kNN based image classification relying on local feature similarity

Giuseppe Amato
ISTI-CNR
via G. Moruzzi, 1
Pisa, Italy
giuseppe.amato@isti.cnr.it

Fabrizio Falchi
ISTI-CNR
via G. Moruzzi, 1
Pisa, Italy
fabrizio.falchi@isti.cnr.it

ABSTRACT

In this paper, we propose a novel image classification approach, derived from the kNN classification strategy, that is particularly suited to be used when classifying images described by local features. Our proposal relies on the possibility of performing similarity search between image local features.

With the use of local features generated over interest points, we revised the single label kNN classification approach to consider similarity between local features of the images in the training set rather than similarity between images, opening up new opportunities to investigate more efficient and effective strategies. We will see that classifying at the level of local features we can exploit global information contained in the training set, which cannot be used when classifying only at the level of entire images, as for instance the effect of local feature cleaning strategies.

We perform several experiments by testing the proposed approach with different types of image local features in a touristic landmarks recognition task.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.1 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Image indexing, image classification, recognition, landmarks, local features

1. INTRODUCTION

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

SISAP '10 , September 18-19, 2010, Istanbul, Turkey.

Traditional kNN classification algorithms decide about the class of an image by searching for the k images of the training set most similar to the image to be classified, and by performing a class weighted frequency analysis. The k closest images are identified relying upon a similarity measure between images.

As an alternative approach, in this paper we propose a new kNN based classification method relying on images represented by means of local features generated over interest points, as for instance SIFT [12] or SURF [6]. With the use of local features and interest points, kNN classification algorithms were revised to consider similarity between local features of the images in the training set rather than similarity between images, opening up new opportunities to investigate more efficient and effective strategies. In fact, direct use of similarity between local features is generally easier to be handled than sets of local features. In addition, we will see that classifying at the level of local features we can exploit global information contained in the training set, which cannot be used when classifying only at the level of entire images.

In this paper, we also study the effect of local feature cleaning strategies and we perform several experiments by testing the proposed approach with different types of image local features in a touristic landmarks recognition task.

The paper is organized as follows. In Section 4 we present an image similarity measures relying on local features to be used with a kNN classification algorithm. Section 5 proposes our new classification approach and defines four local feature classifiers. Section 6 discusses the use of the proposed approach for training local features cleaning. Finally, Sections 7 and 8 presents the experiments that we carried out to assess the proposed technique.

2. RELATED WORK

The first approach to recognizing location from mobile devices using image-based web search was presented in [16]. 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 [14] methods for placing photos uploaded to Flickr on the Wolrd 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 [11] a combination of context- and content-based tools were used to generate representative sets of images for lo-

cation driven features and landmarks. SIFT features were used to establish links between different images which contain views of a single location. However, this information is combined with the textual metadata while we are only considering content-based classification.

In [18], 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 [1]. 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.

An important survey of local features detectors is [15]. However, the various local features are not compared. In this paper we decided to use for each local feature the detector proposed by the authors of each feature.

In [8] 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 [10], various MPEG-7 descriptors have been used to build kNN classifier committees and test were performed on a slabs of stones dataset. In [7] 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 [12] and SURF [6] local features. In this section, we briefly describe both of them.

3.1 SIFT

The Scale Invariant Feature Transformation (SIFT) [12] 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 [3].

3.2 SURF

The basic idea of Speeded Up Robust Features (SURF) [6] 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 interest points and extracting the SURF features, we used the public available noncommercial software developed by the authors [4].

4. THE BASELINE

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. This 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) [9] 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 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 labeled documents d_i belonging to the *training set* ordered with respect to 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, labeled 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,

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

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 [12]. 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.

4.2.1 Local Feature Similarity

The computer vision literature related to local features, uses generally 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 [12, 6]. Let $d(p_1, p_2) \in [0, 1]$ be the normalized distance between two local features p_1 and p_2. We 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.

4.2.2 Local Features Matching

A useful aspect that is often used when dealing with local features is the concept of local feature matching. In [12], a distance ratio matching scheme was proposed that has also been adopted by [6] 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:

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

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 [12], Lowe proposed to use $c = 0.8$ reporting that this threshold allows to eliminate 90% of the false matches while discarding less than 5% of the correct matches. In Section 8 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. However, the best performing classifier defined in Section 5.1.4 does not require this parameter because it is based on fuzzy matching.

