Multimed Tools Appl
DOI 10.1007/s11042-013-1766-4

Image orientation detection using LBP-based features and logistic regression

Gianluigi Ciocca · Claudio Cusano · Raimondo Schettini

Abstract Many imaging applications require that images are correctly orientated with respect to their content. In this work we present an algorithm for the automatic detection of the image orientation that relies on the image content as described by Local Binary Patterns (LBP). The detection is efficiently performed by exploiting logistic regression. The proposed algorithm has been extensively evaluated on more than 100,000 images taken from the Scene UNderstanding (SUN) database. The results show that our algorithm outperformed similar approaches in the state of the art, and its accuracy is comparable with that of human observers in detecting the correct orientation of a wide range of image contents.

Keywords Image orientation detection · Low-level features · Local binary patterns · Logistic regression · Image classification

1 Introduction

Almost all imaging applications and photo-management systems require that images are correctly oriented before processing and visualization. For example, most of the applications for image detection and scene classification, heavily rely on the fact that the given images are up-side.

G. Ciocca · R. Schettini
Department of Informatics, Systems and Communication (DISCo), Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy

G. Ciocca
e-mail: ciocca@disco.unimib.it

R. Schettini
e-mail: schettini@disco.unimib.it

C. Cusano (✉)
Department of Electrical, Computer and Biomedical Engineering, Università degli Studi di Pavia, via Ferrata 1, 27100 Pavia, Italy
e-mail: claudio.cusano@unipv.it

Published online: 19 November 2013

The correct orientation of an image is defined as the orientation in which the scene originally occurred [21, 23]. When no correction is applied, the orientation of a photograph is determined by the rotation of the camera at the moment the picture was taken. Even though any angle is possible, rotations by multiple of 90° are the most common. They are also straightforward to correct once detected. Therefore, it is common to assume that the images have been taken in one of the four orientations 0°, 90°, 180°, 270° (that sometimes are called 'North', 'West', 'South' and 'East').

Information about the orientation of a photograph may be obtained from sensors incorporated into the camera and recorded in the EXIF [8] meta data tags. However this information is often missing on low-end digital cameras or could have been removed by photo editing software. In these cases the user's intervention is required.

Humans can identify the correct orientation of photographs by exploiting their image understanding capabilities. An extensive study on the psychophysical aspects on image orientation recognition was presented in [15]. Using a panel of 26 observers that evaluated 1,000 images, the authors gained a number of interesting insights. They observed that for typical images, accuracy is close to 98 % when using all available semantic cues from high-resolution images, and 84 % when using only low-level vision features and coarse semantics from thumbnails. Some semantic cues stood out as being very important for the correct orientation recognition (e.g. sky, and people). The same study also shows that an image resolution of 256×384 is enough for humans in order to achieve a high accuracy.

The manual correction of image orientation is a tedious, time-consuming and error-prone activity. This is particularly true when large collections of photographs have to be processed. For these cases (digital archives, websites, content-base retrieval systems, workflow management for professional photographers...) an automatic approach would be helpful. Devising a computational approach for automatic detection of image orientation mimicking the high-level human understanding capabilities is a challenging task. Several semantic cues' detectors would be required to cope with the great variability of image content. Therefore, this approach tend to be computationally expensive. Moreover, its accuracy would greatly depend on the capability of bridging the semantic gap between the high-level cues and low-level features [7].

In this work we show that it is possible to devise an image orientation detection algorithm based purely on low-level features whose performances are comparable with those of human observers. The features are derived from Local Binary Patterns (LBP) [17], that are efficiently processed by a linear classifier obtained by logistic regression.

We have used a sub-set of the SUN image database [24] to test our proposal. This set contains 108,754 images divided into 397 scene categories. The experiments assessed the performances of our orientation detection algorithm with respect to specific scene types also taking into account the influence of color, images' resolution, and size of the training set. Our algorithm outperforms similar approaches in the state of the art, and shows an accuracy comparable with that reported by Luo et al. [15] for human observers.

1.1 Related work

Some orientation detection methods in the state of the art rely on low-level features to represent those cues that can be analyzed by a classifier to predict the most probable orientation. For instance, Vailaya et al. [21] used color moments, color histograms, edge direction histograms, and MSAR texture features to described the images after their subdivision in 10×10 blocks. They used a learning vector quantizer to extract a small codebook that they used to estimate the class-conditional densities of the observed features needed for the

Bayesian methodology. They reported 97 % of classification accuracy, obtained on a subset of high quality images from the Corel photo collection.

Wang and Zhang [23] exploited both chrominance and luminance information. Color moments are computed over 48 peripheral sub-blocks of a 8×8 blocks image subdivision, while edge direction histogram is used to characterize the image structure and texture. This information is then processed by different SVM (Support vector Machine) classifiers. Static classifier combination and hierarchical trainable classifier combination approaches are investigated. They reported an accuracy of 78 % on another subset of the Corel images.

Lyu et al. [16] proposed a method based on a set of natural image statistics collected from a multi-scale multi-orientation image decomposition. A two-stage hierarchical classification with binary SVM classifiers is employed to determine image orientation. Experiments performed on 18,040 natural images of different source and contents showed that the proposed method achieved about 60 % accuracy.

Lumini and Nanni [13] used color moments, Harris corner, phase symmetry, and edge direction histograms to describe the images. They then used Borda count to combine different classifiers based on Support Vector Machines, Parzen windows, and statistical classifiers. They obtained a 62 % accuracy on 6,000 images scanned from 350 rolls of film.

Baluja [3] used hundreds of classifiers trained with AdaBoost to determine the upright orientation of an image. 3,930 features related to color and edge information, are extracted from image subregions. Weak binary classifiers are used each built to compare a pair of features. The best set of 1,000 weak classifiers are then selected using the AdaBoost algorithm and combined to obtain a strong classifier. He reported the results obtained on several data sets, the accuracy on the largest one (Corel Disk-6, 15,888 images) is 61.9 %. A combination of 180 different strong classifier is also investigated and the accuracy on the same data set increased to 65 %. If a rejection rule is introduced, the accuracy on the Corel Disk-6 data set increases to 80.3 %.

Tolstaya's [20] approach is based on the assumption that the area in the lower part of an image has more texture that the other regions. Features are computed on local regions of the image and comprise luminance, chrominance and texture information. A two stage classification approach based on AdaBoost is used to detect the image orientation. A rejection scheme is also introduced. At the lowest rejection rate, the accuracy obtained is 87 % on a data set of 861 outdoor images.

A method explicitly designed to require low computational resources is the method proposed by Appia et al. [2]. Their algorithm is based on simple gradient and intensity features extracted from peripheral image sub-blocks. The orientation is determined by a set of heuristic rules, and a rejection threshold is used to discard ambiguous results. A test on 200 consumer images showed an accuracy of 74 % without the rejection threshold and 86 % with the rejection threshold.

