Image Classification via Adaptive Ensembles of Descriptor-Specific Classifiers[1]

T. Fagni, F. Falchi, and F. Sebastiani

Istituto di Scienza e Tecnologia dell' Informazione Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi 1, 56124 Pisa, Italy

e-mail: fabrizio.sebastiani@isti.cnr.it; tiziano.fagni@isti.cnr.it; fabrizio.falchi@isti.cnr.it

Abstract—An automated classification system usually consists of (i) a supervised learning algorithm for automatically generating classifiers from training data, and (ii) a representation scheme for converting the training objects into vectorial representations of their content. In this work, we take a detour from this tradition and present an approach to image classification based on an adaptive ensemble of classifiers, each specialized on classifying images based on a single "descriptor." Each descriptor focuses on a different aspect, or perspective, of images; an ensemble of descriptor-specific classifiers can thus be seen as a committee of experts, each viewing the problem to be solved with a different slant, of from a different viewpoint. We test four different ways to set up such an ensemble, based on different ways of leveraging on the individual responses returned by each member of the ensemble, and on how confident these members are on their responses. We test this approach by using five different MPEG-7 descriptors on the task of assigning photographs of stone slabs to classes representing different types of stones. Our experimental results show important accuracy improvements with respect to a baseline in which a single classifier, working an all five descriptors at the same time, is employed.

Keywords: image classification, supervised learning, classifier committees, classifier ensembles, metric spaces, *k*-nearest neighbours classifier, MPEG-7.

DOI: 10.1134/S1054661810010025

1. INTRODUCTION

An automated classification system is usually described by specifying two essential components. The first such component is a scheme for internally representing the data items that are the objects of classification. This representation scheme, that is usually vectorial in nature, is usually such that a suitable notion of similarity (or closeness) between the representations of two data items can be defined. Here, "suitable" means that similar representations must be attributed to data items that are perceived to be similar. If so, a classifier may identify, within the space of all the representations of the data items, a limited region of space where the objects belonging to a given class lie; here, the assumption of course is that data items that belong to the same class are "similar." The second component is a supervised learning algorithm that takes as input the representations of training data items and generates a classifier from them.

In this work, we address *single-label image classification*, i.e., the problem of setting up an automated system that classifies an image into exactly one from a predefined set of classes. Image classification has a long history (see e.g., [11]), most of which has produced systems that conform to the pattern described at the beginning of this section.

In this paper, we take a detour from this pattern and present an approach to image classification based on an adaptive ensemble (or "committee") of classifiers, each specialized on classifying images based on a different representation of the same image. Each such representation may be seen as describing the image from a different viewpoint, or as looking at the image with a different slant; these different viewpoints will here be called *descriptors*. An ensemble of descriptor-specific classifiers can thus be seen as an ensemble of experts, in which each expert views the problem to be solved from a different perspective, brings to bear a different type of expertise, and returns an independent opinion. Of course, all these independent opinions finally need to be combined into (i.e., need to contribute to) a final decision by a *combination rule*, i.e., an algorithm that specifies how the final decision depends on the opinions of the ensemble members. In this work we test four different combination rules, based on different ways of leveraging (i) on the individual responses returned by each member of the ensemble and (ii) on how confident these members were on their responses.

The ensembles that we use are adaptive, in the sense that, for each image to be classified, they dynamically decide which among the ensemble members should be entrusted with the classification decision, or decide whose decisions should be trusted more. We study experimentally four different techniques of combining the decisions of the individual classifiers.

[1]The article is published in the original.

As a technique for generating the individual members of the classifier ensemble we use *distance-weighted k-nearest neighbours*, a well-known example-based learning technique. While other methods might have been used in principle, this method has the advantage that it does not require a vectorial representation of data items to be defined, since it simply requires that, given two data items, a distance between them is defined. In the discussion that follows this will allow us to abstract away from notions having to do with the vectorial representation of our data items (and from the fact that these representations have a vectorial nature at all), and simply specify our methods in terms of distance functions between data items.

In this work, we experimentally test our approach on a dataset consisting of about 2,500 photographs of stone slabs, each classified as belonging to one of 37 different types of stone. As the descriptors that make up our classifier ensemble we choose five different visual descriptors from the MPEG-7 standard. Finally, since distance computation is so fundamental to our methods (we will see that it plays a key role in the implementation of both the individual ensemble members and the combination rule), we also study how to compute distances between data items efficiently, and implement an efficient system that makes use of metric data structures explicitly devised for "nearest-neighbour search."

