Landmark Recognition in VISITO Tuscany *

Giuseppe Amato, Fabrizio Falchi, Fausto Rabitti

ISTI-CNR, Pisa, Italy
{giuseppe.amato, fabrizio.falchi, fausto.rabitti}@isti.cnr.it

Abstract. This paper discusses and compares various approaches to automatic landmark recognition in pictures, based upon image content analysis and classification. The paper first compares various visual features and image similarity functions based on local features. Finally it discusses and compares a new classification technique to decide the landmark contained in an image that first classifies the local features of the image and then uses this result in order to take a final decision on the entire image. As the experiments demonstrate, this last approach is the most effective one. The discussed techniques were used and tested in the VISITO Tuscany project.

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: H.3.3 Information Search and Retrieval;
Keywords: Image classification, Content Based Retrieval

1 Introduction

An emerging challenge that is recently attracting attention in the field of multimedia information retrieval is that of landmark recognition [15]. It consists in automatically recognizing the landmark (a building, a square, a statue, a monument, etc.) appearing in a non annotated picture. Landmark recognition is particularly appealing in applications for mobile devices, where one wants to obtain information on monuments by simply taking a picture.

The VISITO Tuscany (VIsual Support to Interactive TOurism in Tuscany[1]) project aims at addressing this interesting issue with the purpose of investigating and realizing technologies able to offer an interactive and customized advanced tour guide service for cities of art in Tuscany. More specifically, it focuses on offering services to be used (see Figure 1):

During the tour – through the use of mobile devices of new generation, in order to improve the quality of the experience. The mobile device is used by the user to get detailed information about what he's watching, or about the context he's placed in. While taking pictures of monuments, places and other close-up objects, the user points out what, according to him, seems to be more interesting. The taken picture is processed by

* This work was partially supported by the VISITO Tuscany project, funded by the Regione Toscana, Italy within the POR CREO FESR program.

[1] http://www.visito-tuscany.it/

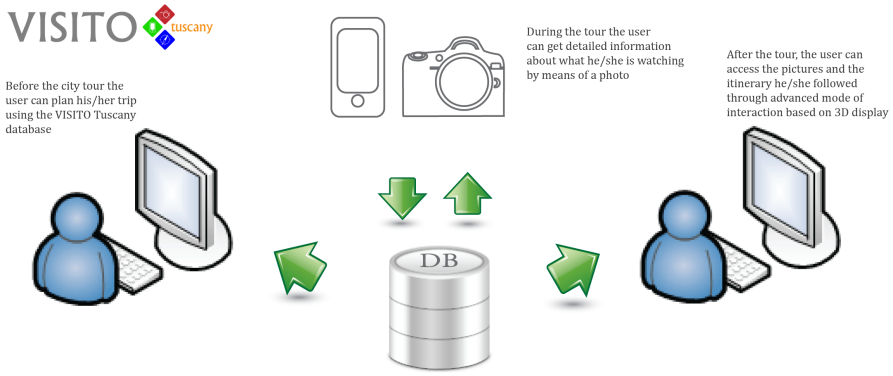

Fig. 1. The VISITO Tuscany project services.

the system to infer which are the user's interests and to provide him relevant and customized information. For example, if a user takes a picture of the bell tower by Giotto, he can get detailed information describing the bell tower, its architectural techniques, etc.

Before the tour – to plan the visit in a better way. The tourist can betteter plan his own visit empoying both the information sent by other users and their experiences, together with the information already included in the database system and, more generally, on the web. The interaction will take place through advanced methods based on 3D graphics.

After the tour – to keep the memory alive and share it with other people. The user can access the pictures and the itinerary he followed through advanced interaction based on 3D graphics. Moreover, he might share his information and experiences with other users by creating social networks.

Even if the general objective of the VISITO Tuscany project is broader, in this paper we will just focus on the automatic landmark recognition in images aspect. In particular after a description of the idea of landmark recognition given in Section 2, Section 3 we first compare the performance of various visual features, considering both global and local features. Then we compare various similarity functions based on local features in Section 4. Finally in Section 5 we discuss a technique for landmark recognition that classifies an image by first classifying its local features.

2 Landmark Recognition

In the last few years, the problem of recognizing landmarks has received growing attention by both research community. As an example, Google presented its approach to building a web-scale landmark recognition engine [15] that was also used to implement the Google Goggles service [10].