Human observers are clearly more accurate in detecting the correct image orientation when they are allowed to take into account high-level semantic cues [15]. For this reason, some works have been proposed to exploit the information obtained by recognizing distinguishable elements in the image such as faces, sky, grass, etc. For instance, Lei Wang [22] used both low-level and high-level features: orientation of faces, position of the sky, brighter regions, textured objects, and symmetry. The cues are combined in a Bayesian framework obtaining an accuracy of 94 % on a data set of 1,287 images.

Luo and Boutell [14] developed a probabilistic approach to image orientation detection via confidence-based integration of low-level and semantic cues within a Bayesian framework. Semantic information is provided by suitable detectors designed to detect faces, blue

and cloudy sky, grass, and ceilings/walls. They reported 90 % accuracy on a set of 3,652 unconstrained consumer photos.

Ciocca et al. [5] combined both low-level features and faces. The approach uses the detection of faces as a hint to deem the image to be upward. When the image does not contain any detectable faces, the orientation is determined by an image classifier based on three low-level features: edge direction histogram, the first two moments in the YCbCr color space and a vertical coherence vector. Classifications performed with AdaBoost algorithm on a set of weak binary classifier. Using a-priori orientation probabilities, on the largest data set composed of about 4,000 images downloaded from the Web, the overall accuracy obtained is 86 %.

Borawski et al. [4] use the region of the sky to distinguish the orientation of outdoor images. The rationale is that the sky visible within an image is different for landscape- and portrait-oriented images. The localization of the sky within an image is based on the color. Fourier analysis is carried out to determine the orientation of the texture in the sky region. The method has been evaluated on 100 digital images containing the sky: 14 images have been rejected and six have been misclassified.

As it can be seen, the methods proposed in the literature show a wide range of accuracy values. Certainly a reason for this is the heterogeneous data sets used in evaluating the methods. Some of these data sets are small or specific for certain image categories that bias the overall results. For some categories such as landscapes, the correct orientation can be easily detected. On the other hand, indoor scenes, close-ups, or images with cluttered background are more difficult to classify since they lack important visual cues. For instance, Zhang et al. [25] separately tested their orientation detector on indoor and outdoor images. The accuracy they obtained on indoor images is much lower than that on outdoor images (48 % vs. 85 %). For this reason they introduced an indoor and outdoor classifier to refine the orientation detection obtaining an accuracy of 81 %.

Table 1 summarizes the aforementioned orientation detection methods.

2 Proposed algorithm

The method we propose is based solely on the information provided by low-level features, that is, features that can be reliably extracted from the images without any *a-priori* knowledge about their content. By not using high-level features, not only we keep manageable the complexity of the algorithm, but we also avoid the inherent sensitivity to the imaging conditions due to the semantic gap between the features and the image semantics. In other words, we hypothesize that full image understanding is not required for a reliable detection of the image orientation, and that the information provided by low-level features, when processed by a suitable classifier, is enough to obtain a good accuracy for a great variety of image contents.

In the literature, most of the methods based on low-level features focused on color and edge/texture information. Intuitively, color distribution is a very useful clue. However, there are several image categories (e.g. indoor images) where it does not help very much. Therefore, we decided to concentrate on a texture descriptor. More in detail, we decided to use features based on the distribution of Local Binary Patterns (LBP). These feature vectors lie in a high-dimensional space, of the kind for which linear classifiers are a very common choice. In this work we built a linear classifiers by using a regularized logistic regression.

Table 1 Summary of the orientation detection methods in the state of the art

Reference		LL	HL	Features	Decision	DB Size	DB Source	Accuracy
Vailaya et al. (2002)	[21]	X	–	Color moments	Mixture of Gaussian with LDA	8,364	Corel	97 %
Zhang et al. (2002)	[25]	X	X	Color moments, edge direction histogram, indoor, outdoor	AdaBoost	10,838	Corel	81 %
Wang et al. (2003)	[22]	X	X	Orientation of faces, position of the sky, brighter regions, and textured objects, and symmetry	Bayesian framework	1,287	Personal	94 %
Wang et al. (2004)	[23]	X	–	Color moments, and edge direction histogram	SVM classifiers	5,422	Corel	78 %
Luo et al. (2005)	[14]	X	X	Color moments, edge direction histogram, face, blue sky, cloudy sky, ceiling/wall, and grass	Bayesian framework	3,652	Personal	90 %
Lyu (2005)	[16]	X	–	Multi-scale multi-orientation image decomposition based on separable quadrature mirror filters	SVM classifiers	18,040	Personal	60 %
Lumini et al. (2006)	[13]	X	–	Color moments, Harris corner, phase symmetry, edge direction histogram	SVM classifiers	6,000	Personal	62 %
Baluja (2007)	[3]	X	–	Mean and variance of simple and normalized R, G, B, Y, I, Q color channels, intensity, horizontal and vertical edges	AdaBoost	15,888	Corel Disk-6	65 %
Tolstaya (2007)	[20]	X	–	Mean and standard deviation of Y, Cb, and Cr color channels, and angle histogram	AdaBoost	861	Personal	87 %

Table 1 (continued)

Reference		LL	HL	Features	Decision	DB Size	DB Source	Accuracy
Ciocca et al. (2010)	[5]	X	X	Edge direction histogram, the first two moments in the YCbCr color space, vertical coherence vector, and faces	AdaBoost	12,009	Fickr	85 %
Appia et al. (2011)	[2]	X	–	Image gradient, and smoothness map	Rule based	200	Personal	92 %
Borawski et al. (2012)	[4]	–	X	Sky	Rule based	100	Personal	86 %
Proposed method		X	–	Local binary patterns and color moments	Logistic regression	108,754	SUN DB	92 %

For each method is indicated a reference, whether it uses low-level (LL) or high-level (HL) features, and the classification framework. Features are usually computed spatially on image's sub-regions following a predefined subdivision schema. Accuracy refers to the best result on the largest data set used and without rejection rule if available

Figure 1 depicts a schematic view of the proposed method that, for the sake of brevity, in the following we will refer to as LBP-LLR (from Local Binary Patterns and Linear Logistic Regression).

2.1 Image features

Local Binary Patterns have shown remarkable discriminative power in different domains due to their invariance with respect to lighting conditions, and robustness with respect to image noise. For example, LBPs have been used in face recognition [1], multi-object tracking [19], and scene classification [11]. For a comprehensive overview about LBP readers can refer to [18].

The LBP descriptor is defined as a histogram of the local patterns surrounding each pixel. These patterns are computed by thresholding the intensity of the neighbors of each pixel with the intensity of the pixel itself (see Fig. 2). More in detail, given a neighborhood size P and a radius R, for each pixel the numerical code $LBP_{P,R}$ is computed as follows:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{1}$$

where g_c is the gray level of the current pixel, $g_0, \ldots, g_{P-1}$ are the gray levels of its neighbors, and s is defined as $s(x) = 1$ if $x \geq 0$, $s(x) = 0$ otherwise. The P neighbors lie on a circular neighborhood, of radius R, of the current pixel: the gray value g_p is obtained by interpolating the intensity image at a displacement $(R\cos(2\pi p/P), R\sin(2\pi p/P))$.