1.1. Outline of the Paper

The rest of the paper is organized as follows. We start by reviewing related work in Section 2. Section 3 describes in detail the various ensemble-based learning algorithms. In Section 4 we move to describing our experiments, and to discussing the conclusions that can be drawn from them. Section 5 deals instead with efficiency issues, discussing how we have implemented efficiently our learning algorithms by recurring to metric data structures. We conclude in Section 6 by pointing out avenues for future work.

2. RELATED WORK

Classification approaches based on classifier ensembles have a long history, which dates back at least to the 70's [18] (see [5, 14] for two general treatments). For instance, even restricting the analysis to ensembles of classifiers generated by neural network technology, ensembles have been proposed in which the classifiers vary in terms of the initial random weights with which the network is initialized, in terms of the network architecture, in terms of the network type, or in terms of the training data [8].

Classifier ensembles for image classification have been proposed before, with applications in handwriting recognition [20], management of remote-sensing data [4, 11, 16], and others. However, our work is unique in this literature in that, for us, the difference between the ensemble members is in the aspect of images they concentrate on, rather than, e.g., on the learning algorithm used.

Image classification based on MPEG-7 visual descriptors is addressed in [17]. However, the approach of [17] is very different from ours, since the authors choose to use a single learning algorithm which takes as input a single representation that combines the contributions of the individual MPEG-7 descriptors; no classifier ensemble is thus involved.

Concerning the classification of photographs of stone slabs, Martinez-Alajarin and colleagues [13] also present an image classification method applied to the classification of photographs of marble slabs. Unfortunately, their evaluation is limited, since their dataset consists of only 75 images subdivided into three classes. Unlike in our dataset, the three classes are not related to the type of stone, but to its quality ("extra," "commercial," "low quality"), which is dependent on texture considerations alone, thus making the use of different visual descriptors useless. Their work is mostly focused on the acquisition of the internal representations of the images, and uses a standard neural network as a learning algorithm.

3. AUTOMATIC IMAGE CLASSIFICATION BY MEANS OF ADAPTIVE, DESCRIPTOR-SPECIFIC ENSEMBLES

Given a set of images D and a predefined set of *classes* (also known as *labels* or *categories*) $C = \{c_1, ..., c_m\}$, *single-label* (also known as 1-of-m, or *multiclass*) *image classification* (SLC) is the task of automatically building a single-label image classifier, i.e., a function $\hat{\Phi}$ that predicts, for any $d_i \in D$, the correct class $c_j \in C$ to which d_i belongs. More formally, the task is that of approximating, or estimating, an unknown *target function* $\Phi: D \longrightarrow C$, that describes how images ought to be classified, by means of a function $\hat{\Phi}: D \longrightarrow C$, called the *classifier*, such that Φ and $\hat{\Phi}$ "coincide as much as possible."[2]

The solutions we will give to this task will be based on automatically generating the classifiers $\hat{\Phi}$ by *supervised learning*. This will require a set Ω of images as input which are manually labelled according to the classes C, i.e., such that for each image $d_i \in \Omega$ the value of the function $\Phi(d_i)$ is known. In the experiments we present in Section 4 the set Ω, will be partitioned into two subsets Tr (the *training set*) and Te (the *test set*), with $Tr \cup Te = \Omega$ and $Tr \cap Te = \emptyset$; Tr will be used in order to generate the classifiers $\hat{\Phi}$ by means of supervised learning methods, while Te will be used in order to test the effectiveness (i.e., accuracy) of the generated classifiers.

[2] Consistently with most mathematical literature we use the caret symbol (^) to indicate estimation.

3.1. Image Classifiers as Ensembles of Single-Descriptor Classifiers

The image classifier $\hat{\Phi}: D \longrightarrow C$ that we will generate will actually consist of a *classifier ensemble* (also known as *classifier committee*), i.e., of a tuple $\hat{\Phi} = (\hat{\Phi}^1, ..., \hat{\Phi}^n)$ of classifiers, where each classifier $\hat{\Phi}^s$ is specialized in analyzing the image from the point of view of a single descriptor $f_s \in F$, where F is a set of image descriptors[3]. For instance, a classifier $\hat{\Phi}^{colour}$ might be set up that classifies the image only according to the distribution of colors within it, and a further classifier $\hat{\Phi}^{texture}$ might be set up that classifies the image according to texture considerations.

The "aggregate" classifier $\hat{\Phi}$ takes its classification decision by combining the decisions returned by the descriptor-specific classifiers $\hat{\Phi}^s$ by means of an *adaptive* combination rule, i.e., a combination rule that pays particular attention to those $\hat{\Phi}^s$'s that are expected to perform more accurately on the particular image that needs to be classified. This is advantageous, since different descriptors could be the most revealing for classifying different types of images; e.g., for correctly recognizing that an image belongs to class c', colour-related considerations might be more important than texture-related ones, while the contrary might happen for class c''. In the techniques that we have used in this work, whether and how much a given descriptor is effective for classifying a given image is automatically detected, and automatically brought to bear in the classification decision.

