrLogos-32 database [23], as it is the most widely used. Due to small logos size, the FlickrLogos-32 is a small-object dataset. The testing/ training scheme is provided by the database creators [23]: 30 images per class for training and 30 images per class for testing phase for a total of 32 classes. Only the crops were used in the training phase, while keypoints are computed for entire images while testing.

BelgaLogos. The BelgaLogos [8] is associated with sports images; most of the logos are on the sportswear being very small and having partial warping. The ones situated on the boards are often occluded and blurred. Many classes have a small number of images which affects the training process. We experimented only with the annotated part: 1937 images containing logos from 37 classes. The training/testing scheme is two-fold, class-wise.

FlickrLogos-27. This database is different from FlickrLogos-32 and it was introduced by Kalantidis et al. [9]. The database is annotated with bounding boxes and it is separated in training and testing sets (fixed at 5 images/class). Multiple choices are presented for the training set, varying from 10 to 40 images per class. We chose the variant with 25 images for training while the rest were added to the initial testing set, as in the database introductory work [9]. Thus, we used 636 images for training and 540 for testing.

3.2 Testing

The purpose of testing is to locate logos and classify them. Given a model for each class, the testing phase tries to match the query image against all the class models as it is shown in figure 4 (b). The matching is done as in the training phase: using SIFT feature matching and RANSAC search for the correct homography.

The Dense SIFT based error map is built for each matching result and its average is used as an indicator of the quality of the image matching: a score smaller than T (found at 0.4) indicates that the logo is present. The system locates the logo position and the homography points to a class model, thus solving the recognition. If, after being confronted to all the class models, no score is large enough, then the test image will be classified as ”no-logo”.

While training, we learned that too small sized logos contain insufficient features for detection. In testing there is no information about logo sizes or locations, so we first try to localize the logos in the images using their natural size following with an up-scaled version of the image by a factor of 4. The upscaling is required for very small logos, encountered especially in the BelgaLogos dataset. The idea behind this up-scaling is to increase the logo detectable size to the minimum training logo size; the precise value was empirically found as a compromise between accuracy and duration.

Table 2: Detection rates (DR) (i.e. localization+classification) on the FlickrLogos-32 database. Lu et al. [17] reports results for BF - brute force and using 4K KD trees to accelerate the process. The proposed method includes both the weighted graph and the inverted colors. Results detailed on individual classes may be seen in figure 5.

Method	[22]	CIM - [19]	[17] - BF	[17] - 4K	[14]
DR[%]	61.14	60.1	65.84	58	78.71
Method	[2]	CI	RESZ.	CIRC.	*Proposed*
DR[%]	84.06	88.43	76.04	90.62	***93.33***

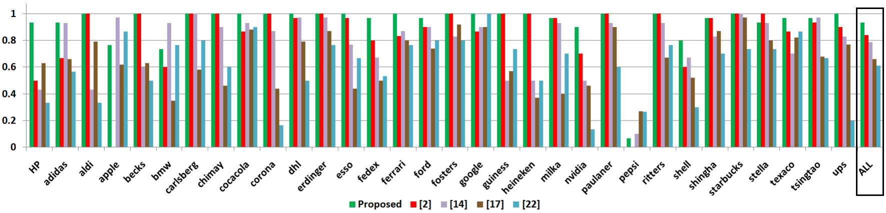


Figure 5: Comparative results on the **FlickrLogos-32** database. We report the results achieved by the proposed method and by [22], [14], [17], [2].

3.3 Results and Discussions

We must point out that our method produces, for the tested databases, only true detections and false negatives (i.e. logo incorrectly labelled as ”no logo”); the logos are not misclassified. Yet, one may imagine a circumstance where two brands have similar enough logos so that the proposed system will be confused by them.

Illustrative examples of the proposed method are in figure 7 while additional results and code are available on the project web-page[1].

FlickrLogos-32. On the FlickrLogos-32 we achieved the average detection rate of 93.33%. Performances for each class, comparative with previous methods [2] are presented in figure 5. A very poor result is achieved on the Pepsi logo, yet this is a trait of all methods and it is caused by the circularity and by the lack of keypoints found.

Numerical comparative results with related systems are in table 2. To show the benefits of critical steps of the proposed system, we considered the central image solely as class model (CIM) as suggested in [19]. We also report the achieved results if we don’t resize small images (marked with *RESZ.*) and if the SIFT vicinities are the standard circular ones (*CIRC.*). We achieve state of the art results, outperforming the previously introduced methods with near 10% and the baseline method of Romberg et al. [22] with more than 30%.

Our previous solution [2] method failed for symmetrical logos (addressed here by the elliptical oriented neighborhoods and showed in the improvement over Apple and BMW), color inverted logos (solved by two prototypes per class and showed by HP or NVidia classes); the overall improvement is also related to the elaborate mode of processing the

[1]The project page is: `http://www.imag.pub.ro/common/staff/rboia/logoRecognition/`
[2]Comparison with state of the art for all the databases uses the results reported by the respective papers with the exception of [2] where we used the code.