The problem of landmark recognition is typically addressed by leveraging on techniques of automatic classification, as for instances kNN Classification [9], applied to image features.

More in details, given a set of documents D and a predefined set of *classes* (also known as *labels*, or *categories*) $C = \{c_1, \ldots, c_m\}$, *single-label document classification* (SLC) is the task of automatically approximating, or estimating, an unknown *target function* $\Phi : D \rightarrow C$, that describes how documents ought to be classified, by means of a function $\hat{\Phi} : D \rightarrow C$, called the *classifier*, such that $\hat{\Phi}$ is an approximation of Φ.

A well-known classification technique, which we have used for landmark recognition tests, is the *single-label distance-weighted k-NN*. It decides about the class of a document in two steps. First it executes a k-NN search between the objects of the *training set*. The result of such operation is a list of labeled documents d_i belonging to the *training set* ordered with respect to the decreasing values of the similarity $s(d_x, d_i)$ between d_x and d_i. The label $\hat{\Phi}^s(d_x)$ assigned to the document d_x by the classifier is the class $c_j \in C$ that maximizes the sum of the similarity between d_x and the documents d_i in the k-NN results list $\chi^k(d_x)$ labeled c_j.

Therefore, first a score $z(d_x, c_i)$ for each label is computed for any label $c_i \in C$:

$$z(d_x, c_j) = \sum_{d_i \in \chi^k(d_x) \,:\, \Phi(d_i) = c_j} s(d_x, d_i) \; .$$

Then, the class that obtains the maximum score is chosen:

$$\hat{\Phi}^s(d_x) = \arg\max_{c_j \in C} z(d_x, c_j) \; .$$

It is also convenient to express a degree of confidence on the answer of the classifier. For the *Single-label distance-weighted kNN* classifier described here we defined the confidence as 1 minus the ratio between the *score* obtained by the second-best label and the best label, i.e,

$$\nu_{doc}(\hat{\Phi}^s, d_x) = 1 - \frac{\max\limits_{c_j \in C - \hat{\Phi}^s(d_x)} z(d_x, c_j)}{\max\limits_{c_j \in C} z(d_x, c_j)} \; .$$

This classification confidence can be used to decide whether or not the predicted label has an high probability to be correct.

The similarity function s between two documents plays a strategic role for the effectiveness of the image classification algorithm. In fact images can be compared on the basis of different visual features and even for the same visual feature, various similarity functions can be defined. In the following we will first test the effectiveness of various visual features for the landmark recognition task, then we compare various similarity measures.

2.1 Landmark recognition test settings

The landmark recognition task was executed using the above mentioned single-label distance-weighted k-NN classification strategy employing specific similarity functions between images depending on the tested visual features.

To compare the various visual features we identified 12 landmarks, and we manually built the training sets for them by identifying a congruous number of pictures representing them. The dataset that we used for our tests is publically available and

composed of 1,227 photos of 12 landmarks located in Pisa and was used also in [5, 3, 4]. The photos have been crawled from Flickr, the well known on-line photo service. The IDs of the photos used for these experiments together with the assigned label and extracted features can be downloaded from [1].

In order to build and evaluating a classifier for these classes, we divided the dataset in a *training set* (Tr) consisting of 226 photos (approximately 20% of the dataset) and a *test set* (Te) consisting of 921 (approximately 80% of the dataset). The image resolution used for feature extraction is the standard resolution used by Flickr i.e., maximum between width and height equal to 500 pixels.

The total number of local features extracted by the SIFT and SURF detectors were about 1,000,000 and 500,000 respectively.

3 Comparisons of Visual Features

Content based retrieval and content based classification techniques typically are not directly applied to images content. Rather, matching and comparisons between low level mathematical descriptions of the images visual appearance, in terms of color histograms, textures, shapes, point of interests, etc., are used. Different visual features represent different visual aspects of an image. All together, different visual features, contribute, not exhaustively, to represent the complete information contained in an image. A single feature is generally able to carry out just a limited amount of this information. Therefore, its performance varies in dependence of the specific dataset used and the type of conceptual information one wants to recognize.

The goal of this section is to compare various visual features or combination of visual features that provides us with the best performance with the landmark recognition task.