For implementing the classifier ensemble, i.e., for combining appropriately the outputs of the $\hat{\Phi}^s$'s, we will experiment with four different techniques. In Sections 3.1.1 to 3.1.4 we will describe these techniques, while in Section 3.2 we will describe how to generate the individual members of these ensembles.

3.1.1. Dynamic classifier selection. The first technique we test is *dynamic classifier selection* (DCS) [7, 10, 19]. This technique consists in

1. identifying the set

$$\chi^w(d_i) = \arg\min_{d_p \in Tr}{}^{w} \delta(d_i, d_p) \quad (1)$$

of the w training examples closest to the test image d_i, where $\delta(d', d'')$ is a (global, i.e., not descriptor-specific) measure of distance that ranges on [0, 1], and to be discussed more in detail in Section 5);

2. attributing to each descriptor-specific classifier $\hat{\Phi}^s$ a score $g(\hat{\Phi}^s, d_i)$ that measures how well it classifies the examples in $\chi^w(d_i)$; see below for details;

3. adopting the decision of the classifier with the highest score; i.e., $\hat{\Phi}(d_i) = \hat{\Phi}^t(d_i)$ where $\hat{\Phi}^t = \arg\max_{\hat{\Phi}^s \in \hat{\Phi}} g(\hat{\Phi}^s, d_i)$.

This technique is based on the intuition that similar images are handled best by similar techniques, and that we should thus trust the classifier which has proven to behave best on the images most similar to the one we need to classify. We compute the score from Step 2 as

$$g(\hat{\Phi}^s, d_i) = \sum_{d_p \in \chi^w(d_i)} (1 - \delta(d_i, d_p))[\hat{\Phi}^s(d_p) = \Phi(d_p)], \quad (2)$$

where $[\alpha]$ is a function defined as

$$[\alpha] = \begin{cases} +1, & \text{if } \alpha = True \\ -1, & \text{if } \alpha = False. \end{cases}$$

Equation (2) thus encodes the intuition that the more examples in $\chi^w(d_i)$ are correctly classified by $\hat{\Phi}^s$ (i.e., are such that $\hat{\Phi}^s(d_p) = \Phi(d_p)$), and the closer they are to d_i (i.e., the lower $\delta(d_i, d_p)$ is), the better $\hat{\Phi}^s$ may be expected to behave in classifying d_i.

3.1.2. Weighted majority voting. The second technique we test is *weighted majority voting* (WMV), a technique similar in spirit to the "adaptive classifier combination" technique of [10]. WMV is different from DCS in that, while DCS eventually trusts a single descriptor-specific classifier (namely, the one that has proven to behave best on images similar to the test image), thus completely disregarding the decisions of all the other classifiers, WMV uses a weighted majority vote of the decisions of *all* the descriptor-specific classifiers $\hat{\Phi}^s \in \hat{\Phi}$, with weights proportional to how well each $\hat{\Phi}^s$ has proven to behave on images similar to the test image. This technique is thus identical to DCS except that Step 3 is replaced by the following two steps:

3. for each class $c_j \in C$, all evidence in favour of the fact that c_j is the correct class of d_i is gathered by summing the $g(\hat{\Phi}^s, d_i)$ scores of the classifiers that believe this fact to be true; i.e.,

$$z(d_i, c_j) = \sum_{f_s \in F: \hat{\Phi}^s(d_i) = c_j} g(\hat{\Phi}^s, d_i) \quad , \quad (3)$$

4. the class that obtains the maximum $z(d_i, c_j)$ score is chosen, i.e.,

$$\hat{\Phi}(d_i) = \arg\max_{c_j \in C} z(d_i, c_j). \quad (4)$$

This method thus encodes the intuition that the more classifiers vote for attributing to d_i a given class, and the better each such classifier has performed on the training documents close to d_i (as encoded in the $g(\hat{\Phi}^s, d_i)$ score), the higher the evidence that that class is the correct one.

[3] More precisely, a classifier committee is a tuple of n classifiers and an adaptive combination rule (see below). The equal sign above is thus a slight abuse of notation.

3.1.3. Confidence-rated, dynamic classifier selection. The third technique we test is *confidence-rated dynamic classifier selection* (CRDCS), a variant of DCS in which the *confidence* with which a given classifier has classified an image is also taken into account. From now on we will indeed assume that, given a test image d_i, a given descriptor-specific classifier $\hat{\Phi}^s$ returns both a class $c_j \in C$ to which it believes d_i to belong *and* a nonnegative real number $\nu(\hat{\Phi}^s, d_i)$ that represents the confidence that $\hat{\Phi}^s$ has in its own decision (high values of ν correspond to high confidence). In Section 3.2 we will see this to be the case for the descriptor-specific classifiers we generate in our experiments. Note also that, with respect to the "standard" version of DCS described in Section 3.1.1, this "confidence-aware" variant is more in line with the developments in computational learning theory of the last 15 years, since confidence is closely related to the notion of "margin," which plays a key role in learning frameworks based on structural risk minimization such as kernel machines and boosting [6].

