On kNN Classification and Local Feature Based Similarity Functions

Giuseppe Amato and Fabrizio Falchi

ISTI-CNR, via G. Moruzzi 1, 56124 Pisa, Italy
{giuseppe.amato,fabrizio.falchi}@isti.cnr.it
http://www.isti.cnr.it

Abstract. In this paper we consider the problem of image content recognition and we address it by using local features and kNN based classification strategies. Specifically, we define a number of image similarity functions relying on local features comparing their performance when used with a kNN classifier. Furthermore, we compare the whole image similarity approach with a novel two steps kNN based classification strategy that first assigns a label to each local feature in the document to be classified and then uses this information to assign a label to the whole image. We perform our experiments solving the task of recognizing landmarks in photos.

Keywords: Image classification, Recognition, Landmarks, Pattern recognition, Machine learning, Local features.

1 Introduction

Image content recognition is a very important issue that is being studied by many scientists worldwide. In fact, with the explosion of the digital photography, during the last decade, the amount of digital pictures available on-line and off-line has extremely increased. However, many of these pictures remain unannotated and are stored with generic names on personal computers and on on-line services. Currently, there are no tools and effective technologies to help users in searching for pictures by real content, when they are not explicitly annotated. Therefore, it is becoming more and more difficult for users to retrieve even their own pictures.

A picture contains a lot of implicit conceptual information that is not yet possible to exploit entirely and effectively. Automatically content based image recognition opens up opportunities for new advanced applications. For instance, pictures themselves might be used as queries on the web. An example in this direction is the service "Google Goggles" [11] recently launched by Google, that allows you to obtain information about a monument through your smartphone using this paradigm.

Note that, even if many smartphones and cameras are equipped with a GPS and a compass, the geo-reference obtained with this is not enough to infer what the user is actually aiming at. Content analysis of the picture is still needed to determine more precisely the user query or the annotation to be associated with a picture. A promising approach toward image content recognition is the use of classification techniques to associate images with classes (labels) according to their content. For instance, if an

J. Filipe and A. Fred (Eds.): ICAART 2011, CCIS 271, pp. 224–239, 2013.

image contains a car, it might be automatically associated with the class *car* (labelled with the label *car*).

In this paper we study the problem of image content recognition by using SIFT [14] and SURF [5] local features, to represent image visual content, and kNN based classifiers to decide about the presence of conceptual content. In more details we will define 20 different functions that measure similarity between images. These functions are defined using various options and combinations of local feature matching and similarities. Some of them also take into consideration geometric properties of the matching local features. These functions are used in combination of a standard Single-label Distance-Weighted kNN algorithm. In addition we also propose a new classification algorithm that extend the traditional kNN classifiers by making direct use of similarity between local features, rather than similarity between entire images. We will see that the similarity functions that also make use of geometric considerations offer a better performance than the others. However, the new kNN based classifier that exploit directly the similarity between local features has an higher performance even without using geometric information.

The paper is organized as follows. In Section 3 we briefly introduce local features. In Section 4 we present various iamge similarity features relying on local features to be used with a kNN classification algorithm. Section 5 propose a novel classification approach. Finally, Sections 6 and 7 presents the experimental results. An earlier version of this paper has been presented at the Third International Conference on Agents and Artificial Intelligence [1].

2 Related Work

The first approach to recognizing location from mobile devices using image-based web search was presented in [17]. Two image matching metrics were used: energy spectrum and wavelet decompositions. Local features were not tested. In the last few years the problem of recognizing landmarks have received growing attention by the research community. In [16] methods for placing photos uploaded to Flickr on the World map was presented. In the proposed approach the images were represented by vectors of features of the tags, and visual keywords derived from a vector quantization of the SIFT descriptors. In [13] a combination of context- and content-based tools were used to generate representative sets of images for location-driven features and landmarks. Visual information is combined with the textual metadata while we are only considering content-based classification. In [19], Google presented its approach to building a web-scale landmark recognition engine. Most of the work reported was used to implement the Google Goggles service [11]. The approach makes use of the SIFT feature. The recognition is based on best matching image searching, while our novel approach is based on local features classification. In [7] a survey on mobile landmark recognition for information retrieval is given. Classification methods reported as previously presented in the literature include SVM, Adaboost, Bayesian model, HMM, GMM. The kNN based approach which is the main focus of this paper is not reported in that survey. In [9], various MPEG-7 descriptors have been used to build kNN classifier committees. However local features were not considered.