4.2.3 Image Similarity Measure

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. 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 as defined in Section 4.2.2. For simplicity, we indicate the number of local features in an image d_x as $|d_x|$.

It is worth to say that this image similarity measure is not metric. In fact, it is not even symmetric. Thus, it could not be efficiently indexed by the various index structure defined for metric spaces (see [17]).

5. LOCAL FEATURE BASED IMAGE CLASSIFICATION

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 (see [17]) 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 baseline approach. In other words we are able to improve at the same time both efficiency and effectiveness.

In Section 5.1 we define four distinct algorithms for local feature classification, i.e. step 1. All the proposed algorithms require searching for similar local features for each of the local features belonging to the image we have to classify. For the second step, we use the confidence-rated majority vote approach reported in Section 5.1.2.

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 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 8 confirm the validity of this assumption.

5.1 Local Features Classification

In the following we propose four different strategies for obtaining local feature (LF) classifiers and the corresponding confidence value.

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.

5.1.1 1-NN LF Classifier – $\hat{\Phi}^f(p_x)$

The simplest way to assign a label to a local feature is to consider the label of its closest neighbor in Tr. The *1-NN Local Features Classifier* assigns to a local feature p_x, the label of the closest neighbor in Tr. The confidence of the classification assigned is the similarity between p_x and is nearest neighbor. Formally:

$$\begin{cases} \hat{\Phi}^f(p_x) = \Phi(NN_1(p_x, Tr)). \\ \nu(\hat{\Phi}^f, p_x) = 1 - d(p_x, NN_1(p_x, Tr)) \end{cases}$$

Please note that this classifier does not require any parameter to be set. Moreover, the similarity search to perform over the local features training set is a simple 1-NN.

5.1.2 Weighted kNN LF Classifier – $\hat{\Phi}^k(p_x)$

This *Weighted kNN LF Classifier* is an extension of the *single-label distance-weighted kNN* reported in Section 4. First a kNN search on the local features of the Tr is executed. The result of such operation is a list of labeled features p_i belonging to Tr ordered with respect to decreasing values of the similarity $s(p_x, p_i)$. The label $\hat{\Phi}^k(p_x)$ assigned to the document d_x by the classifier is the class $c_j \in C$ that maximizes the sum of the similarity between p_x and the features p_i, labeled c_j, in the kNN results list $\chi^k(p_x)$. The confidence is then based on the ratio between second best and best class. Formally, we have to compute a score $z^k(p_x, c_j)$ for each class:

$$z^k(p_x, c_j) = \sum_{p_i \in \chi^k(p_x) \ : \ \Phi(p_i) = c_j} s(p_x, p_i)$$

Then the predicted label $\hat{\Phi}^k$ and the confidence ν are defined as follows:

$$\begin{cases} \hat{\Phi}^k(p_x) = \arg\max_{c_j \in C} z^k(p_x, c_j) \\ \nu(\hat{\Phi}^k, p_x) = 1 - \dfrac{\arg\max_{c_j \in C - \hat{\Phi}^k(p_x)} z^k(p_x, c_j)}{\arg\max_{c_i \in C} z^k(p_x, c_i)} \end{cases}$$

Note that for $k = 1$ we have $\hat{\Phi}^k(p_x) = \hat{\Phi}^f(p_x)$, while the measure of confidence is different. In fact, $\hat{\Phi}^k$ always assigns 1 as confidence when $k = 1$ is set while $\hat{\Phi}^f$ considers the first nearest neighbor similarity as measure of confidence. We will see in Section 5.2 that this difference is important for the whole image classification

This classifier requires the parameter k to be chosen.

5.1.3 LF Matching Classifier – $\hat{\Phi}^m(p_x)$

The *Local Feature Matching Classifier* decides the candidate label similarly to the *1-NN Local Features Classifier*, i.e.:

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

The very difference is the computation of the confidence value of the selected label which is evaluated using the idea of the distance ratio discussed in Section 4.2.2.