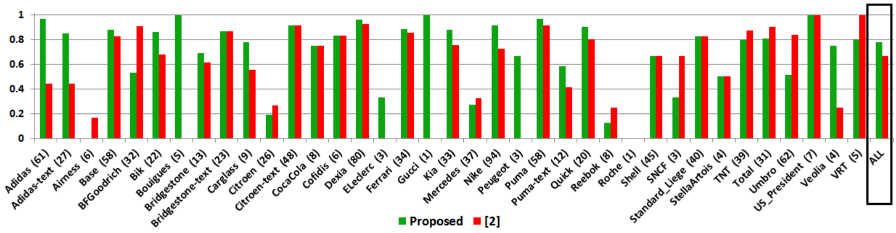


Figure 6: Comparative results on the **BelgaLogos** database. We present our results and the ones using the code from the method of Boia and Florea [2], which achieve an overall accuracy of 66.63%. On the bottom, in parenthesis, we marked the number of images available in each class for training. Poor performance is associated with classes with too few images.

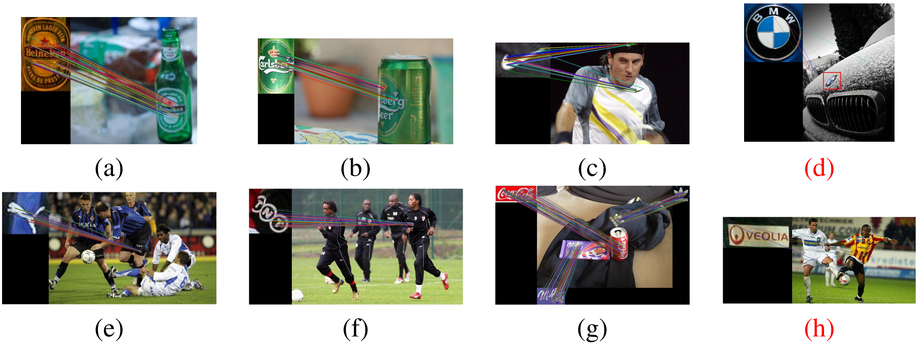


Figure 7: Correct detections: (a) blurry logo; (b) occluded logo (c) simultaneous normal and color inverted logo; (e), (f) very small logos in sport images; (g) multiple classes. Failed detections: (d) Extreme perspective distortion; (h) too small and blurred

class graph. However both methods rely on SIFT based descriptors which are less accurate in describing the Pepsi logo while the HoG descriptor is [17].

BelgaLogos. Here, the achieved detection rate is **78.09%**. Usually the logos in this dataset are very small since the images present sport scenes with logos on players sportswear. Illustrative detections are shown in figures 7 (d),(e),(f). To our best knowledge no other method reports detection on this database, but only retrieval. We used the code for our older method [2] to test in the same scenario as we did here. The individual class results are detailed in figure 6. Again the major differences are on classes that require an inverted model. On the negative side, we mistakenly construct an inverted class for BFGoodrich.

The annotated part of BelgaLogos dataset is unbalanced as for certain classes only few examples exist: 15 classes have less than 10 training /testing images, while ”Gucci” and ”Roche” have just one image. Due to insufficient examples to learn the structure of the logo the method gives poor results in such cases.

The algorithm cannot detect logos in cases when the logo instance is very blurred and at a small scale, since the keypoints extracted in its area are not enough to represent its shape (figure 7 (h)). Extreme warping (figure 7 (d)) also causes problems. Also in some cases, the sportswear carrying a logo is sufficiently distorted so that it exceeds the dealing capacity of the homography model; these cases will result in failures.

Table 3: Detection rates (DR) on the **FlickrLogos-27** database. Values marked with * are estimated from author graphs. In [9] results for a standard BoW algorithm are reported also.

Method	BOW - [9]	[9]	[24]	[28]	[2]	*Proposed*
DR[%]	56.8*	57.8*	71.6*	71.5*	50.7	72.8

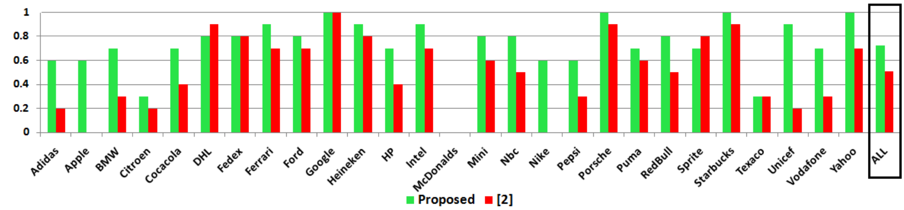


Figure 8: Comparative results on the **FlickrLogos-27** database.