With P neighbors there are 2^P possible patterns, but not all of them are equally significant. Usually, only patterns describing a somewhat regular neighborhood are considered. These patterns are called "uniform" and are defined as those patterns for which there are at most two transitions (bitwise 0/1 changes) between adjacent bits in the code. For instance, the pattern '00011100' is uniform, while the pattern '11001000' is not uniform because it includes three transitions. The number of uniform patterns is $2 + P(P-1)$. In fact, are uniform patterns those consisting of k zeros and $P-k$ ones, where all the zeros or all the ones are consecutive. There is one pattern for $k = 0$, and one for $k = P$. For each value of k in the range $\{1, \ldots, P-1\}$ there are P patterns, each one corresponding to a different rotation of the bits (see [18] for more details).

The circular shape of the neighborhood makes rotation invariance easy to achieve. However, we decided to not exploit this property of the LBP approach because rotation invariance would obviously discard important information about the orientation of the image.

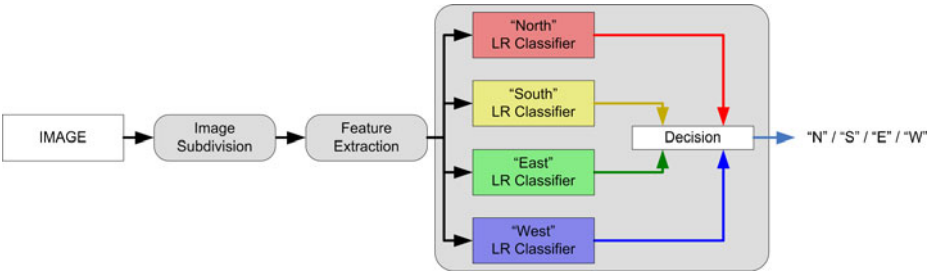

Fig. 1 The proposed LBP-LRR method for the detection of image orientation

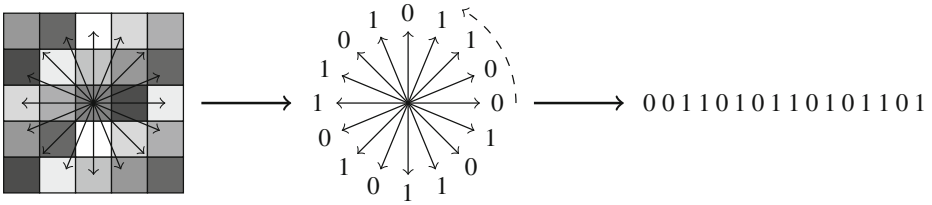

Fig. 2 The first steps of the Local Binary Pattern extraction. For each pixel, a circular neighborhood is considered. Each neighbor is thresholded by the intensity of the central pixel determining a binary response. The pattern is formed by concatenating the resulting bits

To form a fixed length feature vector the patterns are aggregated into one or more histograms. Histograms are formed by counting the occurrences of each uniform pattern in a given region of the image. Non-uniform patterns are not ignored, but they are all accounted for in a single bin. The final descriptor is the concatenation of the normalized histograms. With H possibly overlapping regions and with P neighbors, the final descriptor length is $H \times (3 + P(P-1))$. In fact each of the H histograms has $2 + P(P-1)$ bins for the uniform patterns and one bin for all the non-uniform ones.

2.2 Orientation recognition

Due to their capability in dealing with high-dimensional feature spaces, linear classifiers have become one of the most popular methods for image classification [6]. In fact, linear classifiers are very fast and very efficient learning methods exist for their training.

Typically, the learning procedure of a binary linear classifier consists in solving the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi(\mathbf{w}; \mathbf{x}_i, y_i), \tag{2}$$

where $\mathbf{x}_i$ denote the training samples ($i = \{1, \ldots, m\}$) and $y_i \in \{-1, +1\}$ are the corresponding class labels. The optimal $\mathbf{w}$ defines a hyperplane that linearly separates positive from negative instances. The loss function ξ penalizes the errors on the training set, weighted by the penalization coefficient C. In practice, the parameter C determines a trade-off between the penalization and the regularization term $\|\mathbf{w}\|^2$ (the norm $\|\cdot\|_1$ instead of the Euclidean can be used as well). Linear Support Vector Machines are an example of linear classifier within this framework.

We used a very fast implementation of a regularized binary linear regression classifier as implemented in the LIBLINEAR package [9]. The penalization function is defined as

$$\xi(\mathbf{w}; \mathbf{x}_i, y_i) = \log\left(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}\right), \tag{3}$$

which is derived from a probabilistic model.

The optimization problem (2) is solved by the LIBLINEAR library using a trust region Newton method [12]. The problem of orientation detection is not binary, since there are four possible orientations. For multi-class problems LIBLINEAR uses the one-against-all strategy: for each class a binary problem is built to discriminate between instances of that class from the instances of all the other classes. Therefore, the classifier consists of the

hyperplanes $\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_K$, one for each of the K classes. Given a new instance $\mathbf{x}$, the predicted label $y \in \{1, \ldots, K\}$ is obtained as:

$$y = \arg\max_j \mathbf{w}_j^T \mathbf{x}. \tag{4}$$

In the case of orientation detection we have $K = 4$ classes, corresponding to rotations of multiples of 90°.

2.3 Computational complexity

The LBP-LRR algorithm is very fast. When LBP histograms are computed on H disjoint regions, their computation is linear with respect to the number N of pixels in the image and to the cardinality P of the neighborhood. As stated before, the dimensionality of the resulting feature vector is $H \times (3 + P(P-1))$, and classification is linear with respect to the dimensionality of the feature space. Therefore, the procedure has a complexity in time of $O(N \times P + H \times P^2)$. Note that the classification of the patterns as uniform or non-uniform can be obtained very quickly by using a precomputed look-up table with 2^P entries.

2.4 Feature selection and tuning of the parameters

The computation of LBP features depends on several parameters: the neighborhood cardinality (P) and size (R), and whether or not they are uniform. Moreover, in order to introduce some locality into the final descriptor, usually histograms of LBPs are computed on different regions of the image, and such a subdivision need to be specified as well. These parameters, and those of the classifier (e.g. the penalization coefficient in (2)) have been tuned by estimating the classification accuracy with a five-fold cross-validation on the training set.

Different combinations of the parameters have been considered, and the best one consisted in using uniform LBPs with a neighborhood of cardinality $P = 16$, size $R = 2$. The best image subdivision resulted in the union of two partitions, one that uniformly divides the image in six horizontal bands, the other that divides it in six vertical bands. Therefore, in total 12 histograms are computed and concatenated to form the final descriptor. During parameters' selection, we observed a good degree of stability with respect to the penalization coefficient: the best result has been obtained for $C = 1$.