The intuition behind the use of these confidence values is that a classifier that has made a correct decision with high confidence should be preferred to one which has made the same correct decision but with a lower degree of confidence; and a classifier that has taken a wrong decision with high confidence should be trusted even less than a classifier that has taken the same wrong decision but with a lower confidence.

CRDCS is thus the same as DCS in Section 3.1.1, except for the computation of the $g(\hat{\Phi}^s, d_i)$ score in Step 2, which now becomes confidence-sensitive. In CRDCS Eq. (2) is thus replaced by

$$g(\hat{\Phi}^s, d_i) = \sum_{d_p \in \chi^w(d_i)} (1 - \delta(d_i, d_p)) \times [\hat{\Phi}^s(d_p) = \Phi(d_p)]\nu(\hat{\Phi}^s, d_p). \quad (5)$$

The intuition here is thus that a classifier $\hat{\Phi}^s$ may be expected to perform accurately on an example d_i when many examples in $\chi^w(d_i)$ are correctly classified by $\hat{\Phi}^s$, when these are close to d_i, *and* when these correct classification decisions have been taken with high confidence.

Steps 1 and 3 from Section 3.1.1 remain unchanged.

3.1.4. Confidence-rated weighted majority voting. The fourth technique we test, *confidence-rated weighted majority voting* (CRWMV), stands to WMV as CRDCS stands to DCS; that is, it consists of a version of WMV in which confidence considerations, as from the previous section, are taken into account. CRWMV is thus similar in form to WMV. The only difference is that in CRWMV the $g(\hat{\Phi}^s, d_i)$ score as from Step 2 is obtained through Eq. (5), which takes into account the confidence with which the $\hat{\Phi}^s$ classifiers have classified the training examples in $\chi^w(d_i)$, instead of Eq. (2), which does not. Steps 1, 3, and 4 as from Section 3.1.2 remain unchanged.

3.2. Generating the Individual Classifiers

Each individual classifier $\hat{\Phi}^s$ (i.e., each member of the various ensembles described in Section 3.1) is generated by means of the well-known (*single-label, distance-weighted*) *k nearest neighbours* (*k*-NN) technique. This technique consists in the following steps; for a test image d_i

1. (similarly to Eq. (1)) identify the set

$$\chi^k(d_i) = \arg\min^k_{d_p \in Tr} \delta_s(d_i, d_p) \quad (6)$$

of the k training examples closest to the test image d_i, where k is an integer parameter and $\delta_s(d', d'')$ is a distance measure (ranging on [0, 1]) between images and in which only aspects specific to descriptor f_s are taken into consideration;

2. for each class $c_j \in C$, gather the evidence $q(d_i, c_j)$ in favour of c_j by summing the complements of the distances between d_i and the images in $\chi^k(d_i)$ that belong to c_j; i.e.,

$$q(d_i, c_j) = \sum_{d_p \in \chi^k(d_i):\, \Phi(d_p) = c_j} (1 - \delta_s(d_i, d_p)); \quad (7)$$

3. pick the class that maximizes this evidence, i.e.,

$$\hat{\Phi}^s(d_i) = \arg\max_{c_j \in C} q(d_i, c_j). \quad (8)$$

Standard forms of distance-weighted k-NN do not usually output a value of confidence in their decision. We naturally make up for this by adding a further step to the process, i.e.,

4. set the value of confidence in this decision to

$$\nu(\hat{\Phi}^s, d_i) = q(d_i, \hat{\Phi}^s(d_i)) - \frac{\sum_{c_j \neq \hat{\Phi}^s(d_i)} q(d_i, c_j)}{m - 1}.$$

That is, the confidence in the decision taken is defined as the strength of evidence in favour of the chosen class minus the average strength of evidence in favour of all the remaining classes.