FlickrLogos-27. The achieved detection rate is **72.8%**. Again, we used the code of [2] and compared with results from [9], [24] and [28], that report detection rates on this database. Previous methods discussed multiple solutions but we consider only the best reported performance in table 3. The individual class results are detailed in figure 8. In this case too, the proposed method reaches the best performance, even that now by a smaller margin.

Failures are associated with logos that are rather different from the training set, as it is for instance the case for McDonalds class where the training set contains the yellow logo on bright background, while the testing set requires an inverted class model that was not built.

Multiple logos/image. The algorithm is able to detect multiple logos in one image, such as in figure 7 (g). This trait is achieved by testing a query image against all class models and retaining all homography matches that produced an error map smaller than chosen threshold.

Logos with inverted color. Creating additional virtual classes for cases which present high variation in color (including both original and negative version) permits simultaneous detection of both, even in the same image. A visual example is shown in figure 7 (c).

4 Conclusions

We have proposed an effective automatic method for logo localization and recognition. The system surpasses many challenges of logo detection in natural images. The achieved results reveal an improvement with more than 10% with respect to the best previous work on FlikrLogo-32 while it reaches state of the art performance on all databases. The major differences with respect to previous similar systems lay in coupling ADoG with elliptical neighborhood and into the analysis of the class graph that may result in a supplementary inverted model.

Acknowledgment This work was supported by the Romanian Sectoral Operational Programme Human Resources Development 2007-2013 through the European Social Fund Financial Agreements POSDRU/159/1.5/S/132395 and POSDRU /159/1.5/S/134398.

References

[1] A. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo. Trademark matching and retrieval in sports video databases. In *ACM MIR*, pages 79–86, 2007.

[2] R. Boia and C. Florea. Homographic class template for logo localization and recognition. In *Proc. of IbPRIA*, pages 487–495, 2015.

[3] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73, 2007.

[4] E. Dubrofsky. Homography estimation. Master's thesis, Carleton University, 2009.

[5] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes VOC challenge. *IJCV*, 1:303–338, 2010.

[6] J. Hare, S. Samangooei, and P. Lewis. Efficient clustering and quantisation of SIFT features: Exploiting characteristics of the sift descriptor and interest region detectors under image inversion. In *ICMR*, 2011.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

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

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

[10] J. Kleban, X. Xie, and Wei-Ying Ma. Spatial pyramid mining for logo detection in natural scenes. In *IEEE ICME*, pages 1470–1477, 2008.

[11] J. Krapac, F. Perronnin, T. Furon, and H. Jegou. Instance classification with prototype selection. In *ACM ICMR*, pages 431 – 4, 2014.

[12] R. Lewis, J. Rao, and D. Reiley. Measuring the effects of advertising: The digital frontier. In *Economics of Digitization*, pages 1–5, 2014.

[13] C. Li and L. Ma. A new framework for feature descriptor based on SIFT. *Pat. Rec. Lett.*, 30(5):544–557, 2009.

[14] K.W. Li, S.Y. Chen, S. Su, D.J. Duh, H. Zhang, and S. Li. Logo detection with extendibility and discrimination. *Multimed Tools Appl.*, 72:1285–1230, 2014.

[15] F. Liu and Y. Chen. Fast and robust storefront logo recognition in shopping centers. In *ICME*, pages 1–7, 2015.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 62(2):91–110, 2004.

[17] V. Lu, I. Endres, M. Stroila, and J. Hart. Accelerating arrays of linear classifiers using approximate range queries. In *WACV*, pages 255–260, 2014.

[18] J.M. Morel and G.Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[19] A. P. Psyllos, C. N. E. Anagnostopoulos, and E. Kayafas. Vehicle logo recognition using a SIFT-based enhanced matching scheme. *IEEE Trans. Intel. Transp. Syst.*, 11(2): 322–328, 2010.

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

[21] C. Ries, F. Richter, S. Romberg, and R. Lienhart. Automatic object annotation from weakly labeled data with latent structured SVM. In *CBMI*, pages 1–4, 2014.

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

[23] S. Romberg, L. Garcia Pueyo, R. Lienhart, and R. van Zwol. Scalable logo recognition in real-world images. In *ACM ICMR*, pages 965–968, 2011.

[24] H. Sahbi, L. Ballan, G. Serra, and A. Del Bimbo. Context-dependent logo matching and recognition. *IEEE Trans. Im. Proc.*, 22:1018–1031, 2013.

[25] R. E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *ACM MM*, pages 1469–1472, 2010.

[27] F. Yang and M. Bansal. Feature fusion by similarity regression for logo retrieval. In *WACV*, 2015.

[28] Y. Zhang, S. Zhang, W. Liang, and Q. Guo. Individualized matching based on logo density for scalable logo recognition. In *ICASSP*, pages 4324–4328, 2014.