One of the possible weaknesses of Local Binary Patterns is that, in their original form, they do not encode any information about the color distribution. While it is clear that for most images gray-level information is enough to unambiguously determine their orientation, color is recognized as an important clue. In fact, most algorithms in the state of the art heavily rely on the information provided by color distribution [5, 13, 14, 20, 21, 23, 25].

To assess the importance of the color information we tried to complement the LBP histograms with various color features (color moments in different color spaces and various kind of color histograms). The best results have been obtained by using color moments (mean and standard deviation) in the YUV color space, with the same image subdivision used for LBP histograms. An alternative method to include color information is to compute the LBP histograms independently on the components of a color space. We implemented this strategy by considering LBPs on the three RGB components (in the following we will refer to this algorithm as LBP-RGB).

3 Experimental results

Most existing orientation detection algorithms have been evaluated on small and homogeneous data sets (e.g. only outdoors images, all images with visible sky, etc.). An algorithm designed to be applied in real applications should be proved to be effective on a large, heterogeneous collection of images. To this end we have chosen to use the SUN image database [24] for our experiments. The database was collected by selecting from the available terms of WordNet [10] those describing concrete scenes, places, and environments. After the removal of synonyms the final set of terms numbered 899 image categories. For each term, images were retrieved from the Web by using different search engines obtaining a total of 130,519 images. As suggested in [24], we considered only those categories containing at least 100 images. The final image data set is thus composed of 108,754 images belonging to 397 categories. Figure 3 shows some representative images taken from different categories in this data set.

We divided the data set into a training and a test set. Starting from the 108,754 images of the SUN database, we randomly selected 2,500 images (about 2.3 % of the whole data set, see Section 3.4 for further considerations about the size of the training set) to form the training set. The remaining 106,254 form the test set, and are used to evaluate the methods. All the 397 categories are represented in the test set.

The orientation of the images have been already corrected by the authors of the SUN database and these images may be in the "landscape" layout (i.e. images which are wider than taller) or in the "portrait" layout. We altered the database to simulate the situation in which the images are taken with a digital camera that does not feature the automatic orientation capability. Images with a "landscape" layout retain their original orientation (i.e. North direction). Portrait images are randomly rotated clockwise or counter-clockwise by 90°, and labeled with the East and West orientations, respectively. No image has been

Fig. 3 Some image categories from the SUN database

labeled with the south orientation, because this would correspond to a picture taken with the camera turned upside down (an unrealistic case). Following this procedure all the images end up having a landscape layout. Of the 2,500 images in the training set 1,841 have been labeled with the North orientation (73.6 %) while the East and the West labels have been assigned to 340 (13.6 %) and 319 (12.8 %) images, respectively. Concerning the 106,254 images in the test set, 77,265 have been labeled as North (72.7 %), 14,621 as East (13.8 %), and 14,368 as West (13.5 %). These figures agree with the distribution reported by other authors. For instance, for consumer photos scanned from film, Luo and Boutell [14] reported 72 % North, 14 % East, 12 % West, and 2 % South (although uncommon it is possible in the case of scanned films).

Some other works in the state of the art preferred the generation of a balanced data set, where each orientation is equally represented. To do so, each image is randomly rotated. We preferred to keep the data set unbalanced because: (i) it better represents the conditions found in real applications; (ii) it keeps the correlation between the content and the layout of the images (after all the "portrait" and "landscape" layout are called this way because they are typically used for that kind of scenes).

3.1 Results

In the first experiment we compared variants of the proposed LBP-LRR method based on different features: LBP histograms, YUV moments, their combination, and LBP histograms on the RGB components. The results are reported in Table 2.

With the combination of LBP histograms and color moments the orientation of 98,200 images, out of the 106,254 images that form the test, has been correctly identified (92.4 %). Slightly worse results have been obtained without color information (91.0 %). This demonstrates that color moments, while not very useful when used alone (83.4 % of classification accuracy), can complement the information encoded by the LBP histograms resulting in a measurable improvement (even if to a limited extent). The use of LBP histograms on the RGB components, instead, did not cause any significant improvement (only 0.1 % better than original LBP histograms).

The SUN database has the advantage of being carefully organized in several semantic categories, making it possible to analyze in detail the behavior of the algorithms when dealing with different image contents. Figures 4 and 5 report the results obtained on the 397 categories by using LBP histograms combined with the color moments (in the following we will implicitly refer to this feature combination when not stated otherwise).

For 17 categories (from 'athletic field' to 'volleyball court' in the figure) the orientation of all the images has been correctly detected. This is quite remarkable because these categories are quite heterogeneous, featuring a high degree of intra-class variability. For other 17 categories (including, for instance, 'cafeteria', 'dam', 'butte', 'planetarium') only one test image has been misclassified. At the other end of the spectrum we have the categories for

Table 2 Classification accuracy obtained on the test set by the variants of the LBP-LRR method

Features	Accuracy (%)
LBP histograms	91.0
Color moments	83.4
LBP hist. + Color mom.	92.4
LBP-RGB histograms	91.1