In [6] the effectiveness of NN image classifiers has been proved and an innovative approach based on Image-to-Class distance that is similar in spirit to our approach has been proposed.

3 Local Features

The approach described in this paper focuses on the use of image local features. Specifically, we performed our tests using the SIFT [14] and SURF [5] local features. In this section, we briefly describe both of them.

The Scale Invariant Feature Transformation (SIFT) [14] 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 by 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. For both detecting keypoints and extracting the SIFT features we used the public available software developed by David Lowe[1].

The basic idea of Speeded Up Robust Features (SURF) [5] 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. For both, detecting keypoints and extracting the SURF features, we used the public available noncommercial software developed by the authors [2].

4 Image Similarity Based Classifier

In this section we discuss how traditional kNN classification algorithms can be applied to the task of classifying images described by local features, as for instance SIFT or SURF. In particular, we define 20 image similarity measures based on local features description. These will be later on compared to the new classification strategy that we propose in Section 5.

4.1 Single-Label Distance-Weighted kNN

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) [8] is the task of automatically approximating, or estimating, an unknown *target function* $\Phi : D \to C$, that describes how documents ought to be classified, by means of a function $\hat{\Phi} : D \to C$, called the *classifier*, such that $\hat{\Phi}$ is an approximation of Φ.

[1] http://people.cs.ubc.ca/~lowe/
[2] http://www.vision.ee.ethz.ch/~surf

A popular SLC classification technique is the *Single-label distance-weighted kNN*. Given a training set Tr containing various examples for each class c_i, it assigns a label to a document in two steps. Given a document d_x (an image for example) to be classified, it first executes a kNN search between the objects of the *training set*. The result of such operation is a list $\chi^k(d_x)$ of labelled 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 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, labelled c_j, in the kNN results list $\chi^k(d_x)$

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,

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

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

4.2 Image Similarity

In order the kNN search step to be executed, a similarity function between images should be defined. Global features, generally, are defined along with a similarity (or a distance) function. Therefore, similarity between images, is computed as the similarity between the corresponding global features. On the other hand, a single image has several local features. Therefore, computing the similarity between two images requires combining somehow the similarities between their numerous local features.

In the following we define a function for computing similarity between images on the basis of their local features that is derived from the work presented in [14]. In the experiments, at the end of this paper, we will compare the performance of the similarity function, when used with the *single-label distance-weighted kNN* classification technique, against the local feature based classification algorithm proposed in Section 5.

Local Feature Similarity. 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 [14,5].

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 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.

Local Features Matching. A useful aspect that is often used when dealing with local features is the concept of local feature matching. In [14], a distance ratio matching scheme was proposed that has also been adopted by [5] 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 [14], $c = 0.8$ was proposed reporting that this threshold allows to eliminate 90% of the false matches while discarding less than 5% of the correct matches. In Section 7 we report an experimental evaluation of classification effectiveness varying c that confirms the results obtained by Lowe. 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.

For Computer Vision applications, the distance ratio described above is used for selecting good candidate matches. More sophisticated algorithms are then used to select actual matches from the selected ones considering geometric information as scale, orientation and coordinates of the interest points. In most of the cases a Hough transform [3] is used to search for keys that agree upon a particular model pose. To avoid the problem of boundary effects in hashing, each match is hashed into the 2 closest bins giving a total of 16 entries for each hypothesis in the hash table. This method has been proposed for SIFT [14] and is very similar to the weak geometry consistency check used in [12].

Thus, we define the set $M_h(d_x, d_y)$ as the matching points in the most populated entry in the Hash table containing the Hough transform of the matches in d_y obtained using the distance ratio criteria.

4.3 Similarity Measures

In this section, we define 5 different image similarity measures approaches and 4 different versions of each of them for a total of 20 measures.

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 in 4.2 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 (see Sec. 4.2).

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 [14] and reported in Section 4.2. 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. As mentioned in Section 4.2, 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 [14], 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.

Managing the Asymmetry. All the proposed similarity functions are not symmetric, i.e., $s(d_x, d_y) = s(d_y, d_x)$ does not hold. Consider the case in which the set of local features belonging to d_x is a subset of the ones belonging to d_y. In this case the similarity $s(d_x, d_y)$ is 1 while the same does not hold for $s(d_y, d_x)$.