Distance-weighted k-NN classifiers have several advantages over classifiers generated by means of other learning methods:

- Very good effectiveness, as shown in several text classification experiments [9, 21–23]; this effectiveness is often due to their natural ability to deal with non-linearly separable classes;
- The fact that they scale extremely well (better than SVMs) to very high numbers of classes [23]. In fact, computing the $|Tr|$ distance scores and sorting them in ascending order (as from Step 1) needs to be performed only once, irrespectively of the number m

Table 1. The Stone dataset, represented as a sequence of (class name, number of training examples, number of test examples) triplets. There are a total of 780 training examples and 1817 test examples distributed across 37 classes

Materials	Training	Test
ANDROMEDA	10	46
ANTIQUE_BROWN	76	138
ARANDIS_YELLOW	39	66
ARCTIC_CREAM	9	25
ASTERIX	9	19
BLACK_COSMIC	6	18
BLU_EYES	6	12
BLU_PEARL	30	76
COL_GOLD	7	13
COLONIAL_DREAM	13	23
COPPER_CANYON	19	46
COSTA_SMERALDA	65	144
DESERT_BROWN	2	10
DIORITE	19	36
FANTASTICO	8	20
GALAXY_BLACK	12	19
GIALLO_ARABESCATO	8	16
GIALLO_IRIS	5	37
GIALLO_ORNAMENTALE	128	311
GIALLO_VENZIANO	30	56
GOLDEN_BEACH	15	44
GOLDEN_FLAKES	5	15
JUP_APRICOT	11	21
JUP_PERSA	55	132
LABRADORITE	7	20
LEMURIAN	8	22
LOTUS	15	31
MAGMA	17	53
MASCARELLO	5	7
MOON_YELLOW	40	85
NERO_AFRICA	20	51
NETTUNO	7	13
ROSA_PORRINO	11	32
STAR_BEACH	14	39
TARN	6	33
TROPIC_BROWN	2	4
VOLGA_BLU	41	84

of classes involved; since this is by far the most computation-intensive step of the method, this means that distance-weighted k-NN scales (wildly) sublinearly with the number of classes involved. On the contrary, learning methods that generate linear classifiers scale linearly, since none of the computation needed for generating a single classifier $\hat{\Phi}'$ (aside from the generation of the vectorial representations) can be reused for the generation of another classifier $\hat{\Phi}''$, even if the same training set Tr is involved.

• The fact that they are parametric in the distance function they use. This allows the use of distance measures customized to the specific type of data involved, which turns out to be extremely useful in our case.

4. EXPERIMENTS

4.1. Experimental Setting

The dataset that we have used for our experiments (here called the Stone dataset)—see Table 1 for details) is a set of 2,597 photographs of stone slabs, subdivided into 37 classes representing different types of stone.[4] The dataset was randomly split into a training set, containing approximately 30% of the entire dataset, and a test set, consisting of the remaining 70%.

[4] The dataset was provided by the Metro S.p.A. Marmi e Graniti company (see http://www.metromarmi.it/), and was generated during their routine production process, according to which slabs are first cut from stone blocks, and then photographed in order to be listed in online catalogues that group together stone slabs produced by different companies.

Table 2. Error rates of the techniques discussed in this paper as tested on the Stone dataset; percentages indicate decrease in error rate with respect to the baseline. The first five results are relative to the five descriptor-specific baselines. Boldface indicates the best performer

CL	CS	EH	HT	SC
0.479	0.318	0.479	0.410	0.419
Baseline	DCS	CRDCS	WMV	CRWMV
0.297	0.183 (−38.4%)	**0.179 (−39.7%)**	0.225 (−24.2%)	0.227 (−23.6%)

As image descriptors we have used five visual "descriptors" as defined in the MPEG-7 standard,[5] each of them characterizing a particular visual aspect of the image. These five descriptors are *ColourLayout* (CL—information about the spatial layout of colour images), *ColourStructure* (CS—information about colour content and its spatial arrangement), *EdgeHistogram* (EH—information about the spatial distribution of five types of edges), *HomogeneousTexture* (HT—texture-related properties of the image), and *ScalableColour* (SC—a colour histogram in the HSV colour space).[6] Concerning the descriptor-specific distance functions we have used, see Section 5.

There are others visual descriptors defined in MPEG-7 that we have chosen not to use. First, we have not used any motion-related descriptors (e.g., *CameraMotion*) because they are only pertinent to video sequences, and not to still images; similarly, we have discarded *Shape3D* from consideration because our dataset consists of 2-D photographs only, and *FaceRecognition* was not used for obvious reasons. Second, we have chosen not to use 2-D shape descriptors (i.e., *ContourShape* and *RegionShape*) meant to describe the shape of regions (sets of pixels) in which an image has been partitioned by a segmentation algorithm; in fact, we deemed that this aspect was not relevant to telling different types of stone apart. We have not used *TextureBrowsing* since no interesting distance function can be defined on it.

As a measure of (in)effectiveness we have used *error rate* (noted E), i.e., the percentage of test images that have been misplaced in a wrong class.