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 $\dot{\sigma}$ 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 [12], 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 similarity ratio $\dot{\sigma}$ as:

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

where $NN_2^*(p_x, Tr)$ is the closest neighbor that is known to be labeled differently than the first as suggested in [12].

Note that searching for $NN_2^*(p_x, Tr)$ can not be directly translated in a standard k nearest neighbors search. However, the kNN implementation in metric spaces is generally performed starting with an infinite range and reducing it during the evaluation, considering at any time the actual NN_k. The very same approach can be used for searching $NN_2^*(p_x, Tr)$. In fact, while k is not known in advance, the actual NN_2^* during the similarity search, can be used to reduce the range of the query. Thus, the similarity search needed for the evaluation of $\dot{\sigma}(p_x, T_r)$ can be implemented slightly modifying the standard algorithms developed for metric spaces (see [17]).

The parameter c used in the definition of the confidence is the equivalent of the one used in [12] and [6]. We will see in Section 8 that $c = 0.8$ proposed in [12] 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.

5.1.4 Weighted LF Distance Ratio Classifier – $\hat{\Phi}^w(p_x)$

The *Weighted LF Distance Ratio Classifier* is an extension of the *LF Matching Classifier* defined in the previous section. However, the confidence is not binary but is a fuzzy measure derived from the distance ratio. Given that the greater the confidence the better the matching, we define the assigned label and the confidence as:.

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

The intuition is that it could be better not to filter non-matching features on the basis of the distance ratio, but to use $1 - \dot{\sigma}(p_x, t_r)$ as a measure of confidence to be used during the classification of the whole image. Then, 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 neither a distance ratio threshold c or k. Thus, this classifier has no parameters at all.

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 8 we will show that the proposed confidence is reasonable.

6. LOCAL FEATURES CLEANING

The number of local features obtained from an image is typically high. For instance, from a 500x500 pixels image we obtained an average of about 1,000 key points. Thus, it would be very important to reduce the number of local features in the training set, still maintaining an high effectiveness. This reduction allows better efficiency and reduce memory occupation. Moreover, deleting worst interest points could also results in better effectiveness.

The strategy that we used to decide which local features of Tr can be eliminated, was to assign a score to each local features of Tr considering Tr itself as training set with one of the local feature classifiers defined Section 5.1.4.

The discrepancy between pre-assigned label in the training set and the label predicted by the classifier together with the confidence expressed in predicting the label, was used as an indication for eliminating the local feature (remember that as reported in Section 5 the label pre-assigned to a local feature is the label assigned to the image it belongs to).

Given the good results reported in Section 8, we decided to test the *Weighted Ratio Based Matching*. We use the *Leave-One-Out* technique for classifying each point p_x in each image $d_x \in Tr$. During the classification we leave out the image d_x. Thus, the classification of the features in d_x is only based on the local features in $Tr - d_x$.

The discrepancy between the predicted label and the pre-assigned label is measured by using a score $e(p_x)$ computed as follows:

Figure 1: Example images taken from the dataset

$$e(p_x) = \begin{cases} \nu(\hat{\Phi}^w, d_x) & \text{if } \hat{\Phi}^w(p_x) = \Phi(p_x) \\ -\nu(\hat{\Phi}^w, d_x) & \text{otherwise} \end{cases}$$

The absolute value of the score of p_x is $\nu(\hat{\Phi}^w, d_x)$ that is the confidence of the predicted label. The sign is positive if the feature was correctly classified and negative otherwise. If the predicted label has an high confidence and it is wrong with respect to the pre-assigned label, then the score is very low and it is a good candidate to be eliminated.

In the experiment section we use a threshold on the score to control the amount of promising local features considered.

7. 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.

7.1 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 [5]. 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 [2]. 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 in the same order as they are reported in the list below:

- *Leaning Tower* (119 photos) – leaning campanile
- *Duomo* (130 photos) – the cathedral of St. Mary
- *Battistero* (104 photos) – the baptistery of St. John
- *Camposanto Monumentale (exterior)* (46 photos)
- *Camposanto Monumentale (field)* (113 photos)
- *Camposanto Monumentale (portico)* (138 photos)
- *Chiesa della Spina* (112 photos) – Gothic church
- *Palazzo della Carovana* (101 photos) – building
- *Palazzo dell'Orologio* (92 photos) – building
- *Guelph tower* (71 photos)

- *Basilica of San Piero* (48 photos) – church of St. Peter
- *Certosa* (53 photos) – the charterhouse

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.