In searching for images similar to d_x, it is not clear in advance whether $s(d_x, d_y)$ or $s(d_y, d_x)$ would be a better similarity measure for the recognition task. Thus, we tested various combinations.

Given an image d_{Te} belonging to Te (i.e., an image that we want to automatically classify), and an image d_{Tr} belonging to Tr (i.e., an image for which the class label is known in advance) we define various versions of the similarities defined before:

- $s^{Te}(d_{Te}, d_{Tr}) = s(d_{Te}, d_{Tr})$ – is the canonical approach which tries to find points in the test image that are similar to the ones in the training one.
- $s^{Tr}(d_{Te}, d_{Tr}) = s(d_{Tr}, d_{Te})$ – is the inverse approach which uses the points in training documents as queries.
- $s^{or}(d_{Te}, d_{Tr}) = \max(s(d_{Te}, d_{Tr}), s(d_{Tr}, d_{Te}))$ – is the fuzzy *or* of s^{Te} and s^{Tr}. This considers equivalent two images if any of the two is a crop of the other.
- $s^{and}(d_{Te}, d_{Tr}) = \min(s(d_{Te}, d_{Tr}), s(d_{Tr}, d_{Te}))$ – is the fuzzy *and* of s^{Te} and s^{Tr}. This never considers equivalent two images if any of the two is a crop of the other.
- $s^{avg}(d_{Te}, d_{Tr}) = (s(d_{Te}, d_{Tr}) + s(d_{Tr}, d_{Te}))/2$ – is the mean of s^{Te} and s^{Tr}.

Thus, we have defined 5 versions of our 4 similarity measures for a total of 20 similarity measures that will be denoted as $s^{m,Te}$, $s^{m,Tr}$, $s^{m,or}$, ..., $s^{h,Te}$, etc.

5 Local Feature Based Image Classifier

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 propose 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 [18] can be used to speed-up classification. The proposed *Local Feature Based Image 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 the 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 baseline approach. 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). Following the notation used in Section 4,

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

In other words, we assume that the local features generated over interest points of the 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. The experimental results reported in Section 7 confirm the validity of this assumption.

As we said before, given $p_x \in d_x$, a classifier $\hat{\Phi}$ returns both a class $\hat{\Phi}(p_x) = c_i \in C$ to which it believes p_x to belong *and* a numerical value $\nu(\hat{\Phi}, p_x)$ that represents the confidence that $\hat{\Phi}$ has in its decision. High values of ν correspond to high confidence.

5.1 Local Feature Classifier

Among all the possible approach for assigning a label to a interest point, the simplest is to consider the label of its closest neighbor in Tr. The confidence value can be evaluated using the idea of the distance ratio discussed in Section 4.2.

We thus define a local feature based classifier $\hat{\Phi}^m(p_x)$ that assign a candidate label $\hat{\Phi}^m(p_x)$ as the one of the nearest neighbor in Tr closest to p_x (i.e., $NN_1(p_x, Tr)$):

$$\hat{\Phi}^m(p_x) = \Phi(NN_1(p_x, Tr))$$

The confidence here plays the role of a matching function, where the idea of the distance ratio is used to decide if the candidate label is a good match:

$$\nu(\hat{\Phi}^m, p_x) = \begin{cases} 1 \text{ if } \dot{\sigma}(p_x, t_r) < c \\ 0 \text{ otherwise} \end{cases}$$

The distance ratio $\dot{\sigma}$ here is computed considering the nearest local feature to p_x and the closest local feature that has a label different than the nearest local feature. This idea follows the suggestion given by Lowe in [14], that whenever there are multiple training images of the same object, then the second-closest neighbor to consider for the distance ratio evaluation should be the closest neighbor that is known to come from a different object than the first. Following this intuition, we define the distance ratio $\dot{\sigma}$ 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 labeled differently than the first as suggested in [14].

The parameter c used in the definition of the confidence is the equivalent of the one used in [14] and [5]. We will see in Section 7 that $c = 0.8$ proposed in [14] by Lowe is able to guarantee good effectiveness. It is worth to note that c is the only parameter to be set for this classifier considering that the similarity search performed over the local features in Tr does not require a parameter k to be set.