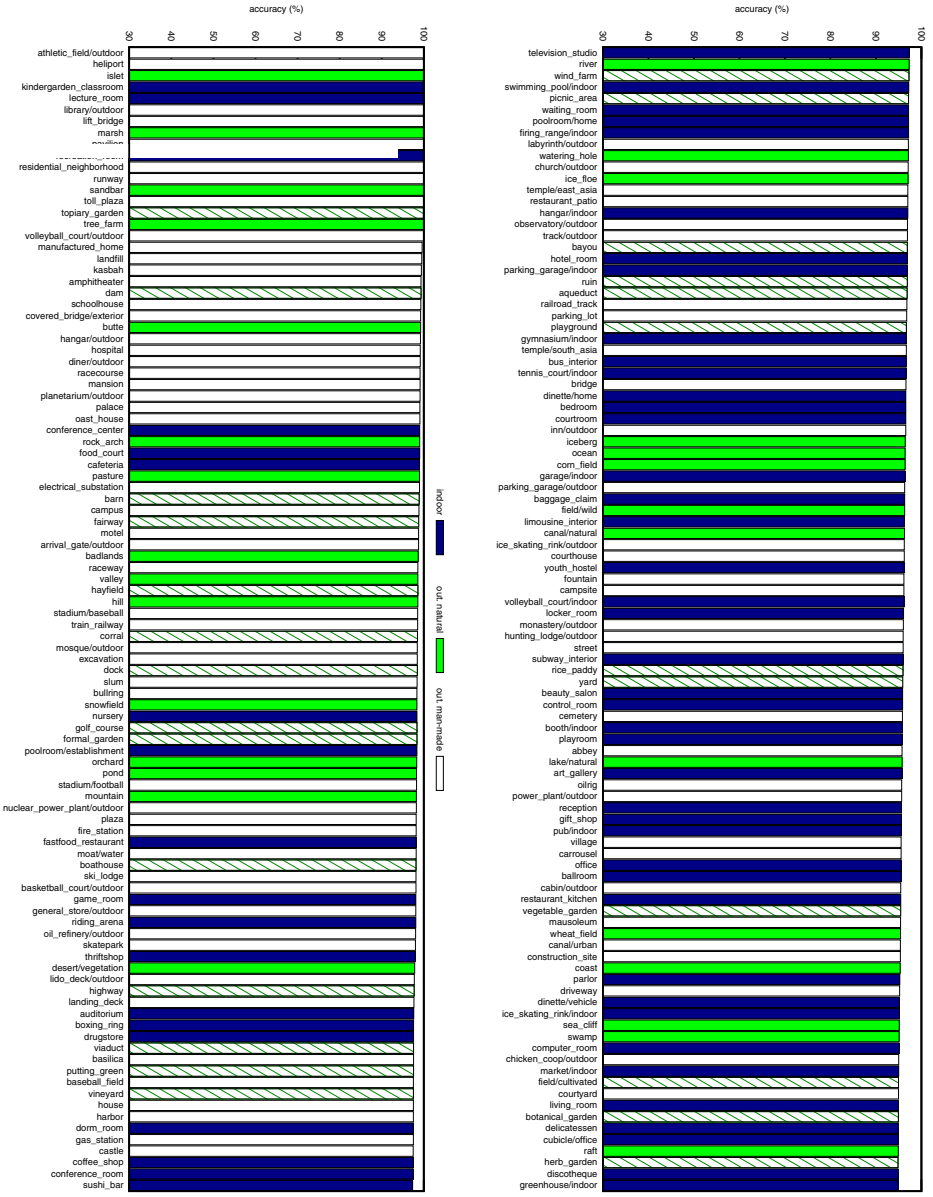

Fig. 4 Detail of the classification accuracy obtained on the 397 categories of the SUN database by the LBP-LRR algorithm (top 200). The categories are listed by decreasing accuracy, and are depicted according to their macro category (indoor, outdoor man-made, outdoor natural). Some categories belong to more than one macro category and are indicated by a hatched bar

which the accuracy of the classifier is very low: 'doorway', 'pulpit', 'apse/indoor' obtained a classification accuracy of less than 60.0 %. These categories contain many images that are cluttered, or underexposed (see the first row in Fig. 3).

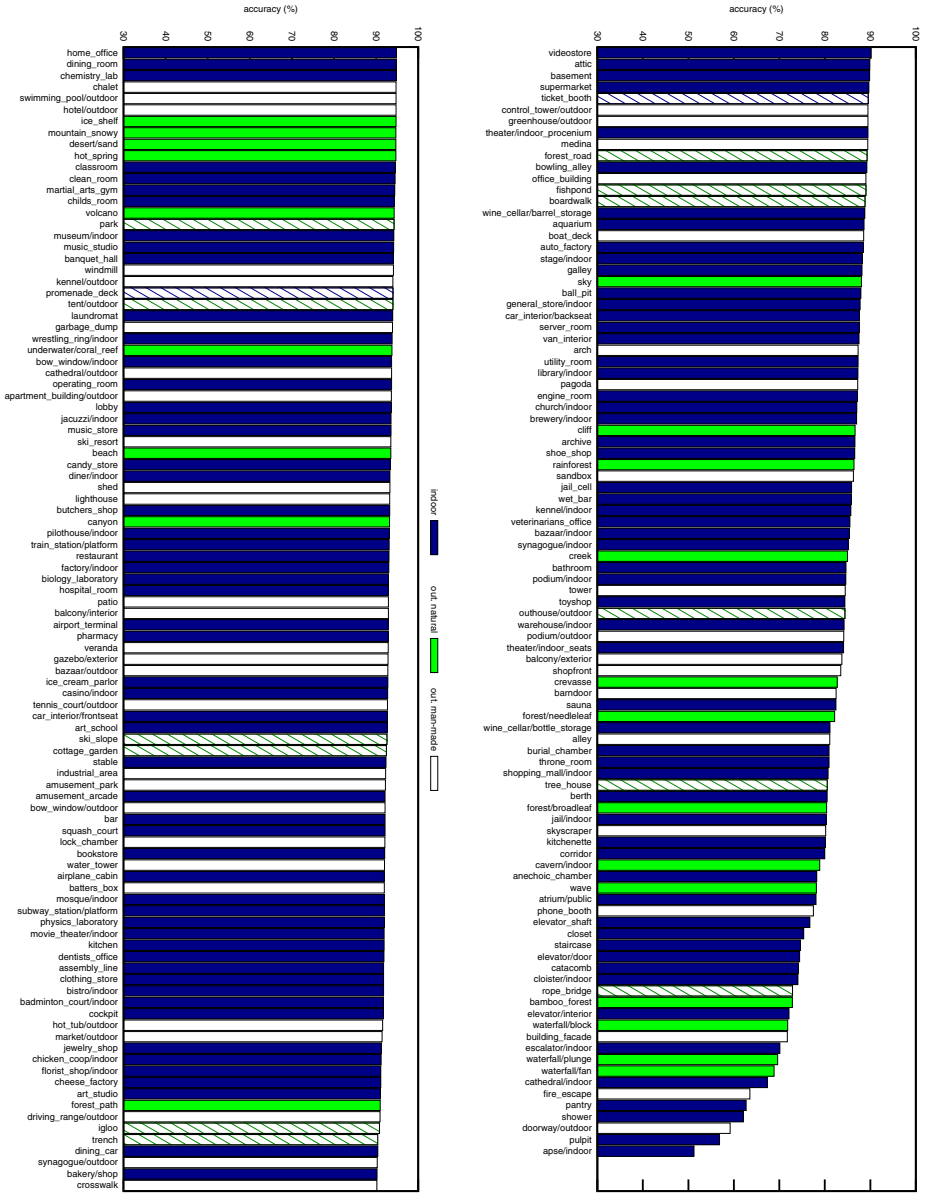

Fig. 5 Detail of the classification accuracy obtained on the 397 categories of the SUN database by the LBP-LRR algorithm (bottom 197). The categories are listed by decreasing accuracy, and are depicted according to their macro category (indoor, outdoor man-made, outdoor natural). Some categories belong to more than one macro category and are indicated by a hatched bar

Of the best 30 categories 27 are outdoor, and 18 of the worst 30 are indoor. This fact seems to confirm previous results in the literature [25] where has been shown that the orientation of indoor images is harder to detect than the orientation of outdoor images. However,

if we look at our results on all the 397 categories we see that the differences between indoor and outdoor are not very evident.

The SUN database is also hierarchically organized: at the first level we have three macro categories, namely indoor, outdoor man-made, and outdoor natural. These are then further divided into several sub-categories. Table 3 reports the results with respect to this categorization. Differently from other studies in the state of the art [25], the performance are quite stable across the indoor/outdoor macro categories. The highest accuracy has been obtained on the 'outdoor man-made' category (93.5 %). On indoor images the accuracy was 90.9 %, which is slightly worse than the 92.7 % obtained on 'outdoor natural' images.