In order to perform our evaluation we choose various global and local visual features. Specifically, we evaluated the performance of the 5 MPEG-7 [11] visual features (Color Layout, Color Structure, Edge Hystogram, Homogeneous Textures, Scalable Colour), the Scale invariant Feature Transform (SIFT) [13], the ColorSIFT [8], and the Speeded Up Robust Features (SURF) [7]. In the following we give a brief description of their principles.

3.1 MPEG-7

MPEG-7 visual descriptors consist of a set of 5 different global descriptors of the low level visual content of an image [11]. These 5 descriptors are mathematical representations of different statistical measures that can be computed analyzing the structure and placement of the colored pixel in an image. In particular:

- Scalable Color is an histogram of the colors of the pixel in an image, when colors are represented in the Hue Saturation Value (HSV) space
- Color Structure expresses local color structure in an image by use of a structuring element that is comprised of several image samples
- Color Layout is a compact description of the spatial distribution of colors in an image

– Edge Histogram descriptor describes edge distribution with a histogram based on local edge distribution in an image, using five types of edges
– Homogeneous Texture descriptor characterizes the properties of the texture in an image.

For extracting the MPEG-7 visual descriptors we made use of the MPEG-7 eXperimental Model (XM) Reference Software [12].

3.2 SIFT

The Scale Invariant Feature Transformation (SIFT) [13] is a representation of the low level image content that is based on a transformation of the image data into scale-invariant coordinates relative to local features. Local feature are low level descriptions of keypoints in an image. Keypoints are interest points in an image that are invariant to scale and orientation. Keypoints are selected byf choosing the most stable points from a set of candidate location. Each keypoint in an image is associated with one or more orientations, based on local image gradients. Image matching is performed by comparing the description of the keypoints in images.

3.3 ColorSIFT

ColorSIFT local features [8] are an extension of the original SIFT definition to also take color into account. Basically, the original SIFT definition describes the local edge distribution around keypoints. The ColorSIFT extends the description of a keypoint also to colors around it. This is obtained by considering color gradients, rather than just intensity gradients. Between the various proposals they made, we tested the colour-based SIFT invariant to shadow and shading effects which performed best in the experiments reported in [8].

3.4 SURF

The basic idea of Speeded Up Robust Features (SURF) [7] is quite similar to SIFT. SURF detects some keypoints in an image and describes these keypoints using orientation information. However, the SURF definition uses a new method for both detection of keypoints and their description that is much faster still guaranteeing a performance comparable or even better than SIFT. Specifically, keypoint detection relies on a technique based on a approximation of the Hessian Matrix. The descriptor of a keypoint is built considering the distortion of Haar-wavelet responses around the keypoint itself.

3.5 Similarity measures

For each feature used in the experiments we need a measure that evaluates the similarity between two photos. For the MPEG-7 visual descriptors we used the distances suggested by the MPEG Group in [12]. Let $d(d_x, d_y)$ be the distance, we defined the similarity between to objects as:

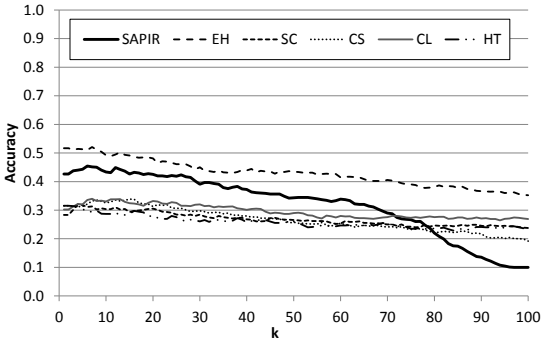

Fig. 2. Micro-averaged accuracy of the classifier for various *k* and various global features (MPEG-7 Visual Descriptors and the combination used in the SAPIR project)

$$s(d_x, d_y) = 1 - w * d(d_x, d_y) \tag{1}$$

where w is a fixed number that guarantees that $w * d(x, y) < 1$ for any d_x and d_y.

In the experiments we also tested the weighted sum distance of these 5 MPEG-7 Visual Descriptors used in the *Search in Audiovisual using Peer-to-Peer Information Retrieval (SAPIR) FP6 European research project* [2]. More information about this combination can be found in [6].