5.2 Whole Image Classification

As we said before, the local feature based feature classification is composed of two steps (see Section 5). In previous section we have dealt with the issue of classifying the local feature of an image. Now, in this section, 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,

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

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:

$$\nu_{img}(\hat{\Phi}, d_x) = 1 - \frac{\arg \max_{c_j \in C - \hat{\Phi}(p_x)} z(d_x, c_j)}{\arg\max_{c_i \in C} z(d_x, c_i)} \; .$$

This whole image classification confidence can be used to decide whether or not the predicted label has an high probability to be correct. In the experimental results Section 7 we will show that the proposed confidence is reasonable.

6 Evaluation Settings

For evaluating the various classifiers we need at least: a data set, an interest points detector, a local feature extractor, some performance measures. In the following, we present all the evaluation setting we used for the experimentation.

6.1 The Dataset

The dataset that we used for our tests is composed of 1,227 photos of landmarks located in Pisa and was used also in [2]. The photos have been crawled from Flickr, the well known on-line photo service. The dataset we built is publicly available. The IDs of the photos used for these experiments together with the assigned label and extracted features can be downloaded from [10]. In the following we list the classes that we used and the number of photos belonging to each class. In Figure 1 we reported an example for each class that are: *Leaning Tower* (119 photos); *Duomo* (130 photos); *Battistero*

Fig. 1. Example images taken from the dataset

(104 photos); *Camposanto Monumentale (exterior)* (46 photos); *Camposanto Monumentale (field)* (113 photos); *Camposanto Monumentale (portico)* (138 photos); *Chiesa della Spina* (112 photos); *Palazzo della Carovana* (101 photos); *Palazzo dell'Orologio* (92 photos); *Guelph tower* (71 photos); *Basilica of San Piero* (48 photos); *Certosa* (53 photos).

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.

6.2 Performance Measures

For evaluating the effectiveness of the classifiers in classifying the documents of the *test set* we use the micro-averaged *accuracy* and micro- and macro-averaged *precision*, *recall* and F_1.

Micro-averaged values are calculated by constructing a global contingency table and then calculating the measures using these sums. In contrast macro-averaged scores are calculated by first calculating each measure for each category and then taking the average of these. In most of the cases we reported the micro-averaged values for each measure.

Precision is defined as the ratio between correctly predicted and the overall predicted documents for a specific class. *Recall* is the ratio between correctly predicted and the overall actual documents for a specific class. F_1 is the harmonic mean of *precision* and *recall*.

Note that for the *single-label* classification task, micro-averaged *accuracy* is defined as the number of documents correctly classified divided by the total number of documents in the *test set* and it is equivalent to the micro-averaged *precision*, *recall* and F_1 scores.

7 Experimental Results

In this section we report the experimental results obtained for all the 20 image similarity based and local feature based classifiers. For the image similarity based classifier results are reported for each similarity measure defined in Section 4.3. We also show that the proposed measure of confidence can be used to improve effectiveness on classified images accepting a small percentage of not classified objects.

7.1 Image Similarity Based Classifiers

In Table 1, *Accuracy* and macro averaged F_1 of the image similarity based classifiers for the 20 similarity functions defined in Section 4 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 (see Section 4). 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 Table 1, 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.

Let's first consider the approach used for managing the asymmetry of the distance functions discussed in Section 4.3. The best approach for all the similarity functions using both SIFT and SURF features is the fuzzy *and*, i.e., $s^{*,and}$. The more traditional approach $s^{*,Te}$ is the second best in most of the cases. On the contrary, $s^{*,Tr}$ always offers the worst performance. In other words, the best results were obtained when the similarity between two images is computed as the minimum of the similarity obtained considering as query in turn the test image local features and the training images. The result is the same both when using SIFT and SURF.

The *Hough Transform Matches Percentage* (s^h) similarity function is the best choice for both SIFT and SURF for all the 5 versions for managing the asymmetry. The geometric information considered by this function allows to obtain significantly better performance in particular for 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 worst 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.

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 test image closest neighbors in the training set are more relevant using SIFT than using SURF.