As a baseline, we have use a "multi-descriptor" version of the distance-weighted k-NN technique of Section 3.2, i.e., one in which the distance function δ mentioned at the end of Section 5, and resulting from a linear combination of the five descriptor-specific δ_s functions, is used in place of δ_s in Eq. (6). For completeness we also report five other baselines, obtained in a way similar to the one above but using in each a descriptor-specific distance function δ_s. In these baselines and in the experiments involving our adaptive classifiers the k parameter has been fixed to 30, since this value has proven the best choice in previous experiments involving the same technique [21, 22]. The w parameter of the four adaptive ensembles has been set to 5, which is the value that had performed best on previous experiments we had run on a different dataset.

4.2. Results

The results of our experiments are reported in Table 2.

From this table we may notice that all four ensembles (2nd row, 2nd to 5th cells) bring about a noteworthy reduction of error rate with respect to the baseline (2nd row, 1st cell); this confirms that splitting the image representation into independent descriptor-specific representations on which descriptor-specific classifiers operate is a good idea, since both the baseline and the four ensemble methods use the same information, and only combine it in different ways. The best performer proves CRDCS, with a reduction in error rate of 39.7% with respect to the baseline, a very noteworthy improvement.

The results also show that confidence-rated methods (CRDCS and CRWMV) are not uniformly superior to methods (DCS and WMV) which do not make use of confidence values. They also show that dynamic classifier selection methods (DCS and CRDCS) are definitely superior to weighted majority voting methods (WMV and CRWMV). This latter result might be explained by the fact that, out of five descriptors, three (CS, CL, SC) are based on colour, and are thus not completely independent from each other; if, for a given test image, colour considerations are not relevant for picking the correct class, it may be difficult to ignore them anyway, since they are brought to bear three times in the linear combination. In this case, DCS and CRDCS are more capable of ignoring colour considerations, since they will likely entrust either the EH- or the HT-based classifier with taking the final classification decision.

The same result also seems to suggest that, for any image, there tends to be a single descriptor that alone is able to determine the correct class of the image, but this descriptor is not always the same, and sharply differs across categories. For instance, the SC-based classifier is the best performer, among the five baseline single-descriptor classifiers, on test images belonging to class GIALLO VENEZIANO ($E = 0.11$), where it largely outperforms the EH-based classifier ($E = 0.55$), but the contrary happens for class ANTIQUE BROWN, where EH ($E = 0.01$) largely outperforms

[5] International Organization for Standardization, *Information technology—Multimedia content description interfaces*, Standard ISO/IEC 15938, 2002.

[6] For definitions of these MPEG-7 visual descriptors see: International Organization for Standardization, *Information technology—Multimedia content description interfaces—Part 3: Visual,* Standard ISO/IEC 15938, 2002.

SC ($E = 0.22$). That no single descriptor alone is a solution for all situations is also witnessed by the fact that all single-descriptor classifiers (1st row of Table 2) are, across the entire dataset, largely outperformed by both the baseline classifier and all the adaptive ensembles.

5. EFFICIENT IMPLEMENTATION OF NEAREST-NEIGHBOUR SEARCH BY METRIC DATA STRUCTURES

In order to speed up the computations of our classifiers we have focused on implementing efficiently *nearest-neighbour search* which can be defined as the operation of finding, within a set of objects, the k objects closest to a given target object, given a suitable notion of distance. The reason we have focused on speeding up this operation is that

(1) it accounts for most of the computation involved in classifying objects through the k-NN method of Section 3.2; Step 1 of this method requires nearest-neighbour search;

(2) it also accounts for most of the computation involved in combining base classifiers through each of the four methods of Section 3.1; Step 1 of each of these four methods also requires nearest-neighbour search.

Efficient implementation of nearest-neighbour search requires data structures in secondary storage that are explicitly devised for this task [2, 15, 24]. As such a data structure we have used an *M-tree* [3],[7] a data structure explicitly devised for speeding up nearest-neighbour search in *metric spaces*, i.e., sets in which a distance function is defined between their members that is a metric.[8] We have been able to use M-trees exactly because

• as the five descriptor-specific distance functions δ_s of Eq. (6), we have chosen the distance measures recommended by the MPEG group (see [12] for details), which are indeed metrics;

• as the global distance function δ of Eq. (1) we have chosen a linear combination of the previously mentioned five δ_s functions, which is by definition also a metric. As the linear combination weights w_s we have simply adopted the weights derived from the study presented in [1], i.e., $w(CL) = 0.007$, $w(CS) = 0.261$, $w(EH) = 0.348$, $w(HT) = 0.043$, $w(SC) = 0.174$. Note that, in reality, the δ_s functions from [12] that we have adopted do *not* range on [0, 1], but on five different intervals $[0, \alpha_s]$; in order to have them all range on [0, 1] we have multiplied all distances by the normalization weights $z(CL) = 0.174$, $z(CS) = 0.075$, $z(EH) = 0.059$, $z(HT) = 0.020$, $z(SC) = 0.001$.