7.2 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.

8. EXPERIMENTAL RESULTS

In this section we report the experimental results obtained for both the image similarity based and local feature based classifiers. We also show that our measure of confidence can be used to improve effectiveness on classified images accepting a small percentage of not classified objects. Finally, we report preliminary results of our local feature cleaning approach.

8.1 The baseline

We first perform an evaluation of the baseline technique ($\hat{\Phi}^s$) based on image similarity. This evaluation will be used to assess the optimal parameters c and k for the baseline approach, to make a fair comparison against the proposed methods in next section.

The image similarity approach ($\hat{\Phi}^s$) uses the matching function defined in Section 4.2.2 that requires a threshold for the distance ratio c to be fixed in advance. In Figure 2 we report the performance obtained varying the matching threshold c. For each matching threshold c we report the best result obtained for k between 0 and 100. As mentioned in Section 4.2.1, in the paper where SIFT [12] were presented, Lowe suggested to use 0.8 as distance reatio threshold (c). The results confirm that the threshold proposed in [12] is the best for both SIFT and SURF and that the algorithm is stable around this values.

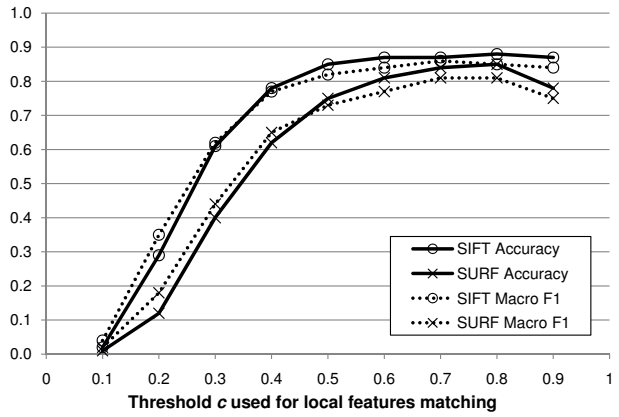

Figure 2: Accuracy and Macro F_1 obtained for various matching threshold by the similarity based approach.

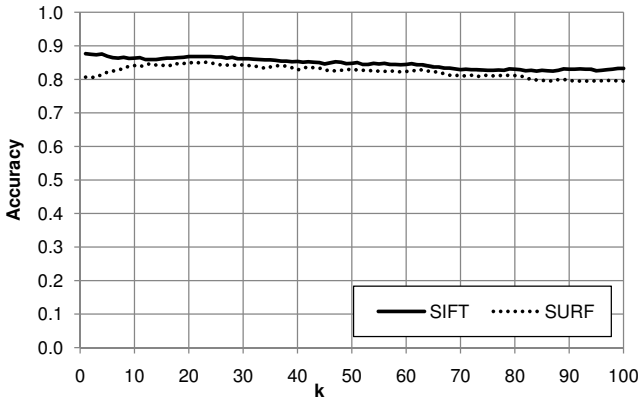

Figure 3: Accuracy obtained for various k using the image similarity based approach.

In Figure 3 we report the *accuracy* obtained for k between 1 and 100 by both SIFT and SURF for $c = 0.8$. The parameter k determines the number of closest neighbors retrieved in order to classify a given image using the *single-label distance-weighted kNN* technique (see Section 4).

The results show that SIFT performs generally better than SURF. Moreover, the k for which the best performance was obtained is typically much higher for SURF than SIFT. In other words, the test image closest neighbor in the training set is more relevant using SIFT than using SURF while the results obtained for higher k are almost the same. Specifically, the best result was obtained with $k = 1$ in case of SIFT and with $k = 20$ in case of SURF.