Table 1. Image similarity based classifier ($\hat{\Phi}^s$) performance obtained using various image similarity functions

similarity function			s^1 - Avg 1-NN					s^m - Perc. of Matches					s^σ - Avg Sim. Ratio					s^h - Hough Transform				
		version	Te	Tr	or	and	avg	Te	Tr	or	and	avg	Te	Tr	or	and	avg	Te	Tr	or	and	avg
Best	Acc	**SIFT**	.75	.52	.55	.85	.82	.88	.80	.81	.90	.88	.89	.80	.81	.91	.89	.92	.88	.88	**.93**	.91
		SURF	.79	.70	.73	.80	.82	.85	.73	.76	.88	.86	.82	.73	.75	.87	.84	.89	.76	.79	**.92**	.86
	F_1	**SIFT**	.72	.55	.56	.84	.84	.86	.80	.80	.89	.86	.87	.80	.81	.91	.88	.90	.87	.86	**.93**	.90
		SURF	.76	.67	.70	.78	.80	.83	.70	.74	.87	.84	.81	.68	.73	.86	.82	.87	.74	.77	**.89**	.85
k=1	Acc	**SIFT**	.73	.52	.55	.85	.82	.88	.78	.80	.90	.88	.89	.78	.80	.91	.88	.91	.87	.87	**.93**	.91
		SURF	.79	.63	.67	.80	.82	.81	.60	.62	.86	.79	.81	.63	.64	.84	.76	.87	.66	.68	**.90**	.81
	F_1	**SIFT**	.72	.55	.53	.84	.84	.86	.78	.80	.89	.86	.87	.79	.80	.90	.87	.90	.86	.86	**.92**	.90
		SURF	.76	.63	.67	.78	.80	.79	.65	.65	.84	.78	.80	.67	.67	.83	.77	.85	.68	.70	**.89**	.81
Best k	Acc	**SIFT**	*9*	*1*	*1*	*1*	*1*	*1*	*7*	*4*	*2*	*3*	*1*	*5*	*5*	*3*	*5*	*2*	*3*	*9*	*2*	*1*
		SURF	*3*	*6*	*8*	*1*	*1*	*20*	*28*	*42*	*14*	*20*	*8*	*23*	*17*	*11*	*14*	*21*	*35*	*39*	*11*	*18*
	F_1	**SIFT**	*1*	*1*	*1*	*1*	*1*	*1*	*7*	*4*	*3*	*3*	*1*	*5*	*5*	*3*	*5*	*2*	*8*	*5*	*9*	*9*
		SURF	*1*	*6*	*3*	*1*	*1*	*18*	*28*	*19*	*23*	*20*	*8*	*5*	*17*	*11*	*14*	*21*	*14*	*30*	*3*	*28*

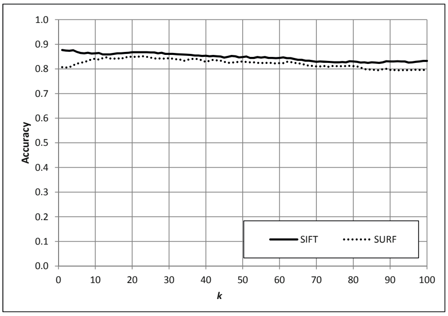

Fig. 2. Accuracy obtained by both SIFT and SURF for various k using the $s^{m,Te}$ similarity function with the image similarity based classifier

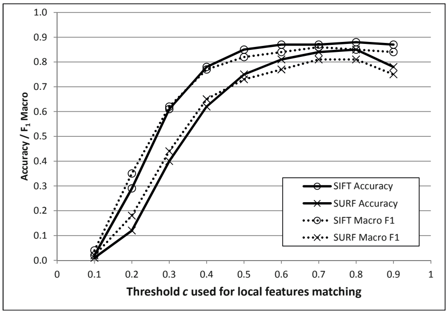

Fig. 3. Accuracy and Macro F_1 for various matching thresholds c, obtained by the image similarity based classifier ($\hat{\Phi}^s$) using the $s^{m,Tr}$ similarity and SIFT

This is more evident in Figure 2 where we report the *accuracy* obtained for k between 1 and 100 by both SIFT and SURF using the $s^{m,Te}$ similarity function. SIFT obtains the best performance for smaller values of k with respect to SURF. Moreover, SIFT performance is generally higher than SURF. It is interesting to note that performance obtained for $k = 1$ is typically just slightly worst than that of the best k. Thus, $k = 1$ gives very good performance even if a better k could be selected during a learning phase.

Two of the similarity measures proposed in Section 4.3 require a parameter to be set. In particular, the similarity measures *Percentage of Matches* (s^m) and *Hough Transform Matches Percentage* (s^h) use the matching function defined in Section 4.2 that requires a threshold for the distance ratio threshold (c) to be fixed in advance.