[7] We have used the publicly available Java implementation of M-trees developed at Masaryk University, Brno; see http://lsd.fi.muni.cz/trac/mtree/.

[8] A *metric* is a function δ on a set of objects X such that, for any $x_1, x_2, x_3 \in X$, it is true that (a) $\delta(x_1, x_2) \geq 0$ (*non-negativity*); (b) $\delta(x_1, x_2) = 0$ if and only if $x_1 = x_2$ (*identity of indiscernibles*); (c) $\delta(x_1, x_2) = \delta(x_2, x_1)$ (*symmetry*); (d) $\delta(x_1, x_3) \leq \delta(x_1, x_2) + \delta(x_2, x_3)$ (*triangle inequality*).

6. CONCLUSIONS

We have proposed an approach to image classification in which an adaptive ensemble of base classifiers, each based on looking at an image under a specific perspective, is entrusted with the classification decision, and we have shown that this approach largely outperforms a classification system that, albeit based on the same learning technology (distance-weighted k nearest neighbours), is more traditionally based on coalescing all these different perspectives into a single representation. We have also shown that there are noteworthy differences among diverse ways to combine the same base classifiers: our experimental results have shown that combination rules based on dynamic classifier selection clearly outperform rules based on weighted majority voting.

We stress that our approach is in no way based on the particular choice of descriptors exemplified in the experiments of Section 4.2. Other MPEG-7 descriptors (or other types of descriptors not from the MPEG-7 family) could be used, depending on the particular type of images being addressed; the only essential property is that a notion of distance can be defined on the descriptor to be used.

In the future we would like to carry out further research on ways to make weighted majority voting perform better. As hinted in Section 4.2, we think that the current inferior performance of WMV methods with respect to DCS methods may derive from the lack of independence among the descriptors used. Lack of independence among descriptors, however, is a phenomenon that cannot be avoided; we would thus like to investigate methods for computing the level of stochastic dependence between two descriptors, and for bringing to bear the computed dependence levels within a WMV ensemble.

ACKNOWLEDGMENTS

This work has been partially supported by Project "Networked Peers for Business" (NeP4B), funded by the Italian Ministry of University and Research (MIUR) under the "Fondo per gli Investimenti della Ricerca di Base" (FIRB) funding scheme. We thank Gianluca Fabrizi and Metro S.p.A. Marmi e Graniti for making the Stone dataset available. Thanks also to Claudio Gennaro and Fausto Rabitti for useful discussions.

REFERENCES

1. G. Amato, F. Falchi, C. Gennaro, F. Rabitti, P. Savino, and P. Stanchev, "Improving Image Similarity Search Effectiveness in a Multimedia Content Management System," in *Proc. 10th Intern. Workshop on Multimedia Information System (MIS'04)* (College Park, US, 2004), pp. 139–146.
2. E. Chávez, G. Navarro, R. Baeza–Yates, and J. L. Marroquín, "Searching in Metric Spaces," ACM Comp. Surveys **33** (3), 273–321 (2001).
3. P. Ciaccia, M. Patella, and P. Zezula, "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces," in *Proc. 23rd Intern. Conf. on Very Large Data Bases (VLDB'97)* (Athens, 1997), pp. 426–435.