A common strategy to compare two images d_x and d_y using local features (e.g., SIFT, ColorSIFT and SURF) is typically the number of keypoints in d_x that have a match in d_y. We translate this information in a similarity function dividing the number of matches by the number of keypoints in d_x. In other words we used the ratio of keypoints in d_x that do have a match in d_y as the similarity between d_x and d_y for all the local features used for the experiments (i.e., SIFT, ColorSIFT and SURF). Later on, in the paper, we will also propose and compare alternative strategies to define local feature based similarity functions.

The algorithms used for matching the keypoints for the various local features are the ones suggested by the features authors and that are also used in their public available implementations. In particular both SIFT and ColorSIFT performs a 2-NN search between the keypoints in d_y for any keypoint in d_x. A match is identified if the 1st result in the 2-NN has a distance from the query keypoint less than 0.6 times the distance of the 2nd result. SURF matching algorithm is very similar except that the distance of the 1st nearest neighbor must be less than $1/\sqrt{2}$. More information can be found in [13, 8, 7].

3.6 Results

In Figure 2 we report the micro-averaged *accuracies* obtained for some MPEG-7 Visual Descriptors and their weighted sum combination used in the SAPIR Project (see

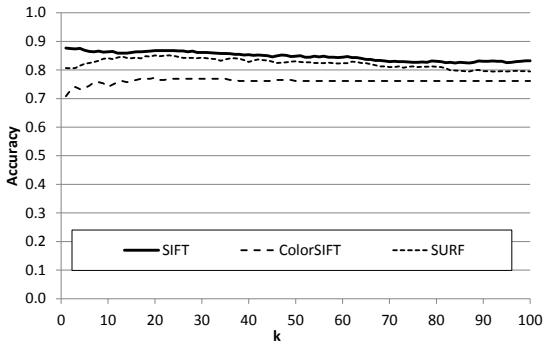

Fig. 3. Micro-averaged accuracy of the classifier for various *k* and various local features (SIFT, Color-SIFT, SURF)

3.5). The best performance is obtained using the EdgeHistogram visual descriptor. The color-based features (i.e., ColorLayout, ColorStructure, ScalableColor) have very similar performance while HomogenousTexture obtained the worst values of *accuracy*. The weighted-sum combination of these visual descriptor performs slightly worst than EdgeHistogram alone. Even if for big values of k the SAPIR metric is preferable, the best *accuracy* for the various k is higher for EdgeHistogram alone.

The *accuracy* obtained for the local features are reported in Figure 3. As expected, all of them perform significantly better than the global features. In fact, the dataset used is specific for landmarks recognition and they are supposed to work well for general recognition tasks. What was not obvious is that SIFT (the oldest) perform better than the others. Both SURF and ColorSIFT are basically extensions of the SIFT but for this specific task they are less effective than SIFT.

4 Comparisons of various local feature based image similarity functions for landmark recognition

In previous section we compared various visual features with a kNN classifier and results proved that the best performance was achieved using local features In particular the best performance was obtained the SIFT local descriptor. The similarity function used with local features was defined as the ratio between the matching keypoints and the total number of keypoints in the compered image. However, additional improvement can be obtained by varying the definition of the similarity function.

In order to define image similarity functions based on local features we first need to define the notion of similarity between local features themselves. The Computer Vision literature related to local features, generally uses the notion of distance, rather than that of similarity. However in most cases a similarity function $s()$ can be easily derived from a distance function $d()$. For both SIFT and SURF the Euclidean distance is typically used as measure of dissimilarity between two features [13, 7].

Let $d(p_1, p_2) \in [0, 1]$ be the normalized distance between two local features p_1 and p_2. We can define the similarity between local features as:

$$s(p_1, p_2) = 1 - d(p_1, p_2)$$

Obviously $0 \leq s(p_1, p_2) \leq 1$ for any p_1 and p_2.