Table 2. Accuracy and Macro F_1 for the local feature based classifiers $\hat{\Phi}^m$ and for the kNN classifiers based on the various image similarity measures proposed for best k and related to the *and* version

	classifier	$\hat{\Phi}^m$	$\hat{\Phi}^s$			
	similarity		$s^{1, and}$	$s^{m, and}$	$s^{\sigma, and}$	$s^{h, and}$
Accuracy	**SIFT**	**.94**	.85	.90	.91	.93
	SURF	**.93**	.80	.88	.87	.92
F_1 Macro	**SIFT**	**.94**	.84	.89	.91	.93
	SURF	**.91**	.78	.87	.86	.89

In Figure 3 we report the performance obtained by using the *Percentage of Matches* classifier, i.e., the image similarity based classifier $\hat{\Phi}^s$ using the similarity measure s^m. For each distance ratio threshold c we report the best result obtained for k between 0 and 100. As mentioned in Section 4.2, in the paper where SIFT [14] was presented, Lowe suggested to use 0.8 as distance ratio threshold (c). The results confirm that the threshold proposed in [14] is the best for both SIFT and SURF and that the algorithm is stable around this values. In Table 1, results were reported for s^m and s^h with $c = 0.8$ for both SIFT and SURF.

Let us now consider the confidence ν_{doc} assigned to the predicted label of each image (see Section 4.1). This confidence can be used to obtain greater *accuracy* at the price of a certain number of false dismissals. In fact, a confidence threshold can be used to filter all the label assigned to an image with a confidence ν_{doc} less than the threshold. In Figure 4 we report the *accuracy* obtained by the $s^{h,and}$ measure using SIFT, varying the confidence threshold between 0 and 1. We also report the percentage of images in Te that were not classified together with the percentage of images that where actually correctly classified but that were filtered because of the threshold. Note that for $\nu_{doc} = 0.3$ the *accuracy* of classified objects rise from 0.93 to 0.99 obtained for $\nu_{doc} = 0$. At the same time the percentage of correctly predicted images that are filtered (i.e., the classifier does not assign a label because of the low confidence threshold ν_{doc}) is less than 10%.

This prove that the measure of confidence defined is meaningful. However, the best confidence threshold to be used depends on the task. Sometimes it could be better to try to *guess* the class of an image even if we are not sure, while in other cases it might be better to assign a label only if the classification has an high confidence.

7.2 Local Feature Based Classifier

In this section we compare the performance of the image similarity based classifiers using the 20 similarity measures defined in Section 4.3 with the local feature based classifier defined in 5.

In Table 2, we report *accuracy* and macro-averaged F_1 obtained by the *Local Feature Based Image Classifier* ($\hat{\Phi}^m$) using both SIFT and SURF together with the results obtained by the image similarity based approach ($\hat{\Phi}^s$) for the various similarity measures. Considering that in the previous section we showed that the fuzzy *and* approach

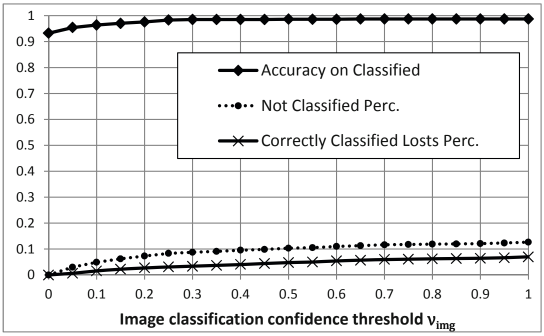

Fig. 4. Results obtained by the image similarity based classifier for similarity $s^{h,and}$ using SIFT, for various classification confidence thresholds (ν_{img})

performs better than the other, we only report the result obtained for the *and* version of each measures and for the best k.

The first observation is that the *Local Feature Based Image Classifier* ($\hat{\Phi}^m$) performs significantly better then any *Image Similarity Based Classifier*. In particular $\hat{\Phi}^m$ performs better then $s^{h,and}$, even if no geometric consistency checks are performed by $\hat{\Phi}^m$ while matches in $s^{h,and}$ are filtered making use of the Hough transform.