4. L. Didaci and G. Giacinto, "Dynamic Classifier Selection by Adaptive *k*-Nearest-Neighbourhood Rule," in *Proc. 5th Intern. Workshop on Multiple Classifier Systems (MCS'04)* (Cagliari, 2004), pp. 174–183.
5. T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Proc. Ist Intern. Workshop on Multiple Classifier Systems (MCS'00)* (Cagliari, 2000), pp. 1–15.
6. R. E. Schapire and Y. Singer, "Improved Boosting Using Confidence-rated Predictions," Machine Learning **37** (3), 297–336 (1999).
7. G. Giacinto and F. Roli, "Adaptive Selection of Image Classifiers," in *Proc. 9th Intern. Conf. on Image Analysis and Processing (ICIAP'97)* (Firenze, 1997), pp. 38–45.
8. G. Giacinto and F. Roli, "Design of Effective Neural Network Ensembles for Image Classification Purposes," Image and Vision Comp. **19**, 699–707 (2001).
9. T. Joachimes, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *Proc. 10th Europ. Conf. on Machine Learning (ECML'98)* (Chemnitz, 1998), pp. 137–142.
10. Y. H. Li and A. K. Jain, "Classification of Text Documents," The Comp. J. **41** (8), 537–546 (1998).
11. D. Lu and Q. Weng, "A Survey of Image Classification Methods and Techniques for Improving Classification Performance," Int. J. Remote Sensing **28** (5), 823–870 (2007).
12. *Introduction to MPEG-7: Multimedia Content Description Interface*, Ed. by B. S. Manjunath, P. Salembier, and T. Sikora (John Wiley & Sons, New York, 2002).
13. J. Martínez-Alajarín, J. D. Luis-Delgado, and L. M. Thomás-Balibrea, "Automatic System for Quality-Based Classification of Marble Textures," IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Review **35** (4), 488–497 (2005).
14. V. D. Mazurov, *The Committee Method in Optimization and Classification Problems* (Nauka, Moscow, 1990) [in Russian].
15. H. Samet, *Foundations of Multidimensional and Metric Data Structures* (Morgan Kaufmann, San Francisco, 2006).
16. P. C. Smits, "Multiple Classifier System for Supervised Remote Sensing Image Classification Based on Dynamic Classifier Selection," IEEE Transactions on Geoscience and Remote Sensing **40** (4), 801–813 (2002).
17. E. Spyrou, H. Le Borgne, T. Mailis, E. Cooke, Y. Avrithis, and N. O'Connor, "Fusing MPEG-7 Visual Descriptions for Image Classification," in *Proc. 15th Intern. Conf. on Artificial Neural Networks (ICANN'05)* (Warsaw, 2005), pp. 847–852.
18. R. Takiyama, "A General Method for Training the Committee Machine," Pattern Recognition **10** (4), 255–259 (1978).
19. K. Woods, W. P. Kegelmeyer, Jr., and K. Bowyer, "Combination of Multiple Classifiers Using Local Accuracy Estimates," IEEE Transactions on Pattern and Machine Intelligence **19** (4), 405–410 (1997).
20. L. Xu, A. Krzyzak, and C. Y. Suen, "Methods for Combing Multiple Classifiers and Their Applications to Handwriting Recognition," IEEE Transactions on System, Man, and Cybernetics **22**, 418–435 (1992).
21. Yiming Yang, "An Evaluation of Statistical Approaches to Text Categorization," Information Retrieval **1** (1/2), 69–90 (1999).
22. Yiming Yang and Xin Liu, "A Re-Examination of Text Categorization Methods," in *Proc. 22nd ACM Intern. Conf. on Research and Development in Informational Retrieval (SIGIR'99)* (Berkeley, 1999), pp. 42–49.
23. Yiming Yang, Jian Zhang, and B. Kisiel, "A Scalability Analysis of Classifiers in Text Categorization," in *Proc. 26th ACM Intern. Conf. on Research and Development in Information Retrieval (SIGIR'03)* (Toronto, 2003), pp. 96–103.
24. P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach* (Springer Verlag, Heidelberg, 2006).

Tiziano Fagni is a Ph.D. student at the University of Pisa and a research fellow at ISTI-CNR in the Text and Opinion Mining group. His main research interest is automatic text classification, on which he has published papers on international journals and conferences.

Fabrizio Falchi has a Ph.D. in Information Engineering from University of Pisa (Italy), and a Ph.D. in Informatics from Faculty of Informatics of Masaryk University of Brno (Czech Republic). In 2003 he received a Research Fellowship from the Networked Multimedia Information System (NMIS) Laboratory of the Information Science and Technologies Institute (ISTI) of the Italian CNR in Pisa, where, from 2006, he is a research collaborator. His interests include similarity search, distributed indexes, multimedia content management systems, content-based image retrieval, Peer-to-Peer systems. He published several papers in peer reviewed international conferences and journals, and co-chaired the track "ELSDS: Engineering Large-ScaleDistributed Systems" at ACM SAC 2008 and the "CHORUS First workshop on peer to peer architectures for multimedia retrieval (1P2P4mm)."

Fabrizio Sebastiani is a Senior Researcher at ISTI-CNR where he leads the Text and Opinion Mining group. He is the author of numerous journal and conference papers on topics at the intersection of information retrieval, machine learning, and computational linguistics. He is the co-Editor-in-Chief of Foundations and Trends in Information Retrieval (Now Publishers}, an associate editor of ACM Transactions on Information Systems, a member of the Editorial Boards of Information Retrieval (Springer) and Information Processing and Management (Elsevier), and a former member of the Editorial Boards of ACM Computing Reviews (ACM Press) and Journal of the American Society for Information Science and Technology (Wiley). He has been the Program Chairman of ECIR 2003 and Program Co-Chairman of ACM SIGIR 2008, and is the Program Co-Chairman of ECDL 2010. He has also been the Vice-Chairman of ACM SIGIR since 2003 to 2007.