Even within each macro category the accuracy on the sub categories are quite regular. Only in two cases the accuracy falls below 90 %: 'cultural' (87.5 %) and 'shops, cities, towns' (88.3 %). Nevertheless, the difference between the hardest and the easiest sub categories is more than 9 %. This fact suggests that results obtained on small data sets, that hardly cover all the sub categories, are prone to be biased. For large data sets the simple subdivision into indoor/outdoor man-made/outdoor natural (or even worse, into indoor/outdoor) is too coarse to fully understand the results.

Figure 6 shows some examples of the errors made by the algorithm. Errors typically occur when the images contain a large amount of details that make very difficult to identify, using only low-level features, those patterns which are clear indicators of the correct orientation. In most cases the correct orientation is difficult to determine "at a glance" even for human observers; on the contrary, there are images where our high-level understanding makes it evident and unambiguous.

Table 3 Classification accuracy on the first two levels of categorization. Note that some images belong to multiple categories and that, to simplify the analysis, they have been ignored, here

Level 1	# Images	Acc. (%)	Level 2	# Images	Acc. (%)
Indoor	46,256	90.9	Shopping and dining	7,152	91.9
			Workplace	6,747	90.5
			Home or hotel	13,085	91.8
			Transportation (interiors)	5,127	92.5
			Sports and leisure	4,273	93.6
			Cultural	6,005	87.5
Outdoor, natural	14,090	92.7	Water, ice, snow	6517	91.8
			Mountains, hills, desert, sky	3,148	93.6
			Forest, field, jungle	3,165	92.5
			Man-made elements	216	94.9
Outdoor, man-made	35,911	93.5	Transportation	4,663	96.6
			Cultural or historical place	8,936	95.3
			Sports fields, parks	5,909	94.9
			Industrial and construction	2,374	95.9
			Houses, cabins, farms	4,282	95.4
			Shops, cities, towns	6,844	88.3

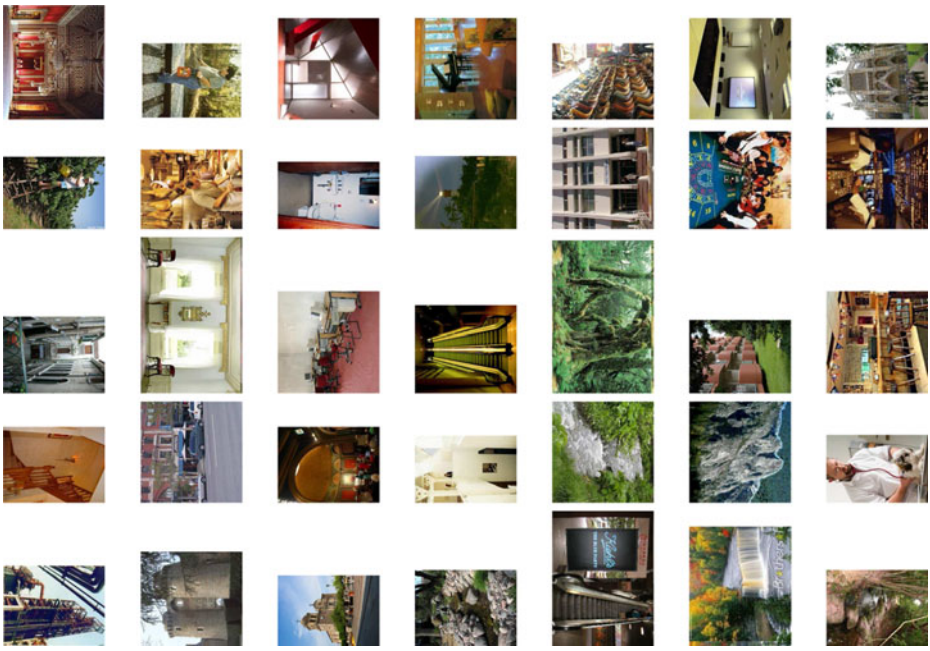

Fig. 6 A random sample of some of the errors made on the test set. Images are rotated according to the orientation detected

3.2 Comparison with other methods

We computed on the SUN database the performance of a selection of alternative methods from the state of the art. In particular, we focused on those methods for which the training procedure can be faithfully replicated without additional information or data. This criterion excludes the methods relying on the high-level features provided by specific object detectors requiring the use of additional data for training. The methods we considered are those proposed by Vailaya et al. [21], Tolstaya [20], Ciocca et al. [5], Appia and Narasimha [2].

The comparison has been obtained by using our own implementations of these methods. See Section 1.1 for a brief description. We used the same experimental protocol described before: training on 2,500 images (possibly with a five-fold cross validation for the model selection step) and test on the 106,254 images of the test set. The classification accuracies obtained are reported in Table 4.

The results are quite clear: the LBP-LRR method outperforms the other methods considered. Note that the performance reported by the original authors may be quite different. For instance, Appia and Narasimha [2] reported a higher performance (74 %) than that shown here. This can be explained by the fact that, in order to achieve a very high processing speed they based their method on reasonable, but simple assumptions (i.e. that high intensity regions stay on top, and that high frequency regions lie at the bottom of the image). These assumptions usually hold for prototypical images (landscapes, indoor images with little or no clutter), but not for most of the images in the SUN database. Another example is the method of Vailaya et al. [21], for which the authors reported an accuracy of 98 % on a set of 8,364 Corel images mostly depicting uncluttered scenes with a clear subject as taken by

Table 4 Classification accuracy obtained on the test sets by the LBP-LRR method, and by four algorithms from the state of the art

Method	Ref.	Acc. (%)
Vailaya et al.	[21]	80.1
Tolstaya	[20]	85.6
Ciocca et al.	[5]	89.3
Appia and Narasimha	[2]	54.3
LBP-LRR	(this work)	92.4

professional photographers. With this method Luo and Boutell obtained 78 % [14] on a collection of 3,652 personal photographs. This difference depends on the properties of the data set used for the evaluation. In our experiments we used a much larger test set (more than 100,000 images) of different quality and resolution obtaining for the Vailaya et al. method a classification accuracy of 80.1 %.

More in detail, Table 5 reports the confusion matrices obtained by the five methods considered. All but one of the methods biased their decisions towards the 'North' orientation. This behavior has been learned from the training set, without any explicit indication. Similarly, the 'South' orientation has been virtually ignored. The method by Appia and Narasimha is an exception, since it is based on rules without any training procedure. The low performance we obtained with their method are also explained by its inability to exploit uneven prior distributions. The design of the method by Ciocca et al. does not completely rule out the 'South' orientation even if no training image has that orientation. However, the 'South' orientation is predicted in less than 1 % of the cases.