Another useful aspect that is often used when dealing with local features is the concept of local feature matching, that is deciding if a given local feature of an image matches a some local feature of another image. In [13], a distance ratio matching scheme was proposed that has also been adopted by [7] and many others. Let's consider a local feature p_x belonging to an image d_x (i.e. $p_x \in d_x$) and an image d_y. First, the point $p_y \in d_y$ closest to p_x (in the remainder $NN_1(p_x, d_y)$) is selected as candidate match. Then, the distance ratio $\sigma(p_x, d_y) \in [0, 1]$ of closest to second-closest neighbors of p_x in d_y is considered. The distance ratio is defined as:

$$\sigma(p_x, d_y) = \frac{d(p_x, NN_1(p_x, d_y))}{d(p_x, NN_2(p_x, d_y))}$$

Finally, p_x and $NN_1(p_x, d_y)$ are considered matching if the distance ratio $\sigma(p_x, d_y)$ is smaller than a given threshold. Thus, a function of matching between $p_x \in d_x$ and an image d_y is defined as:

$$m(p_x, d_y) = \begin{cases} 1 \text{ if } \sigma(p_x, d_y) < c \\ 0 \text{ otherwise} \end{cases}$$

In [13], $c = 0.8$ was proposed reporting that this threshold allows to eliminate 90% of the false matches while discarding less than 5% of the correct matches. Please note, that this parameter will be used in defining the image similarity measure used as a baseline and in one of our proposed local feature based classifiers.

In the following, we finally define 5 different approaches to compute image similarity measures relying on local features.

1-NN Similarity Average – s^1 The simplest similarity measure only consider the closest neighbor for each $p_x \in d_x$ and its distance from the query point p_x. The similarity between two documents d_x and d_y can be defined as the average similarity between the local features in d_x and their closest neighbors in d_y. Thus, we define the *1-NN Similarity Average* as (for simplicity, we indicate the number of local features in an image d_x as $|d_x|$):

$$s^1(d_x, d_y) = \frac{1}{|d_x|} \sum_{p_x \in d_x} \max_{p_y \in d_y} (s(p_x, p_y))$$

Percentage of Matches – s^m A reasonable measure of similarity between two image d_x and d_y is the percentage of local features in d_x that have a match in d_y. Using the distance ratio criterion described above for individuating matches, we define the *Percentage of Matches* similarity function s^m as follows:

$$s^m(d_x, d_y) = \frac{1}{|d_x|} \sum_{p_x \in d_x} m(p_x, d_y)$$

where $m(p_x, d_y)$ is 1 if p_x has a match in d_y and 0 otherwise.

Distance Ratio Average – s^{σ} The matching function $m(p_x, d_y)$ used in the *Percentage of Matches* similarity function is based on the ratio between closest to second-closest neighbors for filtering candidate matches as proposed in [13]. However, this distance ratio value can be used directly to define a *Distance Ratio Average* function between two images d_x and d_y as follows:

$$s^{\sigma}(d_x, d_y) = \frac{1}{|d_x|} \sum_{p_x \in d_x} \sigma(p_x, d_y)$$

Please note that function does not require a distance ratio c threshold.

Hough Transform Matches Percentage – s^{h} An Hough transform is often used to search for keys that agree upon a particular model pose. The Hough transform can be used to define a *Hough Transform Matches Percentage*:

$$s^{h}(d_x, d_y) = \frac{|M_h(d_x, d_y)|}{|d_x|}$$

where $M_h(d_x, d_y)$ is the subset of matches voting for the most voted pose. For the experiments, we used the same parameters proposed in [13], i.e. bin size of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum model dimension for location.

4.1 Results

In Table 1, *Accuracy* and macro averaged F_1 of the image similarity based classifiers for the 4 similarity functions are reported. Note that the *single-label distance-weighted kNN* technique has a parameter k that determines the number of closest neighbors retrieved in order to classify a given image. This parameter should be set during the training phase and is kept fixed during the test phase. However, in our experiments we decided to report the result obtained ranging k between 1 and 100. For simplicity, in the Table, we report the best performance obtained and the k for which it was obtained. Moreover, we report the performance obtained for $k = 1$ which is a particular case in which the kNN classifier simply consider the closest image.