8.2 Local Feature Based Image Classification

In this section we compare the baseline approach against the proposed method for local feature based image classification. In this comparison we use the optimal settings of the parameters for the baseline approach, discussed in previous section. Specifically, we set c to 0.8, for both SIFT and SURF, while we use $k = 1$ for SIFT and $k = 20$ for SURF. We use $c = 0.8$ also for the *Local Feature Matching Classifier* ($\hat{\Phi}^m$).

									best	baseline
	classifier	$\hat{\Phi}^f$	$\hat{\Phi}^1$	$\hat{\Phi}^5$	$\hat{\Phi}^{10}$	$\hat{\Phi}^{25}$	$\hat{\Phi}^{50}$	$\hat{\Phi}^m$	$\hat{\Phi}^w$	$\hat{\Phi}^s$
Accuracy	**SIFT**	0.901	0.901	0.855	0.818	0.756	0.691	0.945	**0.952**	0.877
	SURF	0.883	0.881	0.841	0.794	0.714	0.668	0.927	**0.928**	0.851
F_1 Macro	**SIFT**	0.806	0.883	0.809	0.748	0.657	0.575	0.940	**0.947**	0.864
	SURF	0.791	0.866	0.804	0.727	0.606	0.542	0.915	**0.922**	0.828

Figure 4: Accuracy and Macro F_1 for the proposed local feature based classifiers and for the baseline Φ^s.

In Figure 5, we report *accuracy* and macro-averaged F_1 obtained by the various classifiers using both SIFT and SURF together with the results obtained by the image similarity based approach ($\hat{\Phi}^s$).

The first observation is that all the local feature based approaches perform significantly better then the baseline ($\hat{\Phi}^s$). In particular, both the *Local Features Matching* ($\hat{\Phi}^m$) and the *Weighted LF Distance Ratio* ($\hat{\Phi}^w$) classifiers outperform all the others. This is true using both SIFT and SURF features. The best overall performance were obtained by $\hat{\Phi}^w$ which is slightly better than its non-weighted counterpart ($\hat{\Phi}^m$). Moreover, it has another and probably even more important advantage – it does not require any parameter to be set.

The performance measures obtained by the *Weighted kNN Local Features Classifier* for various k ($\hat{\Phi}^1$, $\hat{\Phi}^5$, $\hat{\Phi}^{10}$, $\hat{\Phi}^{20}$ and $\hat{\Phi}^{50}$) show that the best results are obtained when considering only the closest neighbor (i.e., $k = 1$). Moreover, $\hat{\Phi}^k$ for $k = 1$ outperforms the *1-NN Local Features Classifier* ($\hat{\Phi}^f$) in terms of F_1 while the *accuracy* values are equivalent.

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 are 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, a metric data structure (see [17] and [13]) could be used to improve efficiency.

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.

Let us now consider the confidence ν_{img} assigned to the predicted label of each image (see Section 5.2). 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_{img} less than the threshold. In Figure 5 we report the *accuracy* obtained by the Φ^w classifier using SURF, 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 $\nu_{img} = 0.5$ the *accuracy* of classified objects rise from 0.928 to 0.982

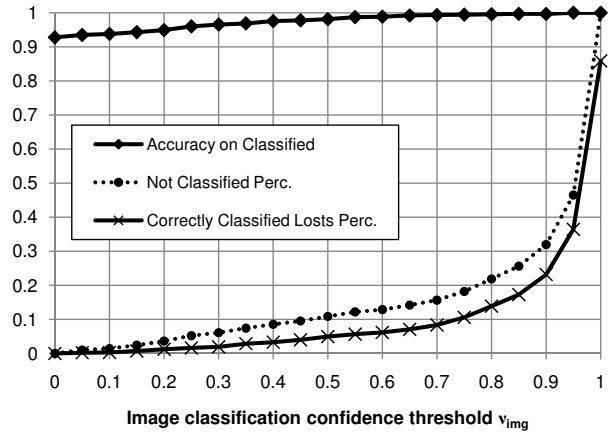

Figure 5: Accuracy on classified obtained by the $\hat{\Phi}^w$ classifier using SURF, for various image classification confidence thresholds.

obtained for $\nu_{img} = 0$. 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_{img}) is less than 5%.

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.