Even if in this paper we did not consider the computational cost of classification, we can make some simple observations. In fact, it is worth saying that the local feature based classifier is less critical from this point of view. First, because closest neighbors of local features in the test image are searched once for all in the Tr and not every time for each image of Tr. Second, because it is possible to leverage on global spatial index for all the features in Tr, to support efficient k nearest neighbors searching. In fact, the similarity function between two local features is the Euclidean distance, which is a metric. Thus, it could be efficiently indexed by using a metric data structures [18,15,4].

Regarding the local features used and the computational cost, we underline that the number of local features detected by the SIFT extractor is twice that detected by SURF. Thus, on one hand SIFT has better performance while on the other hand SURF is more efficient.

8 Conclusions

In this paper we addressed the problem of image content recognition using local features and kNN based classification techniques. We defined 20 similarity functions and compared their performance on a image content landmarks recognition task. We found that a two-way comparison of two images based on fuzzy and allows better performance than the standard approach that compares a query image with the ones in a training set. Moreover, we showed that the similarity functions relying on matching of local features that makes use of geometric constrains perform slightly better than the others.

Finally, we defined a novel kNN classifier that first assigns a label to each local feature of an image and then labels the whole image by considering the labels and the confidences assigned to its local features.

The experiments showed that our proposed local features based classification approach outperforms the standard image similarity kNN approach in combination with any of the defined image similarity functions, even the ones considering geometric constrains.

Acknowledgements. This work was partially supported by the VISITO Tuscany project, funded by Regione Toscana, in the POR FESR 2007-2013 program, action line 1.1.d, and the MOTUS project, funded by the Industria 2015 program.

References

1. Amato, G., Falchi, F.: Local feature based image similarity functions for kNN classfication. In: Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011), vol. 1, pp. 157–166. SciTe Press (2011)
2. Amato, G., Falchi, F., Bolettieri, P.: Recognizing landmarks using automated classification techniques: an evaluation of various visual features. In: Proceeding of the Second Interantional Conference on Advances in Multimedia (MMEDIA 2010), Athens, Greece, June 13-19, pp. 78–83. IEEE Computer Society (2010)
3. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111–122 (1981)
4. Batko, M., Novak, D., Falchi, F., Zezula, P.: Scalability comparison of peer-to-peer similarity search structures. Future Generation Comp. Syst. 24(8), 834–848 (2008)
5. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
6. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
7. Chen, T., Wu, K., Yap, K.H., Li, Z., Tsai, F.S.: A survey on mobile landmark recognition for information retrieval. In: MDM 2009: Proc. of the Tenth International Conference on Mobile Data Management, pp. 625–630. IEEE (2009)
8. Dudani, S.: The distance-weighted k-nearest-neighbour rule. IEEE Transactions on Systems, Man and Cybernetics SMC-6(4), 325–327 (1975)
9. Fagni, T., Falchi, F., Sebastiani, F.: Image classification via adaptive ensembles of descriptor-specific classifiers. Pattern Recognition and Image Analysis 20, 21–28 (2010), `http://dx.doi.org/10.1134/S1054661810010025`
10. Falchi, F.: Pisa landmarks dataset (2011), `http://www.fabriziofalchi.it/pisaDataset/` (last accessed on March 3, 2011)
11. Google: Google Goggles (2011), `http://www.google.com/mobile/goggles/` (last accessed on March 3, 2011)
12. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. Int. J. Comput. Vision 87(3), 316–336 (2010)
13. Kennedy, L.S., Naaman, M.: Generating diverse and representative image search results for landmarks. In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 297–306. ACM Press, New York (2008)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

15. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers Inc., San Francisco (2005)
16. Serdyukov, P., Murdock, V., van Zwol, R.: Placing flickr photos on a map. In: Allan, J., Aslam, J.A., Sanderson, M., Zhai, C., Zobel, J. (eds.) SIGIR, pp. 484–491. ACM (2009)
17. Yeh, T., Tollmar, K., Darrell, T.: Searching the web with mobile images for location recognition. In: CVPR (2), pp. 76–81 (2004)
18. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. In: Advances in Database Systems, vol. 32. Springer, Heidelberg (2006)
19. Zheng, Y., Song, M.Z., Adam, Y., Buddemeier, H., Bissacco, U., Brucher, A., Chua, F., Neven, T.S., Tour, H.: The world: Building a web-scale landmark recognition engine. In: CVPR, pp. 1085–1092. IEEE (2009)