The *Hough Transform Matches Percentage* (s^h) similarity function is the best choice for both SIFT and SURF. The second best is *Distance Ratio Average* (s^{σ}) which only considers the distance ratio as matching criterion. Please note that s^{σ} does not require a distance ratio threshold (c) because it weights every match considering the distance ratio value. Moreover, s^{σ} performs sightly better than *Percentage of Matches* (s^m) which requires the threshold c to be set. The results obtained by the *1-NN Similarity Average* (s^1) function show that considering just the distance between a local features and its closest neighbors gives worse performance than considering the distance ratio s^{σ}. In other words, the similarity between a local feature and its closest neighbor is meaningful only if compared to the other nearest neighbors, which is exactly what the distance ratio does.

similarity function			s^1 Avg 1-NN	s^m Perc. of Matches	s^σ Avg Sim. Ratio	s^h Hough Transf.
Best	Acc	**SIFT**	0.75	0.88	0.89	**0.92**
		SURF	0.79	0.85	0.82	**0.89**
	F_1	**SIFT**	0.72	0.86	0.87	**0.90**
		SURF	0.76	0.83	0.81	**0.87**
k=1	Acc	**SIFT**	0.73	0.88	0.89	**0.91**
		SURF	0.79	0.81	0.81	**0.87**
	F_1	**SIFT**	0.72	0.86	0.87	**0.90**
		SURF	0.76	0.79	0.80	**0.85**
Best k	Acc	**SIFT**	*9*	*1*	*1*	*2*
		SURF	*3*	*20*	*8*	*21*
	F_1	**SIFT**	*1*	*1*	*1*	*2*
		SURF	*1*	*18*	*8*	*21*

Table 1. Comparisons of the performance of the various image similarity functions based on local features

Regarding the parameter k it is interesting to note that the k value for which the best performance was obtained for each similarity measure is typically much higher for SURF than SIFT. In other words, the closest neighbors in the training set are more relevant using SIFT than using SURF.

5 kNN based on local feature similarity

In the previous section, we considered the classification of an image d_x as a process of retrieving the most similar ones in the *training set* Tr and then applying a kNN classification technique in order to predict the class of d_x.

In this section, we discuss a new approach that first assigns a label to each local feature of an image. The label of the image is then assigned by analyzing the labels and confidences of its local features.

This approach has the advantage that any access method for similarity search in metric spaces (see [14]) can be used to speed-up classification.

The proposed *Local Feature Based Classifiers* classify an image d_x in two steps:

1. first each local feature p_x belonging to d_x is classified considering the local features of images in Tr;
2. second the whole image is classified considering the class assigned to each local feature and the confidence of the classification.

Note that classifying individually the local features, before assigning the label to an image, we might loose the implicit dependency between interest points of an image.

However, surprisingly, we will see that this method offers better effectiveness than the other approaches presented before. In other words we are able to improve at the same time both efficiency and effectiveness.

In the following, we assume that the label of each local feature p_x, belonging to images in the training set Tr, is the label assigned to the image it belongs to (i.e., d_x):

$$\forall p_x \in d_x,\ \forall d_x \in Tr\ , \Phi(p_x) = \Phi(d_x).$$

In other words, we assume that the local features generated over interest points of images in the training set can be labeled as the image they belong to. Note that the noise introduced by this label propagation from the whole image to the local features can be managed by the local features classifier. In fact, we will see that when very similar training local features are assigned to different classes, a local feature close to them is classified with a low confidence.

Given $p_x \in d_x$, a local feature classifier $\hat{\Phi}_l$ returns both a class $\hat{\Phi}_l(p_x) = c_i \in C$ to which it believes p_x to belong *and* a numerical value $\nu(\hat{\Phi}_l, p_x)$ that represents the confidence that $\hat{\Phi}$ has in its decision. High values of ν correspond to high confidence. These are defined as follows:

$$\begin{cases} \hat{\Phi}^l(p_x) = \Phi(NN_1(p_x, Tr)) \\ \nu(\hat{\Phi}^l, p_x) = (1 - \dot{\sigma}(p_x, T_r))^2 \end{cases}$$

Where $NN_1(p_x, Tr)$ is the local feature of Tr most similar to p_x and $\dot{\sigma}$ is defined as

$$\dot{\sigma}(p_x, T_r) = \frac{d(p_x, NN_1(p_x, Tr))}{d(p_x, NN_2^*(p_x, Tr))}$$

where $NN_2^*(p_x, Tr)$ is the closest neighbor that is known to be labelled differently than the first as suggested in [13].

The intuition is that we use use $1 - \dot{\sigma}(p_x, t_r)$ that basically is a distance ratio, as a measure of confidence to be used during the classification of the whole image. The value is squared to emphasize the relative importance of greater distance ratios.

Please note that for this classifier we do not have to specify any parameter at all.