Table 5 Confusion matrices of the methods compared in experiments. Results are expressed in percentage. The diagonal elements (corresponding to correct classifications) are reported in bold

		Predicted orientation			
Method	True or.	North	West	South	East
Vailaya et al. [21]	North	**93.9**	3.2	–	2.9
	West	49.4	**42.7**	–	7.9
	East	48.8	7.3	–	**43.9**
Tolstaya [20]	North	**96.8**	1.4	–	1.8
	West	53.6	**33.2**	–	13.2
	East	61.1	14.6	–	**24.3**
Ciocca et al. [5]	North	**96.3**	1.6	0.4	1.7
	West	40.1	**52.5**	0.2	7.2
	East	40.1	7.8	0.2	**51.9**
Appia and Narasimha [2]	North	**54.3**	14.0	18.1	13.6
	West	12.8	**54.5**	13.0	19.7
	East	12.7	20.4	12.8	**54.1**
LBP-LRR	North	**98.2**	1.1	–	0.7
	West	10.1	**78.1**	–	11.8
	East	10.7	13.1	–	**76.2**

The effectiveness of the proposed approach is mostly due, in our opinion, to the design of the image descriptor. Local binary Patterns, in fact, are robust against several categories of image transformations. For instance, they are left unchanged by monotonic transformations of the pixels (see (1)), such as those caused by changes in the lighting conditions. moreover, the aggregation of the patterns into histograms makes the descriptor robust against small translations and scalings.

On the other hand, the descriptor is sensitive to rotations of the image plane. More in detail, rotations by multiples of 90° result in permutations of the feature vectors: the uniformity of the patterns is not affected, but the directionality of the uniform patterns changes according to the angle of rotation (see Fig. 2); due to the way in which the image is subdivided, the order in which the histograms are concatenated also changes in a predictable way (e.g. the first horizontal band becomes the first vertical after a counter-clockwise rotation of 90°, the last horizontal after a rotation of 180°, and the last vertical after a rotation of 270°).

In our framework, the choice of the classifier is not as important as the design of the descriptor. To verify this, we repeated the experiment by using different classifiers: linear and non-linear (Gaussian RBF) Support Vector Machines (SVM), and a nearest neighbor classifier. Their parameters have been selected by five-fold cross validation in the same way described before for the logistic linear regression. Table 6 reports the result obtained: with SVMs the accuracy is just a bit lower than with logistic regression. Clearly worse results have been obtained, instead, with the nearest neighbor classifier (using a k-NN with $k > 1$ did not bring any improvement).

3.3 Resolution of the images

Image resolution clearly influences the accuracy of the detection of the correct orientation. A psychological study about this issue has been conducted by Luo et al. [15]. They asked 26 subjects to detect the orientation of 1,000 images at five different resolution levels (24×36, 64×96, 128×192, 256×384, 512×768). They conclude that the performance of human observers can be considered as an upper bound for computer vision algorithms. This bound would be 84 % when coarse semantics is used (64×96 pixels, in their experiment) and 96 % when all the semantics are considered (512×768 pixels). Of course these figures depend on the data set considered.

To verify how much the resolution of the images influences the performance of our algorithm we measured its performance at the same resolution levels used by Luo et al.. Before training and test, images have been resampled in such a way that their longest side is 36, 96, 192, 384, or 768, according to the resolution level under consideration. The other side is changed to preserve the aspect ratio. Three variants of the algorithms are considered: LBP histograms combined with color moments, LBPs only, and color moments only. The resulting classification accuracies are reported in Fig. 7. To allow a rough comparison with

Table 6 Classification accuracy obtained on the test set by different classifiers

Classifier	Accuracy (%)
Logistic linear regression	92.4
Linear SVM	91.7
Non-linear SVM	91.2
Nearest neighbor	85.7

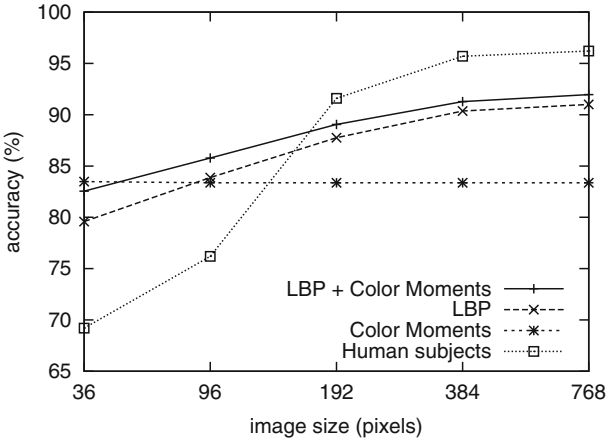

Fig. 7 Performance of the LBP-LRR method, varying the resolution of the images. Three variant are considered: LBP histograms combined with color moments, LBPs only, color moments only. The plot reports also the performance obtained by human subjects, as taken from [15]

the performance of human subjects the results obtained by Luo et al. are also shown, even though they have been obtained on a different data set.

The results clearly show that the LBP-LRR method takes advantage of the additional information provided by the higher level of resolution. As expected, color moments are virtually invariant with respect to the image resolution. For the lowest level of resolution, LBP features perform worse than color moments, but they quickly improve and are clearly better at medium and high resolutions. At the highest level, the performance of LBPs are still increasing, even though the behavior of the plot suggest that they are converging to a maximum. By using a combination of LBPs and color moments better results are obtained than by using a single feature.

3.4 Size of the training set

One of the advantage of using a large data set is that it allows to reliably assess how the performance depend on the size of the training set. To do so, we subdivided the data set into training and test sets of different sizes: the 2,500 images used before are not considered here (but we use the parameters found with the cross validation on those images). The remaining 106,254 images have been randomly partitioned into training and test set pairs. The cardinalities of the training sets are the powers of two from 32 to 65, 536. The test sets correspond to the complements of the training sets.

Figure 8 reports the results obtained with LBP histograms, color moments, and their combination. In all the three cases, the classification accuracy increases with the size of the training set. With color moments, no significant improvement is observed for more than 8,192 images. With LBPs the performance corresponding to the largest training set (65,536 images) are close to 94 %. We believe that an even larger training sets would allow the combination of the two features to match the performance obtained by human subjects.

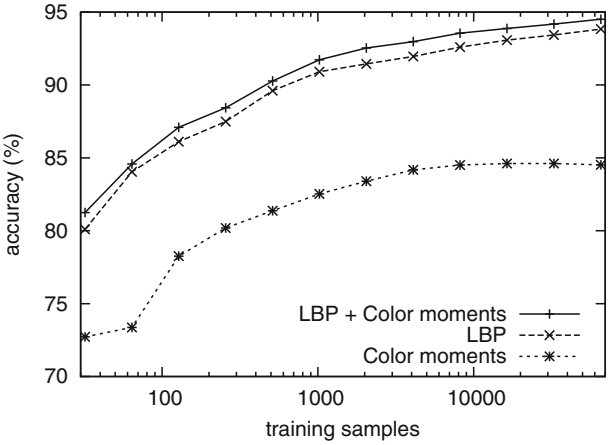

Fig. 8 Performance of the LBP-LRR method, as a function of the number of images in the training set (logarithmic scale). Three variant are considered: LBP histograms combined with color moments, LBPs only, color moments only

4 Conclusions

In this paper we have investigated the automatic detection of image orientation. We have shown that it is possible to devise an effective algorithm based purely on low-level features extracted from gray level images. More in detail, we have proposed the use of Local Binary Patterns for the description of the image content, and of a linear classifier obtained by regularized logistic regression. With this approach we obtained a remarkable classification accuracy (91.0 %). Only slightly better results (92.4 %) have been obtained by combining the LBP features with the color moments. In both the configurations the algorithm outperformed all the other detection algorithms considered, and it is close to the human performance as reported in the state of the art. Our findings are supported by the use of a large collection of images (more than 100,000) presenting a wide range of scene categories. The use of this data set allowed us to obtain reliable and insightful results: the accuracy of the algorithm is quite stable across the categories of the SUN database. About 75 % of the 397 categories have a detection accuracy above 90 %. In particular, unlike most algorithms in the state of the art, the performance on indoor and outdoor images are very similar (about 91 % vs. 93 %).