8.3 Local Features Cleaning

We now consider the local features cleaning algorithm defined in Section 6. The goal of this process is the reduction of the total number of local features while maintaining good results. In Figure 6 we report the performance measures obtained by the $\hat{\Phi}^w$ classifier on using only local features from the *training set* that were evaluated higher than various local features evaluation thresholds e. Together with *accuracy* and macro-averaged F_1, we report the percentage of the local features in Tr that were used, i.e., obtained an evaluation of more than e. Note that, as defined in Section 6, the local features were evaluated only considering Tr while the performance is evaluated on Te.

Removing the local features with an evaluation threshold $e < 0$ we maintain just 40 percent of the total images in Tr, while both *accuracy* and macro F_1 only slightly decreases. Moreover, for $e < 0.2$ we maintain just 20 percent of the total features in Tr while the accuracy decreased of about 0.1. We consider this result to be very promising.

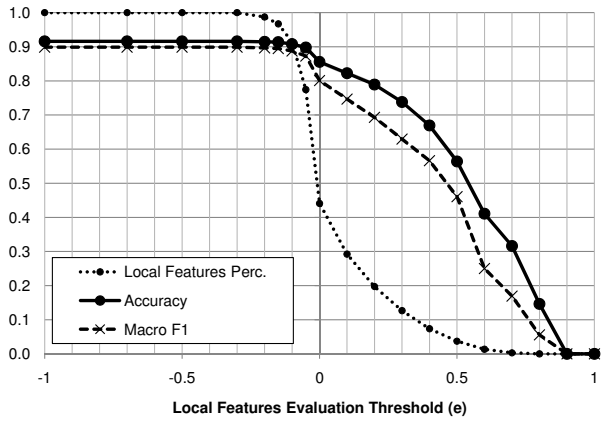

Figure 6: Accuracy and Macro F_1 obtained by $\hat{\Phi}^w$ using only SURF local features from the *training set* that were evaluated more than e.

9. CONCLUSIONS

In this paper, we defined a novel image classification approach, derived from the kNN classification strategy that classify images in two steps: first each local feature is classified considering the local features of a training set; second the whole image is classified considering the class assigned to each local feature and the confidence of these classifications. Four algorithms for the classification of local features were defined and tested.

The experimental results proved that this novel approach outperforms traditional approaches based on image similarity functions. Moreover, the confidence measure for the assigned label proved to be meaningful and useful in improving the accuracy of the classification.

It worths noting that the best results have been obtained by the Weighted LF Distance Ration Classifier ($\hat{\Phi}^w$). The nice feature of this classifier, in addition to be the best performing, is that it does not require any parameter to be set and tuned. Therefore, it is also the most easy and intuitive to be used, and the less prone to tuning errors.

Finally, we also defined a local features cleaning algorithm based on a features evaluation that can be used to reduce the number of local features in a training set at the price of slight efficacy degradation.

10. ACKNOWLEDGMENTS

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.

11. REFERENCES

[1] Google goggles. `http://www.google.com/mobile/goggles/`. last accessed on 30-March-2010.

[2] Pisa landmarks dataset. `http://www.fabriziofalchi.it/pisaDataset/`. last accessed on 30-March-2010.

[3] SIFT keypoint detector. `http://people.cs.ubc.ca/~lowe/`. last accessed on 30-March-2010.

[4] SURF detector. `http://www.vision.ee.ethz.ch/~surf/`. last accessed on 30-March-2010.

[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] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[7] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*. IEEE Computer Society, 2008.

[8] T. Chen, K. Wu, K.-H. Yap, Z. Li, and F. S. Tsai. A survey on mobile landmark recognition for information retrieval. In *MDM '09*, pages 625–630. IEEE Computer Society, 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] T. Fagni, F. Falchi, and F. Sebastiani. Adaptive committees of feature-specific classifiers for image classification. In *Image Mining. Theory and Applications. Proceedings of the 2nd International Workshop on Image Mining Theory and Applications (IMTA-09)*, pages 113–122. INSTICC Press, 2009.

[11] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 297–306, New York, NY, USA, 2008. ACM.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[14] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 484–491, New York, NY, USA, 2009. ACM.

[15] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, 2008.

[16] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *CVPR (2)*, pages 76–81, 2004.

[17] 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.

[18] 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 *CVPR*, pages 1085–1092. IEEE, 2009.