As we said before, the local feature based feature classification is composed of two steps . We have just dealt with the issue of classifying every local feature of an image. Now we discuss the second phase of the local feature based classification of images. In particular we consider the classification of the whole image given the label $\hat{\Phi}(p_x)$ and the confidence $\nu(\hat{\Phi}, p_x)$ assigned to its local features $p_x \in d_x$ during the first phase.

To this aim, we use a confidence-rated majority vote approach. We first compute a score $z(p_x, c_i)$ for each label $c_i \in C$. The score is the sum of the confidence obtained for the local features predicted as c_i. Formally,

$$z(d_x, c_i) = \sum_{p_x \in d_x, \hat{\Phi}(p_x) = c_i} \nu(\hat{\Phi}_l, p_x)\ .$$

Then, the label that obtains the maximum score is chosen:

$$\hat{\Phi}(d_x) = \arg\max_{c_j \in C} z(d_x, c_j)\ .$$

As measure of confidence for the classification of the whole image we use ratio between the predicted and the second best class:

$$\nu_{img}(\hat{\Phi}, d_x) = 1 - \frac{\max\limits_{c_j \in C - \hat{\Phi}(p_x)} z(d_x, c_j)}{\max\limits_{c_j \in C} z(d_x, c_j)} .$$

This whole image classification confidence can be used to decide whether or not the predicted label has an high probability to be correct.

5.1 Results

Also in this case we report the *Accuracy* and macro averaged F_1 of the classifier. Results are shown in Table 2. Comparing these results with those reported in Table 1 it is evident that the local feature based kNN classifier is better than the single-label distance weighted kNN classifier applied to the best similarity function both using SIFT and SURF. In fact, best accuracy was 0.92 for SIFT and 0.89 for SURF, while the new classifier offers an accuracy of 0.95 for SIFT and 0.93 for SURF. Similar considerations can be done for the F_1 measure.

		$\hat{\Phi}$
Accuracy	**SIFT**	**0.95**
	SURF	**0.93**
F_1 Macro	**SIFT**	**0.95**
	SURF	**0.92**

Table 2. Performance of the local feature classifier

6 Conclusions and future work

This paper presented the techniques for landmark recognition that were used in the project VISITO Tuscany. An extensive evaluation was performed to compare the various techniques and to asses the most effective method. In particular the paper performed a comparisons of various image visual features and various similarity functions used to build the classifier. In addition we also proposed a new classification method based on the idea of first classifying the local feature of images and to use this result to classify an entire image. Our experiments proved that this was the most effective method and that it also opens up new opportunities for efficient implementation of the landmark recognition approach on a large scale.

References

1. Pisa landmarks dataset. http://www.fabriziofalchi.it/pisaDataset/.
2. Search in Audiovisual using Peer-to-Peer Information Retrieval (SAPIR) FP6 European research project. http://www.sapir.eu.
3. G. Amato and F. Falchi. kNN based image classification relying on local feature similarity. In *SISAP '10: Proceedings of the Third International Conference on SImilarity Search and APplications*, pages 101–108, New York, NY, USA, 2010. ACM.
4. G. Amato and F. Falchi. Local feature based image similarity functions for kNN classfication. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, pages 157–166. SciTePress, 2011. Vol. 1.
5. G. Amato, F. Falchi, and P. Bolettieri. Recognizing landmarks using automated classification techniques: an evaluation of various visual features. In *in Proceeding of The Second Interantional Conference on Advances in Multimedia (MMEDIA 2010)*, pages 78–83. IEEE Computer Society, 2010.
6. M. Batko, F. Falchi, D. Novak, R. Perego, F. Rabitti, J. Sedmidubsky, and P. Zezula. Building a web-scale image similarity search system. *Multimedia Tools and Applications*, to appear.
7. H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.
8. G. J. Burghouts and J. M. Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113:48–62, 2009.
9. S. Dudani. The distance-weighted k-nearest-neighbour rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):325–327, 1975.
10. Google. Goggles. http://www.google.com/mobile/goggles/, 2011.
11. ISO/IEC. Information technology - Multimedia content description interfaces, 2003. 15938.
12. ISO/IEC. Information technology - Multimedia content description interfaces. Part 6: Reference Software, 2003. 15938-6:2003.
13. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
14. P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer-Verlag, 2006.
15. Y. Zheng, M. Z. 0003, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1085–1092, 2009.