We also investigated the influence of image resolution on the algorithms performance: at lower resolution (i.e. 36 pixels), color seems to be more important than structure. Notwithstanding this, even without color, the accuracy is about 80 %. Concerning the size of the training set, we observed that even with very few (i.e. 32) training samples we can achieve a detection accuracy of more than 80 %.

The results obtained on the hierarchically organized categories allowed us to identify those types of scenes that are more problematic for our algorithm. These results are very insightful in that they provide directions for further improvements of our algorithm.

On the basis of these results we believe that the algorithm is suitable for the application in a variety of scenarios. We will make available the source code of our algorithms and the lists of images we used for training and test.

Acknowledgments We would like to thank Dr. Vikram Appia for the support to the implementation of his method.

References

1. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
2. Appia VV, Narasimha R (2011) Low complexity orientation detection algorithm for real-time implementation. In: Proceedings of SPIE-IS & T electronic imaging on real-time image and video processing, vol 7871, p 787108
3. Baluja S (2007) Automated image-orientation detection: a scalable boosting approach. Pattern Anal Appl 10(3):247–263
4. Borawski M, Frejlichowski D (2012) An algorithm for the automatic estimation of image orientation. In: Perner P (ed) Machine learning and data mining in pattern recognition, of lecture notes in computer science, vol 7376. Springer, Berlin, pp 336–344
5. Ciocca G, Cusano C, Schettini R (2010) Image orientation detection using low-level features and faces. In: Society of photo-optical instrumentation engineers (SPIE) conference series, of society of photo-optical instrumentation engineers (SPIE) conference series, vol 7537, pp 75370R–75370R–8
6. Deng J, Berg AC, Li K, Fei-Fei L (2010) What does classifying more than 10,000 image categories tell us? In: Computer vision–ECCV 2010, pp 71–84
7. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
8. Exchangeable image file format for digital still cameras (2002) EXIF version 2.2. JEITA CP-3451, Standard of Japan Electronics and Information Technology Industries Association
9. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
10. Fellbaum C (1998) Wordnet: an electronic lexical database. Bradford Books
11. Huttunen S, Rahtu E, Kunttu I, Gren J, Heikkil J (2011) Real-time detection of landscape scenes. In: Heyden A, Kahl F (eds) Image analysis of lecture notes in computer science, vol 6688. Springer, Berlin, pp 338–347
12. Lin C-J, Weng RC, Sathiya Keerthi S (2008) Trust region Newton method for large-scale logistic regression. J Mach Learn Res 9:627–650
13. Lumini A, Nanni L (2006) Detector of image orientation based on Borda count. Pattern Recogn Lett 27(3):180–186
14. Luo J, Boutell M (2005) Automatic image orientation detection via confidence-based integration of low-level and semantic cues. IEEE Trans Pattern Anal Mach Intell 27(5):715–726
15. Luo J, Crandall D, Singhal A, Boutell M, Gray RT (2003) Psychophysical study of image orientation perception. Spat Vis 16(5):429–457
16. Lyu S (2005) Automatic image orientation determination with natural image statistics. In: Proceedings of the 13th annual ACM international conference on multimedia, MULTIMEDIA '05. ACM, pp 491–494
17. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recognit 29(1):51–59
18. Pietikäinen M, Zhao G, Hadid A, Ahonen T (2011) Computer vision using local binary patterns. Number 40 in Computational Imaging and Vision, Springer
19. Takala V, Pietikainen M (2007) Multi-object tracking using color, texture and motion. In: IEEE conference on computer vision and pattern recognition, 2007. CVPR '07, pp 1–7
20. Tolstaya E (2007) Content-based image orientation recognition. In: Proceedings of the international conference on computer graphics and vision, GraphiCon 2007, pp 158–161
21. Vailaya A, Zhang H, Yang C, Liu F-I, Jain AK (2002) Automatic image orientation detection. IEEE Trans Image Process 11(7):746–755
22. Wang L, Liu X, Xia L, Xu G, Bruckstein A (2003) Image orientation detection with integrated human perception cues (or which way is up). In: Proceedings of the 2003 international conference on image processing 2003, ICIP 2003, vol 2–3, pp II–539–542

23. Whang Y, Zhang H (2004) Detecting image orientation based on low-level visual content. Comp Vision Image Underst 93(3):328–346
24. Xiao J, Hays J, Ehinger KA, Oliva A, Torralba A (2010) Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR), pp 3485–3492
25. Zhang L, Li M, Zhang H-J (2002) Boosting image orientation detection with indoor vs. outdoor classification. In: Proceedings of the sixth IEEE workshop on applications of computer vision, pp 95–99

Gianluigi Ciocca took his degree (Laurea) in Computer Science at the University of Milan in 1998, and since then he has been a fellow at the Institute of Multimedia Information Technologies of the Italian National Research Council, where his research has focused on the development of systems for the management of image and video databases and the development of new methodologies and algorithms for automatic indexing. He is currently a researcher in computer science at DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione) of the University of Milano-Bicocca, working on video analysis and abstraction.

Claudio Cusano is assistant professor at the Dep. of Electrical, Computer and Biomedical Engineering of the University of Pavia. He took his Ph.D. in 2006 at the the University of Milano-Bicocca. Since April 2001 he has been a fellow of the the ITC Institute of the Italian National Research Council. The main topics of his current research concern 2D and 3D imaging, with a particular focus on image analysis and classification, and on face recognition.

Raimondo Schettini is a professor at the University of Milano Bicocca (Italy). He is Vice-Director of the Department of Informatics, Systems and Communication, and head of Imaging and Vision Lab (www.ivl.disco.unimib.it). He has been associated with Italian National Research Council (CNR) since 1987 where he has leaded the Color Imaging lab from 1990 to 2002. He has been team leader in several research projects and published more than 200 refereed papers and six patents about color reproduction, and image processing, analysis and classification. Raimondo Schettini has been recently elected Fellow of the International Association of Pattern Recognition (IAPR) for his contributions to pattern recognition research and color image analysis.